Neural Operator for Accelerating Coronal Magnetic Field Model

Yutao Du
NJIT
Newark, New Jersey, USA
yd288@njit.edu

Qin Li
NJIT
Newark, New Jersey, USA
ql47@njit.edu

Raghav Gnanasambandam
Virginia Tech
Blacksburg, Virginia, USA
raghavg@vt.edu

Mengnan Du
NJIT
Newark, New Jersey, USA
mengnan.du@njit.edu

Haimin Wang
NJIT
Newark, New Jersey, USA
haimin.wang@njit.edu

Bo Shen*
NJIT
Newark, New Jersey, USA
bo.shen@njit.edu

ABSTRACT

Studying the sun's outer atmosphere is challenging due to its complex magnetic fields impacting solar activities. Magnetohydrodynamics (MHD) simulations help model these interactions but are extremely time-consuming (usually on a scale of days). Our research applies the Fourier Neural Operator (FNO) to accelerate the coronal magnetic field modeling, specifically, the Bifrost MHD model. We apply Tensorized FNO (TFNO) to generate solutions from partial differential equations (PDEs) over a 3D domain efficiently. TFNO's performance is compared with other deep learning methods, highlighting its accuracy and scalability. Physics analysis confirms that TFNO is reliable and capable of accelerating MHD simulations with high precision. This advancement improves efficiency in data handling, enhances predictive capabilities, and provides a better understanding of magnetic topologies.

CCS CONCEPTS

• **Applied computing** → *Astronomy*; • **Mathematics of computing** → *Partial differential equations*; • **Computing methodologies** → *Machine learning approaches.*

KEYWORDS

Tensorized Fourier Neural Operator (TFNO), Solar Outer Atmosphere Simulation, Magnetohydrodynamics (MHD)

ACM Reference Format:
Yutao Du, Qin Li, Raghav Gnanasambandam, Mengnan Du, Haimin Wang, and Bo Shen. 2024. Neural Operator for Accelerating Coronal Magnetic Field Model. In . ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Studying the solar outer atmosphere has been a longstanding challenge in astrophysics, with its complex magnetic structures playing a crucial role in various solar phenomena [3, 16, 34]. In the convection zone, the gas pressure dominates the magnetic field pressure, causing the plasma to move and carry the field along with it. These movements generate energetic flows and mass transfer from the chromosphere to the corona. Most of the energy conveyed to the outer solar atmosphere, as a result of exertion on the magnetic fields, is emitted in the chromosphere. In addition, it is in the chromosphere that the dynamics transition from being dominated by gas pressure to being dominated by magnetic force. Consequently, understanding and precisely simulating these intricate interactions are imperative for the progression of our understanding of space weather and its ramifications on the planet Earth.

Coronal magnetic field modeling is mainly divided into magnetohydrodynamics (MHD) models [4, 17, 38], magnetohydrostatics (MHS) models [31], force-free models [18, 19, 37], and potential-field models [36]. These models solve specific partial differential equations (PDEs), ranging from complex magnetohydrodynamics processes to simplified assumptions of current-free conditions, to simulate and understand the structure and dynamics of the coronal magnetic field. These equations are discretized and solved on a grid, but the wide range of spatial and temporal scales and the complex nonlinear interactions between these scales require fine grids and high resolution, making these modeling extremely expensive. In recent years, there have been many attempts by researchers to simulate the solar atmosphere by computing codes for MHD. Leenaarts et al. [24] simulated the solar atmosphere from the convection zone to the corona. Den [7] presented a three-dimensional MHD simulation code designed to simulate space plasma phenomena with applications in modeling the solar surface and global solar wind structure. Carlsson et al. [6] used the Bifrost MHD to provide the community with a realistic simulated magnetic field of the Sun's outer atmosphere by elucidating the complex interactions between the magnetic field and the plasma. The Bifrost MHD are also used for performance evaluation of coronal field reconstruction using various approaches and tools [9, 10]. The Bifrost-based MHD model incorporates many physical processes and offers higher spatial and temporal resolution, focusing on the chromosphere and corona. This makes it well-suited for our study of predicting magnetic field topologies in corona.

However, MHD simulations are extremely time-consuming. To reduce the computational cost of MHD simulations, researchers have attempted to solve the MHD equations using artificial intelligence (AI)/machine learning (ML) methods to simulate the solar exosphere. AI/ML methods can solve the PDE equations several orders

[*] Corresponding Author.

], ACM Conference,
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

of magnitude faster compared to the traditional approach by providing a fast solver that approximates or augments the traditional approach [2, 14, 27, 28]. More and more researchers have been trying to use ML methods to solve MHD equations. For instance, Yang and Shen [39] devised a method for globally mapping solar wind parameters at the source surface using an Artificial Neural Network (ANN) approach. This model efficiently tackles the MHD equations, employing ANN to adeptly forecast solar wind conditions. However, a notable limitation of ANN models, as highlighted in their research, is their inherent challenge in fully capturing the complex and nonlinear dynamics of solar wind acceleration and interactions within the corona. Similarly, Baty and Vigon [1] presents a groundbreaking method to simulate solar coronal magnetic fields using Physics-Informed Neural Network (PINN) [12, 20]. This technique integrates the core partial differential equations of solar physics directly into the neural network, facilitating accurate modeling of MHD equilibria in the solar corona. This progress highlights the powerful application of AI/ML techniques to improve the modeling and predictive analysis of magnetic fields in the solar coronal environment. However, a significant limitation of PINNs is their inability to adapt to changes in system configurations or parameters without retraining. For instance, when solving the high-dimensional Navier-Stokes equations, each parameter change typically requires a new and specifically trained model. This requirement not only increases the computational cost but also restricts the method's capability to efficiently handle and capture multi-scale phenomena.

Fourier neural operator (FNO) [26] has been introduced to solve this issue and improve the scalability of deep learning methods. The FNO represents a significant advancement over traditional physics-based modeling, as it is mesh-free, can transfer solutions between different grid geometries, and is significantly faster than conventional PDE solvers. In addition, FNO demonstrates computational efficiency surpassing standard vision transformers, and it effectively captures global dependencies compared to convolutional neural networks, graphical neural networks, and other deep learning methods. Despite the numerous advantages associated with FNOs, there is no documented application of FNOs to the solar coronal magnetic field modeling.

