Reduced-Order Probabilistic Emulation of Physics-Based Ring Current Models: Application to RAM-SCB Particle Flux

Alfredo A. Cruz [1], Piyush M. Mehta [1], Steven K. Morley [2], Humberto C. Godinez [3], Vania K. Jordanova [2]

[1]Department of Mechanical and Aerospace Engineering
West Virginia University
Morgantown, WV, USA

[2]Space Science and Applications
Los Alamos National Laboratory
Los Alamos, NM, USA

[3]Applied Mathematics and Plasma Physics
Los Alamos National Laboratory
Los Alamos, NM, USA

Key Points:

- A novel discrete sampling methodology is developed to select event intervals that generate the training, validation, and test datasets.
- Data-driven basis functions model the spatial variations and correlations and a Long-Short Term Memory (LSTM) models the temporal dynamics.
- Hierarchical ensemble of LSTMs provides a probabilistic emulator of the ring current particle flux with a robust and reliable uncertainty.

Corresponding author: A. A. Cruz, aac00009@mix.wvu.edu

Abstract

We present a proof of concept for the probabilistic emulation of the Ring current-Atmosphere interactions Model with Self-Consistent magnetic field (RAM-SCB) particle flux. We extend the workflow developed by Licata and Mehta (2023) by applying it to the ring current and further developing its uncertainty quantification methodology. We introduce a novel approach for sampling over 20 years of solar and geomagnetic activity to identify 30 simulation periods, each one week long, to generate the training, validation, and test datasets. Large-scale physics-based simulation models for the ring current can be computationally expensive. This work aims at creating an emulator that is more efficient, capable of forecasting, and provides an estimate on the uncertainty of its predictions, all without requiring large computational resources. We demonstrate the emulation process on a subset of the RAM-SCB particle flux data product, where we define this subset as a single energy channel of omnidirectional flux. A principal component analysis (PCA) is used for the dimensionality reduction into the reduced-space, and the dynamic modeling is performed with a recurrent neural network. A hierarchical ensemble of Long-Short Term Memory (LSTM) neural networks provides the statistics needed to produce a probabilistic output, resulting in a reduced-order probabilistic emulator (ROPE) that performs time-series forecasting of the ring current's particle flux with an estimate on its uncertainty distribution. The resulting ROPE from this smaller subset of RAM-SCB particle flux provides dynamic predictions with errors less than 11% and calibration scores under 10%, demonstrating that this workflow can provide a probabilistic emulator with a robust and reliable uncertainty estimate when applied to the ring current.

Plain Language Summary

The ring current is a region of the inner magnetosphere where space weather events affect the charging environment experienced by spacecraft. Running large-scale physics-based simulation models in domains such as the ring current can be computationally expensive. This work aims at creating an emulator that runs much faster, is capable of forecasting, and can provide an estimate on the uncertainty of its predictions, all without requiring large computational resources. It is important to note that emulators are not developed to replace physics-based models but rather enable a higher adoption rate and usage for more system-wide investigations. To begin, a subset of the particle flux data product is converted into a reduced, simpler form. A neural network is then implemented to model the ring current environment in this reduced form and trained on a set of week-long simulations derived from a newly developed sampling methodology. An ensemble of these neural networks is then combined into a single predictor. The resulting reduced-order probabilistic emulator (ROPE) provides time-series predictions with error estimates, which together define a probabilistic output. The presented ROPE can make predictions with errors less than 11% with calibration scores under 10%, ultimately demonstrating that this workflow can provide a probabilistic emulator of the ring current with a robust and reliable uncertainty estimate.

1 Introduction

The motivation for this work stems from the plasma populations that can detrimentally affect spacecraft, specifically those contributing to the charging environment. Green et al. (2017) describes the various anomalies that have impacted the satellite industry, where surface and internal charging were dominant issues (Koons et al., 1999). Anomalies such as these can damage electrical components & thermal coatings, destroy sensors and/or scientific instruments, interfere/spoof communication signals, and potentially leave a spacecraft completely inoperable. Modeling of the inner magnetosphere has been used to investigate the potential cause of a detected anomaly (Koons & Fennell, 2006; Ganushkina et al., 2017) but can also aid spacecraft designers and operators in mit-

igating potential damage or disruptions to their spacecraft. Yu et al. (2019) illustrates a recent competition designed to assess the capabilities of current inner magnetosphere models in determining the surface charging environment during the 17 March 2013 geomagnetic storm. Large-scale physics-based simulation models provide invaluable insight into the physical evolution of dynamical systems such as the ring current. Their use in an operational setting, however, can sometimes be limited by computational restrictions, inviting faster, more efficient models to take their place. Development of more efficient models has gained popularity in the thermosphere (Mehta et al., 2018; Gondelach & Linares, 2021; Licata & Mehta, 2023), so our work aims to extend this application and provide an emulator to the Space Weather community capable of an efficient and probabilistic prediction of ring current particle flux using the Ring current-Atmosphere interactions Model with Self-Consistent magnetic field (RAM-SCB) (Engel et al., 2019; Jordanova et al., 2006; Jordanova, Morley, et al., 2022).

The solar wind (SW) is the primary source of energy deposition that drives the Earth's magnetospheric dynamics (Pulkkinen et al., 2007). Since the near-Earth environment is mostly comprised of charged particles in the form of plasma, there are inevitable and unpredictable hazards that come with operating in this type of environment (Green et al., 2017). The inner magnetosphere is a domain in which the Earth's magnetic field lines are closed and charged particles are trapped within these magnetic fields. In this region, Earth's magnetic field closely resembles that of a dipole magnetic field and spans from the dayside magnetopause to the outer transition region (Spence et al., 1989), roughly 10-12 Earth radii (R_E) (Russell et al., 2016; Daglis et al., 1999; Spence et al., 1989; Ganushkina et al., 2017). The trapped particles form different plasma populations that both reside and overlap with each other, which not only complicates the physical processes governing them but also creates a dynamically coupled system (Russell et al., 2016; Yu et al., 2012).

The primary plasma populations found in the inner magnetosphere are the plasmasphere, ring current, and radiation belts. They all coexist together but are typically differentiated by the range of particle energies within each population. The plasmasphere contains cold, dense plasma with energies of a few electronvolts (eV), and its constituents generally originate from the ionosphere (Daglis et al., 1999; Russell et al., 2016; Fok et al., 2021). The plasmasphere is not known to directly affect the Earth's magnetic configuration, but its high density has been known to propagate electromagnetic waves, which can influence both the ring current and radiation belt populations (Daglis et al., 1999; Jordanova, Thorne, et al., 2010; Jordanova et al., 2012; Yu et al., 2012; Ganushkina et al., 2017). The radiation belts are two lobed regions separated by a small gap called the slot region and typically are the most energetic population in the inner magnetosphere (Russell et al., 2016; Li & Hudson, 2019). This region consists of energetic ions and relativistic electrons that range anywhere from ∼500 keV to a few MeV (Russell et al., 2016; Li & Hudson, 2019; Fok et al., 2021). The radiation belts are also known to be highly variable during geomagnetic storms (Friedel et al., 2002; Thorne, 2010). The ring current has energies roughly in-between these two populations, ∼10–400 keV, and is generated by the movement of charged particles experiencing a gradient-curvature drift (Daglis et al., 1999; Jordanova et al., 2014; Russell et al., 2016; Fok et al., 2021).

During geomagnetic activity, the ring current gains population from plasma that is accelerated by reconnection in the magnetotail, making it the population that carries the majority of pressure and current directly into the inner magnetosphere (Daglis et al., 1999; Jordanova et al., 2014; Ganushkina et al., 2017). These accelerated particles experience a nonuniform magnetic field as they travel inward from the magnetotail that causes them to drift in opposite directions (gradient-curvature drift), inducing a current, with the ions moving towards the dusk-side and electrons towards the dawn-side of Earth. This induced westward current, called the ring current, is the main contributor to the

magnetic depression observed by ground-based magnetometers during geomagnetic storms (Daglis et al., 1999; Ganushkina et al., 2017; Fok et al., 2021).

2 Methodology

This work leverages reduced-order modeling (ROM) with machine learning (ML) techniques to significantly decrease the computational cost of physics-based simulation models while maintaining their high fidelity. Note: Emulators are not developed to replace physics-based models but rather enable a higher adoption rate and usage for more system-wide investigations. A ROM parses out which modes of variability are most influential (Mehta et al., 2018; Mehta & Linares, 2017) and then operates in this reduced space, or lower-dimensional representation. Figure 1 shows a high-level overview of the emulation process, where the following steps are covered in more detail:

1) Event Selection in Section 2.1
2) Simulate Events in Section 2.2
3) Dataset Creation in Section 2.3
4) Dimensionality Reduction in Section 2.5
5) Dynamic Modeling in Section 2.6
6) Model Ensemble in Section 2.7
7) Uncertainty Quantification in Section 2.8

Steps that are developed in either the physical or reduced space are color coded as blue and orange, respectively. To begin, a novel discrete sampling methodology is introduced to determine a set of geomagnetic storms that encompasses a wide range of solar and geomagnetic activity. This list of storms is then run through RAM-SCB to produce simulation outputs that generate the ML datasets used to develop the emulator. A dimensionality reduction is applied that identifies the dominant spatial modes of variability and transforms the ML datasets into the reduced space. This is done to enable future data assimilation applications by significantly simplifying the calculations needed for high-dimensional systems (Mehta & Linares, 2018; Maulik et al., 2022). A dynamic model, in this case a recurrent neural network, is then developed to predict the system's temporal variations in the reduced space, where the inclusion of a neural network enables nonlinear modeling. The resulting dynamic model is deterministic, meaning that it only provides a point estimate. Thus, we leverage an ensemble of deterministic models to compute an uncertainty quantification (UQ). The final step is to then reconstruct the model ensemble's predictions and uncertainty statistics back into the physical space by reversing the dimensionality reduction transformation. It is important to note that any development in the reduced space can be evaluated in the physical space by utilizing this reconstruction step.

2.1 Event Selection

The first and arguably most important step of any ML-based model development is to build proper training, validation, and test datasets. Here, we use the definitions common in ML literature where the validation dataset refers to out-of-sample data not seen by the model during training that can be used to measure performance, optimize methods, and make decisions. The test dataset is also out-of-sample but is only used to measure model performance. Using NASA's SPDF (Space Physics Data Facility) OMNIWeb database, we analyze solar wind and geomagnetic data from 2000-2020, all at a 1-minute cadence. The following solar wind parameters were queried: velocity components (V_x, V_y, V_z) in GSE coordinates, interplanetary magnetic field (IMF) components (B_x, B_y, B_z) in GSM coordinates, proton density, proton temperature, and flow pressure. The AL and SYM-H geomagnetic indices were also included in the query. Simulating this en-

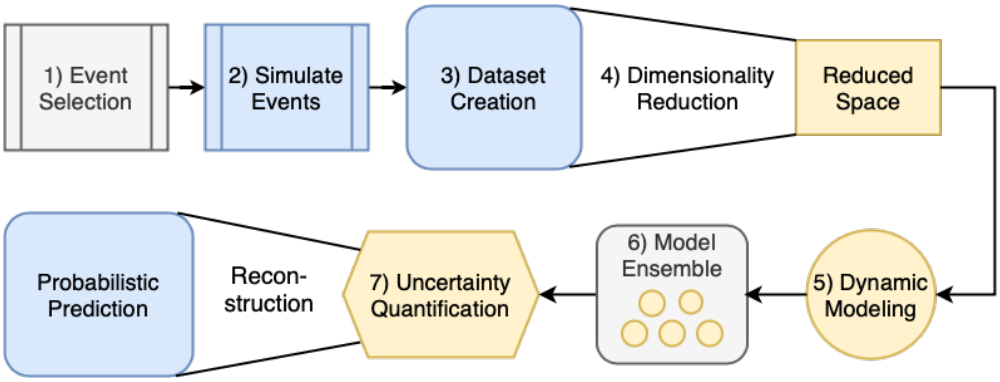

Figure 1. Overview of emulator workflow from creation of the ML datasets, through the reduced-order dynamic modeling, culmination of the model ensemble, and ending with the final probabilistic output. Steps developed in either the physical or reduced space are color coded in blue and orange, respectively.

tire span in physics-based models such as RAM-SCB would be extremely challenging and computationally expensive. Therefore, we developed a custom discrete sampling methodology to determine a set of random events that adequately covers this entire span of solar wind drivers and ring current responses.

