



Cruz, Alfredo A, Siddalingappa, Rashmi ORCID logoORCID: <https://orcid.org/0000-0001-9786-8436>, Mehta, Piyush M. ORCID logoORCID: <https://orcid.org/0000-0002-5240-2322>, Morley, Steven K. ORCID logoORCID: <https://orcid.org/0000-0001-8520-0199>, Godinez, Humberto C. ORCID logoORCID: <https://orcid.org/0000-0001-7138-3481> and Jordanova, Vania K. ORCID logoORCID: <https://orcid.org/0000-0003-0475-8743> (2024) Reduced-Order Probabilistic Emulation of Physics-Based Ring Current Models: Application to RAM-SCB Particle Flux. Space Weather, 22 (6).

Downloaded from: <https://ray.yorks.ac.uk/id/eprint/12871/>

The version presented here may differ from the published version or version of record. If you intend to cite from the work you are advised to consult the publisher's version:

<http://dx.doi.org/10.1029/2023sw003706>

Research at York St John (RaY) is an institutional repository. It supports the principles of open access by making the research outputs of the University available in digital form. Copyright of the items stored in RaY reside with the authors and/or other copyright owners. Users may access full text items free of charge, and may download a copy for private study or non-commercial research. For further reuse terms, see licence terms governing individual outputs. [Institutional Repositories Policy Statement](#)

RaY

Research at the University of York St John

For more information please contact RaY at
ray@yorks.ac.uk

Space Weather®



RESEARCH ARTICLE

10.1029/2023SW003706

Key Points:

- A probabilistic emulator for the ring current-atmosphere interactions model with self-consistent magnetic field (RAM-SCB) particle flux is designed with a focus on reducing computational resources of time and space
- An ensemble of long-short term memory captures temporal variations and provides a probabilistic emulator
- The emulator makes a week-long prediction for single energy flux in ~ 20 s providing a significant speed-up over the full RAM-SCB run

Correspondence to:

P. M. Mehta,
piyush.mehta@mail.wvu.edu

Citation:

Cruz, A. A., Siddalingappa, R., Mehta, P. M., Morley, S. K., Godinez, H. C., & Jordanova, V. K. (2024). Reduced-order probabilistic emulation of physics-based ring current models: Application to RAM-SCB particle flux. *Space Weather*, 22, e2023SW003706. <https://doi.org/10.1029/2023SW003706>

Received 9 SEP 2023
Accepted 23 MAY 2024

Author Contributions:

Conceptualization: Alfredo A. Cruz, Piyush M. Mehta
Data curation: Alfredo A. Cruz, Rashmi Siddalingappa
Formal analysis: Alfredo A. Cruz
Funding acquisition: Piyush M. Mehta
Investigation: Alfredo A. Cruz, Rashmi Siddalingappa, Piyush M. Mehta, Steven K. Morley, Humberto C. Godinez, Vania K. Jordanova
Methodology: Alfredo A. Cruz, Piyush M. Mehta, Steven K. Morley, Humberto C. Godinez
Project administration: Piyush M. Mehta
Resources: Piyush M. Mehta
Supervision: Piyush M. Mehta
Validation: Alfredo A. Cruz, Rashmi Siddalingappa

© 2024. The Authors.

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial-NoDerivs License](#), which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

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

Alfredo A. Cruz¹, Rashmi Siddalingappa¹, Piyush M. Mehta¹ , Steven K. Morley² , Humberto C. Godinez³ , and Vania K. Jordanova² 

¹Department of Mechanical and Aerospace Engineering, West Virginia University, Morgantown, WV, USA, ²Space Science and Applications, Los Alamos National Laboratory, Los Alamos, NM, USA, ³Computational Physics and Methods, Applied Mathematics and Plasma Physics, Los Alamos National Laboratory, Los Alamos, NM, USA

Abstract In this work, we address the computational challenge of large-scale physics-based simulation models for the ring current. Reduced computational cost allows for significantly faster than real-time forecasting, enhancing our ability to predict and respond to dynamic changes in the ring current, valuable for space weather monitoring and mitigation efforts. Additionally, it can also be used for a comprehensive investigation of the system. Thus, we aim to create an emulator for the Ring current-Atmosphere interactions Model with Self-Consistent magnetic field (RAM-SCB) particle flux that not only improves efficiency but also facilitates forecasting with reliable estimates of prediction uncertainties. The probabilistic emulator is built upon the methodology developed by Licata and Mehta (2023), <https://doi.org/10.1029/2022sw003345>. A novel discrete sampling is used to identify 30 simulation periods over 20 years of solar and geomagnetic activity. Focusing on a subset of particle flux, we use Principal Component Analysis for dimensionality reduction and Long Short-Term Memory (LSTM) neural networks to perform dynamic modeling. Hyperparameter space was explored extensively resulting in about 5% median symmetric accuracy across all data sets for one-step dynamic prediction. Using a hierarchical ensemble of LSTMs, we have developed a reduced-order probabilistic emulator (ROPE) tailored for time-series forecasting of particle flux in the ring current. This ROPE offers accurate predictions of omnidirectional flux at a single energy with no pitch angle information, providing robust predictions on the test set with an error score below 11% and calibration scores under 8% with bias under 2% providing a significant speed up as compared to the full RAM-SCB run.

Plain Language Summary Our study tackles the challenges of simulating the ring current in space and time, using a model called RAM-SCB. By making our simulations faster, we can predict changes in the ring current faster than in real time. This is crucial for monitoring space weather and taking actions to protect satellites and other space assets. Our goal is to create a computer program (emulator) that is efficient, can forecast changes accurately, and gives reliable estimates of its predictions. We follow a method developed by Licata and Mehta (2023), <https://doi.org/10.1029/2022sw003345> and extend it to the ring current. Discrete sampling is used to pick specific periods for our simulations. We focus on a part of the simulation data and use an advanced method, a hierarchical ensemble of Long Short-Term Memory (LSTM) to build our emulator. This is called a ROPE, that can predict the ring current's behavior by providing forecasts with errors below 11% and calibration scores under 8% (how aligned the predictions are with the actual).

1. Introduction

Earth's protective magnetic field shields it from harmful solar wind particles. Some of these particles, upon trapping in the Earth's magnetic field, form a “ring current.” The ring current, among the four major particle populations in the inner magnetosphere—comprising the cold plasmasphere, the warm plasma cloak, and the relativistic radiation belts—plays a crucial role in regulating energy density and exerting feedback to the magnetospheric magnetic field configuration (Daglis et al., 1999; Fok et al., 2021; Russell et al., 2016). Containing ions and electrons of a few keV to a few 100 keV flowing at geocentric distances of ~ 2 –6 Earth radii (R_E), the ring current interacts with other inner magnetospheric populations, creating a dynamic environment (Jordanova et al., 2020; Spence et al., 1989). The ring current has energies ~ 10 –400 keV, and is generated by the movement of charged particles experiencing both a convective and a gradient-curvature drift. During geomagnetic activity, the ring current gains population from plasma that is accelerated by reconnection in the magnetotail and transported

Writing – original draft: Alfredo A. Cruz
Writing – review & editing:
Rashmi Siddalingappa, Piyush M. Mehta,
Steven K. Morley, Humberto C. Godinez,
Vania K. Jordanova

earthward by the cross-tail convective electric field, making it the population that carries most of the plasma pressure and current directly into the inner magnetosphere (Ganushkina et al., 2017; Jordanova et al., 2012; Jordanova, Thorne, et al., 2010; Yu et al., 2012). These accelerated particles experience a non-uniform 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 toward the dusk side and electrons toward the dawn side of Earth. This induced westward current aka ring current, is the main contributor to the magnetic depression observed by ground-based magnetometers during geomagnetic storms. The ring current plays a crucial role in space weather, influencing particle populations, magnetic and electric fields, and various space phenomena. It significantly impacts both the Dst and Sym-H indices, key indicators of geomagnetic storms, and is highly correlated with space weather events. The storm-time ring current influences the ionosphere and thermosphere, contributing to geomagnetically induced currents through region 2 field-aligned currents (Jordanova, et al., 2020).

Various models simulate Earth's magnetospheric conditions, with two main types: magnetohydrodynamic models and kinetic models. Magnetohydrodynamic models are effective for the outer magnetosphere but fall short in describing the inner magnetosphere due to the dominance of energy-dependent drifts, as noted by DeZeeuw et al. (2004). Kinetic approaches are favored for the inner magnetosphere and ring current modeling, offering better suitability for capturing the intricate particle dynamics. Kinetic ring current models, pioneered by Harel et al. (1981) and R. Wolf et al. (1982), have been in use for over three decades and continue to be refined, for example (Fok et al., 2021; Jordanova, Thorne, et al., 2010; Toffoletto et al., 2003). These models use diverse methodologies to compute self-consistent electric and/or magnetic fields crucial for simulating the inner magnetosphere. The self-consistent magnetic field is derived by solving the force balance equation $J \times B = \nabla \cdot P$, where J is the current density and P is the plasma pressure tensor (Parker, 1987). While strictly valid in subsonic regions where plasma inertia can be neglected, this force balance equation remains valuable for ring current models due to their alignment within this domain (R. A. Wolf & Kamide, 1983).

In this paper, we consider a well-developed and community-accepted kinetic convection-diffusion ring current-atmosphere interactions model (RAM) (Jordanova et al., 1994; Jordanova, Thorne, et al., 2010) with a self-consistent magnetic (B) field (SCB) (Jordanova et al., 2006; Zaharia et al., 2006). The Ring current-atmosphere interactions model with self-consistent magnetic field (RAM-SCB) began as a research-based code with limited options but is now a powerful and highly configurable open-source software that supports parallelization (Engel et al., 2019; Jordanova & Engel et al., 2022; Jordanova & Morley et al., 2022). By default, RAM-SCB models four species of charged particles (H^+ , He^+ , and O^+ , and e^-) in energies ranging from about 100 eV to 500 keV and the outer flux boundary is set using the Denton et al. (2016) model. An earlier (Kp-parameterized) version of the Denton model is used by the community coordinated modeling center runs-on-request version of RAM (S. K. Morley et al., 2023); the vBz-parameterized version has also been used previously to help specify the outer flux boundary (Yu et al., 2022) and was also used for the near-realtime version described in Jordanova et al. (2023). RAM-SCB provides a configurable spatial domain, but we use the default grid spanning from 2 to 6.5 R_E with a default setting utilizing 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), R_E , energy (E), and pitch angle (PA). The model provides invaluable insight into the physical evolution of dynamical systems such as the ring current. Yet, for real-time applications, the computational cost associated becomes a bottleneck, inviting faster, more efficient models to take their place. Development of more efficient models has gained popularity in the thermosphere (D. J. Gondelach & Linares, 2021; Licata & Mehta, 2023; Mehta & Linares, 2018). Therefore, our objective is to enhance this application and create an emulator that holds wider applicability within the Space Weather community. To begin with, we present a probabilistic forecast of the ring current, showcasing its functionality on a limited subset of RAM-SCB particle flux—specifically, a single energy channel of omnidirectional flux.