In this study, Tensorized Fourier Neural Operator (TFNO) [23] is employed to accelerate coronal magnetic field modeling (i.e., the Bifrost-based MHD model). TFNO incorporates tensor decomposition [22, 32] to model the kernel operator, which is data efficient and highly parallelizable with reduced memory requirement and better generalization. We use data from Carlsson et al. [6] to demonstrate the effectiveness and efficiency of TFNO by comparing it with state-of-the-art deep learning methods such as the Vision Transformer [8], CNN-RNN [35], and CNN-LSTM [15]. In addition to the evaluation metrics from an AI/ML perspective, evaluation from a physics perspective is conducted to verify that our prediction from TFNO is reliable.

2 PROBLEM SETTING

2.1 Bifrost-based MHD Model

Bifrost is a flexible and massively parallel code described in detail in [13]. The Bifrost-based MHD model represents the realistic simulations of the magnetic solar outer atmosphere, with the 3D magnetic topologies on an enhanced area on the Sun. Bifrost has evolved from earlier numerical codes developed by Nordlund et al. [29] and Galsgaard and Nordlund [11], these codes share a common core. One of the key features of the Bifrost-based MHD model is its ability to simulate the multi-scale dynamics of solar phenomena, ranging from small-scale magnetic reconnection events to large-scale coronal mass ejections (CMEs). It incorporates realistic physics processes, such as radiative transfer, non-equilibrium ionization, and thermal conduction, to provide a comprehensive understanding of solar atmospheric phenomena. Specifically, a staggered mesh explicit code that solves the standard MHD PDEs on a Cartesian grid as follows

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \mathbf{u}) \tag{1}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} = -\nabla \cdot (\rho \mathbf{u}\mathbf{u} - \tau) - \nabla P + \mathbf{J} \times \mathbf{B} + \rho \mathbf{g} \tag{2}$$

$$\mu \mathbf{J} = \nabla \times \mathbf{B} \tag{3}$$

$$\mathbf{E} = \eta \mathbf{J} - \mathbf{u} \times \mathbf{B} \tag{4}$$

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E} \tag{5}$$

$$\frac{\partial e}{\partial t} = -\nabla \cdot (e\mathbf{u}) - P\nabla \cdot \mathbf{u} + Q \tag{6}$$

where ρ, $\mathbf{u}$, e, $\mathbf{B}$ are the density, the velocity vector, the internal energy per unit volume, and the magnetic field intensity respectively. τ, P, $\mathbf{J}$, $\mathbf{g}$, μ, $\mathbf{E}$ and η are the stress tensor, the gas pressure, the electric current density vector, the gravitational acceleration, the vacuum permeability, the electric field vector and the magnetic diffusivity respectively. The quantity Q can contain many terms, depending on the individual experiment. It could for instance contain a term from the chosen Equation Of State (EOS), a term containing the effect of the Spitzer thermal conductivity, a term from radiative transfer, etc. The EOS needed to close this set of equations can be anything from a simple ideal gas EOS to a complex EOS including detailed microphysics [13].

2.2 Tensorized Fourier Neural Operators

The idea of the Fourier Neural Operator (FNO) is motivated by Green's function [40]. To solve a differential equation $Lu = f(\mathbf{x})$ with an inhomogeneous term $f(\mathbf{x})$ and a linear operator L, one would express the solution $u(\mathbf{x})$ as an integral over the domain D where the problem is defined:

$$u(\mathbf{x}) = \int_D G(\mathbf{x}, \mathbf{y}) f(\mathbf{y})\, d\mathbf{y} \tag{7}$$

where $\mathbf{x}, \mathbf{y} \in \mathcal{D} \subseteq \mathbb{R}^d$ and G is the Green's function of the operator L. If there exists a function g such that $G(\mathbf{x}, \mathbf{y}) = g(\mathbf{x} - \mathbf{y})$, Equation (7) reduces to a convolution operation, and G serves as the convolution kernel. FNO learns the operator kernel by directly optimizing the kernel within the Fourier frequency domain using the Fast Fourier Transform (FFT), which has proved to be an efficient way to reduce the computational cost. For a parametric PDE, Green's function also depends on the parameter. According to [26], the kernel integral operator is defined by

$$(\mathcal{K}_\theta(v) h_i)(\mathbf{x}) = \int_D \mathcal{K}_\theta(\mathbf{x}, \mathbf{y}, v(\mathbf{x}), v(\mathbf{y})) h_i(\mathbf{y})\, d\mathbf{y} \tag{8}$$

where $v(\mathbf{x})$ is the input function and the kernel integral operator $\mathcal{K}_\theta$ is parameterized by a neural architecture with learnable parameters θ. In particular, h_i denotes the latent representation generated by a Fourier layer in a sequence of Fourier layers indexed by i.

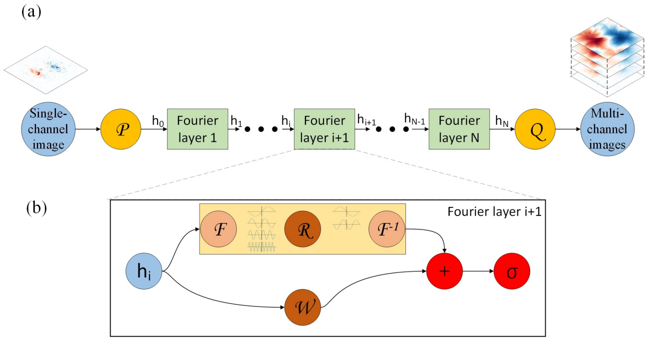


Figure 1: (a) The architecture of the fourier neural operators; (b) Fourier layer.