The 21 years of OMNIWeb data from 2000-2020 are split into smaller, more manageable weekly segments, each representing a candidate simulation interval. These 7-day intervals are long enough to encompass a space weather event & recovery period but short enough to minimize the likelihood that separate events would be grouped together. When initializing large-scale physics-based simulations, the initial condition should be set to low activity levels so that the internal components can stabilize before the system is perturbed. RAM-SCB is known to not perform well when simulations are initialized with heightened activity levels (Jordanova, Engel, et al., 2022; Jordanova et al., 2014). Therefore, we filter out weekly intervals that begin with radial SW velocities (V_x) exceeding 500 km/s. A 7-day sliding window is implemented to avoid disqualifying events solely based on this initialization criteria, which is marched daily and identified 7,664 candidates. We limit the amount of missing data in each candidate interval to a cumulative total of 36 hours (1.5 days) for any given parameter, which amounts to roughly 21% of the data within that week. Any smaller gaps that pass through this filter are linearly interpolated using the entire weekly timeseries. Applying these two filters reduced the number of possible candidates down from 7,664 to 2,839 weekly intervals.

This work introduces a novel custom discrete sampling methodology that efficiently and effectively samples our full parameter space. Each of the 2,839 week-long candidate intervals are located in a 4-dimensional parameter space using a set of summary statistics: 1) minimum SYM-H, 2) mean AL, 3) mean V_x, and 4) minimum B_z. The strength of the ring current disturbance and overall geomagnetic activity is captured by taking the minimum SYM-H. The mean AL is used to describe the impulsive energy dissipation and injection of plasma into the inner magnetosphere. The strength of the SW drivers are characterized by the mean V_x and minimum B_z. We then leverage concepts behind Latin hypercube sampling (LHS) that normally aim to efficiently reproduce the underlying probability distributions (Deutsch & Deutsch, 2012) but instead utilize them to provide sufficient coverage of our parameter space. In lieu of splitting each parameter's distribution into evenly-spaced probability intervals, we take the full range of each parameter and separate it into 10 linearly-space bins. Each bin is then assigned an equal probability, and a bin index is randomly drawn with replacement. In the event that a

bin for any given parameter is empty, another index is randomly selected until a bin with at least one candidate event is chosen. Once an occupied bin is identified, a candidate interval is then randomly selected, with uniform probability, from the bin. This is repeated for each parameter, providing a pool of 4 candidate intervals. To finalize a selection, an interval from this pool is then randomly selected, removed from each of the parameter spaces, and then the selection process is repeated for the number of desired samples. This differs from LHS, which is typically used to efficiently sample continuous probability distributions that contain the majority of samples in the high-probability regions of the parameter space. Instead, we are aiming for a more uniform converge of the parameter space to avoid a heavily imbalanced training dataset dominated by quiescent times.

Table 1. Training Events Identified by the Novel Sampling Methodology.

Event	Start Date	min(SYM-H)	mean(AL)	mean(V_x)	min(B_z)
TRNG 1	2001-03-31	-437.0	-216.3	-580.4	-44.4
TRNG 2	2001-04-07	-280.0	-272.2	-605.7	-20.3
TRNG 3	2001-10-16	-219.0	-173.8	-379.6	-17.8
TRNG 4	2001-11-24	-234.0	-77.9	-506.1	-26.6
TRNG 5	2002-09-05	-168.0	-224.2	-440.7	-22.8
TRNG 6	2003-03-14	-67.0	-283.4	-670.2	-7.4
TRNG 7	2003-11-09	-134.0	-412.9	-638.5	-8.5
TRNG 8	2003-11-20	-490.0	-251.5	-542.9	-51.3
TRNG 9	2004-07-19	-168.0	-287.0	-505.4	-18.6
TRNG 10	2005-07-08	-114.0	-253.4	-435.7	-18.9
TRNG 11	2005-09-10	-137.0	-381.8	-706.5	-6.5
TRNG 12	2005-11-30	-25.0	-102.5	-607.2	-3.6
TRNG 13	2007-11-13	-24.0	-53.0	-516.9	-5.6
TRNG 14	2008-07-12	-41.0	-116.9	-566.1	-7.7
TRNG 15	2009-03-08	-45.0	-79.3	-409.8	-10.2
TRNG 16	2009-09-09	-20.0	-50.5	-332.4	-6.1
TRNG 17	2011-01-07	-49.0	-82.1	-531.2	-4.6
TRNG 18	2012-05-02	-32.0	-53.1	-305.2	-8.3
TRNG 19	2013-01-17	-58.0	-62.9	-376.7	-12.3
TRNG 20	2013-10-30	-57.0	-84.0	-348.6	-8.1

A total of 30 events were selected using this sampling methodology, with 20 used for the training (TRNG) dataset (see Table 1) and 5 used for each of the validation (VAL) and test (TST) datasets (see Table 2). Figure 2 displays the training, validation, test, and remaining samples (SAMP) in red, green, orange, and dark blue, respectively. Histograms of each sample parameter's distribution are shown on the diagonal plots. The panels below the diagonal show 2-D scatter plots between the various parameter pairs, and the bivariate kernel density estimates (KDE) (Węglarczyk, 2018; Waskom, 2021) are plotted above the diagonal. This split leads to a training/validation/test ratio of 66/17/17%. The events in each dataset were selected such that they contained a wide range of activity levels, with the training dataset having the largest possible range in each of the parameter spaces. The VAL 2 and TST 2 simulations begin only days apart, so the extrema in SYM-H and B_z are the same for both events because of this overlap. However, due to their offset, the initial state and evolution of each week-long interval will differ. These two events constitute a period of prolonged geomagnetic activity where two significant storms occurred within a few days of each other. Each storm is captured very differently in the two intervals, although the most severe activity overlaps into both events and is emphasized by the minimum statistic.

Table 2. Validation and Test Events Identified by the Novel Sampling Methodology.

Event	Start Date	min(SYM-H)	mean(AL)	mean(Vx)	min(Bz)
VAL 1	2003-05-05	-93.0	-297.5	-670.1	-7.5
VAL 2	2004-11-05	-394.0	-409.3	-542.7	-44.7
VAL 3	2005-01-12	-107.0	-251.9	-618.7	-12.3
VAL 4	2012-11-09	-118.0	-101.0	-357.5	-17.4
VAL 5	2017-12-01	-47.0	-129.5	-443.0	-11.1
TST 1	2002-04-19	-185.0	-206.3	-482.8	-13.7
TST 2	2004-11-03	-394.0	-277.3	-475.3	-44.7
TST 3	2005-08-24	-179.0	-164.9	-479.3	-32.4
TST 4	2013-04-24	-52.0	-132.2	-435.1	-12.8
TST 5	2017-03-26	-86.0	-259.1	-586.1	-9.2

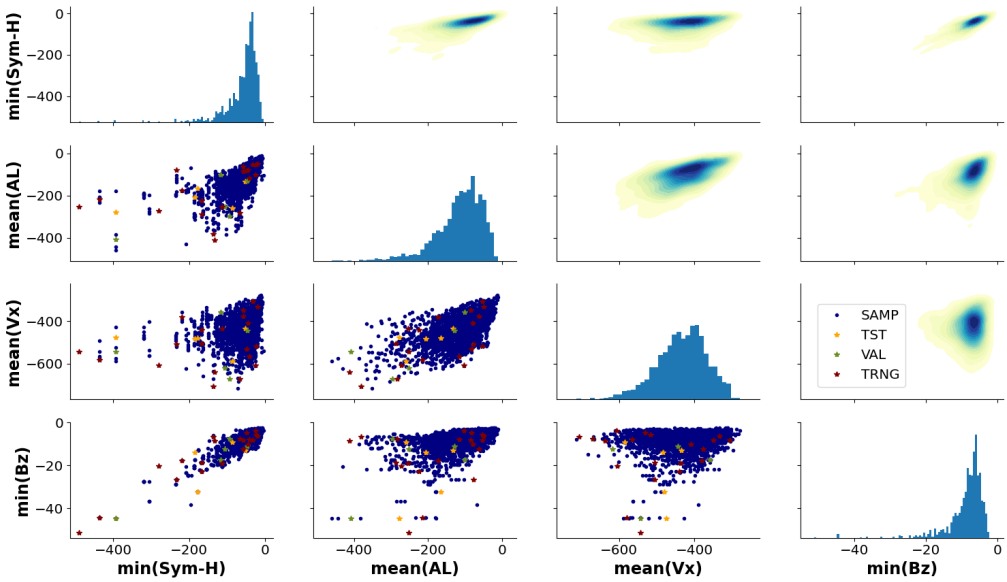

Figure 2. Pairplot displaying the TRNG, VAL, and TST events identified by the novel custom discrete sampling methodology. It visualizes the sampling taken within each parameter's distribution, where histograms of each parameter are shown on the diagonal plots. The panels below and above the diagonal show 2-D scatter plots between parameter pairs and the bivariate KDEs, respectively. The remaining samples (SAMP) are shown in dark blue.

2.2 Simulate Events

RAM-SCB is a unique inner magnetosphere model developed at Los Alamos National Laboratory (LANL) that combines a kinetic ring current plasma model (RAM) (Jordanova, Zaharia, & Welling, 2010; Jordanova, Engel, et al., 2022) with a 3-D self-consistent magnetic field model (SCB) (Zaharia et al., 2006; Jordanova et al., 2006). RAM and SCB are two separate components that are two-way coupled for self-consistent evolution (Jordanova, Engel, et al., 2022). RAM-SCB began as a research-based code with limited options but is now a powerful and highly configurable open-source software that is highly parallelizable (Engel et al., 2019; Jordanova, Engel, et al., 2022). By default, RAM-SCB models 4 species of charged particles (H^+, He^+, and O^+, and e^-) in energies ranging from 100 eV to 500 keV. Its spatial domain spans from 2 to 6.5 R_E with a 0.25 R_E resolution along the magnetic equatorial plane. One of its many data products is the equatorial particle flux, which is provided in terms of magnetic local time (MLT), radial distance (R_E), energy (keV), and pitch angle (PA) (Jordanova, Engel, et al., 2022).

All 30 events (20 training, 5 validation, and 5 test) were run using WVU's Thorny Flat cluster, each with an identical configuration. All system environment information and input files are provided for reproducibility purposes (Cruz et al., 2023). Each simulation utilizes 13 CPU cores, is run in its own standalone run directory, and outputs 92 GB of data. The total 210 days of simulation time were completed in just under 48 days of computational time, resulting in an average speed of 4.4x real-time. An overall wall time of 16 days was ultimately needed because multiple simulations were run simultaneously over several compute nodes on the Thorny Flat cluster. The total amassed outputs for the set of 30 simulations was 3 TB.

2.3 Dataset Creation

RAM-SCB outputs equatorial, directional differential flux as a 4-D hypercube for each various plasma species identified in its setting file (PARAM.in), which we set to include all default species (H^+, He^+, and O^+, and e^-) for each simulation. There are 72 pitch angles over 35 energy channels with spatial dimensions of 25 MLTs and 20 radial distances, equating to a data shape of (72, 35, 25, 20) per timestep. Each 7-day simulation has outputs at a 10-minute cadence, resulting in 1,008 timesteps per simulation. RAM-SCB's particle flux is saved in NetCDF files at the output cadence, meaning there are 1,008 individual flux files per simulation, each roughly 40 MB. The resulting data shape for an entire simulation of particle species comes out to be (1008, 72, 35, 25, 20). We decided to develop this proof of concept using protons (H^+) since they are known to be the most dominant species for convection in Earth's ring current (Daglis et al., 1999; Jordanova et al., 2012, 2014). Concatenating the 20 training simulations all together creates a data structure with shape (20160, 72, 35, 25, 20) that occupies roughly 101 GB of physical memory. Any operation (add, subtract, mean, etc.) roughly doubles the memory usage to around ∼200 GB, requiring significant computational resources to work directly on a data structure this size.

In creating new datasets, there are many unforeseen steps needed in order to get the data in a suitable state for analysis. To start, our RAM-SCB simulations are all run using double precision, thus small numbers (i.e. 10^{-300}) are found in the loss cone and at the inner boundary. To mitigate the propagation of these small numbers as well as reduce memory usage, we converted our data to single precision, which resets the minimum threshold to around 10^{-45}. In addition, RAM-SCB uses ghost cells for the inner radial boundary condition at 1.75 RE, across all pitch angles and energy channels that should not be included in physical analyses. To remove ghost cells and reduce the emulated area, we truncated all radial distances below 3 R_E, resulting a data shape of (20160, 72, 35, 25, 15) that occupies 71 GB of physical memory.

Because of this dataset's size, our emulator is developed using only a subset of the RAM-SCB particle flux data product. Developing an emulator on a smaller subset of the data has the benefit of speeding calculations up because there is less data, thus making each step in the workflow both simpler and faster. Once the emulation process is demonstrated on this smaller subset, it can then be expanded to incorporate RAM-SCB's full 4-D data product. Since maintaining the spatial information is key for modeling the system's dynamics, we decided to only use a single energy channel and integrate the pitch angle distribution to obtain omnidirectional flux. The 208 keV energy channel was selected since the differential flux is already separated by energy. We then integrated directional flux into omnidirectional flux (normalized per steradian) following Bourdarie et al. (2012) to further reduce the dimensionality:

$$j_{omni} = \frac{\int_0^\pi j(E,\alpha)\ sin(\alpha)\ d\alpha}{\int_0^\pi sin(\alpha)\ d\alpha} \tag{1}$$

This results in omnidirectional differential flux (j_{omni}) with units of $cm^{-2}\ s^{-1}\ sr^{-1}\ keV^{-1}$, where α is the pitch angle and $j(E,\alpha)$ is the directional differential flux at a specific energy (E) and pitch angle (α). By removing the pitch angle information and selecting a single energy channel, the training data is now reduced to just the spatial dimensions with a shape of (20160, 25, 15) that occupies 30 MB of physical memory. This same process is also applied to the validation and test datasets.