We employ reduced order modeling (ROM) to create efficient representations of the complex dynamics of the system, offering a cost-effective approach. Lumley (Lumley (1967)) pioneered model order reduction techniques, initially applied to gain insights into turbulence mechanisms and coherent structures in fluid flow. Subsequent refinements and improvements in these techniques have been explored in geophysical applications (Audouze et al., 2009; Carlberg & Farhat, 2011; Issan & Kramer, 2023; Licata & Mehta, 2023; Mehta et al., 2018; Puzyrev et al., 2018; Rekoske et al., 2023). One of the most challenging aspects of ROM in space weather systems is to properly adjust the timing of the temporal variation predictions with the corresponding input driver(s) (Mehta & Linares, 2017). Nevertheless, utilizing the learned reduced-state features, we construct a reduced-order

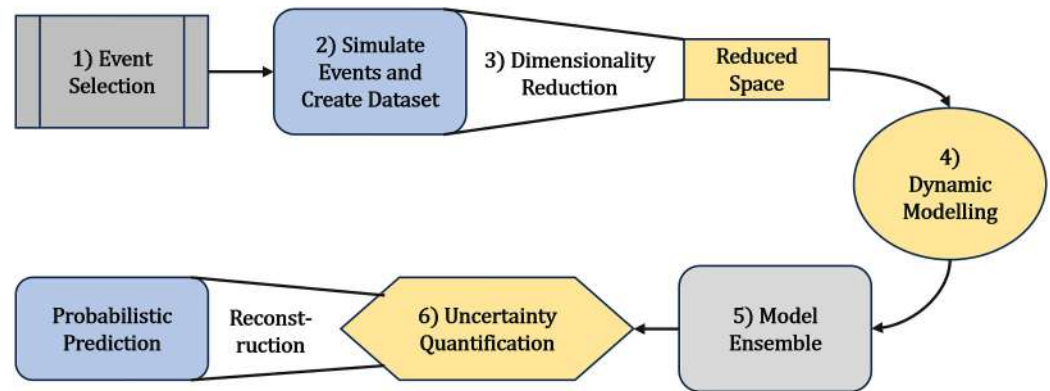


Figure 1. Emulator workflow overview: creation of machine learning data sets, reduced-order dynamic modeling, model ensemble assembly, and generation of the final probabilistic output. Physical and reduced space steps are color-coded in blue and orange, respectively.

probabilistic emulator (ROPE) using an ensemble of Long Short-Term Memory (LSTMs) for time-series forecasting of ring current particle flux, incorporating an uncertainty distribution estimate. The key reasons for the importance of such a system are; (a) This represents a significant advancement, addressing the computational demands associated with traditional simulation-based models. (b) Speed-up allows comprehensive investigation of the system in terms of sensitivity and variance studies. (c) The reduced space simplifies data assimilation as previously demonstrated in Mehta and Linares (2017). (d) It also enables uncertainty quantification (UQ), crucial for operational use. We consider this an important advancement since physical models require significant processing power to execute a simulation.

2. Methodology

The work employs reduced-order modeling (ROM) and machine learning (ML) techniques to significantly cut down the computational cost of physics-based simulation models while preserving their accuracy. Note: Emulators are not developed to replace physics-based models but rather enable a higher adoption rate and usage for more system-wide investigations. 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. Data set creation in Section 2.2
3. Dimensionality reduction in Section 2.3
4. Dynamic modeling in Section 2.4
5. Model ensemble in Section 2.5
6. UQ in Section 2.6

The workflow employs color-coded steps in blue for physical space and orange for reduced space. It starts with discrete sampling to identify geomagnetic storms, simulating them with RAM-SCB to generate ML data sets for the emulator. Dimensionality reduction transforms ML data sets into a reduced space, and ROM identifies influential variability modes (Mehta & Linares, 2017; Mehta et al., 2018). Operating in this space aids data assimilation for high-dimensional systems (D. J. Gondelach & Linares, 2020; Mehta & Linares, 2018; Maulik et al., 2022). A recurrent neural network (RNN) predicts temporal variations in the reduced space, enhancing nonlinear modeling. This deterministic model provides a single estimate, therefore an ensemble of deterministic models is created for UQ thereby developing a ROPE. In the final step, predictions and uncertainties from the model ensemble are reconstructed into the physical space by reversing the dimensionality reduction. It's essential to note that developments in the reduced space can be evaluated in the physical space using this reconstruction step.

2.1. Event Selection

The initial and perhaps the most crucial stage in developing any ML model involves constructing appropriate training, validation, and test data sets. To create the data sets, we used NASA's space physics data facility

OMNIWeb database (OMNIWeb, 2020), focusing on solar wind and geomagnetic data spanning 2000–2020, with a 1-min cadence. We included various solar wind parameters such as 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, flow pressure, and the geomagnetic indices AL and SYM-H. The decision to retrieve velocity components in GSE coordinates, rather than GSM coordinates, is constrained by the fact that the OMNI web interface provides velocity components in GSE. We note the coordinate systems to help any users of the open data associated with this work avoid the pitfalls of incorrect coordinate system assumptions. Here, the transverse components of velocity are not used in the modeling, only V_x . As the X-axis of GSM and GSE are the same, our input data to RAM-SCB is consistent. However, anyone choosing to use these input files to run a different model would need to be aware of the systems used so that they could convert the transverse components to GSM. Given the computational challenges and expenses involved in simulating this extensive time frame using physics-based models like RAM-SCB, we devised a custom discrete sampling methodology. This approach enabled us to identify a set of random events that sufficiently capture the entire range of solar wind drivers and ring current responses.

The 21 years of OMNIWeb data from 2000 to 2020 are split into smaller, more manageable weekly segments, each representing a candidate simulation interval. The 7-day intervals are long enough to capture both a space weather event and its recovery phase and short enough to avoid grouping unrelated events. We then applied two filters; (a) 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. The predictive performance of RAM-SCB is known to be diminished when simulations are started under strongly disturbed conditions (Jordanova, Zaharia, & Welling, 2010). Therefore, we drop weekly intervals that begin with radial SW velocities (V_x) exceeding 500 km/s. (b) We restrict missing data within each interval to a cumulative total of 36 hr (1.5 days) for any parameter. Smaller gaps passing this filter are linearly interpolated using the entire weekly time series. These filters reduce the number of possible candidates from 7,664 to 2,839 at 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 weeks-long candidate intervals is located in a 4-dimensional (4-D) parameter space using a set of summary statistics: (a) minimum SYM-H, (b) mean AL, (c) mean V_x , and (d) minimum B_z . These parameters describe the strength of the ring current disturbance and overall geomagnetic activity, impulsive energy dissipation and injection of plasma into the inner magnetosphere, and strength of the solar wind drivers, respectively. Traditional Latin hypercube sampling principles (Deutsch & Deutsch, 2012) use equal probability bins in continuous probability distributions. We have discrete samples and require even coverage over parameter space, hence we opt for an independent binning approach for each parameter. The full range of each parameter is divided into 10 linearly spaced bins, each assigned equal probability. In the sampling process, a bin index is randomly drawn with replacements for each parameter, ensuring coverage. If a bin is empty, the process is reiterated until a non-empty bin is found. Subsequently, a candidate interval is randomly chosen, maintaining uniform probability within the bin. This procedure is repeated for all parameters, yielding a pool of four candidate intervals. The final selection involves randomly choosing an interval from this pool and removing it from each parameter space. In our sampling approach, we independently bin each parameter space, providing a flexible and controlled method to ensure sufficient coverage of each parameter's range. Binning directly in the 4-D space would present challenges in balancing the representation of parameters, especially with varying ranges. We aimed to address two key objectives with our discrete sampling method. Firstly, we sought to comprehensively and uniformly cover the entire parameter space, ensuring a thorough representation of geomagnetic activity. Secondly, we aimed to mitigate, by construction, the common issue of data imbalance in space weather data sets. The sampling uses two summary statistics that represent magnetospheric response (minimum Sym-H and mean AL), and two that represent the solar wind driver (mean V_x and minimum B_z). Uniform sampling from this 4-D parameter space will include intervals driven by solar wind transients, although the method does not specifically select based on the solar wind transients. The approach covers the span of drivers from weak to strong (as represented by the summary measures), as well as the span of responses (as represented by the geomagnetic index summaries). While it is unsurprising that strong activity and strong drivers are associated with solar wind transients, the approach does not specifically select these. Thus, the observed prevalence of these events in our selected data intervals is coincidental in that regard. Note that we also draw events from bins with less geoeffective drivers and minimal response—this part of parameter space may have non-geoeffective

Table 1
Training Events Identified by the Novel Sampling Methodology

Event	Start date (YYYY-MM-DD)	Min (SYM-H) (nT)	Mean (AL) (nT)	Mean (Vx) (km/s)	Min (Bz) (nT)
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

transients but highlights that we are not selecting based on the presence of transients. Our custom discrete sampling method addresses the discrete and event-driven nature of space weather, providing adaptability and more uniform coverage in the 4-D parameter space, mitigating bias toward less dynamic states/quiescent times. This discrete sampling method will also contribute to improved generalization in the ROPE process.

A total of 30 events were selected using this sampling methodology, with 20 used for the training (TRNG) data set (see Tables 1 and 5) used for each of the validation (VAL) and test (TST) data sets (see Table 2). Upon initial sampling, it was observed that the TST2 case (TST 2, 2004-11-03, (394.0, (277.3, (475.3, (44.7) was being overlapped, with the values closely aligning with the VAL2 data (VAL 2, 2004-11-05, (394.0, (409.3, (542.7, (44.7). Recognizing the potential for data leakage during the model evaluation phase, we opted to address this concern by replacing the TST2 data with a new event selected from the same bin where the original TST2 was located. This proactive step ensures the integrity of our model evaluation process by mitigating the risk of inadvertent overlap between training and validation data sets. By selecting a replacement event with similar characteristics to TST2, we maintain the coherence of the data set while minimizing the possibility of bias in our results. 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 (Waskom, 2021; Węglarczyk, 2018) are plotted above the diagonal. This split leads to a training/validation/test ratio of 66/17/17%.

2.2. Data Set Creation

Simulations were performed on WVU's Thorny Flat cluster with identical configurations, utilizing 13 CPU cores (Cruz et al., 2024). The simulations generated 92 GB of data over 210 days in nearly 48 computational days at an average speed of 4.4x real-time, with a final output of 3 TB. RAM-SCB produced equatorial, directional differential fluxes as a 4-D hypercube for various plasma species (H^+ , He^+ , O^+ , and e^-) for each

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

Event	Start date (YYYY-MM-DD)	Min (SYM-H) (nT)	Mean (AL) (nT)	Mean (Vx) (km/s)	Min (Bz) (nT)
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	2005-05-12	(305.0	(273.9	(562.6	(36.7
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

simulation, resulting in a data shape of (72, 35, 25, 20) per timestep. Each 7-day simulation outputted at a 10-min cadence, yielding 1,008 timesteps. The data structure for protons (H^+) from 20 training simulations had a shape of (20,160, 72, 35, 25, 20). Preprocessing steps included converting to single precision, resetting the minimum threshold, and truncating radial distances below $3 R_E$, which resulted in a final shape of (20,160, 72, 35, 25, 15). To maintain spatial information, we used a single energy channel and integrated directional flux into omnidirectional flux (normalized per steradian) at the 208 keV energy channel following Bourdarie et al. (2012):

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

This yields omnidirectional differential flux (j_{omni}) in $\text{cm}^{-2} \text{s}^{-1} \text{sr}^{-1} \text{keV}^{-1}$, where $j(E, \alpha)$ is the directional differential flux at a specific energy (E) and pitch angle (α). By focusing on a single energy channel and eliminating α information, the training data is condensed to spatial dimensions with a shape of (20,160, 25, 15), occupying 30 MB of physical memory. The same process is applied to validation and test data sets.

2.3. Dimensionality Reduction

The next step in the emulation process is to reduce the dimensionality of the data sets. Principal Component Analysis (PCA) (Hotelling, 1933), aka Empirical Orthogonal Function analysis (Bjornsson & Venegas, 1997; Matsuo & Forbes, 2010), is the first step applied in ROM to reduce the spatial dimensionality of the system. A system's spatial variations are normally represented by a set of basis vectors that are both independent in time and mutually orthogonal (Carlberg & Farhat, 2011; D. J. Gondelach & Linares, 2020; D. Wilks, 2011). The temporal variations $c_i(t)$ are then added in as weights to the spatial variations, which we will be referring to as the reduced-order temporal coefficients (Licata et al., 2022). This is shown in Equation 2, where $\mathbf{X} \in \mathbb{R}^n$, $\mathbf{s}$ represents the spatial domain, t represents the temporal domain, and $\mathbf{U}$ contains the spatial modes.

$$\mathbf{X}(\mathbf{s}, t) = \bar{\mathbf{X}}(\mathbf{s}) + \tilde{\mathbf{X}}(\mathbf{s}, t) \quad \text{where} \quad \tilde{\mathbf{X}}(\mathbf{s}, t) \approx \sum_{i=1}^r c_i(t) \mathbf{U}_i(\mathbf{s}) \quad (2)$$

We convert the data into a 2-D array as a last step performed to prep the data for the PCA. 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 (20,160, 375). Our PCA is implemented using a singular value decomposition solver (Pedregosa et al., 2011).