For a better understanding of the structure of a Fourier layer, we visualize it in Figure 1. The input at the Fourier layer $i + 1$, $h_i(\mathbf{x})$ undergoes two transformations. $h_i(\mathbf{x})$ is transformed to the Fourier space where a linearly transform $\mathcal{R}$ filters out the high-level Fourier modes, and finally the inverse Fourier transform $\mathcal{F}^{-1}$ converts back into the original space. Unlike [25] in which message passing is used as in a graph neural network, FNO leverages the convolution theorem, which states convolution is equivalent to point-wise multiplication in the Fourier space. Therefore, FNO initializes complex-valued learnable parameters directly in the Fourier space. Following the notations in [26], Fourier integral operator is defined by

$$\mathcal{K}_\theta(v)h_i = \mathcal{F}^{-1}(\mathcal{F}(\mathcal{K}_\theta) \cdot \mathcal{F}(h_i)) = \mathcal{F}^{-1}(R_\phi \mathcal{F}(h_i)) \qquad (9)$$

where R_ϕ is the Fourier transform of $\mathcal{K}$, i.e., $R_\phi = \mathcal{F}(\mathcal{K}_\theta)$, and output is real-valued in the latent space. In the implementation, the fast Fourier transform and its inverse is used to preserve computational efficiency. The bottom path input $h_i(x)$ applies a local linear transform $\mathcal{W}$, where the discretized $\mathcal{W}$ is a matrix of learnable parameters. The results of the two paths are then added together, followed by a nonlinear activation function σ.

With the Fourier layer defined, Figure 1 also shows the architecture of FNO. FNO incorporates a lifting layer $\mathcal{P}$ which lifts the input tensor into high-dimensional space. A sequence of Fourier layers follows to update the latent representations by

$$h_{i+1} = \sigma(\mathcal{W}_{i+1}h_i(\mathbf{x}) + \mathcal{K}_{\theta_{i+1}}(v)h_i)(\mathbf{x}) \qquad (10)$$

with $i = 1, 2, \ldots, N - 1$. Since R_ϕ, $\mathcal{W}$ differ in each layer, they are indexed by i as well. Finally, layer $\mathcal{Q}$ projects h_N to the solution function space denoted as $u(\mathbf{x})$. In our case, the output is multi-channel images.

As FNO learns the operator to map from input space v to the solution space u, it solves PDEs with variable parameters [26]. For comparison, frameworks such as PINN [30] only solve one specific setting of the PDEs by fixing the boundary/initial conditions and coefficients in PDEs. TFNO extends FNO by parameterizing the learnable weights $\mathcal{W}$ within the Fourier domain with a low-rank tensor factorization (e.g., Tucker, Tensor Train). Thus, TFNO imposes a low-rank constraint on the Fourier domain representation of the convolutional weights within the FNO. This factorization leads to a significant reduction in the number of parameters required to represent the operator, enhancing the model's data efficiency and generalization capabilities [23].

3 EXPERIMENTS

All data used in the experiments are available from Hinode Science Data Centre Europe (https://sdc.uio.no/search/simulations). In our study, we will focus on the 3D magnetic data cube $\mathbf{B}$ along the x, y, and z axes—denoted as B_x, B_y, and B_z, respectively. These components depict the 3D magnetic environment of the region of interest, in our case, an enhanced network area on the Sun. The data are located in the "en024048_hion", a collection of 3D simulated magnetic data cubes generated using the Bifrost code to simulate a computational volume with a magnetic field topology similar to an enhanced network area on the Sun [5]. We use the dataset of B_z as one example to illustrate the data formats (the same applies to B_x and B_y). The first snapshot of B_z was captured at $t = 3850$ seconds and the last snapshot of B_z was captured at $t = 5410$ seconds. Snapshots were recorded at 10-second intervals, resulting in 157 cubes. Therefore, B_z dataset contains $\{B_z^j\}_{j=1}^{157}$. The size of each cube B_z^j is $504 \times 504 \times 496$. The third dimension (i.e., 496) represents the height, which extends from the upper convection zone to the corona and is aimed at aiding the study of the solar chromosphere, a region that is challenging to model due to its complex dynamics and physics [6]. The simulation volume measures 24 Mm × 24 Mm horizontally (504 × 504 pixels), with a depth of 2.4 Mm below the solar surface, and extends 14.4 Mm above, covering the upper part of the convection zone, the photosphere, chromosphere, transition region, and corona.

3.1 Data Prepossessing

Given the constraints (limited memory space) of our GPU server, we downsampled the cube from $504 \times 504 \times 496$ to $504 \times 504 \times 100$ by selecting every five slices with index $1, 6, 11, \ldots, 496$. **Our goal here is to learn the mapping from the bottom-most slice (the boundary condition of PDEs) to the remaining 99 slices as shown in Figure 1 (a)**. The deep learning method inputs the bottom-most image of size 504×504 and outputs a 3D cube of size $504 \times 504 \times 99$. In addition, data normalization is implemented. 90% of the data is randomly selected for training.

3.2 Experiment Settings

In our experiments, we implemented three other deep learning methods to compare with TFNO.

3.2.1 *TNFO Structure.*

The code (https://github.com/neuraloperator/neuraloperator) we used is publicly available. To increase the complexity of the neural network, and thus its expressive power, we set the number of modes retained in each dimension in the Fourier layer to $(64, 64)$ and set the hidden channel to 64, thus enabling the model to efficiently address the complex spatial structure inherent in the data. The model uses the Tucker decomposition with a rank$= 0.42$, which balances expressiveness with computational and storage efficiency. It captures the most important interactions between input data elements while discarding less important ones to create a more compact model. In addition to the basic Lifting Layer, FNO Blocks, and Projection Layer mentioned in Section 2.2, a layer of Multi-Layer Perceptrons (MLPs) is added after the FNO Blocks to output the desired size of the 3D cube (504 × 504 × 99) since the original code can only output 2D image. Additionally, it enables an additional non-linear processing stage, improving its capability to model complex functions and enhancing its ability to learn a diverse array of PDE solutions.

3.2.2 Vision Transformer (ViT). The ViT [8] used in our paper is a modification of the transformer architecture traditionally used for Natural Language Processing (NLP). Our ViT is adapted to process the input image in the form of patches, which are embedded with masks and then passed through the transformer encoder. The model processes 504×504 images using 8×8 patches. With 128-dimensional embedding and a multi-head attention mechanism spanning 2 layers of 8 heads. A learnable mask embedding is added to the patch embedding sequence, and positional embedding is added to the spatial information lost during the patch embedding process. After converter processing, the output is summarized and decoded to the desired number of output channels for our prediction.

