5-31-2024

Neural operator for accelerating coronal magnetic field computations in bifrost MHD model

Yutao Du
New Jersey Institute of Technology, yd288@njit.edu

Recommended Citation

Du, Yutao, "Neural operator for accelerating coronal magnetic field computations in bifrost MHD model" (2024). *Theses*. 2584.
https://digitalcommons.njit.edu/theses/2584

ABSTRACT

NEURAL OPERATOR FOR ACCELERATING CORONAL MAGNETIC FIELD COMPUTATIONS IN BIFROST MHD MODEL

by
Yutao Du

The application of the Tensorized Fourier Neural Operator (TFNO) to significantly enhance the computational efficiency of coronal magnetic field calculations within the Bifrost Magnetohydrodynamics (MHD) model is introduced in this study. Leveraging simulated data from the European Sunrise Science Data Center, the TFNO— an extension of the Fourier Neural Operator (FNO) that incorporates tensor decomposition for improved handling of high-dimensional data—is employed to solve time-varying partial differential equations (PDEs) over a 3D domain. The performance of the TFNO is compared with traditional machine learning methods, including Vision Transformer and CNN-RNN (encoder-decoder) architectures, to demonstrate its accuracy, computational efficiency, and scalability. A physics-analysis of the TFNO predictions is also performed to demonstrate the reliability of the method. This advancement not only accelerates the simulation of solar atmospheric phenomena but also provides more reliable prediction capabilities, thus greatly contributing to the understanding of space weather dynamics and its impact on Earth.

NEURAL OPERATOR FOR ACCELERATING CORONAL MAGNETIC FIELD COMPUTATIONS IN BIFROST MHD MODEL

by
Yutao Du

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Fulfillment of the Requirements for the Degree of
Master of Science in Engineering Management

Department of Mechanical and Industrial Engineering

May 2024

APPROVAL PAGE

NEURAL OPERATOR FOR ACCELERATING CORONAL MAGNETIC FIELD COMPUTATIONS IN BIFROST MHD MODEL

Yutao Du

Bo Shen, Thesis Advisor Date
Assistant Professor of Mechanical and Industrial Engineering

Sanchoy Das, Committee Member Date
Professor of Mechanical and Industrial Engineering, NJIT

SangWoo Park, Committee Member Date
Assistant Professor of Mechanical and Industrial Engineering, NJIT

Haimin Wang, Committee Member Date
Distinguished Professor of Physics, NJIT

BIOGRAPHICAL SKETCH

Author:	Yutao Du
Degree:	Master of Science
Date:	May 2024

Undergraduate and Graduate Education:

- Master of Science in Engineering Management,
  New Jersey Institute of Technology, Newark, NJ, 2024
- Bachelor of Science in Electrical, Electronics and Communications Engineering,
  Chongqing University of Post and Telecommunications, Chongqing, China, 2022
- Bachelor of Science in Electrical, Electronics and Communications Engineering,
  Brunel University London, Chongqing, China, 2022

Major:	Engineering Management

Presentations and Publications:

Yutao Du, "A research on the architectures of large capacity optical access networks," *2021 2nd International Conference on Computing and Data Science (CDS)*, pp 460-465, 2021.

In the vast universe of deep learning networks, we are like explorers navigating uncharted waters, where every algorithm update and parameter adjustment is a brave attempt to probe the unknown. This field, filled with complex theories and endless possibilities, seems daunting at first glance. Yet, it is these very challenges that ignite scientists' relentless pursuit and deep contemplation. On this journey, we need not only the precision of algorithms and the vastness of data but also a curiosity for the unknown world and a deep desire for knowledge. Just as deep learning networks demonstrate incredible ability in processing data, extracting patterns, and knowledge from chaos, I too hope to discover new realms untouched by science and technology during this process. I am grateful for all the mentors and peers on this path; it is your wisdom and persistence that illuminate this challenging and miraculous journey of exploration.

ACKNOWLEDGMENT

I am profoundly grateful to my advisor, Bo Shen, for his guidance and support throughout this project. His expertise and dedication have not only fueled my research but have also been fundamental to my academic growth. This journey has been enriched immeasurably by his continued encouragement.