2.4 Metrics

The metric used to describe error in the physical space is the median symmetric accuracy (MdSA; S. K. Morley et al., 2018). Ring current particle flux spans many orders of magnitude, is strictly positive, and has a physically meaningful zero value (Zheng et al., 2019). Normally, datasets with large ranges utilize relative error metrics, such as the percent error, that are able to scale values over these large ranges. The mean absolute percent error (MAPE) is widely used in space science data analysis (S. K. Morley et al., 2018; Zheng et al., 2019) but has drawbacks. The MdSA metric was developed to help mitigate many of these concerns (S. K. Morley et al., 2018), aimed at inner magnetospheric flux data. First, it is a relative error metric that penalizes over- and under-estimations equally. The median is also used instead of the mean because it is a robust central tendency statistic that is resistant to outliers and bad data. For the development of the uncertainty quantification in Section 2.8, the median statistic will be used whenever an average is taken over the temporal range (t), since outliers are expected to arise during the highest solar and geomagnetic activity levels. Lastly, MdSA is easily interpreted as a straight-forward accuracy, or percent error. Equation 2 shows how to compute the MdSA, where $Q = \frac{\text{pred}}{\text{truth}}$ is defined as the accuracy ratio.

$$MdSA = 100\ (\ \exp(\ \text{Median}(\ |log_e(Q)|\ )\ )\ - 1) \tag{2}$$

The metric used to determine the bias in either the physical or reduced spaces is the symmetric signed percentage bias (SSPB; S. K. Morley et al., 2018). Similarly to the MdSA, the SSPB is a relative error metric that penalized over & under estimations equally. The median is also used in its calculations as a robust central tendency statistic that is resistant to outliers and bad data. The SSPB metric is interpreted like a mean percentage error where an unbiased prediction is at 0% SSPB and an over- or under-prediction produces positive and negative SSPB, respectively.

$$SSPB = 100\ \mathrm{sgn}(\ \mathrm{Median}(log_e(Q))\ )\ (\ \exp(\ |\ \mathrm{Median}(log_e(Q))\ |\ )\ - 1) \tag{3}$$

The standard metric of mean squared error (MSE) is used to describe the errors of the temporal coefficients in the reduced space (ref. Equation 5). It is also key to note that the MSE optimized in the dynamic models (Section 2.6) will have gone through multiple transformations (logarithmic, dimensional reduction, and standardization), making it extremely difficult to interpret. Thus, any model performance metrics must be determined post-process by reconstructing the predictions back into the physical space. This is one of the unique challenges of working with ROMs: the ML algorithms analyze the reduced-space representations of the data, which are not necessarily interpretable.

The reliability metric used for the UQ is the calibration error score (CES). It is used for consistency with developments in the thermosphere (Licata, Mehta, Tobiska, and Huzurbazar (2022); Licata, Mehta, Weimer, et al. (2022); Licata and Mehta (2022, 2023)) and is a relative metric that is easily interpreted as a percent error. The CES measures the deviation of the observed cumulative probability $p(\hat{\alpha}_{r,m})$ from the expected cumulative probability $p(\alpha_{r,m})$. The above probabilities are calculated using the process described in Section 2.5.1 of Licata, Mehta, Tobiska, and Huzurbazar (2022), where the prediction intervals span from 5-99% in increments of 5%. The reliability of the uncertainty estimates is visualized by plotting $p(\hat{\alpha}_{r,m})$ against $p(\alpha_{r,m})$, also known as a calibration curve. The calibration curves presented in this work are under the assumption of a Gaussian distribution, and the reliability under non-Gaussian distributions will require further investigation. An uncertainty estimate that matches a normal distribution is indicated by a 45° line (i.e., $y = x$) on the calibration curve. Any deviation from this line indicates an over or underestimation of the uncertainty for a curve that is above or below the line, respectively. Here, the calibration curves and CES are all calculated in the reduced space. The CES calculation is shown in Equation 4,

$$CES = \frac{100\%}{r \cdot m} \sum_{r} \sum_{m} \left| p(\alpha_{r,m}) - p(\hat{\alpha}_{r,m}) \right| \tag{4}$$

where r is the number of reduced-space coefficients and m is the number of prediction intervals used to determine the cumulative probabilities.

2.5 Dimensionality Reduction

The next step in the emulation process is to reduce the dimensionality of the datasets. A system's spatial variations are normally represented by a set of basis vectors that are both independent in time and mutually orthogonal, or what is commonly known as empirical orthogonal functions (EOF) (Bjornsson & Venegas, 1997; D. Wilks, 2011). The temporal variations $\alpha_i(t)$ are then added in as weights to the spatial EOFs (Mehta & Linares, 2017; Mehta et al., 2018; Licata, Mehta, Tobiska, & Huzurbazar, 2022), which we will be referring to as the reduced-order temporal coefficients. This is shown in Equation 5, where $\mathbf{X} \in \mathbf{R}^n$, $\mathbf{s}$ represents the spatial domain, t represents the temporal domain, and U contains the spatial modes.

$$\mathbf{X}(\mathbf{s}, t) = \overline{\mathbf{X}}(\mathbf{s}) + \widetilde{\mathbf{X}}(\mathbf{s}, t) \quad \text{where} \quad \widetilde{\mathbf{X}}(\mathbf{s}, t) \approx \sum_{i=1}^{r} \alpha_i(t)\, U_i(\mathbf{s}) \tag{5}$$

One of the most challenging aspects of ROM on space weather systems is to properly adjust the timing of the temporal variation predictions with the corresponding in-

puts driver(s) (Mehta & Linares, 2017). The resulting reduced-space transformation has a controlled loss of accuracy with respect to the physical model, through optimized truncation, along with the benefit of being in a much more manageable & practical form for analysis (Mehta et al., 2018). Before the dimensionality can be reduced, though, a logarithmic transformation (log_{10}) is normally applied (Zheng et al., 2019). Transformations using logarithms not only reshape skewed distributions into more normalized distributions but also significantly reduce their value range (D. S. Wilks, 2011). This also implies that the antilogarithm must be taken directly after the dimensional reduction is reversed during any reconstructions back into the physical space.

The ROM process begins by reducing the spatial dimensionality of the system by applying a principal component analysis (PCA). PCA is an unsupervised method used to map high-dimensional data into an uncorrelated lower-dimensional space by means of a linear rotation and scaling. In some literature, PCA and EOF can be used interchangeably (Bjornsson & Venegas, 1997). PCA is a popular starting point for reducing the dimensionality of space weather domains because it is a simple yet powerful method (McGranaghan et al., 2015; Mehta & Linares, 2017; Licata & Mehta, 2022; Licata, Mehta, Tobiska, & Huzurbazar, 2022; Licata & Mehta, 2023). Once the logarithmic transformation ($\log_{10}$) has been applied, the next step is to remove the spatial mean $\overline{\mathbf{X}}(\mathbf{s})$ from the training data (see Equation 5), which is referred to as centering the data. We use the spatial mean because the mean is taken over the temporal dimension, and it is this mean of the training dataset that is used when transforming any and all data between the physical and reduced spaces. The last preparation step before performing the actual PCA is to convert the data into a 2-D array (Bjornsson & Venegas, 1997; D. Wilks, 2011). Since we are analyzing only a single energy of omnidirectional flux, the spatial dimensions (25, 15) will be collapsed into a single array of size $n = 375$, resulting in a data shape of (20160, 375). Our PCA is implemented using a singular value decomposition (SVD) solver (Pedregosa et al., 2011),

$$\widetilde{\mathbf{X}} = U\Sigma V^T \quad \text{where} \quad \widetilde{\mathbf{X}} = \begin{bmatrix} | & | & & | \\ \widetilde{\mathbf{x}}_1 & \widetilde{\mathbf{x}}_2 & ... & \widetilde{\mathbf{x}}_n \\ | & | & & | \end{bmatrix}, \tag{6}$$

where U contains the left singular vectors of $\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^T$, V contains the right singular vectors of $\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}}$, Σ is a diagonal matrix containing the squares of the corresponding eigenvalues, and all are arranged in descending order. We use this PCA decomposition to transform the ML datasets into the reduced-space representation.

The spatial modes of variability identified by the PCA decomposition often reveal or resemble known physical processes and phenomenon (McGranaghan et al., 2015). Direct interpretations, however, are not necessarily guaranteed since each mode may contain multiple processes or various combinations of physical processes. Figure 3 shows the mean and first 7 right singular vectors from the PCA, or spatial modes of variability, on RAM-SCB's grid (for the 208 keV proton flux). Upon visual inspection, there are roughly 3 trends: 1) radial falloff, 2) symmetric rings, and 3) asymmetric structures. The mean and Mode 1 are both examples of the radial falloff and reminiscent of the ring current's expected location. During quiescent times, the ring current is normally confined to radial distances under 4.5 R_E ($R < 4.5$ R_E) for high-energy protons ($E > 200$ keV) (Jordanova et al., 2014), which is validated by the mean plot. During the main phase of a geomagnetic storm, most all particle fluxes are reduced at radial distances $R > 4.5$ R_E, and the ring current is compressed closer towards the Earth (Jordanova et al., 2012). Mode 1 agrees with this reduction and compression, which by definition is also the most dominant mode of variability. The symmetric rings in Modes 2, 4, and 7 seem to simply resemble basis functions for the symmetric ring current, which becomes more defined at higher parti-

cle energies. During the main and recovery phases of a geomagnetic storm, each particle's drift is known to vary radially (Jordanova et al., 2012), creating similar symmetric rings. The asymmetric structures in Modes 3, 5, and 6 are more difficult to interpret and will require further analysis because the ring current is comprised of both a symmetric and asymmetric portion, or partial ring current (Daglis et al., 1999; Russell et al., 2016), as well as drifting injected particles. Most of the asymmetric modes show variations between dawn and dusk, which is the expected drift path for ions (H^+) in the ring current.

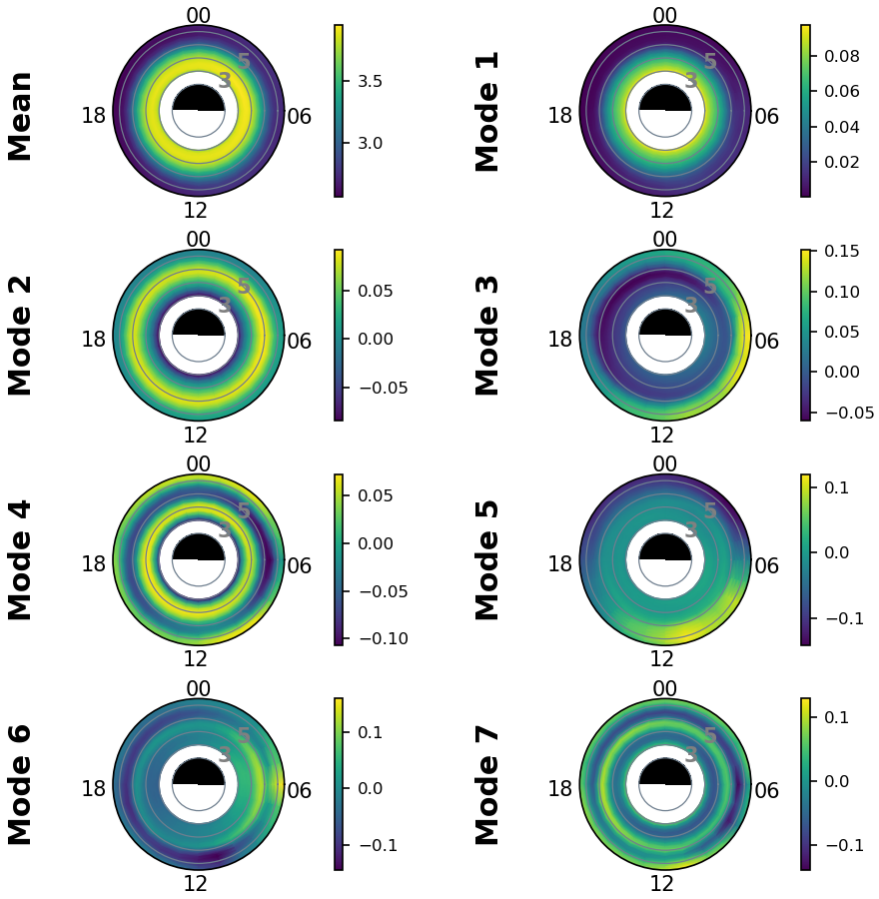

Figure 3. Mean and first 7 spatial modes of variability identified by the PCA from the right singular vectors plotted on RAM-SCB's grid. The modes are ordered in terms of importance, meaning the mean is the most dominant followed by mode 1, and so on.