3.2.3 CNN-RNN and CNN-LSTM. We use two different encoder-decoder structures: The CNN-RNN [35] and CNN-LSTM [15]. We use Convolutional Neural Networks (CNN) for feature extraction and Recurrent Neural Networks (RNN) or Long Short-Term Memory Networks (LSTM) for modeling sequential images. The CNN component consists of three convolutional layers that progressively increase the number of feature channels from 1 to 256 while downsampling the image dimensions by a factor of 2 after each layer. This is done via convolution with a stride of 2, effectively reducing the spatial dimensions and extracting higher-level features. In CNN-RNN, the main component of the recurrent network is a series of gated recurrent units (GRUs), which are chosen for their ability to efficiently model time series. The hidden state size of the GRU units in this model is 256, which reflects a balance between model complexity and computational efficiency, allowing the network to capture a large amount of temporal information without being too computationally demanding. Meanwhile, the CNN-LSTM uses LSTM cells with a hidden state size of 256, which is consistent with the output dimension of the CNN.

3.2.4 Optimizer and Scheduler. All deep learning methods are trained using the Adam optimizer [21] and the ReduceLROnPlateau scheduler with 300 epochs and a batch size of 10. An Adam optimizer was used with an initial learning rate of 0.001 and a weight decay of 0.0001. The ReduceLROnPlateau scheduler was used to adjust the learning rate according to the performance of the model, with a learning rate reduction factor of 0.8 and a patience of 2. Huber Loss [33] is used for training for all the methods to approximate PDE solutions. It is a hybrid loss function, known for blending the properties of mean squared error and mean absolute error, demonstrating robustness against outliers—a feature when dealing with the intrinsic discontinuities and sharp gradients characteristic of PDEs.

3.3 Results

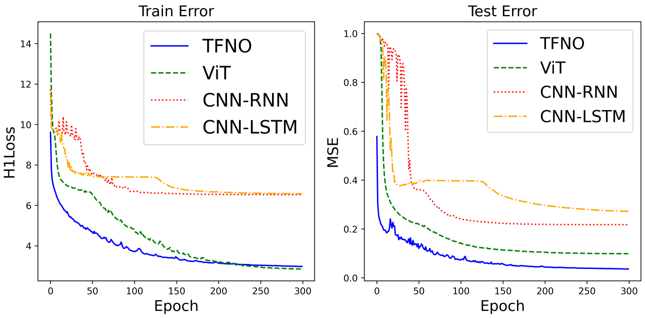


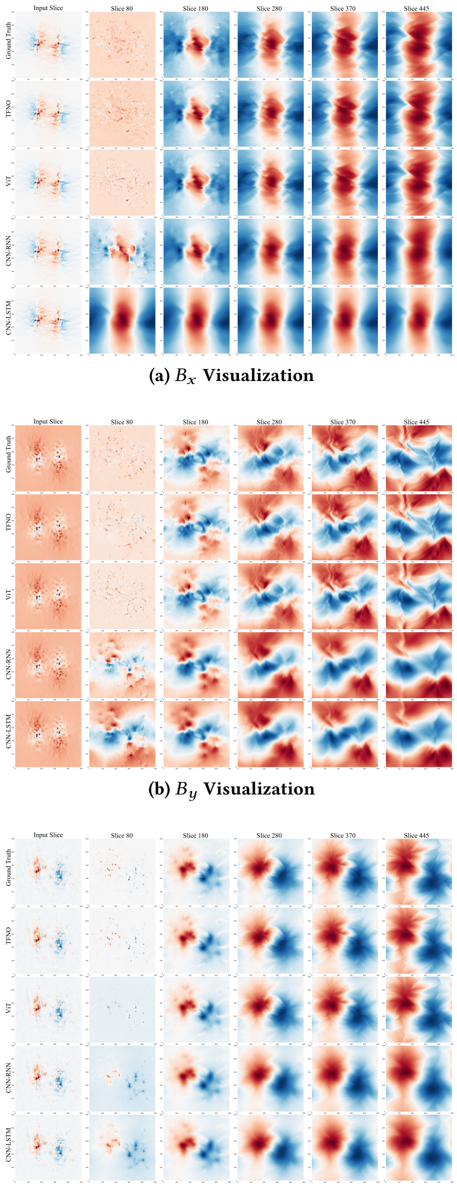
Figure 2: Training and test error curves for different epochs of TFNO, ViT, CNN-RNN, and CNN-LSTM.

3.3.1 Evaluation from an AI/ML Perspective. Figure 2 shows the training error, i.e., Huber Loss, and test error, i.e., Mean Squared Error (MSE), over 300 epochs for the TFNO, ViT, CNN-RNN, and CNN-LSTM. The training error of TFNO and ViT shows an overall decreasing trend and converges to a much lower error than CNN-RNN and CNN-LSTM. Meanwhile, TFNO shows a very stable and smooth training error compared to all other deep learning methods. In addition, we observe a divergence in the generalization capabilities between the ViT and the TFNO. TFNO shows a lower training error before 220 epochs compared to ViT while ViT shows a slightly lower training error after 220 epochs. However, this improvement does not translate equivalently to the test phase, where ViT has a significantly higher test error compared to TFNO. This is caused by overfitting of the ViT, a condition where a model's excessive complexity captures not only the underlying data distribution but also the noise within the training set. This is potentially caused by ViT's parameter-rich architecture that encompasses the largest number of model parameters among all deep learning methods. Conversely, TFNO has consistently better test performance, suggesting a more robust generalization capacity. This pattern highlights the potential advantages of TFNO to mitigate overfitting. While CNN-RNN and CNN-LSTM have similar training errors, CNN-RNN outperforms CNN-LSTM in test error. However, both methods have much higher test errors compared to TFNO and ViT.

Table 1: Test performance comparison of TFNO, ViT, CNN-RNN, and CNN-LSTM for B in terms of Average Training Time (one epoch), MSE, R^2, RE, and MAE.