Special thanks are extended to Professors Sanchoy Das, SangWoo Park, and Haimin Wang for their valuable time and insightful feedback during my defense. Their detailed and thoughtful critiques have significantly enhanced the quality of my work, providing clarity and direction at critical stages.

Further appreciation is due to Bo Shen for the scholarship that has financially supported my research endeavors. This generosity has enabled me to pursue my scientific inquiries with unwavering vigor and dedication, a privilege that I cherish deeply.

I am also indebted to the High-Performance Computing (HPC) Center at NJIT for providing the computational resources essential for my research. Their state-of-the-art facilities have not only supported but propelled the advancement of my work, allowing me to explore and innovate beyond boundaries.

I extend my deepest gratitude to Professor Qin Li for his invaluable contributions to the physics analysis of my experimental results. His expertise and meticulous attention to detail have greatly enhanced the depth and accuracy of this study, shaping its final outcomes.

Additional thanks go to Chenyang Li and Marco Marena, whose assistance and collaboration were invaluable during challenging times in my research journey. Their readiness to help and share their expertise has left an indelible impact on my academic career.

Lastly, I must express my heartfelt gratitude to my parents, Yinghao Du and Biyun Chen. Their unwavering support and belief in my pursuits have been my source of strength and inspiration throughout my studies abroad.

Each of the individuals mentioned has played a pivotal role in this academic journey, and I am eternally grateful for their support and belief in my potential. This thesis stands as a testament to their collective contribution to my personal and professional development.

TABLE OF CONTENTS

Chapter **Page**

LIST OF TABLES

LIST OF FIGURES

CHAPTER 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 [1]. In the convection zone, the gas pressure surpasses the magnetic pressure in all magnetic flux concentrations except for the most intense ones, causing the plasma to move the field. These movements initiate the generation of 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. Due to the complicated nature of chromospheric physics and the many transitions involved, the chromosphere is the least understood region of the Sun [2]. 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 [3], magnetohydrostatics (MHS) models [4], force-free models [5], and potential-field models [6]. 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 computations 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. (2007) simulated the solar atmosphere from the convection zone to the corona. The study demonstrates the effects of nonequilibrium hydrogen ionization on the dynamical structure of the solar atmosphere and the formation of hydrogen lines, specifically $H\alpha$ lines, and quantifies these effects [7]. Den et al. (2015) present a three-dimensional magnetohydrodynamic (MHD) simulation code designed to simulate space plasma phenomena with applications in modeling the solar surface and global solar wind structure [8]. The goal of [9] is to provide the community with a realistic simulated magnetic field of the Sun's outer atmosphere. This will allow a detailed comparison of existing and upcoming observations with simulated synthetic observations, to elucidate the complex interactions between the magnetic field and the plasma. Their study uses the radiative MHD code Bifrost to simulate a computational volume with a magnetic field topology similar to the extended net region on the Sun. These models are often based on physics-driven models such as Bifrost, which provide valuable insights but have significant computational cost and time requirements.

To reduce the computational cost and time of MHD simulations, researchers have attempted to solve the MHD equations using machine learning methods to simulate the solar exosphere. The machine learning (ML) approach solves the PDE equations several orders of magnitude faster compared to the traditional approach by providing a fast solver that approximates or augments the traditional approach [10] [11] [12] [13]. More and more researchers have been trying to use ML methods to solve MHD equations. For instance, Yang and Shen (2019) devised a method for globally mapping solar wind parameters at the source surface using an Artificial Neural Network (ANN) approach. This model presents an efficient way to tackle 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 [14]. Jarolim et al. (2023) established a method to extrapolate the solar coronal magnetic field using Physics-Informed Neural Networks (PINNs), integrating observational data with force-free magnetic field theory. This approach, demonstrated through the simulation of active region NOAA 11158, enables real-time magnetic field extrapolations. Validated against EUV observations, their method offers a significant advancement in solar magnetic field modeling, promising enhanced accuracy in solar research and space weather forecasting [15]. Similarly, Baty et al. (2024) present a groundbreaking method to simulate solar coronal magnetic fields using PINNs. 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 machine learning techniques to improve the modeling and predictive analysis of magnetic fields in the solar coronal environment [16]. However, one major drawback of these methods is the difficulty of transferring knowledge between different configurations. For example, when solving the high-dimensional Navier-Stokes equation, PINNs might require a dedicated model for each desired resolution, increasing the computational effort and potentially limiting its effectiveness in capturing multi-scale phenomena.