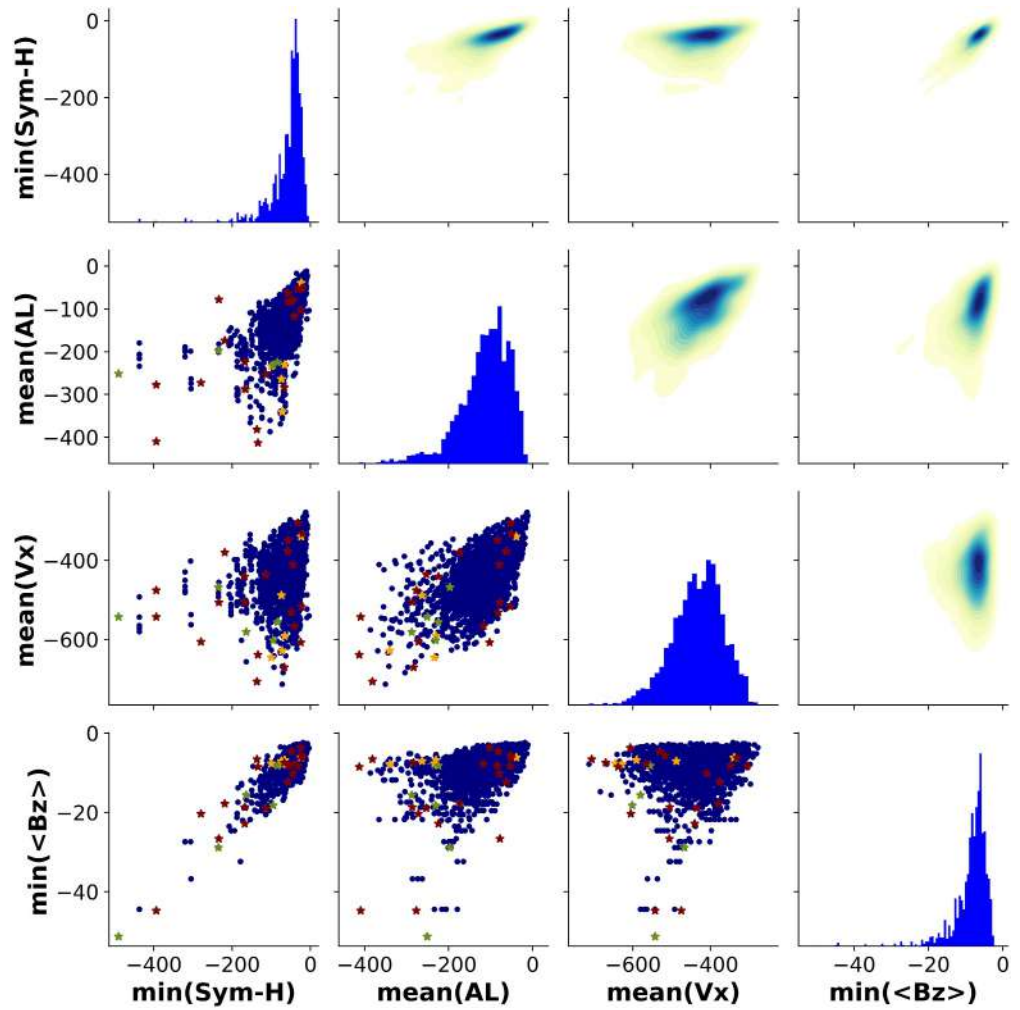


Figure 2. Pairplot displaying the TRNG (red), VAL (green), and TST (orange) events identified by the novel custom discrete sampling methodology. It visualizes the sampling taken within each parameter's distribution, where histograms of each parameter are shown on the diagonal plots. The panels below 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.

$$\tilde{\mathbf{X}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad \text{where} \quad \tilde{\mathbf{X}} = \begin{bmatrix} | & | & & | \\ \tilde{\mathbf{x}}_1 & \tilde{\mathbf{x}}_2 & \dots & \tilde{\mathbf{x}}_n \\ | & | & & | \end{bmatrix}, \quad (3)$$

where $\mathbf{U}$ contains the left singular vectors of $\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T$, $\mathbf{V}$ contains the right singular vectors of $\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$, $\mathbf{\Sigma}$ 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 data sets into the reduced-space representation.

The PCA decomposition identifies spatial modes of variability, resembling known physical processes, but direct interpretations are not guaranteed as each mode may encompass multiple processes or combinations of physical phenomena (McGranaghan et al., 2015). Figure 3 shows the mean and first seven 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 three trends: (a) radial falloff, (b) symmetric rings, and (c) asymmetric structures. The mean and Mode 1 exhibit radial falloff, corresponding to the expected location of the ring current during quiescent and storm phases. 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, Zaharia, & Welling, 2010), which is consistent

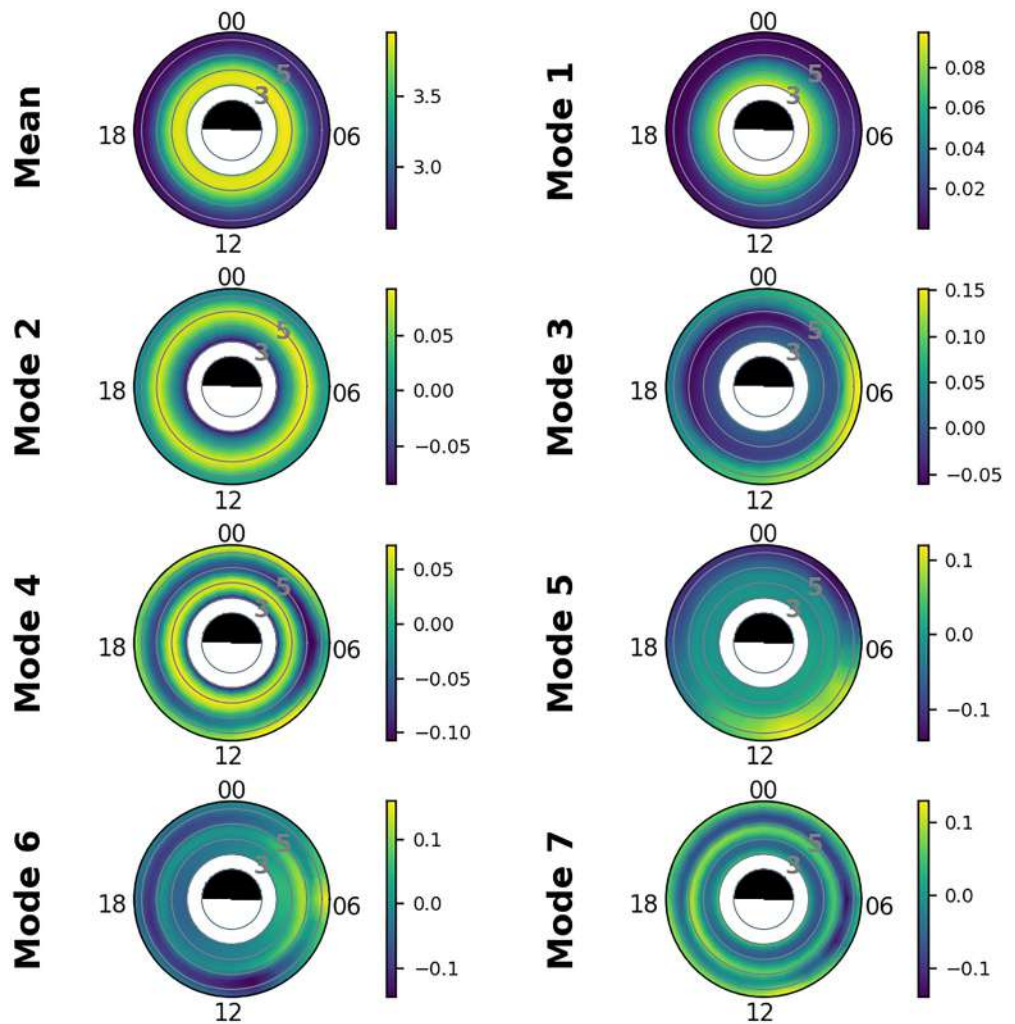


Figure 3. Mean and first 7 spatial modes of variability identified by the Principal Component Analysis from the right singular vectors plotted on RAM-SCB's grid. The modes are ordered in terms of importance, meaning the mean is the most dominant followed by mode 1, and so on.

with 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 toward 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 particle energies. During the main and recovery phases of a geomagnetic storm, each particle's drift is known to vary radially (Jordanova, Zaharia, & Welling, 2010), creating similar symmetric rings. The asymmetric structures in Modes 3, 5, and 6 are more difficult to interpret but are probably related to the asymmetric particle injection at the night-side boundary. 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 probably due to the westward drift path for high-energy ions in the ring current.

We decided to truncate our PCA at 20 modes ($r = 20$), which reduces the spatial dimensionality from $\tilde{\mathbf{X}} \in \mathbf{R}^n$ to $\tilde{\mathbf{X}} \in \mathbf{R}^r$. This is essentially a balance between minimizing the amount of reconstruction error and reducing the dimensionality of the system for enough observability in later data assimilation applications (Mehta & Linares, 2018). The cumulative variance contribution is plotted on the left axis of 3, 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% median symmetric accuracy (MdSA) (S. K. Morley, Brito, & Welling, 2018). Please refer to Section 2.7 for more details.

To illustrate the robustness of the PCA decomposition, three 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 before and after the storm are in the same order as the truncation error, with an MdSA of 3.4% and 1.6%, respectively. Geomagnetic storms are dynamic and complex phenomena characterized by non-linear interactions between solar wind and Earth's magnetosphere. While linear PCA provides valuable insights into the principal modes of variability within our data set, it may not fully capture the intricate non-linear dynamics exhibited during geomagnetic storms. As such, errors in the PCA reconstruction are expected to increase under such conditions, highlighting the limitations of linear methods in capturing non-linear relationships within the system.

2.4. Dynamic Modeling

We implement a LSTM (Gers et al., 2002; Hochreiter & Schmidhuber, 1997) RNN to model and predict RAM-SCB's temporal variations (Wang et al., 2022). We chose LSTM for dynamic modeling for 2 reasons: (a) Since magnetospheric responses tend to lag behind their SW drivers (Bargatze et al., 1985; Borovsky, 2020), 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). (b) 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; Kozyra et al., 2002; S. K. Morley & Lockwood, 2006) adds another layer of complexity to how the LSTM learns the dynamics of this system. LSTMs require a unique input structure, containing 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).

Figure 6 shows a pictorial representation of different components in an LSTM cell. The LSTM cell processes input x_t alongside the previous state outputs (C_{t-1} , H_{t-1}) and current state input, with t and $t-1$ denoting current and previous steps. Three σ nodes represent internal layers, using sigmoid activation, and tanh indicates a layer or point-wise operation. Sigmoid layers act as binary gates, allowing open (1) or closed (0) states. In the input gate, the sigmoid and tanh layers process input and previous output. The forget gate employs a sigmoid layer to influence the previous cell state. The output gate decides how much of the output should proceed to the next layer. The internal cell state undergoes updates, and the information proceeds to the downstream layer, ensuring LSTMs effectively manage long-term information.