Model	Time ↓	MSE ↓	R^2 ↑	RE ↓	MAE ↓
TFNO	**31.96s**	**0.0502**	**0.9498**	**0.2205**	**0.0831**
ViT	39.25s	0.1240	0.8759	0.3498	0.1417
CNN-RNN	32.93s	0.2514	0.7485	0.4995	0.2230
CNN-LSTM	37.10s	0.3257	0.6743	0.5686	0.2940

In addition to MSE, Table 1 summarizes the performance of different deep learning methods in terms of Average training time of one epoch, R-square (R^2), Relative Error (RE), and Mean Absolute Error (MAE). TFNO shows the best test performance among all

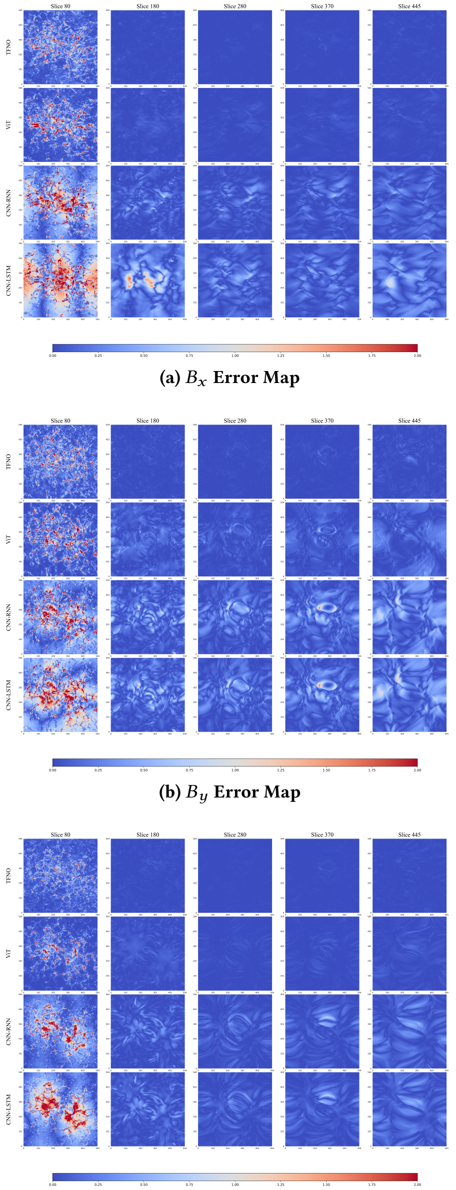


(a) B_x Visualization

(b) B_y Visualization

(c) B_z Visualization

Figure 3: The visualization of prediction from TFNO, ViT, CNN-RNN, and CNN-LSTM, compared with ground truth at different heights: (a) B_x; (b) B_y; (c) B_z.



(a) B_x Error Map

(b) B_y Error Map

(c) B_z Error Map

Figure 4: The visualization of error maps from TFNO, ViT, CNN-RNN, and CNN-LSTM at different heights: (a) B_x; (b) B_y; (c) B_z.

deep learning methods. TFNO has the lowest MSE, RE, and MAE. TFNO achieves $R^2 = 0.9498$ while the ViT achieves the second-best $R^2 = 0.8759$. This verifies that ViT has a suboptimal generalization capability indicated by the lowest training error but a higher test error compared to TFNO. On the other hand, the CNN-RNN and CNN-LSTM (i.e., CNN-based recurrent architectures) show the second-worst and worst test performance, respectively, which are not comparable with TFNO and ViT. Meanwhile, they do not offer computational advantages over ViT as reflected in their average training time for each epoch. Overall, TFNO not only shows the best test performance but also shows the best training time efficiency.

Figure 3 shows the visualization of prediction from TFNO, ViT, CNN-RNN, and CNN-LSTM, compared with ground truth at different heights. In Figures 3a, 3b, and 3c, the selected slices are captured at different heights of the cube. The first column in each Figure is the input data of the deep learning methods. The second through fifth columns correspond to heights of approximately 0 Mm, 2 Mm, 4 Mm, 6 Mm, and 10 Mm, respectively. These slices were chosen to demonstrate the diversity of magnetic field strengths at different heights. The first row represents the "ground truth", which serves as a benchmark for evaluating the subsequent rows – TFNO, ViT, CNN-RNN, and CNN-LSTM.

From these three figures, we can observe that the prediction results of TFNO and ViT are almost the same as the ground truth while CNN-RNN and CNN-LSTM reveal remarkable discrepancies in fine details when compared with the ground truth at heights =0, 2 Mm, 4 Mm, 6 Mm, and 10 Mm. Specifically, CNN-RNN and CNN-LSTM cannot learn the pattern of the ground truth at heights = 0 Mm.

To better visualize the predictions of different deep learning methods, Figure 4 shows the error maps of B_x, B_y, and B_z from TFNO, ViT, CNN-RNN, and CNN-LSTM at different heights of approximately 0 Mm, 2 Mm, 4 Mm, 6 Mm, and 10 Mm. These different heights are corresponding to the first, second, third, fourth, and fifth columns, respectively. These three figures are obtained by subtracting the predicted results from the ground truth and then taking the absolute value. The blue color represents a small error, indicating that the predicted values are closer to the actual values. These regions can be interpreted as areas where the deep learning method has reached a high level of accuracy, successfully capturing the underlying patterns and dynamics of the data. Conversely, the red color represents a large error, indicating an error between the output from the deep learning method and the ground truth. The darkness of the red color is proportional to the magnitude of the error, with darker red indicating a larger error.

Figure 4 shows that there are prediction errors for all deep learning methods around a height of approximately 0 Mm. However, TFNO and ViT still perform much better compared to CNN-RNN and CNN-LSTM. This observation can be attributed to several potential factors, including the inherent complexity of the data. While it may be challenging to find differences between TFNO and ViT in Figures 3a, 3b, and 3c, Figures 4a, 4b, and 4c allow us to observe that TFNO only has the smallest prediction errors at heights of approximately 2 Mm, 4 Mm, 6 Mm, and 10 Mm, where the error map is mostly dark blue (i.e., no errors). ViT shows serpentine, thin-line errors, indicating a deficiency in capturing fine details as compared to TFNO. The CNN-RNN and CNN-LSTM exhibit learning capabilities but present significantly larger errors than TFNO and ViT, which are not only quantitatively substantial but also lack meaningful physical interpretation.

In conclusion, TFNO stands out for its robustness and ability to capture detailed features with the lowest test error, indicating its potential suitability for complex tasks requiring high precision. In contrast, while ViT shows promise in certain aspects, its performance is shadowed by its shortcomings in complex structure predictions.