Fourier neural operator (FNO) has been introduced to solve this issue and improve the scalability of deep learning methods [17]. The FNO represents a significant advancement over traditional methods, as it is mesh-independent, can transfer solutions between different grid geometries, and is significantly faster than conventional PDE solvers. Moreover, FNO exhibits superior accuracy compared to previous learning-based methods under fixed resolution and is the first machine learning-based method to successfully model turbulent flows with zero-shot super-resolution [17]. Consequently, aside from its mesh-independence, FNO demonstrates computational efficiency surpassing standard visual transformers (with quadratic time complexity), and it effectively captures global dependencies compared to convolutional neural networks, graphical neural networks, and other local models (Falk et al., 2019) [18]. Despite the numerous advantages associated with FNOs, there is no documented application of FNOs to the calculation of the solar coronal magnetic field.

In this study, I use FNO to accelerate coronal magnetic field calculations in the Bifrost MHD model. I use an extended FNO architecture called Tensorized Fourier Neural Operator (TFNO, [19]), which introduces tensor decomposition to model the kernel operator and thus enhances its ability to efficiently handle high-dimensional data. I employed simulated data from Carlsson et al. (2015) [9] to solve a time-varying PDE over a 3D domain using the TFNO. after which I used traditional machine algorithms such as Vision Transformer, and CNN-RNN (encoder-decoder) to compare with the TFNO. In addition, I verify the scalability of FNO and analyze the predicted data physically to verify the reliability of its predictions.

CHAPTER 2

NEURAL OPERATOR ARCHITECTURE

2.1 Problem Formulation

Bifrost, drawing from a lineage of numerical codes spanning several generations, incorporates the Oslo Stagger Code as its latest iteration. These codes, rooted in the method developed by Nordlund and Galsgaard in 1995 [20] and further refined by Galsgaard and Nordlund in 1996 [21], share a common foundation. At the core of the code, a staggered mesh explicit code that solves the standard MHD PDEs on a Cartesian grid:

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \mathbf{u}) \tag{2.1}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} = -\nabla \cdot (\rho \mathbf{u}\mathbf{u} - \tau) - \nabla P + \mathbf{J} \times \mathbf{B} + \rho \mathbf{g} \tag{2.2}$$

$$\mu \mathbf{J} = \nabla \times \mathbf{B} \tag{2.3}$$

$$\mathbf{E} = \eta \mathbf{J} - \mathbf{u} \times \mathbf{B} \tag{2.4}$$

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E} \tag{2.5}$$

$$\frac{\partial e}{\partial t} = -\nabla \cdot (e \mathbf{u}) - P \nabla \cdot \mathbf{u} + Q \tag{2.6}$$

where ρ, $\mathbf{u}$, e, $\mathbf{B}$ are the density, the velocity vector, the internal energy per unit volume, and the magnetic flux density vector 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 a number of 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 [22].

2.2 Fourier Neural Operator

The idea of the Fourier Neural Operator (FNO) is derived from the Green's function [23]. To solve a differential equation $Lu = f(x)$ with an inhomogeneous term $f(x)$ and a linear operator L, one would express the solution $u(x)$ as an integral over the domain D where the problem is defined:

$$u(x) = \int_D G(x, y) f(y)\, dy \tag{2.7}$$

where $x, y \in \mathcal{D} \subseteq \mathbb{R}^d$. If there exists a function g such that $G(x, y) = g(x - y)$, Equation (2.7) reduces to a convolution operation, and G serves as the convolution kernel. For a parametric PDE, Green's function also depends on the parameter. For example, the magnetic flux density vector B is a parameter in the MHD equation. In this framework, the parameter is assumed to be a function of x. According to [17], the kernel integral operator is defined by

$$(K_\theta(B) h_i)(x) = \int_D \mathcal{K}_\theta(x, y, B(x), B(y)) h_i(y)\, dy \tag{2.8}$$

where B is the PDE parameter, and the kernel integral operator K_θ is parameterized by a neural architecture with learnable parameters θ. On the right-hand side, $\mathcal{K}$ approximates the (convolution) kernel G. In particular, h_i denotes the latent representation generated by a Fourier layer in a sequence of Fourier layers indexed by i.