PCA's ability to reduce the dimensionality of a dataset comes into play when the modes that contribute the least to the system's variability are identified and removed. Determining the point of truncation for an emulator is a balance between minimizing the amount of reconstruction error and reducing the dimensionality of the system for enough observability (Mehta & Linares, 2018) in later data assimilation applications. Typically, the truncation point is set to where the reconstruction error is on the order of a few percent and the dimensionality is reduced to around 10. We decided to truncate our PCA at 20 modes (r=20), which reduces the spatial dimensionality from $\widetilde{\mathbf{X}} \in \mathbf{R}^n$ to $\widetilde{\mathbf{X}} \in \mathbf{R}^r$. The cumulative variance contribution is plotted on the left axis of Figure 4, where the first 20 modes are shown to capture 82.9% of the variability. Figure 4 also reveals

that the truncation error (right axis) from the reconstruction back to the physical space using 20 PCA modes is 2.9% MdSA.

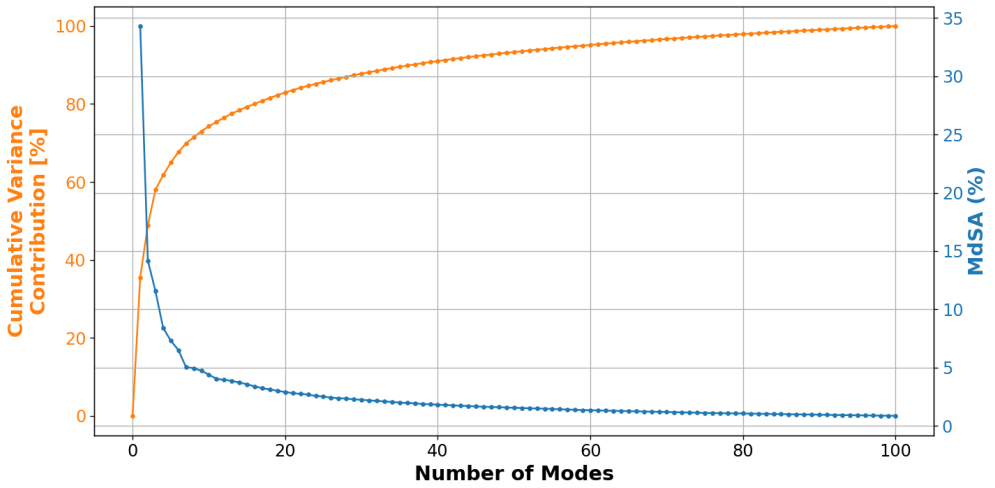

Figure 4. The cumulative variance contribution (orange) for each mode of the PCA, and truncation error (blue) of the reconstruction back into the physical space using the specified number of modes on the training dataset.

To illustrate the robustness of the PCA decomposition, 3 different timesteps from the VAL 4 simulation (see Table 2) are reconstructed back to the physical space and shown in Figure 5. Timesteps were chosen before, during, and after the geomagnetic storm, and the resulting truncation errors between the actual (left plots) and reconstructed (middle plots) fluxes are plotted on the right. The errors in the plots for before and after the storm are on the same order as the truncation error, with an MdSA of 3.4% and 1.6%, respectively. However, errors are expected to increase during the geomagnetic storm, since the linear PCA would not be able to capture any nonlinearities in the system's dynamics. Even though local errors rose up to 33% during the storm, the MdSA only increased a few percent to 5.9%.

2.6 Dynamic Modeling

For dynamic models such as RAM-SCB, ML algorithms capable of capturing the temporal evolution of these systems are required. A class of neural network that is well suited for modeling time-series data is a recurrent neural network. We implement a Long-Short Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997; Gers et al., 2002) recurrent neural network to model and predict RAM-SCB's temporal variations (Wang et al., 2022; Licata & Mehta, 2023). Since magnetospheric responses tend to lag behind their SW drivers (Bargatze et al., 1985; Mehta et al., 2018), an LSTM copes with this temporal hysteresis by incorporating knowledge of previous timesteps, often referred to as the lookback period, in its short-term memory while still maintaining information on any long-term trends in its cell state (Licata & Mehta, 2023). An LSTM can also capture nonlinear relationships between the input drivers and reduced-space temporal coefficients. The preconditioning of the inner magnetosphere (Kozyra et al., 1998, 2002; S. Morley & Lockwood, 2006) adds another layer of complexity on how the LSTM learns the dynamics of this system. The ability to capture nonlinear correlations while also managing the aforementioned temporal hysteresis and preconditioning is why we chose an LSTM for the dynamic modeling of our emulator. LSTMs require a unique input structure, con-

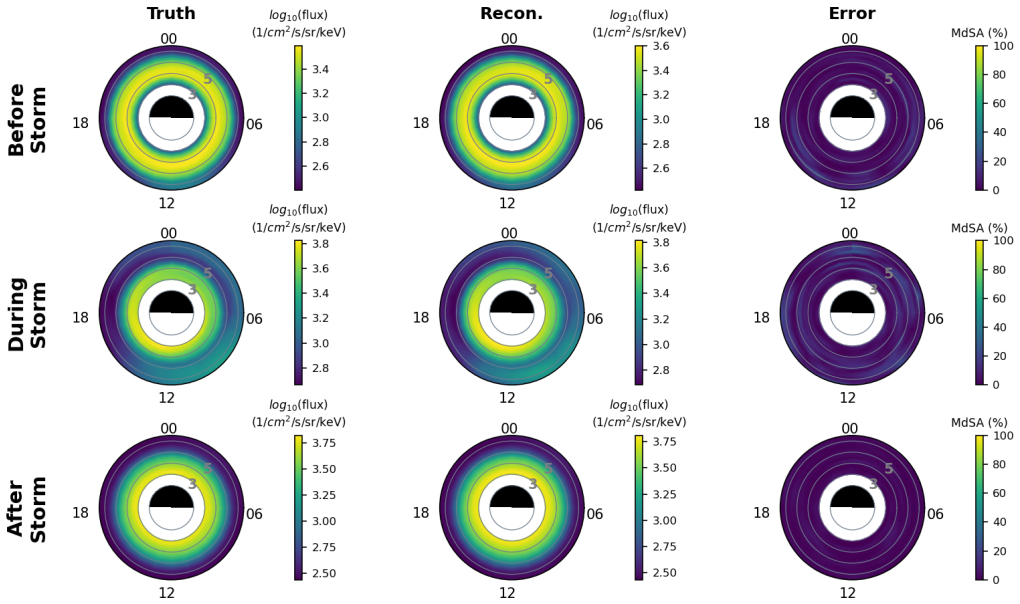

Figure 5. Snapshots taken before, during, and after the geomagnetic storm of the VAL 4 simulation with the truncation errors (right) between the actual (left) and reconstructed (middle) fluxes. The truncation errors for the before, during and after snapshots are 3.4%, 5.9%, and 1.6% MdSA, respectively.

taining the reduced-space temporal coefficients as well as a set of user-defined input drivers. We chose the same parameters used during the discrete sampling in Section 2.1 (SYM-H index, AL index, IMF B_z, and SW V_x) as input drivers with the addition of the SW density. The LSTM input structures are built following the process outlined in Section 2.2 of Licata and Mehta (2023).

2.6.1 Hyperparameter Tuner

We implement a hyperparameter tuner to identify suitable LSTM architectures using TensorFlow's (Abadi et al., 2015) API and Keras Tuner (O'Malley et al., 2019). Normally, each layer of a neural network is configured with a set of specific settings (activation function, number of neurons, input shape, etc.). A systematic grid search of these settings is then performed that builds many different combinations to train and test. Instead, a hyperparameter tuner not only automates this grid search but also applies an optimization scheme to determine an optimal set of hyperparameters (Goodfellow et al., 2016; O'Malley et al., 2019). When developing a tuner, each setting of interest is instead replaced with a range of values that the tuner can search. We utilize a Bayesian Optimization (O'Malley et al., 2019) scheme, which begins by estimating distributions for each hyperparameter from the processed trials and computes expected distributions for the next trial (Snoek et al., 2012). A set of hyperparameters with the highest probability of improving the objective performance is then selected from each expected distribution (Snoek et al., 2012) to begin training the next trial. The method used in Keras Tuner begins with a random search of the hyperparameter space for a select number of initial trials to develop the hyperparameter distributions and then applies the Bayesian optimization scheme on the remaining trials. Our hyperparameter tuner is setup to perform 50 total trials, with the first 25 being a random grid search and the final 25 trials using the Bayesian Optimization scheme.

A summary of our hyperparameter tuner's configuration is shown in Table 3. Normally, datasets with a large number of samples, or timesteps in our case, are split into smaller batches (Wilson & Martinez, 2003; Montavon et al., 2012). We split our datasets by cutting each simulation in half. Splitting the data into batches also allows for the order in which the batches are trained to be shuffled during each epoch of training. This batch shuffling has the added benefit of better generalizing a model (Montavon et al., 2012; Goodfellow et al., 2016; Licata & Mehta, 2023). Splitting the data into batches, however, has the drawback of truncating additional data because each batch requires a lookback period of a few timesteps to predict the initial epoch. Our hyperparameter tuner is also set to perform 2 separate executions per trial to help mitigate any potential performance degradation from the weight initialization (O'Malley et al., 2019; Licata & Mehta, 2023). This increases the tuner's overall runtime but is a much more robust configuration. Lastly, a callback to terminate the training of any individual model if a loss of NaN is returned is used as a precautionary measure to mitigate the effects of exploding gradients (Goodfellow et al., 2016).

Table 3. Hyperparameter Tuner Configuration.

Setting	**Choice**
Scheme	Bayesian Optimization
Total Trials	50
Initial Search	25
Repeats per Trial	2
Epochs per Trial	50
Shuffle Batches	Yes
Termination	NaN
Loss Metric	MSE
Minimization Parameter	Validation MSE

A summary of the hyperparameter space is shown in Table 4. To start, we include hyperparameters that determine how deep the neural network can go by choosing the number of LSTM and fully-connected, or dense, layers to include in the architecture for each trial. Each of these layers then has its own set of hyperparameters from which to choose from. Immediately following each dense layer is a dropout layer, which randomly shuts off neurons to help generalize a model by encouraging connections to take different paths (G. E. Hinton et al., 2012). The choice of an optimizer is also treated as a hyperparameter, where the tuner is given choices of: AdaGrad (Duchi et al., 2011), RMSprop (G. Hinton et al., 2012), AdaDelta (Zeiler, 2012), and Adam (Kingma & Ba, 2014). To end, we include a custom hyperparameter to determine the LSTM's lookback period because the inner magnetosphere's responses have varying lag times with each of the solar wind drivers (Bargatze et al., 1985; Maggiolo et al., 2017; Stumpo et al., 2020). This presented an additional challenge in that the LSTM's input shape needs to be changed for each trial of the hyperparameter tuner.

2.6.2 LSTM Training

During training, an LSTM typically make predictions using the true values of both the input drivers and state outputs, or reduced-space coefficients in our case. The true state outputs are available because the training, validation, and test datasets are all predetermined from the simulations. This evaluation method of using the true input drivers and state outputs to predict each timestep is known as a one-step prediction method. In operations, however, the true state output is not always available. When forecasting, the predicted state outputs are instead used to predict future timesteps, as outlined in

Table 4. Hyperparameter Space.

	Hyperparameter	**Range**
Architecture:		
	No. of LSTM Layers	[1, 2]
	No. of Dense Layers	[1, 3]
	Lookback Period	[3, 24]
	Optimizer	AdaGrad, AdaDelta, RMSProp, Adam
LSTM Layer:		
	Neurons	[32, 300]
	Activation Func.	Tanh, Sigmoid, SoftSign
Dense Layer:		
	Neurons	[64, 600]
	Activation Func.	ReLu, Elu, Sigmoid, SoftSign, SoftPlus
Dropout Layer:		
	Dropout Rate	[0.01, 0.50]

Figure 3 of Licata and Mehta (2023). After the current timestep t is predicted, the lookbacks are marched forward for the next timestep t+1. The corresponding lookback for t is then updated with the predicted output. The next timestep t+1 can then be predicted, and the lookbacks are again marched forward for the following timestep $t+2$. Now, any lookbacks corresponding to the previous two timesteps are updated with their respective predictions. This process is repeated for the length of the forecast window. This evaluation method is known as a dynamic prediction and is one of the advantages gained by developing an emulator.