3.3.2 *Evaluation from a Physics Perspective.*

When assessing the performance of deep learning methods in reproducing the MHD outputs, it is vital to check whether these methods can successfully reproduce the comprehensive physical quantities found in the AI-generated data, similar to the ground truth. We examine key physical quantities–specifically, the magnetic field strength and current density as a function of height, to see the capabilities of different methods in reflecting the magnetic field strength at different heights. In addition, by analyzing the magnetic field orientation through the inclination and azimuthal angles as a function of height, we can assess the models' proficiency in capturing the geometric properties of the field. These metrics collectively demonstrate the methods' competencies not only in emulating the MHD model outputs but also in accurately rendering the comprehensive physics quantities and magnetic field orientation at various atmospheric levels, including the surface and coronal height.

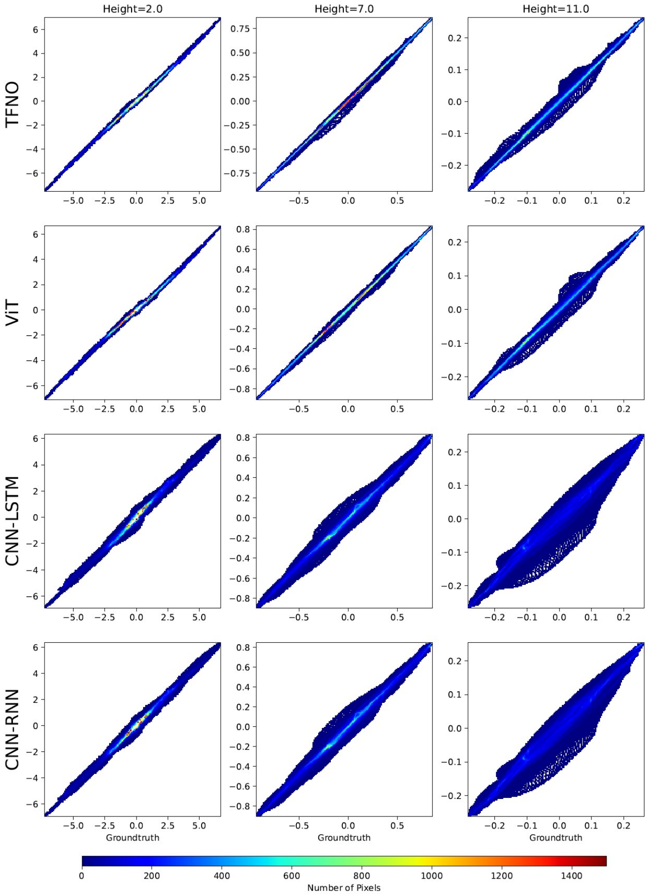


Figure 5: 2D histogram of TFNO, ViT, CNN-RNN, and CNN-LSTM to the ground truth, at height = 2.0 Mm, 7.0 Mm, and 11.0 Mm. Color gradient denotes the number of data points.

Figure 5 shows the 2D histograms that compare predictions from different deep learning methods (TFNO, ViT, CNN-RNN, and CNN-LSTM) to the ground truth across various height = 2.0 Mm, 7.0 Mm, and 11.0 Mm. Each method demonstrates a scattered distribution of points along the diagonal, indicating a linear correlation between predicted and true magnetic field strengths. At a height of 2.0 Mm, closer to the surface, the TFNO and ViT show a denser clustering of points around the diagonal, a pattern less concentrated in the CNN-LSTM and CNN-RNN. At greater heights of 7.0 Mm and 11.0 Mm, TFNO and ViT consistently keep points near the diagonal, showcasing sustained prediction accuracy. On the contrary, CNN-LSTM and CNN-RNN exhibit a wider spread of points, hinting at a decline in predictive precision.

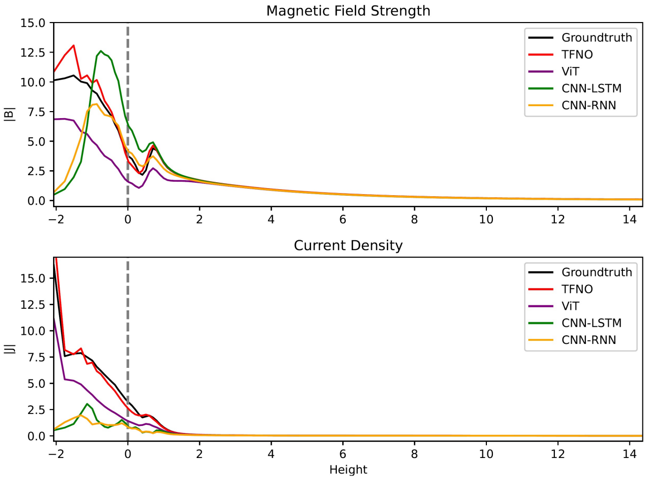


Figure 6: Comparison of physics quantities derived from TFNO, ViT, CNN-RNN, and CNN-LSTM. Upper panel: Variation of magnetic field strength as a function of height. Lower panel: Variation of current density as a function of height.

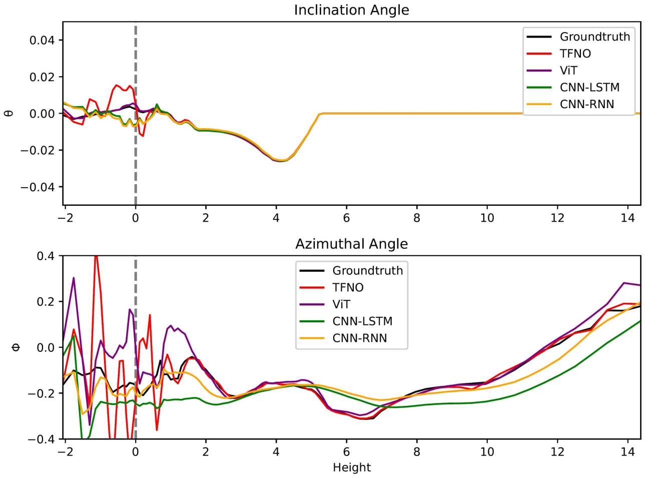


Figure 7: Comparison of field orientation derived from TFNO, ViT, CNN-RNN, and CNN-LSTM. Upper panel: Variation of inclination angle as a function of height. Lower panel: Variation of azimuthal angle as a function of height.