For a better understanding of the structure of a Fourier layer, we visualize it in Figure 2.1. The flow of $h_i(x)$ bifurcates into two paths. The upper path input $h_i(x)$ first applies the Fourier transform $\mathcal{F}$, then linearly transforms $\mathcal{R}$ for the low-level Fourier modes and filters out the high-level modes, and finally applies the inverse

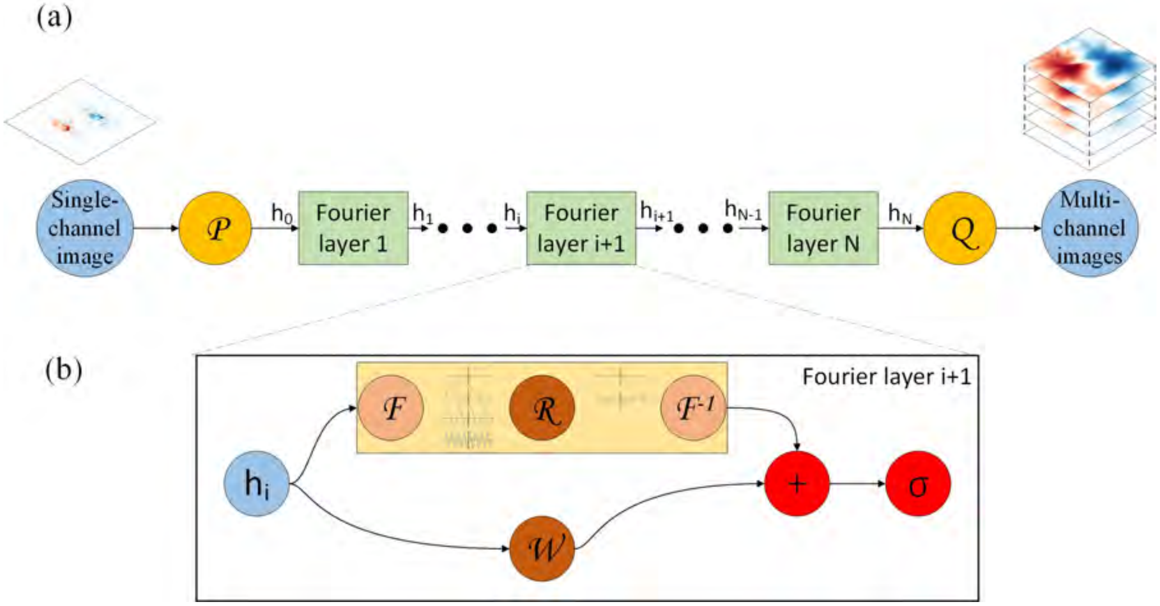

Figure 2.1 **(a)** The architecture of the neural operators; **(b)** Fourier layer.

Fourier transform $\mathcal{F}^{-1}$. Unlike [24] 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. Hereby, FNO initializes complex-valued learnable parameters directly in the Fourier space. Following the notations in [17], Fourier integral operator is defined by

$$K_\theta(B)h_i = \mathcal{F}^{-1}(\mathcal{F}(\mathcal{K}_\theta) \cdot \mathcal{F}(h_i)) = \mathcal{F}^{-1}(R_\phi \mathcal{F}(h_i)) \tag{2.9}$$

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 (FFT) and its inverse are used to preserve computational efficiency. The bottom path input $h_i(x)$ applies a local linear transform $\mathcal{W}$, wherein 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 2.1 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

$$h_{i+1} = \sigma(\mathcal{W}_{i+1} h_i(x) + \mathcal{K}_{\theta_{i+1}}(B) h_i)(x) \tag{2.10}$$

with $i = 1, 2, ..., N$. 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. In our case, the output is multi-channel images.