Our hyperparameter tuner is implemented with a fixed number of epochs so that it can search the entire hyperparameter space in a reasonable amount of time. This, however, does not guarantee that these models have converged, so we included optimizers in the tuner that utilize momentum (Goodfellow et al., 2016; Montavon et al., 2012), which helps mitigate the effects of local minima in the loss function. The top architectures identified by the tuner are then put through a more rigorous training. Each of these architectures is allowed to reach a maximum of 1,000,000 epochs, but this value does not have to be reached because an early stopping (Goodfellow et al., 2016) callback with a patience period (Montavon et al., 2012) was implemented to prevent any overfitting. This is a much more robust training but requires additional computational resources and time, which is why it was not implemented in the hyperparameter tuner.

2.7 Model Ensemble

Our emulator implements a model ensemble to not only provide an uncertainty estimate but also increase overall model performance. An ensemble of models typically outperforms a single model (Weigel et al., 2008; Kioutsioukis & Galmarini, 2014; Xiao et al., 2018; S. Morley et al., 2018; Elvidge et al., 2016, 2023) due to the fact that a diverse set of models will normally contain individual models that predict certain portions of the training data better than others. Combining models in a way that emphasizes the best performing model will ultimately increase performance. Since the predictions of the LSTM models from the hyperparameter tuner are deterministic, a model ensemble provides the ability to compute statistics from multiple models to determine an error distribution.

To encourage diversity in our model ensemble, 5 separate instances of the top 5 architectures are trained from scratch, providing an ensemble of 25 models. This increase in the number of architectures is an enhancement to the method developed by Licata and Mehta (2023). Models trained with the same architectures will differ because the weight initialization is random, dropout is included, and the batches are shuffled during training (Goodfellow et al., 2016; Montavon et al., 2012). This provides confidence that models within an architecture contain enough diversity and statistics to determine an error distribution. Also, the top models from a hyperparameter tuner are normally identified by their performance on the validation dataset, which in our case is the MSE of the reduced-space temporal coefficients. Instead, we determine the tuner's top architectures by analyzing the validation dataset's performance using the physical-space metric (MdSA), which may not yield the same results.

2.8 Uncertainty Quantification

The emulator's last step is to combine the ensemble of deterministic models into a single probabilistic model, where we leverage the 3-tier hierarchical approach of Licata and Mehta (2023) to produce a robust and reliable uncertainty estimate. Multi-model ensembles have a history of applying a 2-tier weighted average method to combine models (Sewell, 2008; Huang et al., 2009; D. S. Wilks, 2011; Elvidge et al., 2016, 2023), but Licata and Mehta (2023) adds another tier to the method while also computing a variance. To begin, each of the 25 models must be evaluated over the training dataset using a dynamic prediction. For better interpretability, the indexes in the next sections have the following definitions: i refers to the architecture, j refers to the individual model within an architecture, k refers to the reduced-space coefficient's index, and t refers to the timestep from the above training dataset evaluation. As stated in Section 2.4, the central tendency metric (mean vs median) used in the UQ calculations varies depending on the dataset. The RAM-SCB dynamic predictions have a small number of timesteps with large errors (see Figure 8), considered to be outliers, which justifies the use of the median statistic whenever an average is taken over the temporal dimension (t). Implementing the median statistic instead of the mean is another modification made to the method developed by Licata and Mehta (2023).

Combining models with a weighted average is more robust than taking a simple average because the weights can be computed to place more emphasis on predictions with a higher accuracy. In Equation 7 (right), the median absolute error (MdAE) is taken over t for each individual model's evaluation and inverted to place more weight on models that have the least error. These weights $\widetilde{w}_{i,j,k}$ are then normalized within each architecture using Equation 7 (left) so that the combination can be calculated as a simple weighted sum.

$$w_{i,j,k} = \frac{\widetilde{w}_{i,j,k}}{\sum_j \widetilde{w}_{i,j,k}} \quad \text{where} \quad \widetilde{w}_{i,j,k} = \frac{1}{\text{MdAE}_{i,j,k}} \tag{7}$$

The resulting weights $w_{i,j,k}$ are then used to calculate the mean prediction and variance for each architecture, creating the 2nd tier of this hierarchical ensemble method. This is done by performing a weighted sum over the individual models within an architecture as shown in Equation 8. In these equations, $\hat{\alpha}_{i,j,k,t}$ are the dynamic predictions from each individual model, $\hat{\alpha}_{i,k,t}$ is the mean prediction for each architecture, and $\hat{\sigma}^2_{i,k,t}$ is each architecture's estimated variance.

$$\hat{\alpha}_{i,k,t} = \sum_j w_{i,j,k}\, \hat{\alpha}_{i,j,k,t} \quad \text{and} \quad \hat{\sigma}^2_{i,k,t} = \sum_j w_{i,j,k}\, \left(\hat{\alpha}_{i,k,t} - \hat{\alpha}_{i,j,k,t}\right)^2 \tag{8}$$

This variance calculation assumes a Gaussian distribution for each architecture, but combining these distributions to develop the final emulator's uncertainty estimate may not end up Gaussian. This is because each architecture's mean and variance may differ, meaning their distributions will not necessarily be independent or uncorrelated with each other, resulting in a non-Gaussian distribution. A visual depiction of this can be found in Figure 7 of Soltanzadeh et al. (2011), which shows the resulting non-Gaussian probability density function (PDF) from a Bayesian model averaging (BMA) ensemble. To provide a more robust and reliable UQ, Licata and Mehta (2023) apply a scaling factor to the uncertainty, called σ-scaling (Laves et al., 2021). The concept behind σ-scaling is to inflate the variance whenever predictions within an architecture are very precise but not accurate. Equation 9 shows how to calculate the σ-scaling factor, $S_{i,k}$, for each architecture and reduced-space coefficient, where $\alpha_{k,t}$ is the training dataset's ground truth (i.e., from the original simulations). This is another deviation from Licata and Mehta (2023) in that we use the median statistic instead of the mean to calculate our scaling factors. Laves et al. (2021) also developed $S_{i,k}$ to be applied to the standard deviation (i.e. σ), but we instead apply $S_{i,k}^2$ to each architecture's variance $\hat{\sigma}_{i,k,t}^2$.

$$S_{i,k} = \sqrt{\text{Median}\left[\frac{\left(\alpha_{k,t} - \hat{\alpha}_{i,k,t}\right)^2}{\hat{\sigma}_{i,k,t}^2}\right]} \tag{9}$$

The mean and variance estimates from each architecture are then combined to determine the ensemble's overall mean $\hat{\alpha}_{k,t}$ and variance $\hat{\sigma}_{k,t}^2$, which define the emulator's probabilistic output. This is also the 3rd and final tier of the hierarchical ensemble method. The calculations are shown in Equation 10, where n_i is the number of architectures, $\hat{\alpha}_{i,k,t}$ is each architecture's mean prediction, and $\hat{\sigma}_{i,k,t}^2$ is the variance estimate for each architecture with the σ-scaling factor already applied. A simple average is used here because this combination is conducted on the 2nd tier of the hierarchical ensemble. Licata and Mehta (2023) demonstrate that if the same number of models are trained within each architecture then the pooled variance calculation simplifies to a simple average. The result is referred to as a probabilistic output because of the included error distribution from the uncertainty estimate.

$$\hat{\alpha}_{k,t} = \frac{1}{n_i}\sum_i \hat{\alpha}_{i,k,t} \quad \text{and} \quad \hat{\sigma}_{k,t}^2 = \frac{1}{n_i}\sum_i \hat{\sigma}_{i,k,t}^2 \tag{10}$$

3 Results

3.1 Hyperparameter Tuner

Keras Tuner typically lists the best models in descending order by the defined metric on the validation dataset. Since our MSE is in the reduced space, the hyperparameter tuner's best models are instead listed in terms of the physical-space metric, MdSA, and shown in Table 5. As seen in the test and validation metrics, most all errors hover around 5% MdSA with biases under ±1% SSPB after only 50 epochs of training, but there is still a bit of diversity seen in these values. This diversity is important when identifying which architectures to include in the model ensemble because equal performance in very similar architectures would not benefit the ensemble. With this said, the top 5 architectures in this table were selected to develop our model ensemble. The tuner settled on a shallow architecture, where all of the top 10 architectures had only 1 LSTM and 1 dense layer. Only the Best Model #2 differed by having 2 dense layers. Each also used the AdaGrad optimizer with a lookback period of 3 timesteps, or 30 minutes of simu-

lation time. These hyperparameters may seem like these architectures are extremely similar, but this is merely a summary of the entire hyperparameter space (see Section 2.6.1). Overall, the performances shown in Table 5 provides strong support that the training data sufficiently sampled the event space to capture the dynamics found in RAM-SCB's particle flux data product. It is also important to note that these metrics are derived from a one-step prediction and not the dynamic prediction, or forecast evaluation method, used for the performance metrics in the next sections.

Table 5. Top 10 LSTM Architectures from the Hyperparameter Tuner.

Best Model	TRNG MSE	TRNG MdSA	TRNG SSPB	VAL MSE	VAL MdSA	VAL SSPB	TEST MSE	TEST MdSA	TEST SSPB
1	0.159	4.22%	0.57%	0.380	5.29%	0.44%	0.269	5.05%	0.35%
2	0.156	4.24%	0.37%	0.398	5.33%	0.69%	0.278	5.23%	0.88%
3	0.150	4.14%	-0.02%	0.395	5.49%	-0.63%	0.274	4.91%	-0.22%
4	0.156	4.26%	-0.36%	0.367	5.58%	-0.61%	0.260	4.85%	0.54%
5	0.187	4.10%	0.73%	0.273	5.59%	0.06%	0.207	5.30%	0.46%
6	0.166	4.29%	-0.03%	0.408	5.72%	-1.54%	0.275	4.96%	-0.32%
7	0.168	4.40%	0.04%	0.394	5.74%	-0.39%	0.276	5.51%	-0.14%
8	0.190	4.32%	-0.11%	0.288	5.80%	-0.58%	0.213	5.42%	0.79%
9	0.206	4.66%	0.70%	0.348	5.91%	0.31%	0.240	5.88%	1.88%
10	0.205	5.15%	0.13%	0.328	6.35%	-0.22%	0.241	6.67%	1.17%

3.2 Dynamic Prediction

Based on a detailed analysis, we found relatively high errors during the initial few hours of each simulation. Figure 6 shows the relative frequency of errors across all 20 training simulations. The simulation time is binned every hour (6 timesteps) while the errors are binned every 10% MdSA. Figure 6 is interpreted as a histogram, where the errors for every hour of each simulation are binned and presented as a percentage. Verified by the mean MdSA in Figure 8, the relative frequencies of low errors (i.e. <10%) are the dominant trend seen in dark blue (Figure 6). The inlay, however, highlights a shorter trend of errors in the initial few hours. A more in-depth look at the input drivers (SYM-H, AL, and B_z) during the onset of each simulation showed that not all parameters began at quiescent levels. This meant that each simulation's initialization, or spin-up, period was set with heightened activity, which is known to affect the simulation results. Since the input drivers of each simulation varied in activity level, the spin-up periods ultimately differed across all simulations, so a simple cutoff time could not be determined. The individual energy channels within each simulation are also expected to have varying spin-up times, so we decided to use this finding as a lesson learned for running large-scale physics-based simulation models such as RAM-SCB. Future work from this project will incorporate a more robust initialization period that allows each simulation to reach a steady state before the event of interest begins. Of course, these initialization periods will not be included when creating the training, validation, and test dataset, but it should mitigate the errors seen in the initial few hours of Figure 6.