Moreover, we have evaluated physics-based measurements varying with height to assess each model's ability to reproduce the MHD model outputs. Figure 6 displays magnetic field strength (B_0) and current density ($|J|$) reveals that, above a height of 2 Mm, all models produce patterns that resemble the ground truth, indicative of a stable environment with low magnetic field strength. Below this height, only the TFNO model's predictions align closely with the actual measurements. Similarly, with current density, TFNO maintains a near-ground truth trend below a height of 1.5 Mm. At a height above 1.5 Mm, characterized by low current, all deep learning methods follow a similar trajectory. Overall, TFNO delivers the most consistent and accurate predictions of these physical quantities, particularly close to the surface where the magnetic settings are complex.

Figure 7 illustrates the variation of the horizontally averaged inclination angle θ and azimuthal angle ϕ as a function of height, derived from different deep learning methods compared to the ground truth. Please note that a binary map has been applied to focus solely on magnetic field strengths that exceed 2.5 times the background level. This approach helps to disregard weaker fields, which tend to be noisier and less significant, while still taking into account their potential influence on the calculations of field orientation metrics. All deep learning methods can reproduce the inclination angles well after height > 1Mm. At a lower height below 1 Mm, ViT outperforms other models regarding the inclination angles. The azimuthal angle variation, however, demonstrates a significant divergence between the deep learning methods and the ground truth at the lower height (below height = 2 Mm). TFNO and ViT appear to align with the ground truth, but TFNO converges faster than ViT. Although CNN-LSTM and CNN-RNN follow the general trend, they do not closely align with the ground truth. At the higher layer (height > 12 Mm), TFNO continues to track the ground truth, whereas ViT starts to diverge. The inclination and azimuthal angles are important for understanding the orientation of the magnetic field. The consistency of the TFNO with the ground truth suggests it may provide a more physically accurate magnetic field orientation with height, compared with other deep learning methods.

4 CONCLUSION

The exploration of neural operator architectures, particularly the TFNO, represents a significant advancement in the computational modeling of the solar coronal magnetic field. TFNO has demonstrated the capability to simulate the complex magnetic solar outer atmosphere compared to state-of-the-art deep learning methods such as ViT, CNN-RNN, and LSTM. The findings from the comparative study reveal that TFNO achieves superior prediction accuracy as evidenced by the best test performance among all methods. Compared to traditional physics-based models like MHD, TFNO only takes a few seconds to generate the 3D cube while MHD takes days to simulate.

From the physics perspective, TFNO stands out for its precision in predicting the magnetic field topologies, crucially maintaining this accuracy across varying atmospheric heights. TFNO exhibits a remarkable adherence to ground truth, particularly below a height of 2Mm, a region characterized by complex magnetic interactions. Its performance is consistently superior to that of other methods, such as the ViT, CNN-RNN, and CNN-LSTM, especially in critical lower atmospheric layers where precise modeling is most needed. Additionally, TFNO's alignment with ground truth in the calculation of field orientation metrics further solidifies its capability to capture the nuanced dynamics of solar phenomena.

ACKNOWLEDGEMENT

This work was supported by the National Science Foundation grants AGS-2229064 and AST-2204384, and National Aeronautics and Space Administration Grant 80NSSC20K1282.

REFERENCES

[1] Hubert Baty and Vincent Vigon. 2024. Modelling solar coronal magnetic fields with physics-informed neural networks. *Monthly Notices of the Royal Astronomical Society* 527, 2 (2024), 2575–2584.

[2] Jens Berg and Kaj Nyström. 2018. A unified deep artificial neural network approach to partial differential equations in complex geometries. *Neurocomputing* 317 (2018), 28–41.

[3] Joseph E Borovsky, Gian Luca Delzanno, Philip J Erickson, Alexa Jean Halford, Benoit Lavraud, and Sabrina Savage. 2024. The future of space physics 2022. , 1403148 pages.

[4] SI Braginskii. 1965. Transport processes in a plasma. *Reviews of plasma physics* 1 (1965), 205.

[5] Mats Carlsson, Viggo H. Hansteen, Boris V. Gudiksen, Jorrit Leenaarts, and Bart De Pontieu. 2016. A publicly available simulation of an enhanced network region of the Sun. *Astronomy & Astrophysics* 585, Article A4 (Jan. 2016), A4 pages. https://doi.org/10.1051/0004-6361/201527226 arXiv:1510.07581 [astro-ph.SR]

[6] Mats Carlsson, Viggo H Hansteen, Boris V Gudiksen, Jorrit Leenaarts, and Bart De Pontieu. 2016. A publicly available simulation of an enhanced network region of the Sun. *Astronomy & Astrophysics* 585 (2016), A4.

[7] Mitsue Den. 2016. Three-dimensional MHD simulation of the solar wind from the solar surface to 400 solar radius using REPPU (REProduce Plasma Universe) code. *PoS* ICRC2015 (2016), 184. https://doi.org/10.22323/1.236.0184

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ArXiv* abs/2010.11929 (2020), 1–22. https://api.semanticscholar.org/CorpusID:225039882

[9] Gregory Fleishman, Ivan Mysh'yakov, Alexey Stupishin, Maria Loukitcheva, and Sergey Anfinogentov. 2019. Force-free Field Reconstructions Enhanced by Chromospheric Magnetic Field Data. *The Astrophysical Journal* 870, 2, Article 101 (Jan. 2019), 101 pages. https://doi.org/10.3847/1538-4357/aaf384 arXiv:1811.02093 [astro-ph.SR]

[10] Gregory D. Fleishman, Sergey Anfinogentov, Maria Loukitcheva, Ivan Mysh'yakov, and Alexey Stupishin. 2017. Casting the Coronal Magnetic Field Reconstruction Tools in 3D Using the MHD Bifrost Model. *The Astrophysical Journal* 839, 1, Article 30 (April 2017), 30 pages. https://doi.org/10.3847/1538-4357/aa6840 arXiv:1703.06360 [astro-ph.SR]

[11] Klaus Galsgaard and Åke Nordlund. 1996. Heating and activity of the solar corona: 1. Boundary shearing of an initially homogeneous magnetic field. *Journal of Geophysical Research: Space Physics* 101, A6 (1996), 13445–13460.