As FNO maps the parameter B to the solution u, it succeeds in solving PDEs with variable parameters. For comparison, frameworks such as [25] only solve one instance of the PDE after fixing the parameter.

FNO extends the concept of neural operators by incorporating the Fourier transform to streamline the learning process. It operates by first transforming the input function into the frequency domain, where a neural network learns the interactions between different frequencies. Subsequently, the output is transformed back into the spatial domain. Leveraging the Fourier space to represent continuous functions inherent in PDEs often results in increased computational efficiency [17].

Another advantage of FNO compared with [26] is the mesh-independence. FNO is designed to be independent of the mesh size, allowing it to be trained using coarser, low-resolution data and then applied to finer, high-resolution data for evaluation. This can be done without the neural operator ever being exposed to high-resolution data during training, enabling it to enhance resolution in a manner that requires no additional high-resolution examples (known as zero-shot super-resolution) [17].

2.3 Tensorized Fourier Neural Operators

Section 2.2 introduces the structure of a basic FNO. TFNO extends FNO by introducing a unified parameterization where the entire operator is represented by a single parameter tensor $\mathcal{T}$. By expressing $\mathcal{T}$ in a factorized form (e.g., Tucker or Canonical Polyadic decomposition), 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 efficiency and generalization capabilities [19].

In practice, we express $\mathcal{T}$ in a low-rank factorized form, e.g. Tucker or CP. In the case of a Tucker factorization with rank $(\mathcal{R}_1, ..., \mathcal{R}_d, \mathcal{R}_L, \mathcal{R}_I, \mathcal{R}_O)$, where $\mathcal{R}_L$ controls the rank across layers, $\mathcal{R}_I = \mathcal{R}_O$) control the rank across the input and output co-dimension, respectively, and $\mathcal{R}_1, ..., \mathcal{R}_d$ control the rank across the dimensions of the operator

$$\begin{aligned}\mathcal{T} = \sum_{r_1=1}^{\mathcal{R}_1} \cdots \sum_{r_d=1}^{\mathcal{R}_d} \sum_{r_i=1}^{\mathcal{R}_I} \sum_{r_o=1}^{\mathcal{R}_O} \sum_{r_l=1}^{\mathcal{R}_L} G(r_1, \ldots, r_d, r_i, r_o, r_l) \cdot \\ \cdot U^{(1)}(:, r_1) \ldots U^{(d)}(:, r_d) \cdot U^{(I)}(:, r_i) \cdot U^{(O)}(:, r_o) \cdot U^{(L)}(:, r_l).\end{aligned} \tag{2.11}$$

Here, G is the core of size $\mathcal{R}_L \times \mathcal{R}_I \times \mathcal{R}_O \times \mathcal{R}_1 \times ... \times \mathcal{R}_d$ and $U^{(L)}, U^{(I)}, U^{(O)}, U^{(1)}, ..., U^{(d)}$ are factor matrices of size $(\mathcal{R}_L \times L), (\mathcal{R}_I \times I), (\mathcal{R}_O \times O), (\mathcal{R}_1 \times \alpha), ..., (\mathcal{R}_d \times \alpha)$, respectively.

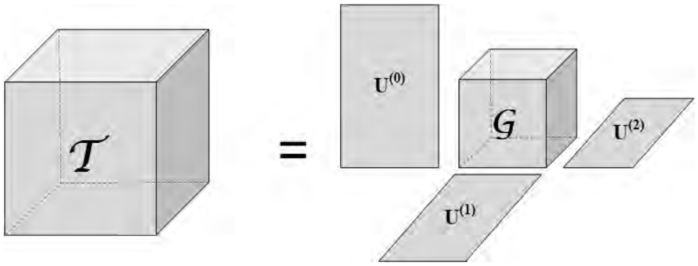

Figure 2.2 Illustration of a Tucker decomposition.