In a data set with n samples, each comprising inputs (n_{inp}) and outputs (n_{out}), concatenation produces an array of shape $n \times (n_{out} + n_{inp})$. For RNNs, the hyperparameter known as the number of lag-steps (n_{LS}) determines the model's consideration of previous time steps, akin to short-term memory in Long Short-Term Memory (LSTM) networks. Data preparation involves stacking rows to arrange outputs and inputs for each lag step, resulting in a shape of $n \times (n_{LS} + 1) (n_{out} + n_{inp})$. The last n_{inp} columns are dropped. Further division yields training inputs and outputs, with the first $n \times n_{LS} (n_{out} + n_{inp})$ columns as inputs and the last n_{out} columns as outputs. The final step involves reshaping the input data to $n \times n_{LS} \times (n_{out} + n_{inp})$.

In our LSTM training process, we employ a many-to-one mode, where states are reset only after each simulation period. Consequently, states undergo multiple resets within a single epoch. We opt not to utilize a dynamic mode during training. Transitioning to evaluation mode, we maintain the many-to-one configuration; however, the "many" aspect begins to incorporate predictions as we progress beyond the prediction epoch.

During LSTM training, the model predicts one step ahead using the last n_{LS} sets of true values for both the input drivers and state outputs (or reduced-space coefficients in our case). The availability of true state outputs stems from the predetermined nature of the training, validation, and test data sets, all derived from simulations. This evaluation method, entailing the utilization of true input drivers and state outputs to forecast each timestep, is commonly referred to as a one-step prediction method.

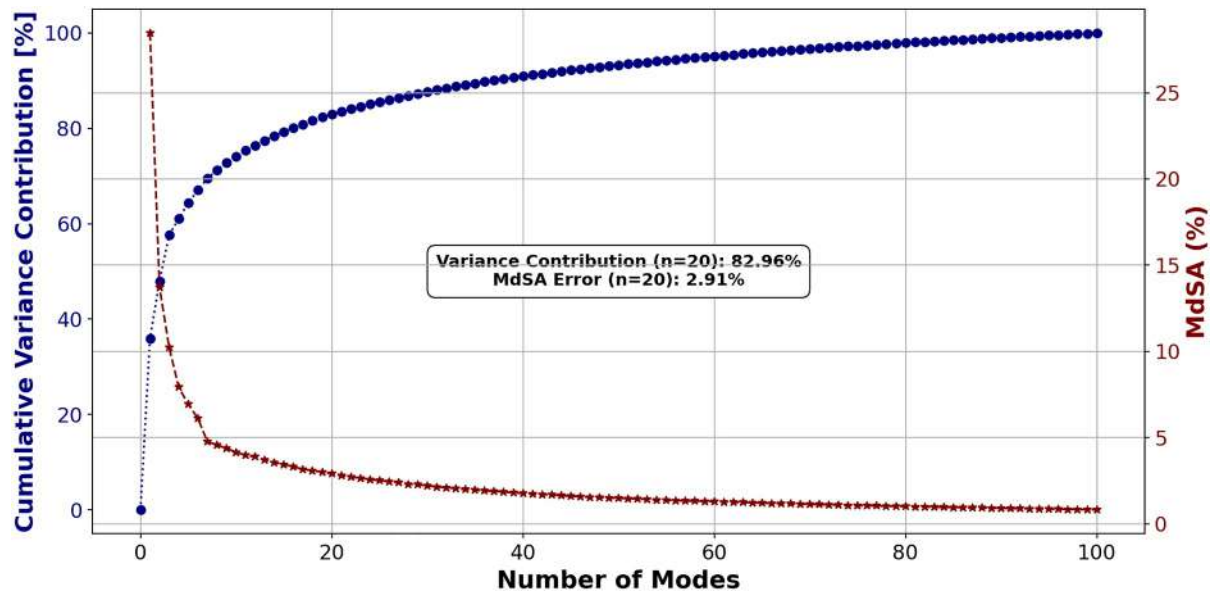


Figure 4. The cumulative variance contribution (orange) for each mode of the Principal Component Analysis, and truncation error (blue) of the reconstruction back into the physical space using the specified number of modes on the training data set.

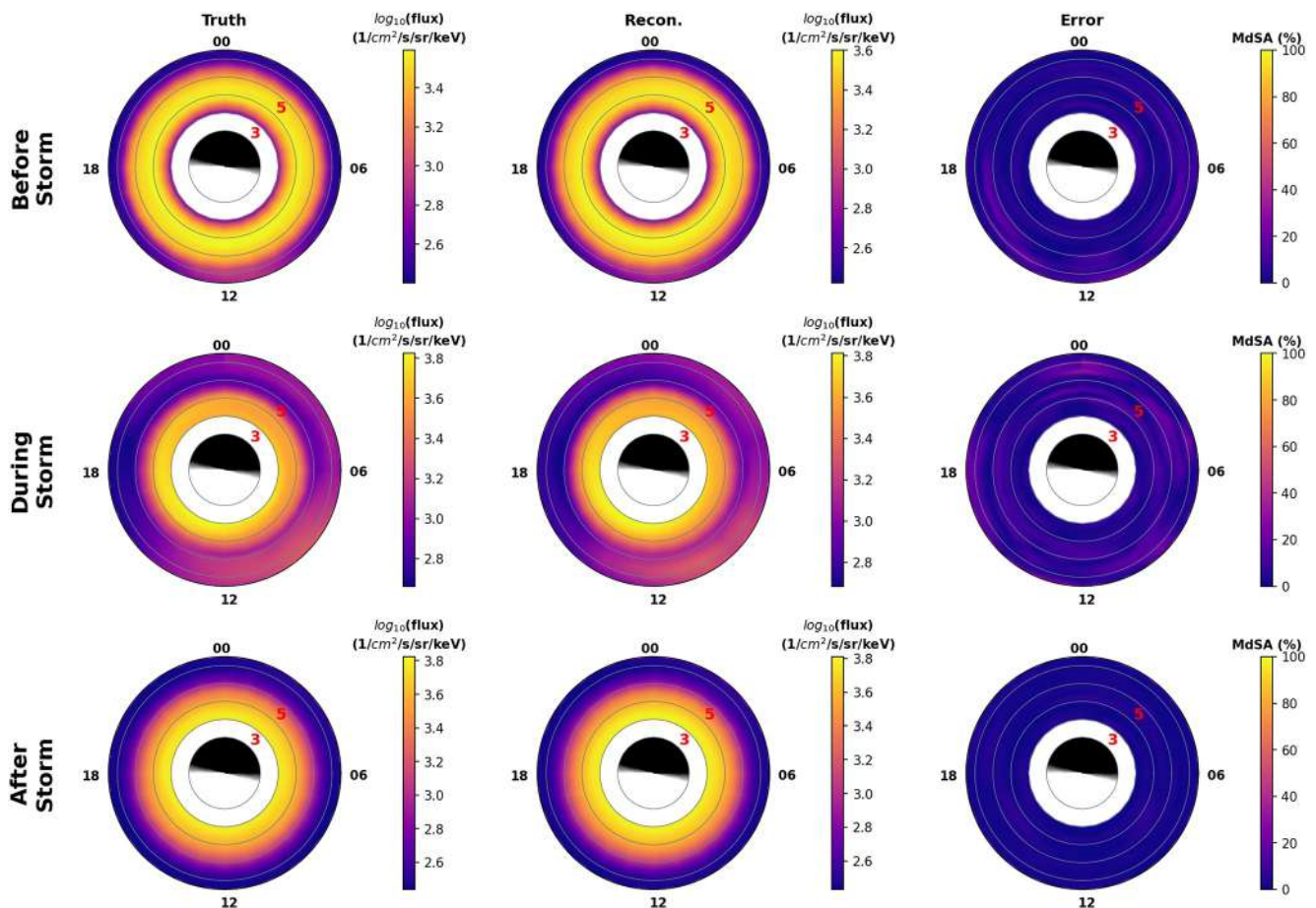


Figure 5. Snapshots taken before, during, and after the geomagnetic storm of the VAL 4 simulation with the truncation errors (right) between the actual (left) and reconstructed (middle) fluxes. The truncation errors for the before, during, and after snapshots are 3.4%, 5.9%, and 1.6% median symmetric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

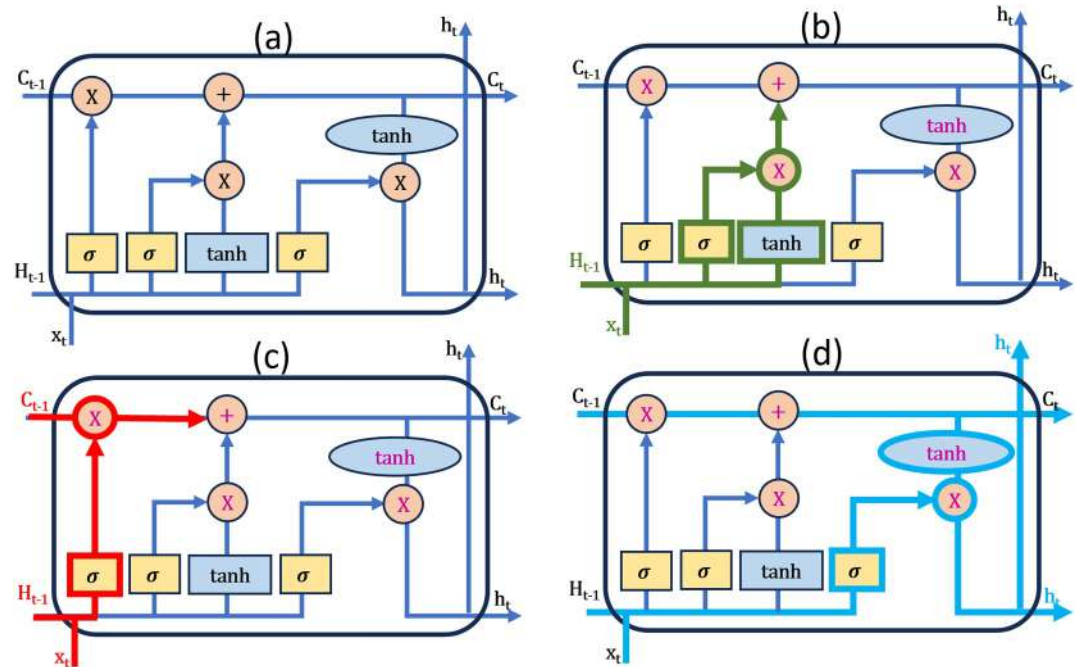


Figure 6. Schematic representation of a typical Long-Short Term Memory (LSTM) cell (a). The components of an LSTM cell such as the input gate (b - green color-coded), forget gate (c - red color-coded), and output gate (d - blue color-coded) are shown. Point-wise operations are denoted in green.

Throughout an epoch, the model iteratively updates weights, encompassing those of the three sigmoid layers and one tanh layer within each LSTM cell. As information traverses through each cell, the LSTM's internal state C evolves. To counter temporal discontinuities, we reset the internal memory of every LSTM cell at the end of each epoch, precluding the utilization of outdated information in predictions. This resetting mechanism proves pivotal for ensuring accurate evaluations across diverse time periods during one-step LSTM training.

In certain scenarios, true state outputs may not be accessible. In such cases, when conducting forecasts, predicted state outputs are utilized for forecasting future timesteps, following the approach outlined in Figure 3 of Licata and Mehta (2023). After predicting the current timestep t , the lookbacks are advanced for the subsequent timestep $t + 1$, with the corresponding lookback for t being updated with the predicted output. This iterative process continues for the forecast window's duration, constituting a dynamic prediction method, which represents one of the advantages conferred by developing an emulator.

For the direct sequence applied to the model, we employ a look-back period of 3, determined through parameter tuning. This suggests that, based on the validation metric, the optimal window for this application spans 3-time steps. We exclusively operate in a stateful mode for a singular dynamic prediction window. For instance, historical information and forecasted drivers initiate the process, with a fresh state commencing the sequential advancement into the future, replacing historical information with predictions in the lookback window to diverge from the prediction epoch. The state remains unreset during this period, retaining system information. Subsequently, if the model is utilized for a different timeframe, the state is reset. Throughout all training and evaluation processes, the state reset only occurs when transitioning to a new window.