As stated in Section 2.7, the top 5 architectures identified by the hyperparameter tuner are processed through a more rigorous training and evaluated using a dynamic prediction. An hourly forecast window was chosen for the dynamic prediction because it seemed natural to forecast double the lookback period. Figure 7 shows the errors of the dynamic prediction evaluation for the TRNG 5 simulation (see Table 1) using the tuner's best model. The SYM-H and IMF B_z drivers are included below the error plot to visually check for correlations between increased errors and heightened activity levels. The errors in Fig-

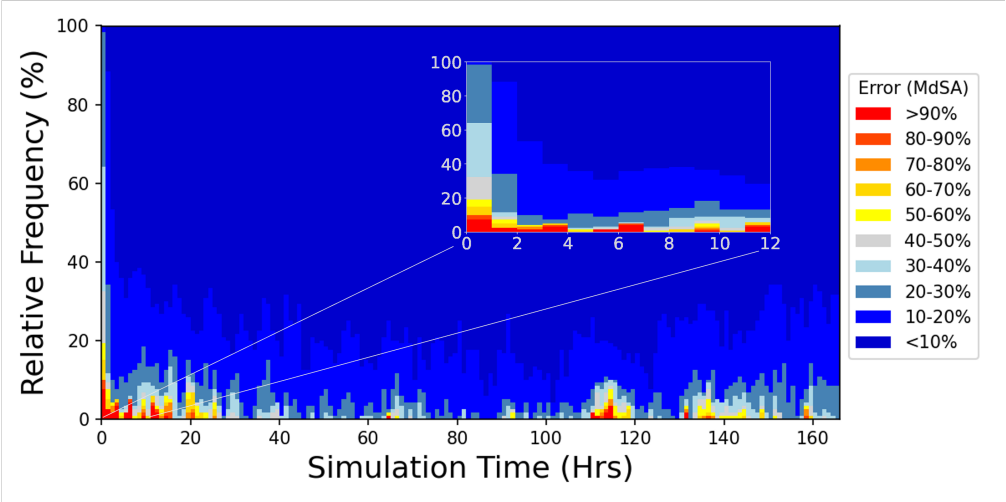

Figure 6. Relative error histogram of dynamic prediction errors from all 20 training simulations. The simulation time is binned hourly, while the errors are binned every 10% MdSA. The inlay highlights the relative high errors seen at the onset of each simulation.

ure 7, visually, almost directly coincide with heightened activity in each of the drivers, which is expected. This LSTM model was able to dynamically predict this week-long simulation in just 22 seconds with a mean MdSA less than 8%, even though the peak error just before the 400^{th} timestep reaches a factor of 2. This mean MdSA error is an average over the simulation period where the reconstructed MdSA is determined at each timestep. The threshold for errors reaching a factor of 2 is important because Boyd et al. (2019) shows that even instruments on the same spacecraft can have flux values of the inner magnetosphere that disagree by a factor of 2. The quartiles (25%, 50%, 75%) for this simulation came out to 3.37%, 5.10%, and 8.82% MdSA, respectively.

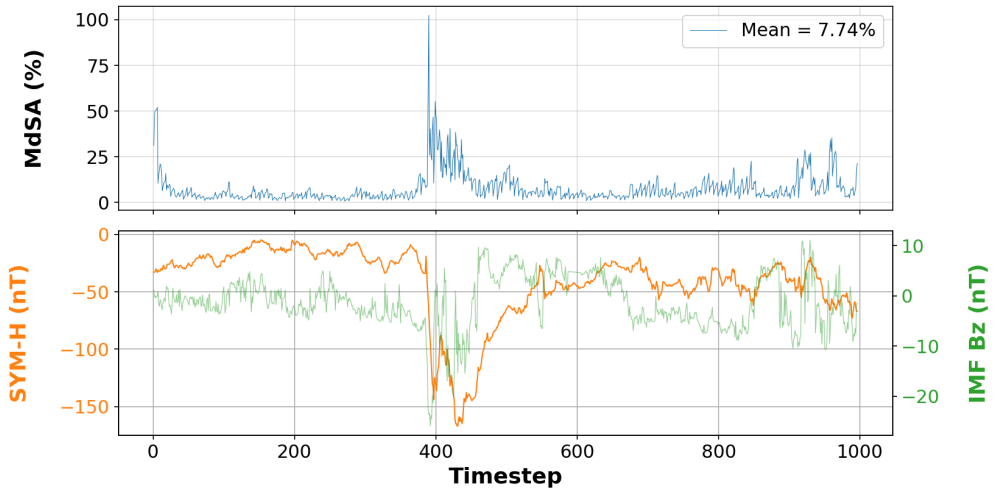

Figure 7. Hourly dynamic prediction results of the TRNG 5 simulation from the LSTM tuner's best model. Reconstructed errors (blue) in the physical space (MdSA) are plotted along with the SYM-H index (orange) and IMF B_z component (green).

Similarly, Figure 8 shows the results of the dynamic prediction evaluation for all 20 training simulations using the hyperparameter tuner's best model. This LSTM model was able to dynamically predict all 20 week-long simulations in approx. 7 minutes with a mean MdSA of 8.5%. This error value is an average over the entire training dataset, where the MdSA is determined from the reconstructed fluxes for each timestep of every simulation. The quartiles (25%, 50%, 75%) came out to 3.57%, 5.66%, and 9.50% MdSA, respectively. This means that more than 75% of the errors in this entire dataset have less than 10% MdSA. As in the single simulation results, Figure 8 has timesteps in which the MdSA peaks during heightened activity levels throughout the various simulations. For instance, errors around 100%, or a factor of 2, can be seen in Simulations 4, 8, 12, 13, and 17. Errors upward of 200% (factors of 3, 4, and 5) can be seen in Simulations 3, 7, 15, 16, and 17. These error spikes must be put into context, though, as Boyd et al. (2019) has shown that even instruments on the same spacecraft can have flux values that disagree by a factor of 2. The SYM-H index and IMF B_z are also plotted directly below the errors to determine if these error spikes visually coincide with heightened activity levels. The largest errors do coincide with the deepest SYM-H depressions, which indicate significant levels of geomagnetic activity. The IMF's B_z component fluctuations line up with the lower error regions (i.e. $< 100\%$ MdSA), although its amplitude ranges on a much smaller scale than that of SYM-H.

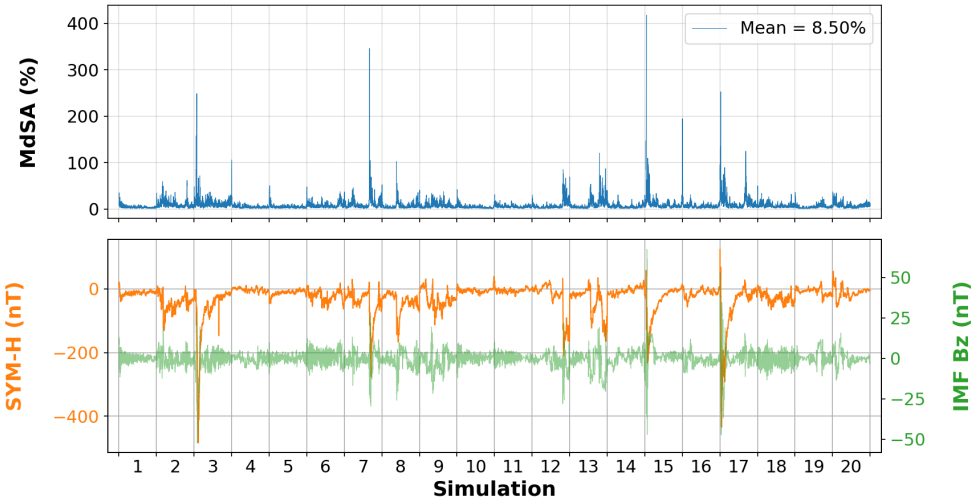

Figure 8. Hourly dynamic prediction results for all 20 training simulations, each one block on the bottom axis, from the LSTM tuner's best model. Reconstructed errors (blue) in the physical space (MdSA) are plotted along with the SYM-H index (orange) and IMF B_z component (green).

3.3 Reduced-Order Probabilistic Emulator

As stated in Section 2.7, a model ensemble is leveraged to create a probabilistic output from a system of deterministic models with the added benefit that an ensemble typically outperforms a single model (Weigel et al., 2008; Kioutsioukis & Galmarini, 2014; Xiao et al., 2018; S. Morley et al., 2018; Elvidge et al., 2016, 2023). The 3-tier hierarchical approach of first combining models within an architecture via a weighted average and then combining the various architectures though a simple mean provides this work's final product, a reduced-order probabilistic emulator (ROPE) of RAM-SCB particle flux.

A summary of our ROPE's final performance metrics are shown in Table 6, where it has an average MdSA of roughly 10% with biases just under 2% SSPB using an hourly

dynamic prediction on both the validation and test datasets. As expected, the model ensemble outperformed the best individual model by a whole percentage point, which is a significant performance bump given the level of accuracy in the ensemble members (see Table 5). The biases stayed about the same between 1-2% SSPB. The ROPE's training, validation, and test quartiles (25%, 50%, 75%) came out to (3.19%, 5.12%, and 9.01%), (3.88%, 6.84%, and 12.25%), and (3.28%, 5.55%, and 10.51%) MdSA, respectively.

Table 6. Hourly dynamic prediction results for both the best individual model (deterministic) and final probabilistic emulator (ROPE) over each of the ML datasets.

		TRNG	VAL	TEST
Indiv. Model:				
	Dyn. Pred. (MdSA)	8.50%	11.44%	11.32%
	Model Bias (SSPB)	-1.80%	1.36%	-1.26%
ROPE:				
	Dyn. Pred. (MdSA)	7.60%	10.34%	10.36%
	Model Bias (SSPB)	-1.53%	-1.97%	-1.80%
	Calibration (CES)	8.97%	7.61%	7.15%

Each of the 25 LSTMs in the model ensemble are evaluated using a dynamic prediction. Running them in parallel took just 110 seconds to predict the 5 simulations found in each of the validation and test datasets. Similarly, running these 5 simulations in RAM-SCB using the same configuration and computational resources as in Section 2.2, also in parallel, takes roughly 38.2 hours. This results in a speed increase of 1,250x between the emulator and RAM-SCB, which highlights the efficiency gained by developing an emulator. The ROPE's predictions (i.e ensemble's combined hourly dynamic predictions) on the TST 3 simulation (see Table 2) are shown in Figure 9 with 2-σ bounds. Upon visual inspection, the first 2 reduced-order coefficients express good agreement with the truth values. Since the PCA coefficients are numbered in descending order, having the best performance in the first few coefficients is ideal, so these are very promising results.

Since our variance calculation assumes a Gaussian distribution (see Equation 8), we expect that approx. 95% of the ROPE's predictions will fall within the 2-σ bounds. The actual observed percentages for the first 2 coefficients (shown in Figure 9) are 93.5% and 92.8%, respectively. This is a slight underestimation of the variance and only a few percentage points off, implying these uncertainty estimates are indeed well-calibrated. Figure 10 demonstrates that the uncertainty is mostly underestimated for the remaining coefficients. The CES for each dataset is provided in Table 6, with scores less than 10%. These scores are interpreted as the emulator's reduced-space predictions have error distributions that deviate less than 10%, on average, from a normal distribution.

Lastly, Figure 11 depicts the evolution of the particle flux predicted by our ROPE through the TST 1 simulation, similar to Figure 5. The before and after storm predictions show a high degree of resemblance between the true and predicted fluxes, with errors of 3.8% and 6.0% MdSA, respectively. These errors are on the order of the truncation error introduced by the PCA decomposition, demonstrating good performance. During the storm, however, local errors climb past 500%, which is expected but still relatively large even given the fact that this is evaluated using a dynamic prediction. The quartiles (25%, 50%, 75%) during the storm came out with errors of 12.68%, 27.73%, and 52.37%, respectively. This translates to 3 out of every 4 flux values, on average, will have an error less than 53% during a storm period where errors are expected to be high, which is well within the threshold of a factor of 2 (Boyd et al., 2019).

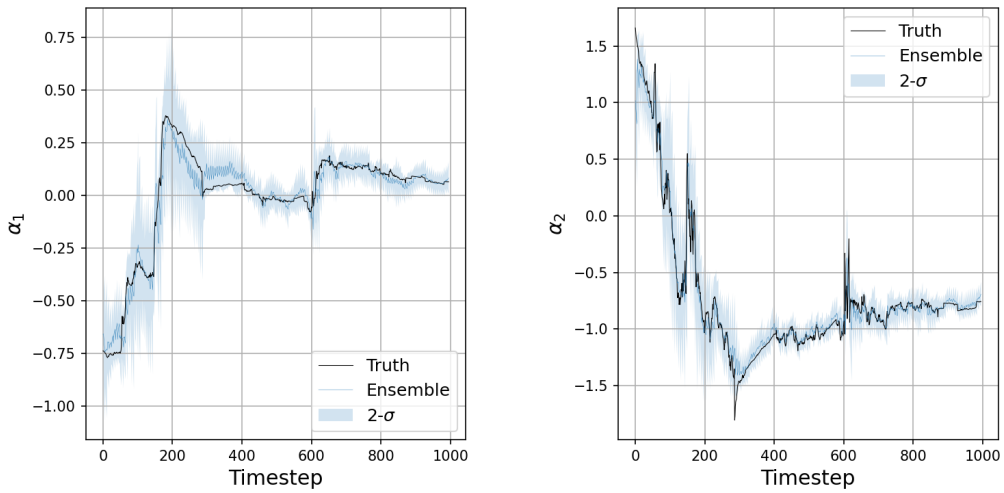

Figure 9. Hourly dynamic predictions of the first 2 reduced-space coefficients (α_1 & α_2) by the ROPE on the TST 3 simulation. The prediction (blue) is plotted at each timestep along with the truth (black) and 2-σ bounds (light blue).

4 Limitations and Future Work

The goal of this work is to apply the emulator workflow (Licata & Mehta, 2023) to the ring current by demonstrating it on a smaller subset of RAM-SCB particle flux, which in this case is a single energy channel of omnidirectional flux. This is our greatest limitation but was chosen to build a solid foundation. Thus, subsequent work will expand this workflow to encompass the full energy spectrum and pitch angle distribution found in the particle flux data product.