[12] Raghav Gnanasambandam, Bo Shen, Jihoon Chung, Xubo Yue, and Zhenyu Kong. 2023. Self-scalable tanh (stan): Multi-scale solutions for physics-informed neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 12 (2023), 15588–15603.

[13] Boris V Gudiksen, Mats Carlsson, Viggo H Hansteen, Wolfgang Hayek, Jorrit Leenaarts, and Juan Martínez-Sykora. 2011. The stellar atmosphere simulation code Bifrost-code description and validation. *Astronomy & Astrophysics* 531 (2011), A154.

[14] Jiequn Han, Arnulf Jentzen, and Weinan E. 2018. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences* 115, 34 (2018), 8505–8510.

[15] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. 2018. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. IEEE, Salt Lake City, US, 6546–6555.

[16] Wen He, Ju Jing, Haimin Wang, Sushree S Nayak, and Avijeet Prasad. 2023. Coronal Magnetic Field Extrapolation and Topological Analysis of Fine-scale Structures during Solar Flare Precursors. *The Astrophysical Journal* 958, 1 (2023), 90.

[17] Satoshi Inoue, Keiji Hayashi, Takahiro Miyoshi, Ju Jing, and Haimin Wang. 2023. A comparative study of solar active region 12371 with data-constrained and data-driven magnetohydrodynamic simulations. *The Astrophysical Journal Letters* 944, 2 (2023), L44.

[18] Ju Jing, Chang Liu, Jeongwoo Lee, Shuo Wang, Thomas Wiegelmann, Yan Xu, and Haimin Wang. 2014. Evolution of a magnetic flux rope and its overlying arcade based on nonlinear force-free field extrapolations. *The Astrophysical Journal Letters* 784, 1 (2014), L13.

[19] Ju Jing, Yuan Yuan, Thomas Wiegelmann, Yan Xu, Rui Liu, and Haimin Wang. 2010. Nonlinear force-free modeling of magnetic fields in a solar filament. *The Astrophysical Journal Letters* 719, 1 (2010), L56.

[20] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. 2021. Physics-informed machine learning. *Nature Reviews Physics* 3, 6 (2021), 422–440.

[21] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014), 1–15. https://api.semanticscholar.org/CorpusID:6628106

[22] Tamara G Kolda and Brett W Bader. 2009. Tensor decompositions and applications. *SIAM review* 51, 3 (2009), 455–500.

[23] Jean Kossaifi, Nikola B. Kovachki, Kamyar Azizzadenesheli, and Anima Anandkumar. 2023. Multi-Grid Tensorized Fourier Neural Operator for High-Resolution PDEs. *ArXiv* abs/2310.00120 (2023), 1–21. https://api.semanticscholar.org/CorpusID:263334252

[24] J Leenaarts, M Carlsson, V Hansteen, and RJ Rutten. 2007. Non-equilibrium hydrogen ionization in 2D simulations of the solar atmosphere. *Astronomy & Astrophysics* 473, 2 (2007), 625–632.

[25] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. 2020. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485* 1, 1 (2020), 1–21.

[26] Zong-Yi Li, Nikola B. Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. 2020. Fourier Neural Operator for Parametric Partial Differential Equations. *ArXiv* abs/2010.08895 (2020), 1–16. https://api.semanticscholar.org/CorpusID:224705257

[27] Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. 2018. Pde-net: Learning pdes from data. In *International conference on machine learning*. PMLR, PMLR, Stockholmsmässan, 3208–3216.

[28] Alaeddin Malek and R Shekari Beidokhti. 2006. Numerical solution for high order differential equations using a hybrid neural network—optimization method. *Appl. Math. Comput.* 183, 1 (2006), 260–271.

[29] Åke Nordlund, Klaus Galsgaard, and R. F. Stein. 1995. Visualizing Astrophysical 3D MHD Turbulence. In *Workshop on Applied Parallel Computin*. Springer, New York, NY, US, 450–461. https://api.semanticscholar.org/CorpusID:37218855

[30] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics* 378 (2019), 686–707.

[31] VD Shafranov. 1958. On magnetohydrodynamical equilibrium configurations. *Soviet Physics JETP* 6, 3 (1958), 1013.

[32] Bo Shen, Weijun Xie, and Zhenyu James Kong. 2022. Smooth robust tensor completion for background/foreground separation with missing pixels: Novel algorithm with convergence guarantee. *Journal of Machine Learning Research* 23, 217 (2022), 1–40.

[33] Jamie M Taylor, David Pardo, and Ignacio Muga. 2023. A deep Fourier residual method for solving PDEs using neural networks. *Computer Methods in Applied Mechanics and Engineering* 405 (2023), 115850.

[34] Jiasheng Wang, Jeongwoo Lee, Chang Liu, Wenda Cao, and Haimin Wang. 2022. A High-resolution Study of Magnetic Field Evolution and Spicular Activity around the Boundary of a Coronal Hole. *The Astrophysical Journal* 924, 2 (2022), 137.

[35] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. 2016. Cnn-rnn: A unified framework for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, Las Vegas, NV, USA, 2285–2294.

[36] Y-M Wang and NR Sheeley Jr. 1992. On potential field models of the solar corona. *Astrophysical Journal, Part 1 (ISSN 0004-637X), vol. 392, no. 1, June 10, 1992, p. 310-319. Research supported by US Navy.* 392 (1992), 310–319.

[37] Lodewijk Woltjer. 1958. A theorem on force-free magnetic fields. *Proceedings of the National Academy of Sciences* 44, 6 (1958), 489–491.

[38] Daiki Yamasaki, Satoshi Inoue, Yumi Bamba, Jeongwoo Lee, and Haimin Wang. 2022. A Data-constrained Magnetohydrodynamic Simulation of the X1. 0 Solar Flare of 2021 October 28. *The Astrophysical Journal* 940, 2 (2022), 119.

[39] Yi Yang and Fang Shen. 2019. Modeling the global distribution of solar wind parameters on the source surface using multiple observations and the artificial neural network technique. *Solar Physics* 294, 8 (2019), 111.

[40] Xuanle Zhao, Yue Sun, Tielin Zhang, and Bo Xu. 2023. Local Convolution Enhanced Global Fourier Neural Operator For Multiscale Dynamic Spaces Prediction. *ArXiv* abs/2311.12902 (2023), 1–10. https://api.semanticscholar.org/CorpusID:265351567