2.4.1. Hyperparameter Tuner

We employ a hyperparameter tuner, using TensorFlow's API (Abadi et al., 2015) and Keras Tuner (O'Malley et al., 2019) with GPU settings using Nvidia's CUDA framework (NVIDIA, 2022), to automate the search for optimal LSTM architectures. The tuner performs 50 trials, starting with 25 random grid searches and concluding with 25 Bayesian Optimization iterations (Snoek et al., 2012). In typical practice, data sets with many samples are split into batches for training efficiency which also enhances the model's generalization capability (Goodfellow et al., 2016; Licata & Mehta, 2023; Montavon et al., 2012). While we adopt this approach by halving our

Table 3
Hyperparameter Tuner Configuration (Left Panel) and Tuner Search Space (Right Panel) for the Long-Short Term Memory Model

Hyperparameter tuner configuration		Hyperparameter search space	
Tuner option	Choice	Parameter	Values/range
Scheme	Bayesian optimization	Number of LSTM layers	[1–2]
Total Trials	50	LSTM neurons	Min = 32, Max = 300, step = 4
Initial Trials	25	LSTM activations	Tanh, Sigmoid, SoftSign
Repeats per trial	2	Number of dense layers	[1–3]
Minimization parameter	Validation MSE	Dense Neurons	Min = 64, Max = 600, step = 4
Epochs per trial	50	Dense activation	ReLU, Elu, Sigmoid, SoftSign, SoftPlus
Early stopping criteria	val_loss	Dense dropout	min = 0.01, max = 0.50, step = 0.01
Early stopping patience	75	Optimizer	AdaGrad, AdaDelta, RMSProp, Adam
		Lookback Period	[3, 24]

simulations, allowing for batch shuffling during each training epoch, it has the drawback of truncating data due to the required lookback period of a few timesteps to predict the initial epoch. Our hyperparameter tuner performs two separate executions per trial to mitigate potential performance issues from weight initialization. Additionally, we implement a callback to terminate training upon encountering a NaN loss, acting as a precaution against exploding gradients (Goodfellow et al., 2016).

Table 3 outlines the hyperparameter space and the tuner configuration settings. The choice of optimizers, such as AdaGrad (Duchi et al., 2011), RMSprop (Hinton et al., 2012), AdaDelta (Zeiler, 2012), and Adam (Kingma & Ba, 2014), is treated as a hyperparameter. We also introduce a custom hyperparameter for the LSTM's lookback period to address variations in the inner magnetosphere's responses to solar wind drivers, requiring changes in the LSTM's input shape for each trial (Bargatze et al., 1985; Maggiolo et al., 2017; Stumpo et al., 2020). However, this presented an additional challenge because LSTM's input shape needs to be changed for each trial of the hyperparameter tuner.

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, which helps mitigate the effects of local minima in the loss function. The top architectures identified by the tuner are then put through 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.

2.5. 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 (Elvidge et al., 2016, 2023; Kioutsoukis & Galmarini, 2014; S. K. Morley, Welling, & Woodroffe, 2018; Weigel et al., 2008; Xiao et al., 2018) because a diverse set of models will normally contain individual models that predict certain portions of the training data better than others. 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 ensure a diverse ensemble, we independently train five instances of the top five architectures from scratch, resulting in 25 models. This approach, different from the one introduced by Licata and Mehta (2023), enhances diversity within architectures. The models, even with the same architecture, exhibit differences due to random weight initialization, dropout inclusion, and batch shuffling during training (Goodfellow et al., 2016; Montavon et al., 2012). Additionally, instead of selecting top models based on mean squared error (MSE) of reduced-space

temporal coefficients in the validation data set, as is common, we identify top architectures using the physical-space metric (MdSA) (refer Section 2.7).

2.6. Uncertainty Quantification

The emulator's last step is to combine the ensemble of deterministic models into a single probabilistic model. Multi-model ensembles have a history of applying a 2-tier weighted average method to combine models (Elvidge et al., 2016, 2023; Huang et al., 2009; Sewell, 2008; D. S. Wilks, 2011;), but Licata and Mehta (2023) adds another tier to the method while also computing a variance, making this 3-tier hierarchical approach a robust and reliable uncertainty estimate. To begin, each of the 25 models must be evaluated over the training data set 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 data set evaluation. The central tendency metric (mean vs. median) used in the UQ calculations varies depending on the data set as indicated in Section 2.7. 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 higher accuracy. In Equation 4 (right), the median absolute error (MdAE) is taken over t for each model's evaluation and inverted to place more weight on models that have the least error. These weights $\tilde{w}_{i,j,k}$ are then normalized within each architecture using Equation 4 (left) so that the combination can be calculated as a simple weighted sum.

$$w_{i,j,k} = \frac{\tilde{w}_{i,j,k}}{\sum_j \tilde{w}_{i,j,k}} \quad \text{where} \quad \tilde{w}_{i,j,k} = \frac{1}{\text{MdAE}_{i,j,k}} \quad (4)$$

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 5. In these equations, $\hat{\alpha}_{i,j,k,t}$ are the dynamic predictions from each model, $\hat{\alpha}_{i,k,t}$ is the mean prediction for each architecture, and $\hat{\sigma}_{i,k,t}^2$ is each architecture's estimated variance.

$$\hat{\alpha}_{i,k,t} = \sum_j w_{i,j,k} \hat{\alpha}_{i,j,k,t} \quad \text{and} \quad \hat{\sigma}_{i,k,t}^2 = \sum_j w_{i,j,k} (\hat{\alpha}_{i,k,t} - \hat{\alpha}_{i,j,k,t})^2 \quad (5)$$

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 from a Bayesian model averaging 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 6 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 data set'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 $\hat{\sigma}_{i,k,t}^2$. This choice was made to focus on capturing the architecture-specific variability within the variance metric.

$$S_{i,k} = \sqrt{\text{Median} \left[\frac{(\alpha_{k,t} - \hat{\alpha}_{i,k,t})^2}{\hat{\sigma}_{i,k,t}^2} \right]} \quad (6)$$

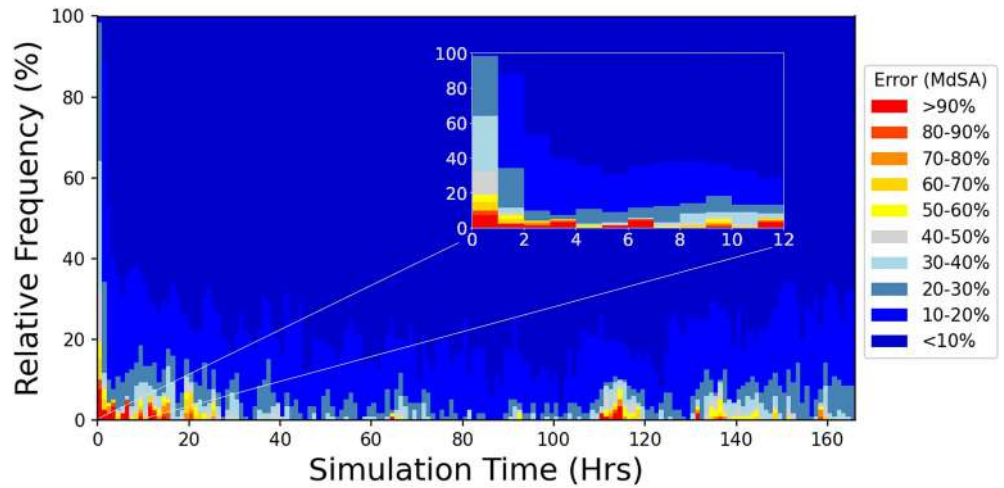


Figure 7. Relative error histogram of dynamic prediction errors from all 20 weeks-long training simulations. The simulation time is binned hourly, while the errors are binned every 10% median symmetric accuracy. The inset highlights the relative high errors seen at the onset of each simulation.

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 7, 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.

$$\hat{\alpha}_{k,t} = \frac{1}{n_i} \sum_i \hat{\alpha}_{i,k,t} \quad \text{and} \quad \hat{\sigma}_{k,t}^2 = \frac{1}{n_i} \sum_i \hat{\sigma}_{i,k,t}^2 \quad (7)$$

2.7. Metrics

The error metric employed for physical space is the MdSA (S. K. Morley, Brito, & Welling, 2018). Given the wide span and strictly positive nature of ring current particle flux (Zheng et al., 2019), conventional relative error metrics like mean absolute percent error face limitations. MdSA addresses these concerns by offering a relative error metric that equally penalizes over- and under-estimations. Utilizing the median instead of the mean enhances robustness, particularly in the presence of outliers or unreliable data. In Section 2.6, when averaging over the temporal range (t) for UQ, the median statistic is preferred due to the expected outliers during peak solar and geomagnetic activity. Equation 8 outlines the MdSA computation, where $Q = \frac{\text{pred}}{\text{truth}}$ represents the accuracy ratio. MdSA provides a straightforward interpretation as a percentage error.

$$\text{MdSA} = 100 \left(\exp \left(\text{Median}(|\log_e(Q)|) \right) \right) (1) \quad (8)$$

The metric used to determine the bias in either the physical or reduced spaces is the symmetric signed percentage bias (SSPB) (S. K. Morley, Brito, & Welling, 2018). Similar to the MdSA, the SSPB is a relative error metric that penalizes over (positive SSPB value) and under (negative SSPB value) estimations equally. An SSPB with 0% indicates an unbiased prediction.

$$\text{SSPB} = 100 \, \text{sgn} \left(\text{Median}(\log_e(Q)) \right) \left(\exp \left(|\text{Median}(\log_e(Q))| \right) \right) (1) \quad (9)$$

The standard metric, MSE, is used to quantify errors in the temporal coefficients in the reduced space (ref. Equation 2). Model performance metrics, optimized in dynamic models (Section 2.4), undergo multiple

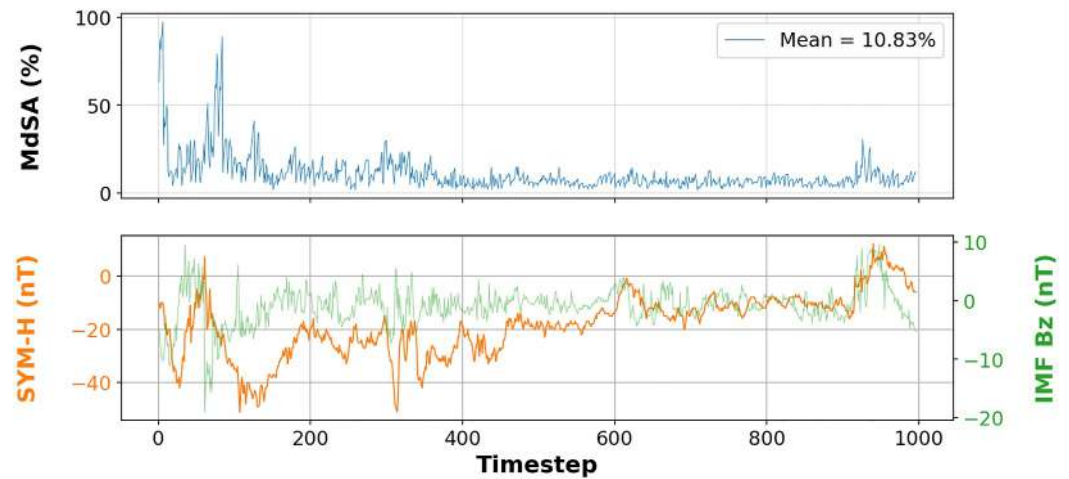


Figure 8. Hourly dynamic prediction results of the TST 5 simulation from the Long-Short Term Memory tuner's best model. Reconstructed errors (blue) in the physical space (median symmetric accuracy) are plotted along with the SYM-H index (orange) and interplanetary magnetic field B_z component (green).

transformations (logarithmic, dimensional reduction, and standardization), posing challenges for direct interpretation. Consequently, post-processing involves reconstructing predictions into the physical space, a unique challenge when working with ROMs as ML algorithms analyze reduced-space representations that may lack interpretability.