A visualization of the Tucker decomposition of a third-order tensor can be seen in Figure 2.2. Note that we can rewrite the entire weight parameter for this Tucker case, equivalently, using the more compact $n - mode$ product as:

$$\mathcal{T} = G \times_1 U^{(1)} \cdots \times_d U^{(d)} \times_{d+1} U^{(I)} \times_{d+2} U^{(O)} \times_{d+3} U^{(L)} \tag{2.12}$$

This joint factorization along the entire operator allows us to leverage redundancies both locally and across the entire operator. This leads to a large reduction in the memory footprint, with only a fraction of the parameter. It also acts as a low-rank regularize on the operator, facilitating training.

2.4 Model Used in this Work

The code we use is from (https://github.com/neuraloperator/neuraloperator), which 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 of 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 above, I added a layer of Multi-Layer Perceptrons (MLPs) after the FNO Blocks. It signifies an additional non-linear processing stage within the network, reinforcing its capability to model complex functions and enhancing its ability to learn a diverse array of PDE solutions.

CHAPTER 3

NUMERICAL EXPERIMENTS

3.1 Data

3.1.1 Data Access

All data used in the experiments are available from the European Sunrise Science Data Center (https://sdc.uio.no/search/simulations). In our study, we predominantly utilized the magnetic field intensity data along the x, y, and z axes—denoted as B_x, B_y, and B_z, respectively—from the 'en024048_hion' dataset. 'en024048_hion' dataset is a simulation cube generated using the Bifrost code to simulate a computational volume with a magnetic field topology similar to an enhanced network area on the Sun. The simulation 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 [9]. The simulation volume measures 24 Mm × 24 Mm horizontally, 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. The simulation saved cubes in a configuration of $504 \times 504 \times 496$ voxels. The first snapshot was captured at $t = 3850$ seconds, and the simulation run concluded at $t = 5410$ seconds. Snapshots were recorded at 10-second intervals, cumulating in a total of 157 cubes.

3.1.2 Data Prepossessing

Initially, the dataset underwent down-sampling. We selected the bottom-most slice as the input for our model. Subsequently, one slice was chosen at every 5 interval, culminating in a total of 99 slices designated as the output. Given the constraints of the experimental environment, utilizing the entire cube as training data was not feasible. Consequently, the largest feasible subset of data was selected for training within the parameters permissible by the experimental environment. This approach was adopted to evaluate the model's capacity to predict the entire cube under these constraints. During the down-sampling process, we computed the mean and standard deviation for each slice and proceeded to normalize them. Our experiments revealed that normalization of the data significantly enhances the model's performance and markedly improves accuracy. Following the down-sampling, we obtained 157 data pairs. We then randomly divided these data into training and testing groups in approximately a 9 : 1 ratio. Finally, for post-physics analysis of the prediction results, we will unnormalize the data.

3.2 Experiment Set-up

In our experiments, we employed three different model architectures to compare with TFNO. This section will introduce the model structures, optimizers, and learning rate schedulers we used.

3.2.1 Visual Transformer (ViT)

The ViT model [27] used in our paper is a modification of the transformer architecture traditionally used for NLP. It processes the input image in the form of patches, which are embedded with markers 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, the model captures complex spatial relationships in image data. In the network, a learnable class marker is added to the patch embedding sequence, and positional embeddings are 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 high-resolution prediction.

3.2.2 Encoder-Decoder

In the paper, we use two different encoder-decoder structures: The CNN-RNN [28] and CNN-LSTM models [29]. We use Convolutional Neural Networks (CNN) for feature extraction and Recurrent Neural Networks (RNN) or Long Short-Term Memory Networks (LSTM) for sequence modeling to process sequential image data.

The CNN component consists of three convolutional layers that progressively increase the number of feature channels from 1 to 256 while down-sampling 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 the CNN-RNN model, the main component of the recurrent network is a series of GRU units, 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 model uses LSTM cells with a hidden state size of 256, which is consistent with the output dimension of the CNN. Adding additional cell states of the same size enhances the ability of the LSTM to regulate the flow of information through the network. The input, output, and forgetting gates of the LSTM play an important role in this process by allowing the model to selectively memorize and forget information.