The use of a linear PCA to reduce the system's dimensionality is another limitation in this work. Expanding to incorporate RAM-SCB's full energy spectrum will require the dimensionality reduction to explore nonlinear techniques and ML methods such as a kernel PCA (k-PCA) or convolutional autoencoder (CAE). Since it is known that this region of the inner magnetosphere contains nonlinear dynamics (Daglis et al., 1999), a nonlinear dimensional reduction will also aid in capturing these dynamics. This can help mitigate the large error spikes seen during periods of heightened solar and geomagnetic activity in this work, which partially stems from the use of a linear PCA method for the dimensionality reduction.

The hierarchical ensemble methodology is still a relatively novel approach for creating probabilistic predictions. There is much to be explored and room for more improvements. Even though the first 2 reduced-space coefficients contained roughly 93% of the ground truth values in their 2-σ bounds, the uncertainties of the other coefficients were all underestimated. Our calibration curves are also under a Gaussian assumption, so measuring the reliability under non-Gaussian distributions will require further investigation. Exploring a debiasing or more sophisticated ensemble method (e.g Elvidge et al. (2023)) may potentially improve the UQ's performance. The emulation process also leveraged reduced-order modeling to facilitate future data assimilation applications. This can enhance the workflow by assimilating an observable, such as the Dst index, back into the emulator to further calibrate it.

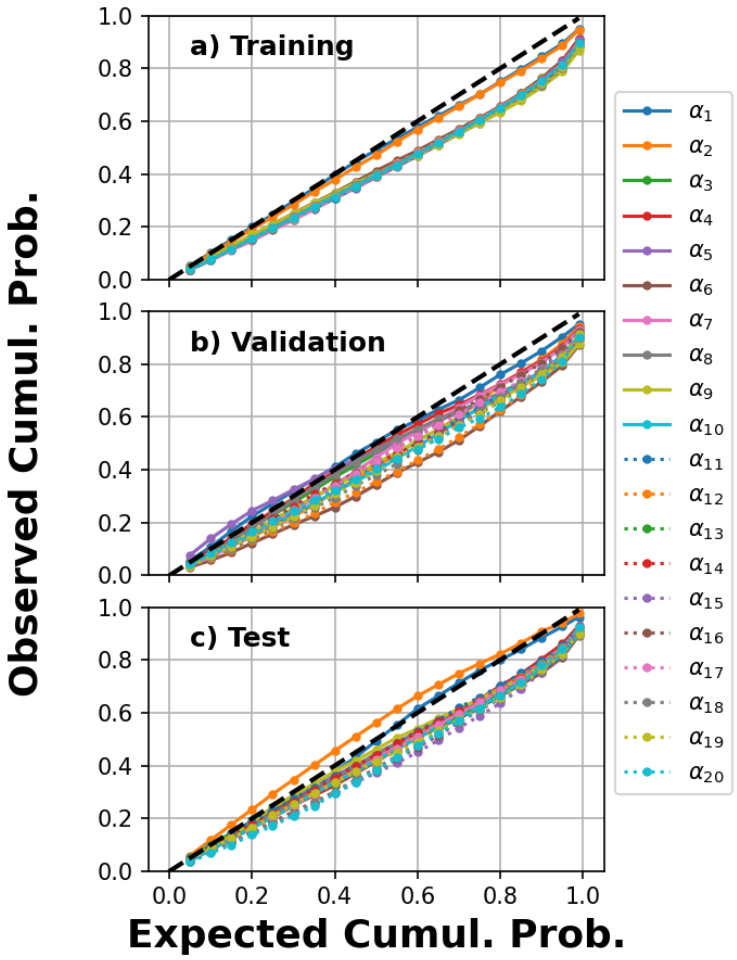

Figure 10. ROPE's calibration curves for the ML datasets (training, validation, and test). Each reduced-space coefficient has its own curve, where the first 10 are plotted in solid lines and the remaining 10 have dotted lines. The black dashed line represents the perfectly calibrated $y = x$ line of the Gaussian assumption.

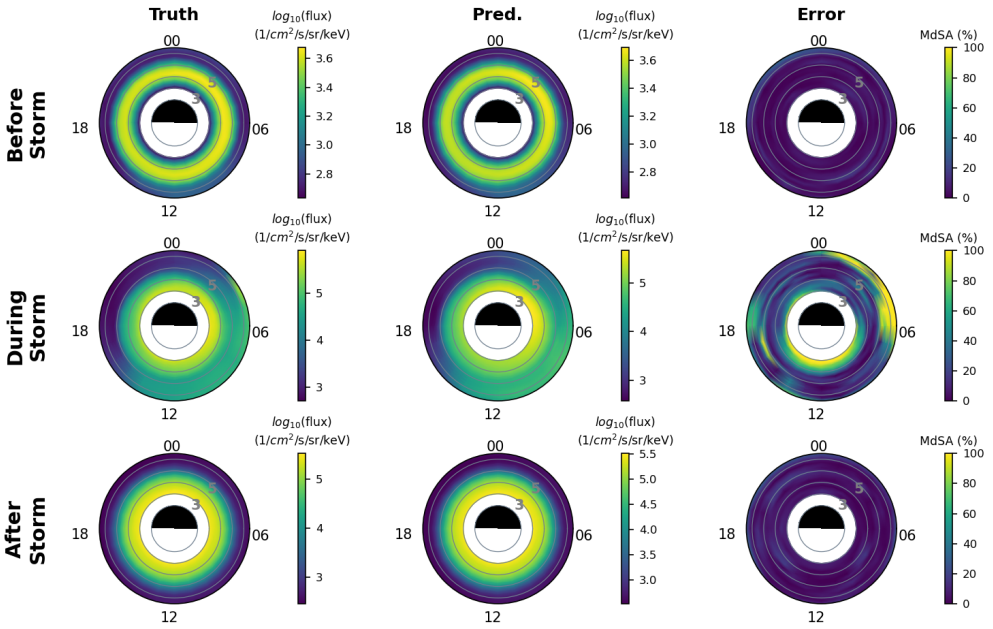

Figure 11. Snapshots taken before, during, and after the geomagnetic storm in the TST 1 simulation with the prediction errors (right) between the actual (left) and reconstructed ROPE hourly dynamic predictions (middle), plotted on RAM-SCB's grid.

5 Summary

This work builds upon the emulation process developed by Licata and Mehta (2023), but now applied to ring current dynamics, and creates a reduced-order probabilistic emulator of the RAM-SCB particle flux data product from the ground up. The resulting ROPE is the culmination of 25 independent LSTM models that are trained on 20 one-week-long simulations from RAM-SCB, where a hierarchical ensemble blends these deterministic LSTMs together into a probabilistic prediction with a robust and reliable uncertainty estimate. The simulations that make up the training, validation, and test datasets are all derived from a novel approach of sampling over 20 years of solar and geomagnetic activity that were transformed into reduced-space representations by a PCA decomposition.

Metrics showcasing low errors throughout each step of the emulation process demonstrate the effectiveness of this workflow. The hyperparameter tuner's performance metrics of roughly 5% MdSA over all ML datasets, evaluated using a one-step prediction, provides significant confidence that the event space was sufficiently sampled. However, more consideration is needed when initializing the simulations to obtain better results. The low truncation error from the PCA of 2.9% MdSA demonstrates its robustness in reducing the dimensionality of this system, although fluxes of H^+ at higher energies (i.e. 208 keV) are undoubtedly easier to capture with PCA than lower energies (e.g. 1-10 keV). The lookback period, number of LSTM layers, and number of dense layers from the hyperparameter tuner results were all lower than expected, but this may have been an artifact from modeling a smaller subset of the RAM-SCB particle flux data product. Once expanded to the full energy spectrum and pitch angle distribution, we expect the hyperparameter tuner to provide a much more diverse set of architectures. The model ensemble is a relatively modern approach for determining the uncertainty of LSTM models and still a novel concept for the ring current, so there is much to be learned and tested from

the ensemble method. Our emulator provides a speed increase of 1,250x over RAM-SCB with an overall accuracy of roughly 10% MdSA using an hourly dynamic prediction.

6 Open Research

The OMNIWeb data used in this paper can be downloaded at `https://omniweb.gsfc.nasa.gov/form/omni_min.html`. The RAM-SCB source code (Jordanova, Engel, et al., 2022) can be found at `https://github.com/lanl/RAM-SCB/`, and the version used in this work was tagged *v.2.1.1*. Both TensorFlow (Abadi et al., 2015) and Keras Tuner (O'Malley et al., 2019) were downloaded using Anaconda (*Anaconda Software Distribution*, 2020). The input files for the RAM-SCB simulations, ML datasets, and code to run ROPE are available at `https://zenodo.org/record/8313973` (Cruz et al., 2023).

Acknowledgments

PMM and AAC would like to acknowledge support from NSF Grant #1929127 and DoE Grant #DE-SC0020294. This research has also been made possible by NASA's West Virginia Space Grant Consortium (WVSGC) Grant #80NSSC20M0055. The authors acknowledge the use of the Thorny Flat HPC cluster which was funded in part by NSF MRI Award #1726534 and WVU. Special thanks to Dr. Guillermo Avendaño-Franco from WVU's Research Computing Department for helping configure a proper system environment to run our RAM-SCB simulations.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... Zheng, X. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems.* Retrieved from `https://www.tensorflow.org/` (Software available from tensorflow.org)

Anaconda software distribution. (2020). Anaconda Inc. Retrieved from `https://docs.anaconda.com/`

Bargatze, L. F., Baker, D., McPherron, R., & Hones Jr, E. W. (1985). Magnetospheric impulse response for many levels of geomagnetic activity. *Journal of Geophysical Research: Space Physics*, *90*(A7), 6387–6394.

Bjornsson, H., & Venegas, S. (1997). *A manual for eof and svd analyses of climate data.* (techreport No. 97-1). Montreal, Quebec.: McGill University.

Bourdarie, S., Blake, B., Cao, J., Friedel, R., Miyoshi, Y., Panasyuk, M., & Underwood, C. (2012). Standard file format guidelines for particle fluxes. [Computer software manual].

Boyd, A. J., Reeves, G. D., Spence, H. E., Funsten, H. O., Larsen, B. A., Skoug, R. M., ... Jaynes, A. N. (2019, nov). RBSP-ECT combined spin-averaged electron flux data product. *Journal of Geophysical Research: Space Physics*, *124*(11), 9124–9136. doi: 10.1029/2019ja026733

Cruz, A. A., Mehta, P. M., Morley, S. K., Godinez, H. C., & Jordanova, V. K. (2023). *Ram-scb rope data and code.* Zenodo. doi: 10.5281/ZENODO.8147672

Daglis, I. A., Thorne, R. M., Baumjohann, W., & Orsini, S. (1999, nov). The terrestrial ring current: Origin, formation, and decay. *Reviews of Geophysics*, *37*(4), 407–438. doi: 10.1029/1999rg900009

Deutsch, J. L., & Deutsch, C. V. (2012, mar). Latin hypercube sampling with multidimensional uniformity. *Journal of Statistical Planning and Inference*, *142*(3), 763–772. doi: 10.1016/j.jspi.2011.09.016

Duchi, J. C., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, *12*, 2121-2159. Retrieved from `http://dblp.uni-trier.de/db/journals/jmlr/jmlr12.html#DuchiHS11`

Elvidge, S., Godinez, H. C., & Angling, M. J. (2016, jul). Improved forecasting of thermospheric densities using multi-model ensembles. *Geoscientific Model Development*, *9*(6), 2279–2292. doi: 10.5194/gmd-9-2279-2016

Elvidge, S., Granados, S., Angling, M., Brown, M., Themens, D., & Wood, A. (2023). Multi-model ensembles for upper atmosphere models. *Space Weather*, *21*(3), e2022SW003356.

Engel, M. A., Morley, S. K., Henderson, M. G., Jordanova, V. K., Woodroffe, J. R., & Mahfuz, R. (2019, jun). Improved simulations of the inner magnetosphere during high geomagnetic activity with the RAM-SCB model. *Journal of Geophysical Research: Space Physics*, *124*(6), 4233–4248. doi: 10.1029/2018ja026260

Fok, M.-C., Kang, S.-B., Ferradas, C. P., Buzulukova, N. Y., Glocer, A., & Komar, C. M. (2021, apr). New developments in the comprehensive inner magnetosphere-ionosphere model. *Journal of Geophysical Research: Space Physics*, *126*(4). doi: 10.1029/2020ja028987

Friedel, R., Reeves, G., & Obara, T. (2002). Relativistic electron dynamics in the inner magnetosphere—a review. *Journal of Atmospheric and Solar-Terrestrial Physics*, *64*(2), 265–282.