We will use the calibration error score (CES) from Licata and Mehta (2022) as a metric for UQ. Calibration error score measures the deviation between observed cumulative probability $p(\hat{\alpha}_{r,m})$ and expected cumulative probability $p(\alpha_{r,m})$, computed as outlined in Section 2.5.1 of Licata et al. (2022), where the prediction intervals span from 5% to 99% in increments of 5%. Reliability is assessed through calibration curves, plotting $p(\hat{\alpha}_{r,m})$ against $p(\alpha_{r,m})$, with curves under Gaussian distribution assumption. A 45° line (i.e., $y = x$) indicates a match with a normal distribution, while deviations imply over or underestimation of uncertainty. Calibration curves and CES are calculated in the reduced space, and the CES calculation is presented in Equation 10.

$$CES = \frac{100\%}{r \cdot m} \sum_r \sum_m |p(\alpha_{r,m}) - p(\hat{\alpha}_{r,m})| \quad (10)$$

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

3. Results

3.1. Hyperparameter Tuner

Table 4 presents the best models from the hyperparameter tuner. As seen in the test and validation metrics, most errors hover around 5% MdSA with biases under $\pm 1\%$ SSPB after only 50 epochs of training. A model ensemble was created by selecting the top 5 architectures from this table. The tuner settled on a shallow architecture, where all of the top 10 architectures had only one LSTM and one dense layer. Only the Best Model #2 differed by having two dense layers. Each model used the AdaGrad optimizer with a lookback period of three timesteps, or 30 min of simulation time. It is 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.

3.2. Dynamic Prediction

Based on a detailed analysis, we found relatively high errors during the initial few hours of each simulation. Figure 7 shows the relative frequency of errors across all 20 weeks-long training simulations. The simulation time

Table 4
Top 10 Long-Short Term Memory 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%	0.41	7.89%	1.81%
2	0.156	4.24%	0.37%	0.398	5.33%	0.69%	0.39	8.49%	0.51%
3	0.150	4.14%	(0.02%	0.395	5.49%	(0.63%	0.44	8.01%	(0.72%
4	0.156	4.26%	0.73%	0.367	5.58%	(0.61%	0.42	7.71%	1.00%
5	0.187	4.10%	(0.36%	0.273	5.59%	0.06%	0.33	7.15%	1.15%
6	0.166	4.29%	(0.03%	0.408	5.72%	(1.54%	0.42	8.70%	(0.87%
7	0.168	4.40%	0.04%	0.394	5.74%	(0.39%	0.46	8.50%	(1.12%
8	0.190	4.32%	(0.11%	0.288	5.80%	(0.58%	0.33	7.10%	1.35%
9	0.206	4.66%	0.70%	0.348	5.91%	0.31%	0.34	7.21%	0.50%
10	0.205	5.15%	0.13%	0.328	6.35%	(0.22%	0.33	7.29%	0.52%

is binned every hour (6 timesteps) while the errors are binned every 10% MdSA. Figure 7 is interpreted as a histogram, where the errors for every hour of each simulation are binned and presented as a percentage. The inset, 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 data set, but it should mitigate the errors seen in the initial few hours of Figure 7.

As stated in Section 2.5, the top 5 architectures identified by the hyperparameter tuner are processed through more rigorous training and evaluated using a dynamic prediction. An hourly forecast window was chosen for the dynamic prediction. The selection of an hourly forecast window for dynamic prediction was guided by an empirical assumption, driven by the intuition that forecasting double the lookback period would be a logical choice in this context. Figure 8 shows the errors of the dynamic prediction evaluation for the TST 5 simulation (see Table 1) using the tuner's best model. The SYM-H and IMF B_z drivers are also plotted directly below the errors to visually show that the error spikes coincide with heightened activity level. This LSTM model was able to dynamically predict this week-long simulation in just 11 s with a mean MdSA less than 11%, even though

the peak error can be observed in the initial timesteps, which is 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 5.48%, 8.09%, and 11.31% MdSA, respectively.

The LSTM model was able to dynamically predict all 5 week-long test simulations using the hyperparameter tuner's best model in approx. 2 minutes with a mean MdSA of 17.9%. This error value is an average over the entire test data set, where the MdSA is determined from the reconstructed fluxes for each timestep of every simulation. The quartiles (25%, 50%, 75%) came out to 6.8%, 11.42%, and 19.74% MdSA, respectively. This means that more than 75% of the errors in this entire data set have less than 20% MdSA.

Table 5
Hourly Dynamic Prediction Results for Both the Best Individual Model (Deterministic) and Final Probabilistic Emulator (Reduced-Order Probabilistic Emulator) Over Each of the Machine Learning Data Sets

	TRNG	VAL	TEST
Indiv. Model:			
Dyn. Pred. (MdSA)	8.50%	11.44%	13.59%
Model Bias(SSPB)	(1.80%	1.36%	(1.56%
ROPE:			
Dyn. Pred. (MdSA)	7.60%	10.34%	10.98%
Model Bias(SSPB)	(1.53%	(1.97%	(1.92%
Calibration (CES)	8.97%	7.61%	7.50%

3.3. Reduced-Order Probabilistic Emulator

The 3-tier hierarchical approach of first combining models within an architecture via a weighted average and then combining the various architectures through a simple mean provides this work's final product, a ROPE of RAM-SCB particle flux.

A summary of our ROPE's final performance metrics is shown in Table 5, 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 data sets. 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 4). The biases stayed about the same between 1% and 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 (3.92%, 6.03%, and 10.83%) MdSA, respectively.

The model ensemble, consisting of 25 LSTMs, delivers a striking result in terms of time efficiency during dynamic predictions. The entire evaluation process for predicting the 5 week-long simulations within each validation and test data set is accomplished in a mere 110 s. In stark contrast, the parallel execution of the same 5 week-long simulations using RAM-SCB, configured with identical settings and computational resources, demands an extensive 38.2 hr. However, it is important to note that while the emulator predicts omnidirectional flux at a single energy level without pitch angle information, RAM-SCB predicts a much higher dimensionality space. Additionally, the ROPE is valid within the training bounds and represents only one configuration of the RAM-SCB model. Although RAM-SCB itself is physics-based and configurable, the emulator offers advantages in routine prediction, low-resource systems, large-scale analyses, and scenarios where adding probabilistic output is advantageous. 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 $2(\sigma)$ bounds. Upon visual inspection, the first two 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 5), we expect that approx. 95% of the ROPE's predictions will fall within the $2\text{-}\sigma$ bounds. The actual observed percentages for the first two 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 data set is provided in Table 5, with scores less than 10%. These scores are interpreted as the emulator's reduced-space predictions having 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. 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).

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 biggest limitation. Thus, subsequent work will expand this workflow to encompass the full energy spectrum and pitch angle distribution found in the particle flux data product.

While acknowledging that the inner magnetosphere contains nonlinear dynamics (e.g., Daglis et al., 2023), necessitating consideration of nonlinear dimensionality reduction methods, our work opts for linear techniques such as PCA. This choice, however, is a limitation. The future expansion to encompass RAM-SCB's full energy spectrum may require exploring nonlinear methods like kernel PCA or convolutional autoencoder (CAE).

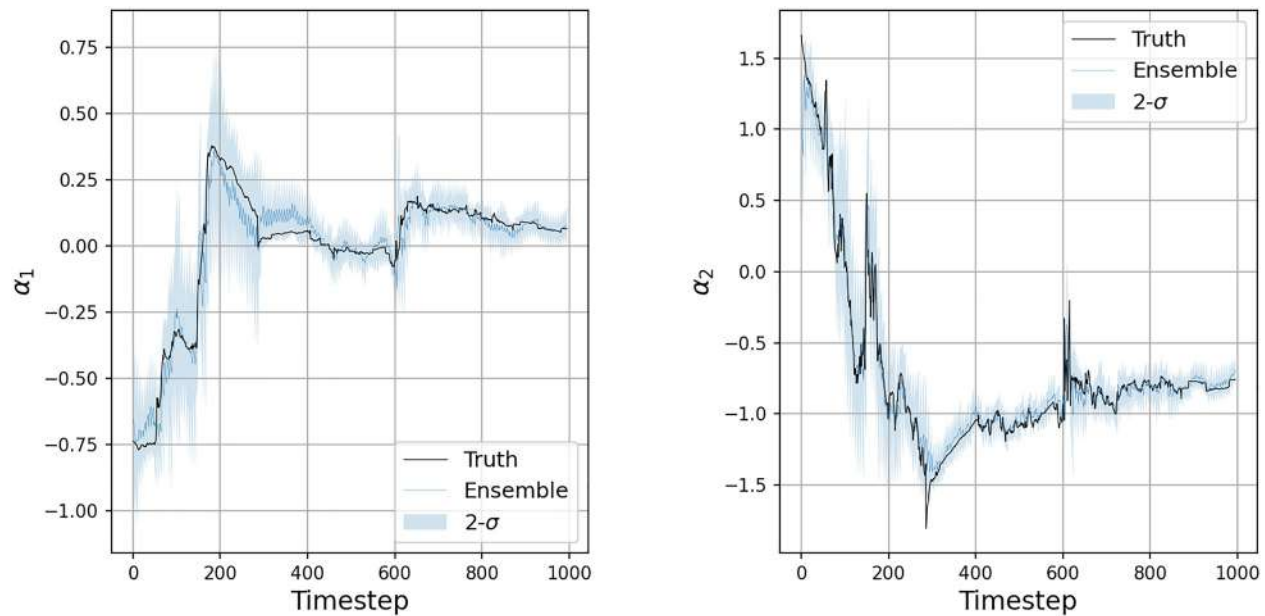


Figure 9. Hourly dynamic predictions of the first 2 reduced-space coefficients (α_1 and α_2) by the reduced-order probabilistic emulator on the TST 3 simulation. The prediction (blue) is plotted at each timestep along with the truth (black) and 2σ bounds (light blue).

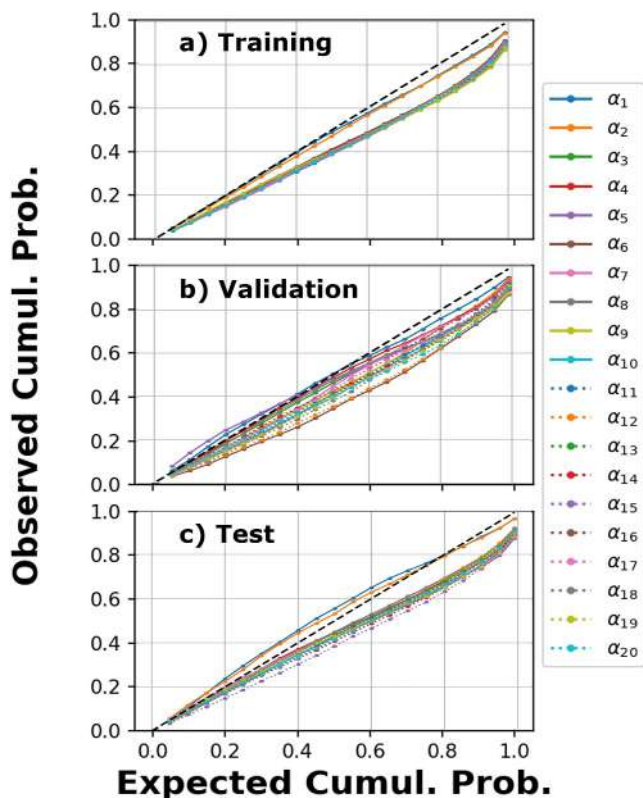


Figure 10. ROPE's calibration curves for the machine learning data sets (training, validation, and test). Each reduced-space coefficient has its own curve, where the first 10 are plotted in solid lines and the remaining 10 have dotted lines. The black dashed line represents the perfectly calibrated $y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

Incorporating nonlinear dimensionality reduction methods could help capture these dynamics and potentially alleviate the significant error spikes observed during periods of heightened solar and geomagnetic activity in our study, which can be attributed, in part, to the use of linear PCA for 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 two reduced-space coefficients contained roughly 93% of the ground truth values in their 2σ 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.

5. Summary

This study extends Licata and Mehta's (2023) emulation process to ring current dynamics, creating a ROPE with 25 LSTM models trained on 20 weeks-long RAM-SCB simulations. A hierarchical ensemble combines these models for a probabilistic prediction with a robust uncertainty estimate. The data that was fed to this mechanism comes from a unique sampling technique where 30 events spanning 20 years of solar and geomagnetic activity are transformed into reduced-space representations through PCA decomposition.

The emulator demonstrates its effectiveness through consistently low errors. The hyperparameter tuner's performance, evaluated with a one-step prediction, yields metrics of around 5% MdSA overall. However, more

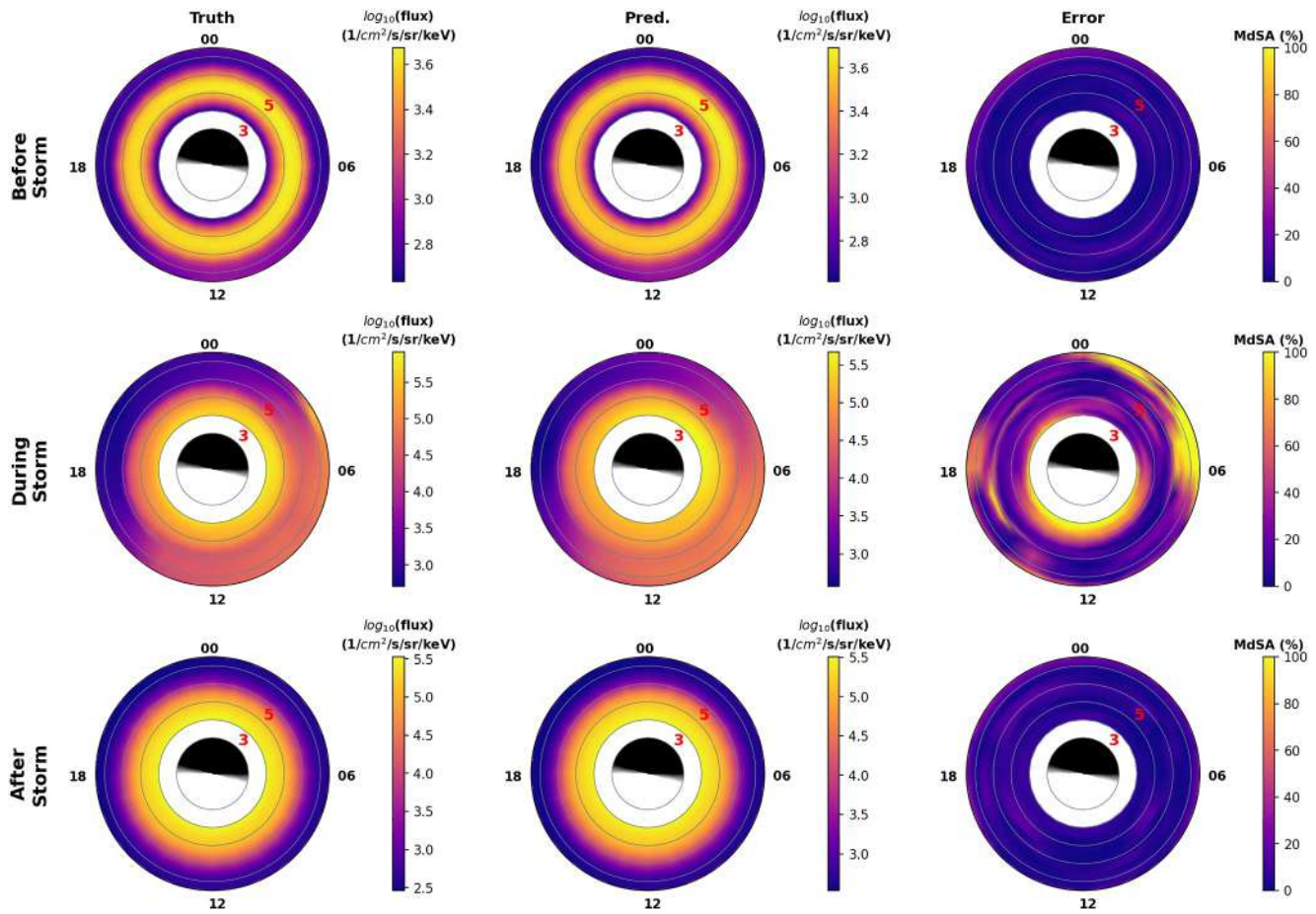


Figure 11. Snapshots taken before, during, and after the geomagnetic storm in the TST 1 simulation with the prediction errors (right) between the actual (left) and reconstructed reduced-order probabilistic emulator hourly dynamic predictions (middle), plotted on RAM-SCB's grid.

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 PA distribution, we expect the hyperparameter tuner to provide a much more diverse set of architectures. With all that said, our emulator achieves a remarkable speed over RAM-SCB, offering approximately 10% MdSA for hourly dynamic predictions. This substantial enhancement was achieved within a mere 110 s, a noteworthy improvement compared to the 38.2 hr required for week-long RAM-SCB simulations. The emulator predicts omnidirectional flux at a single energy level, contrasting with RAM-SCB's prediction of a higher dimensionality space; however, it offers advantages such as routine prediction and low-resource system suitability due to its probabilistic output capabilities and efficient performance.

Data Availability Statement

The OMNIWeb data used in this paper can be downloaded at (OMNIWeb, 2020). The RAM-SCB source code (Jordanova, Engel et al., 2022; Jordanova & Morley et al., 2022) can be found at Github (2022) (Github repo.Lanl 732 RAM-SCB, 2022), 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 data sets, and code to run ROPE are available at Cruz et al. (2024).

Acknowledgments

PMM and AAC would like to acknowledge support from NSF Grant 1929127 and DoE Grant DE-SC0020294. Contributions by SKM, HCG, and VKJ were performed at Los Alamos National Laboratory under the auspices of the US Department of Energy. This research has also been made possible by NASA's West Virginia Space Grant Consortium 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., et al. (2015). TensorFlow: Large-Scale machine learning on heterogeneous systems. Retrieved from [https://www.tensorflow.org/\(Softwareavailablefromtensorflow.org\)](https://www.tensorflow.org/(Softwareavailablefromtensorflow.org))
- Anaconda Software Distribution. (2020). Anaconda Inc. Retrieved from <https://docs.anaconda.com/>
- Audouze, C., de Vuyst, F., & Nair, P. B. (2009). Reduced-order modeling of parameterized PDEs using time-space-parameter principal component analysis.
- 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*, 90(A7), 6387–6394. <https://doi.org/10.1029/JA090iA07p06387>
- Bjornsson, H., & Venegas, S. (1997). *A manual for EOF and SVD analyses of climate data*. McGill University.
- Borovsky, E. J. (2020). A survey of geomagnetic and plasma time lags in the solar-wind-driven magnetosphere of Earth. *Journal of Atmospheric and Solar-Terrestrial Physics*, 208, 105376. <https://doi.org/10.1016/j.jastp.2020.105376>
- 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., et al. (2019). RBSP-ECT combined spin-averaged electron flux data product. *Journal of Geophysical Research: Space Physics*, 124(11), 9124–9136. <https://doi.org/10.1029/2019ja026733>
- Carlberg, K., & Farhat, C. (2011). A low-cost, goal-oriented “compact proper orthogonal decomposition” basis for model reduction of static systems. *International Journal for Numerical Methods in Engineering*, 86(3), 381–402. <https://doi.org/10.1002/nme.3074>
- Cruz, A. A., Siddalingappa, R., Mehta, P., Morley, S. K., Godinez, H. C., & Jordanova, V. K. (2024). RAM-SCB ROPE dataset and machine learning models with code. *Zenodo*. <https://doi.org/10.5281/zenodo.10798662>
- Daglis, I. A., Thorne, R. M., Baumjohann, W., & Orsini, S. (1999). The terrestrial ring current: Origin, formation, and decay. *Reviews of Geophysics*, 37(4), 407–438. <https://doi.org/10.1029/1999rg900009>
- Denton, M. H., Henderson, M. G., Jordanova, V. K., Thomsen, M. F., Borovsky, J. E., Woodroffe, J., et al. (2016). An improved empirical model of electron and ion fluxes at geosynchronous orbit based on upstream solar wind conditions. *Space Weather*, 14(7), 511–523. <https://doi.org/10.1002/2016SW001409>
- Deutsch, J. L., & Deutsch, C. V. (2012). Latin hypercube sampling with multidimensional uniformity. *Journal of Statistical Planning and Inference*, 142(3), 763–772. <https://doi.org/10.1016/j.jspi.2011.09.016>
- DeZeeuw, D. L., Sazykin, S., Wolf, R. A., Gombosi, T. I., Ridley, A. J., & Tóth, G. (2004). Coupling of a global MHD code and an inner magnetospheric model: Initial results. *Journal of Geophysical Research*, 109(A12). <https://doi.org/10.1029/2003JA010366>
- 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. <https://doi.org/10.5555/1953048.2021068>
- Elvidge, S., Godinez, H. C., & Angling, M. J. (2016). Improved forecasting of thermospheric densities using multi-model ensembles. *Geoscientific Model Development*, 9(6), 2279–2292. <https://doi.org/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. <https://doi.org/10.1029/2022SW003356>
- Engel, M. A., Morley, S. K., Henderson, M. G., Jordanova, V. K., Woodroffe, J. R., & Mahfuz, R. (2019). 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. <https://doi.org/10.1029/2018ja026260>
- Fok, M.-C., Kang, S.-B., Ferradas, C. P., Buzulukova, N. Y., Gloer, A., & Komar, C. M. (2021). New developments in the comprehensive inner magnetosphere-ionosphere model. *Journal of Geophysical Research: Space Physics*, 126(4). <https://doi.org/10.1029/2020ja028987>
- Ganushkina, N., Jaynes, A., & Liemohn, M. (2017). Space weather effects produced by the ring current particles. *Space Science Reviews*, 212(3–4), 1315–1344. <https://doi.org/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. <https://doi.org/10.1162/153244303768966139>
- Github, R. (2022). *Lanl RAM-SCB*. Github Repository. Retrieved from <https://github.com/lanl/RAM-SCB/>
- Gondelach, D. J., & Linares, R. (2020). Real-time thermospheric density estimation via two-line element data assimilation. *Advances in the Astronautical Sciences*, 18(2). <https://doi.org/10.1029/2019SW002356>
- Gondelach, D. J., & Linares, R. (2021). Real-time thermospheric density estimation via radar and GPS tracking data assimilation. *Space Weather*, 19(4), e2020SW002620. <https://doi.org/10.1029/2020SW002620>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. Retrieved from <http://www.deeplearningbook.org>
- Harel, M., Wolf, R., Spiro, R., Reiff, P., Chen, C., Burke, W., et al. (1981). Quantitative simulation of a magnetospheric substorm 2. Comparison with observations. *Journal of Geophysical Research*, 86(A4), 2242–2260. <https://doi.org/10.1029/ja086ia04p02242>
- 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.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6), 417–441. <https://doi.org/10.1037/h0071325>
- Huang, F., Xie, G., & Xiao, R. (2009). Research on ensemble learning. In *International conference on artificial intelligence and computational intelligence* (Vol. 3, pp. 249–252). <https://doi.org/10.1109/AICI.2009.235>
- Issan, O., & Kramer, B. (2023). Predicting solar wind streams from the inner-heliosphere to Earth via shifted operator inference. *Journal of Computational Physics*, 473, 111689. <https://doi.org/10.1016/j.jcp.2022.111689>
- Jordanova, V. K., Engel, M. A., Morley, S. K., Welling, D. T., Yu, Y., Yakymenko, K., et al. (2022a). RAM-SCB. *Zenodo*. <https://doi.org/10.5281/ZENODO.6977287>
- Jordanova, V. K., Ilie, R., & Chen, M. W. (2020). *Ring current investigations: The quest for space weather prediction*. Elsevier. <https://doi.org/10.1016/C2017-0-03448-1>
- Jordanova, V. K., Kozyra, J. U., Khazanov, G. V., Nagy, A. F., Rasmussen, C. E., & Fok, M.-C. (1994). A bounce-averaged kinetic model of the ring current ion population. *Geophysical Research Letters*, 21(25), 2785–2788. <https://doi.org/10.1029/94GL02695>