Ganushkina, N., Jaynes, A., & Liemohn, M. (2017, oct). Space weather effects produced by the ring current particles. *Space Science Reviews*, *212*(3-4), 1315–1344. doi: 10.1007/s11214-017-0412-2

Gers, F. A., Schraudolph, N. N., & Schmidhuber, J. (2002). Learning precise timing with lstm recurrent networks. *Journal of machine learning research*, *3*(Aug), 115–143.

Gondelach, D. J., & Linares, R. (2021). Real-time thermospheric density estimation via radar and gps tracking data assimilation. *Space Weather*, *19*(4), e2020SW002620.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. (`http://www.deeplearningbook.org`)

Green, J., Likar, J., & Shprits, Y. (2017). Impact of space weather on the satellite industry. *Space Weather*, *15*(6), 804–818.

Hinton, G., Srivastava, N., & Swersky, K. (2012). Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, *14*(8), 2.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). *Improving neural networks by preventing co-adaptation of feature detectors*. arXiv. doi: 10.48550/ARXIV.1207.0580

Hochreiter, S., & Schmidhuber, J. (1997, nov). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Huang, F., Xie, G., & Xiao, R. (2009). Research on ensemble learning. In *2009 international conference on artificial intelligence and computational intelligence* (Vol. 3, pp. 249–252).

Jordanova, V. K., Engel, M. A., Morley, S. K., Welling, D. T., Yu, Y., Yakymenko, K., ... Junghans, C. (2022). *Ram-scb*. Zenodo. doi: 10.5281/ZENODO.6977287

Jordanova, V. K., Miyoshi, Y. S., Zaharia, S., Thomsen, M. F., Reeves, G. D., Evans, D. S., ... Fennell, J. F. (2006, oct). Kinetic simulations of ring current evolution during the geospace environment modeling challenge events. *Journal of Geophysical Research*, *111*(A11). doi: 10.1029/2006ja011644

Jordanova, V. K., Morley, S. K., Engel, M. A., Godinez, H., Yakymenko, K., Henderson, M. G., ... Miyoshi, Y. (2022). The ram-scb model and its applications to advance space weather forecasting. *Advances in Space Research*.

Jordanova, V. K., Thorne, R. M., Li, W., & Miyoshi, Y. (2010, may). Excitation of whistler mode chorus from global ring current simulations. *Journal of Geophysical Research: Space Physics*, *115*(A5), n/a–n/a. doi:

10.1029/2009ja014810

Jordanova, V. K., Welling, D. T., Zaharia, S. G., Chen, L., & Thorne, R. M. (2012, may 16). Modeling ring current ion and electron dynamics and plasma instabilities during a high-speed stream driven storm. *Journal of Geophysical Research: Space Physics*, *117*(A9), n/a–n/a. doi: 10.1029/2011ja017433

Jordanova, V. K., Yu, Y., Niehof, J. T., Skoug, R. M., Reeves, G., Kletzing, C. A., ... Spence, H. E. (2014). Simulations of inner magnetosphere dynamics with an expanded ram-scb model and comparisons with van allen probes observations.. doi: 10.1002/

Jordanova, V. K., Zaharia, S., & Welling, D. T. (2010, dec). Comparative study of ring current development using empirical, dipolar, and self-consistent magnetic field simulations. *Journal of Geophysical Research: Space Physics*, *115*(A12), n/a–n/a. doi: 10.1029/2010ja015671

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kioutsioukis, I., & Galmarini, S. (2014). De praeceptis ferendis: good practice in multi-model ensembles. *Atmospheric Chemistry and Physics*, *14*(21), 11791–11815.

Koons, H., & Fennell, J. (2006). Space weather effects on communications satellites..

Koons, H., Mazur, J., Selesnick, R., Blake, J., & Fennell, J. (1999). *The impact of the space environment on space systems* (Tech. Rep.). El Segundo, CA: Aerospace Corp.

Kozyra, J., Borovsky, J., Chen, M., Fok, M.-C., & Jordanova, V. (1998). Plasma sheet preconditioning, enhanced convection and ring current development. *Substorms-4*, *238*, 755.

Kozyra, J., Liemohn, M., Clauer, C., Ridley, A., Thomsen, M., Borovsky, J., ... Gonzalez, W. (2002). Multistep dst development and ring current composition changes during the 4–6 june 1991 magnetic storm. *Journal of Geophysical Research: Space Physics*, *107*(A8), SMP–33.

Laves, M.-H., Ihler, S., Fast, J. F., Kahrs, L. A., & Ortmaier, T. (2021, April). Recalibration of aleatoric and epistemic regression uncertainty in medical imaging. *Journal of Machine Learning for Biomedical Imaging*, *Special Issue: Medical Imaging with Deep Learning (MIDL)*(008), 1-26. doi: 10.48550/ARXIV.2104.12376

Li, W., & Hudson, M. (2019, nov). Earth's van allen radiation belts: From discovery to the van allen probes era. *Journal of Geophysical Research: Space Physics*, *124*(11), 8319–8351. doi: 10.1029/2018ja025940

Licata, R. J., & Mehta, P. M. (2022, may). Uncertainty quantification techniques for data-driven space weather modeling: thermospheric density application. *Scientific Reports*, *12*(1). doi: 10.1038/s41598-022-11049-3

Licata, R. J., & Mehta, P. M. (2023, may). Reduced order probabilistic emulation for physics-based thermosphere models. *Space Weather*, *21*(5). doi: 10.1029/2022sw003345

Licata, R. J., Mehta, P. M., Tobiska, W. K., & Huzurbazar, S. (2022, apr). Machine-learned HASDM thermospheric mass density model with uncertainty quantification. *Space Weather*, *20*(4). doi: 10.1029/2021sw002915

Licata, R. J., Mehta, P. M., Weimer, D. R., Tobiska, W. K., & Yoshii, J. (2022, nov). MSIS-UQ: Calibrated and enhanced NRLMSIS 2.0 model with uncertainty quantification. *Space Weather*, *20*(11), e2022SW003267. doi: 10.1029/2022sw003267

Maggiolo, R., Hamrin, M., Keyser, J. D., Pitkänen, T., Cessateur, G., Gunell, H., & Maes, L. (2017, nov). The delayed time response of geomagnetic activity to the solar wind. *Journal of Geophysical Research: Space Physics*, *122*(11). doi: 10.1002/2016ja023793

Maulik, R., Rao, V., Wang, J., Mengaldo, G., Constantinescu, E., Lusch, B., ... Kotamarthi, R. (2022). Efficient high-dimensional variational data assimilation with machine-learned reduced-order models. *Geoscientific Model Development*, *15*(8), 3433–3445.

McGranaghan, R., Knipp, D. J., Matsuo, T., Godinez, H., Redmon, R. J., Solomon, S. C., & Morley, S. K. (2015). Modes of high-latitude auroral conductance variability derived from dmsp energetic electron precipitation observations: Empirical orthogonal function analysis. *Journal of Geophysical Research: Space Physics*, *120*(12), 11–013.

Mehta, P. M., & Linares, R. (2017, oct). A methodology for reduced order modeling and calibration of the upper atmosphere. *Space Weather*, *15*(10), 1270–1287. doi: 10.1002/2017sw001642

Mehta, P. M., & Linares, R. (2018). A new transformative framework for data assimilation and calibration of physical ionosphere-thermosphere models. *Space Weather*, *16*(8), 1086–1100.

Mehta, P. M., Linares, R., & Sutton, E. K. (2018, may). A quasi-physical dynamic reduced order model for thermospheric mass density via hermitian space-dynamic mode decomposition. *Space Weather*, *16*(5), 569–588. doi: 10.1029/2018sw001840

Montavon, G., Orr, G., & Müller, K.-R. (2012). *Neural networks: tricks of the trade* (Vol. 7700). springer.

Morley, S., & Lockwood, M. (2006). A numerical model of the ionospheric signatures of time-varying magnetic reconnection: Iii. quasi-instantaneous convection responses in the cowley-lockwood paradigm. In *Annales geophysicae* (Vol. 24, pp. 961–972).

Morley, S., Welling, D., & Woodroffe, J. (2018). Perturbed input ensemble modeling with the space weather modeling framework. *Space Weather*, *16*(9), 1330–1347.

Morley, S. K., Brito, T. V., & Welling, D. T. (2018, jan). Measures of model performance based on the log accuracy ratio. *Space Weather*, *16*(1), 69–88. doi: 10.1002/2017sw001669

O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., & Invernizzi, L. (2019). *Kerastuner*. `https://github.com/keras-team/keras-tuner`.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pulkkinen, T., Palmroth, M., Tanskanen, E., Ganushkina, N. Y., Shukhtina, M., & Dmitrieva, N. (2007). Solar wind—magnetosphere coupling: a review of recent results. *Journal of Atmospheric and Solar-Terrestrial Physics*, *69*(3), 256–264.

Russell, C. T., Luhmann, J. G., & Strangeway, R. J. (2016). *Space physics: An introduction*. Cambridge University Press.

Sewell, M. (2008). Ensemble learning. *RN*, *11*(02), 1–34.

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In F. Pereira, C. Burges, L. Bottou, & K. Weinberger (Eds.), *Advances in neural information processing systems* (Vol. 25). Curran Associates, Inc. Retrieved from `https://proceedings.neurips.cc/paper_files/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf`

Soltanzadeh, I., Azadi, M., & Vakili, G. A. (2011, jul). Using bayesian model averaging (BMA) to calibrate probabilistic surface temperature forecasts over iran. *Annales Geophysicae*, *29*(7), 1295–1303. doi: 10.5194/angeo-29-1295-2011

Spence, H. E., Kivelson, M. G., Walker, R. J., & McComas, D. J. (1989). Magnetospheric plasma pressures in the midnight meridian: Observations from 2.5 to 35 re. *Journal of Geophysical Research*, *94*(A5), 5264. doi: 10.1029/ja094ia05p05264

Stumpo, M., Consolini, G., Alberti, T., & Quattrociocchi, V. (2020, feb). Measuring information coupling between the solar wind and the magnetosphere–ionosphere system. *Entropy*, *22*(3), 276. doi: 10.3390/e22030276

Thorne, R. M. (2010). Radiation belt dynamics: The importance of wave-particle interactions. *Geophysical Research Letters*, *37*(22).

Wang, P., Chen, Z., Deng, X., Wang, J., Tang, R., Li, H., ... Wu, Z. (2022, mar). The prediction of storm-time thermospheric mass density by LSTM-based ensemble learning. *Space Weather*, *20*(3). doi: 10.1029/2021sw002950

Waskom, M. L. (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021. Retrieved from `https://doi.org/10.21105/joss.03021` doi: 10.21105/joss.03021

Węglarczyk, S. (2018). Kernel density estimation and its application. *ITM Web of Conferences*, *23*, 00037. doi: 10.1051/itmconf/20182300037

Weigel, A. P., Liniger, M., & Appenzeller, C. (2008). Can multi-model combination really enhance the prediction skill of probabilistic ensemble forecasts? *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, *134*(630), 241–260.

Wilks, D. (2011). Principal component (eof) analysis. In *International geophysics* (pp. 519–562). Elsevier. doi: 10.1016/b978-0-12-385022-5.00012-9

Wilks, D. S. (2011). *Statistical methods in the atmospheric sciences*. Elsevier.

Wilson, D., & Martinez, T. R. (2003, dec). The general inefficiency of batch training for gradient descent learning. *Neural Networks*, *16*(10), 1429–1451. doi: 10.1016/s0893-6080(03)00138-2

Xiao, Y., Wu, J., Lin, Z., & Zhao, X. (2018). A deep learning-based multi-model ensemble method for cancer prediction. *Computer methods and programs in biomedicine*, *153*, 1–9.

Yu, Y., Jordanova, V., Zaharia, S., Koller, J., Zhang, J., & Kistler, L. M. (2012, mar). Validation study of the magnetically self-consistent inner magnetosphere model RAM-SCB. *Journal of Geophysical Research: Space Physics*, *117*(A3), n/a–n/a. doi: 10.1029/2011ja017321

Yu, Y., Rastätter, L., Jordanova, V. K., Zheng, Y., Engel, M., Fok, M.-C., & Kuznetsova, M. M. (2019, feb). Initial results from the GEM challenge on the spacecraft surface charging environment. *Space Weather*, *17*(2), 299–312. doi: 10.1029/2018sw002031

Zaharia, S., Jordanova, V. K., Thomsen, M. F., & Reeves, G. D. (2006, oct). Self-consistent modeling of magnetic fields and plasmas in the inner magnetosphere: Application to a geomagnetic storm. *Journal of Geophysical Research*, *111*(A11). doi: 10.1029/2006ja011619

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zheng, Y., Ganushkina, N. Y., Jiggens, P., Jun, I., Meier, M., Minow, J. I., ... Kuznetsova, M. M. (2019, oct). Space radiation and plasma effects on satellites and aviation: Quantities and metrics for tracking performance of space weather environment models. *Space Weather*, *17*(10), 1384–1403. doi: 10.1029/2018sw002042