- Jordanova, V. K., Miyoshi, Y. S., Zaharia, S., Thomsen, M. F., Reeves, G. D., Evans, D. S., et al. (2006). Kinetic simulations of ring current evolution during the geospace environment modeling challenge events. *Journal of Geophysical Research*, 111(A11). <https://doi.org/10.1029/2006ja011644>
- Jordanova, V. K., Morley, S. K., Engel, M. A., Godinez, H., Yakymenko, K., Henderson, M. G., et al. (2022). The RAM-SCB model and its applications to advance space weather forecasting. *Advances in Space Research*, 72(12), 5596–5606. <https://doi.org/10.1016/j.asr.2022.08.077>
- Jordanova, V. K., Thorne, R. M., Li, W., & Miyoshi, Y. (2010). Excitation of whistler mode chorus from global ring current simulations. *Journal of Geophysical Research*, 115(A5). <https://doi.org/10.1029/2009ja014810>
- Jordanova, V. K., Welling, D. T., Zaharia, S. G., Chen, L., & Thorne, R. M. (2012). Modeling ring current ion and electron dynamics and plasma instabilities during A high-speed stream driven storm. *Journal of Geophysical Research*, 117(A9). <https://doi.org/10.1029/2011ja017433>
- Jordanova, V. K., Zaharia, S., & Welling, D. T. (2010). Comparative study of ring current development using empirical, dipolar, and self-consistent magnetic field simulations. *Journal of Geophysical Research*, 115(A12). <https://doi.org/10.1029/2010ja015671>
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. <https://doi.org/10.48550/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. <https://doi.org/10.5194/acp-14-11791-2014>
- Kozyra, J., Borovsky, J., Chen, M., Fok, M.-C., & Jordanova, V. (1998). Plasma sheet preconditioning, enhanced convection and ring current development (Vol. 238, pp. 755–760). Substorms-4. https://doi.org/10.1007/978-94-011-4798-9_158
- Kozyra, J., Liemohn, M., Clauer, C., Ridley, A., Thomsen, M., Borovsky, J., et al. (2002). Multistep Dst development and ring current composition changes during the 4–6 June 1991 magnetic storm. *Journal of Geophysical Research*, 107(A8), SMP–33. <https://doi.org/10.1029/2001JA000023>
- Laves, M.-H., Ihler, S., Fast, J. F., Kahrs, L. A., & Ortmaier, T. (2021). 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(MIDL 2020), 1–26. <https://doi.org/10.48550/ARXIV.2104.12376>
- Licata, R. J., & Mehta, P. M. (2022). Uncertainty quantification techniques for data-driven space weather modeling: Thermospheric density application. *Scientific Reports*, 12(1), 7256. <https://doi.org/10.1038/s41598-022-11049-3>
- Licata, R. J., & Mehta, P. M. (2023). Reduced order probabilistic emulation for physics-based thermosphere models. *Space Weather*, 21(5). <https://doi.org/10.1029/2022sw003345>
- Licata, R. J., Mehta, P. M., Tobiska, W. K., & Huzurbazar, S. (2022). Machine-learned HASDM thermospheric mass density model with uncertainty quantification. *Space Weather*, 20(4). <https://doi.org/10.1029/2021sw002915>
- Lumley, J. L. (1967). The structure of inhomogeneous turbulent flows.
- Maggiolo, R., Hamrin, M., Keyser, J. D., Pitkänen, T., Cessateur, G., Gunell, H., & Maes, L. (2017). The delayed time response of geomagnetic activity to the solar wind. *Journal of Geophysical Research: Space Physics*, 122(11). <https://doi.org/10.1002/2016ja023793>
- Matsuo, T., & Forbes, J. M. (2010). Principal modes of thermospheric density variability: Empirical orthogonal function analysis of CHAMP 2001–2008 data. *Journal of Geophysical Research*, 115(A7). <https://doi.org/10.1029/2009ja015109>
- Maulik, R., Rao, V., Wang, J., Mengaldo, G., Constantinescu, E., Lusch, B., et al. (2022). Efficient high-dimensional variational data assimilation with machine-learned reduced-order models. *Geoscientific Model Development*, 15(8), 3433–3445. <https://doi.org/10.5194/gmd-15-3433-2022>
- 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. <https://doi.org/10.1002/2015JA021828>
- Mehta, P. M., & Linares, R. (2017). A methodology for reduced order modeling and calibration of the upper atmosphere. *Space Weather*, 15(10), 1270–1287. <https://doi.org/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. <https://doi.org/10.1029/2018SW001875>
- Mehta, P. M., Linares, R., & Sutton, E. K. (2018). A quasi-physical dynamic reduced order model for thermospheric mass density via hermitian space-dynamic mode decomposition. *Space Weather*, 16(5), 569–588. <https://doi.org/10.1029/2018sw001840>
- Montavon, G., Orr, G., & Müller, K.-R. (2012). *Neural networks: Tricks of the trade* (Vol. 7700). Springer. <https://doi.org/10.1007/978-3-642-35289-8>
- Morley, S. K., Brito, T. V., & Welling, D. T. (2018). Measures of model performance based on the log accuracy ratio. *Space Weather*, 16(1), 69–88. <https://doi.org/10.1002/2017sw001669>
- Morley, S. K., Jordanova, V. K., Zheng, Y., & Petrenko, M. (2023). RAM-SCB runs on request at CCMC. *Space Weather*, 21(12). <https://doi.org/10.1029/2023SW003771>
- Morley, S. K., & 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. *Annales Geophysicae*, 24(3), 961–972. <https://doi.org/10.5194/angeo-24-961-2006>
- Morley, S. K., Welling, D., & Woodroffe, J. (2018). Perturbed input ensemble modeling with the space weather modeling framework. *Space Weather*, 16(9), 1330–1347. <https://doi.org/10.1029/2018SW002000>
- NVIDIA. (2022). NVIDIA corporation. In *CUDA toolkit 11.0*. NVIDIA Corporation. Retrieved from <https://developer.nvidia.com/cuda-toolkit>
- O'Malley, T., Bursztin, E., Long, J., Chollet, F., Jin, H., & Invernizzi, L. (2019). KerasTuner. Retrieved from <https://github.com/keras-team/keras-tuner>
- OMNIWeb. (2020). NASA space physics data facility. Retrieved from https://omniweb.gsfc.nasa.gov/form/omni_min.html
- Parker, E. N. (1987). Sweet's mechanism for merging magnetic fields in conducting fluids. *Journal of Geophysical Research*, 62(4), 509–520. <https://doi.org/10.1029/JZ062i004p00509>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Puzyrev, V., Ghommem, M., Meka, S., & Calo, V. (2018). Reduced order modeling in geophysics: PyROM open-source framework. In *80th EAGE conference and exhibition 2018* (Vol. 2018, pp. 1–5). European Association of Geoscientists and Engineers. <https://doi.org/10.3997/2214-4609.201801350>
- Rekoske, J. M., Gabriel, A. A., & May, D. A. (2023). Instantaneous physics-based ground motion maps using reduced-order modeling. arXiv preprint arXiv. 2212. <https://doi.org/10.1029/2023JB026975>.11335
- 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. <https://doi.org/10.48550/arXiv.1206.2944>
- Soltanzadeh, I., Azadi, M., & Vakili, G. A. (2011). Using bayesian model averaging (BMA) to calibrate probabilistic surface temperature forecasts over Iran. *Annales Geophysicae*, 29(7), 1295–1303. <https://doi.org/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. <https://doi.org/10.1029/ja094ia05p05264>
- Stumpo, M., Consolini, G., Alberti, T., & Quattrociochi, V. (2020). Measuring information coupling between the solar wind and the magnetosphere–ionosphere system. *Entropy*, 22(3), 276. <https://doi.org/10.3390/e22030276>
- Toffoletto, F., Sazykin, S., Spiro, R., & Wolf, R. (2003). Inner magnetospheric modeling with the rice convection model. *Space Science Reviews*, 107(1/2), 175–196. <https://doi.org/10.1023/A:1025532008047>
- Wang, P., Chen, Z., Deng, X., Wang, J., Tang, R., Li, H., et al. (2022). The prediction of storm-time thermospheric mass density by LSTM-based ensemble learning. *Space Weather*, 20(3). <https://doi.org/10.1029/2021sw002950>
- Waskom, M. L. (2021). Seaborn: Statistical data visualization. *Journal of Open Source Software*, 6(60), 3021. <https://doi.org/10.21105/joss.03021>
- Weglarczyk, S. (2018). Kernel Density Estimation and its Application. *ITM Web of Conferences*, 23, 00037. <https://doi.org/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. <https://doi.org/10.1002/qj.210>
- Wilks, D. (2011). Principal component (EOF) analysis. In *International geophysics* (pp. 519–562). Elsevier. <https://doi.org/10.1016/b978-0-12-385022-5.00012-9>
- Wilks, D. S. (2011). *Statistical methods in the atmospheric sciences*. Elsevier.
- Wolf, R., Harel, M., Spiro, R., Voigt, G., Reiff, P., & Chen, C. (1982). Computer simulation of inner magnetospheric dynamics for the magnetic storm of July 29, 1977. *Journal of Geophysical Research*, 87(A8), 5949–5962. <https://doi.org/10.1029/ja087ia08p05949>
- Wolf, R. A., & Kamide, Y. (1983). Inferring electric fields and currents from ground magnetometer data: A test with theoretically derived inputs. *Journal of Geophysical Research*, 88(A10), 8129–8135. <https://doi.org/10.1029/JA088iA10p08129>
- 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. <https://doi.org/10.1016/j.cmpb.2017.09.005>
- Yu, Y., Jordanova, V., Zaharia, S., Koller, J., Zhang, J., & Kistler, L. M. (2012). Validation study of the magnetically self-consistent inner magnetosphere model RAM-SCB. *Journal of Geophysical Research*, 117(A3). <https://doi.org/10.1029/2011ja017321>
- Yu, Y., Su, S., Cao, J., Jordanova, V. K., & Denton, M. H. (2022). Improved boundary conditions for coupled geospace models: An application in modeling spacecraft surface charging environment. *Space Weather*, 20(9). <https://doi.org/10.1029/2022SW003178>
- Zaharia, S., Jordanova, V. K., Thomsen, M. F., & Reeves, G. D. (2006). Self-consistent modeling of magnetic fields and plasmas in the inner magnetosphere: Application to a geomagnetic storm. *Journal of Geophysical Research*, 111(A11). <https://doi.org/10.1029/2006ja011619>
- Zeiler, M. D. (2012). *Adadelta: An adaptive learning rate method*. arXiv preprint arXiv:1212.5701. <https://doi.org/10.48550/arXiv.1212.5701>
- Zheng, Y., Ganushkina, N. Y., Jiggins, P., Jun, I., Meier, M., Minow, J. I., et al. (2019). 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. <https://doi.org/10.1029/2018sw002042>