3.2.3 Optimizer and Scheduler

All models are trained using the Adam optimizer [30] and the ReduceLROnPlateau scheduler. Adam stands for "Adaptive Moment Estimation" and combines the advantages of two other extensions of stochastic gradient descent, namely AdaGrad and RMSProp. ReduceLROnPlateau scheduler adjusts the learning rate based on the validation loss. If no improvement is seen for a “patience” number of epochs, the learning rate is reduced.

3.3 Experiment & Result

3.3.1 Prediction Analysis

In the following experiments, we used four different models for training. Each model was trained 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. The training error is measured by Huber Loss (H1 Loss), and the testing error by Mean Squared Error (MSE).

In the context of training FNO for the approximation of PDE solutions, the utilization of Huber Loss serves as a judicious choice [31]. This hybrid loss function, known for blending the properties of mean squared error and mean absolute error, demonstrates enhanced robustness against outliers—a quintessential feature when dealing with the intrinsic discontinuities and sharp gradients characteristic of PDEs. Huber Loss facilitates the maintenance of numerical stability during training by moderating the impact of large errors, thereby enabling the model to effectively capture the smooth underlying structure of PDE solutions while concurrently accommodating the potential for abrupt transitions. Furthermore, the quadratic behavior of Huber Loss in regions of low error expedites convergence, offering a computationally efficient pathway to optimize the FNO parameters, ensuring a smooth yet flexible learning process that is well-suited for the complex landscape of PDE problem spaces.

Figure 3.1 delineates the training and testing error trajectories over 300 epochs for the TFNO, ViT, CNN-RNN, and CNN-LSTM models. The training error is

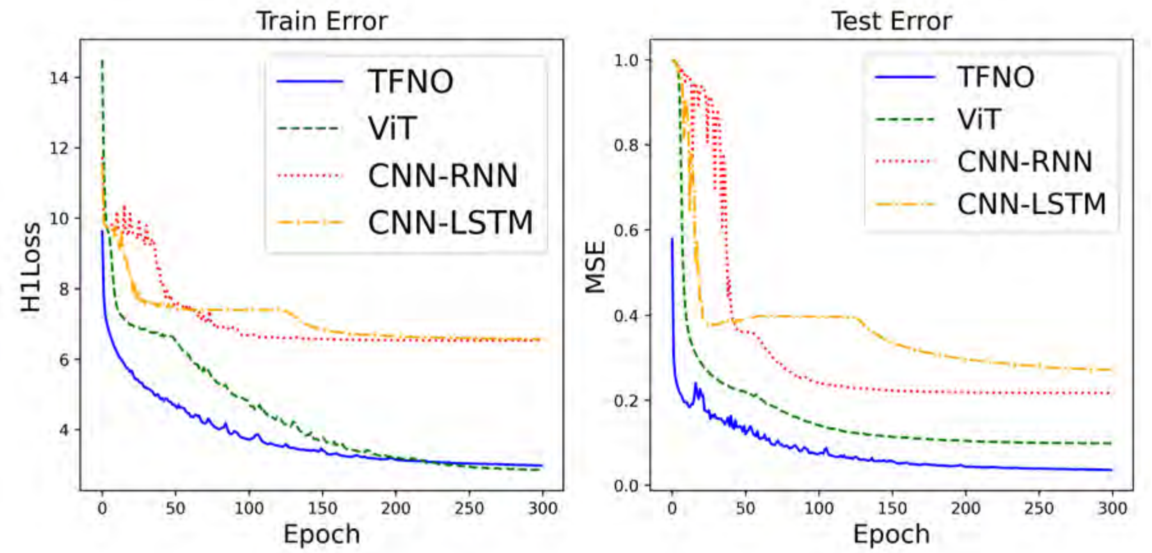

Figure 3.1 Evolution of the training loss and the validation loss as a function of epochs for TFNO, ViT, CNN-RNN, CNN-LSTM.

denoted by H1 Loss and the testing error is denoted by MSE, both errors are closer to 0 indicating better model performance. The TFNO architecture consistently demonstrates superior performance with the lowest error rates in both the training and testing phases, indicating its robustness and efficiency in learning from image data. In addition, we observe a divergence in the generalization capabilities between the ViT and the TFNO models. The ViT model demonstrates an admirable reduction in error rates on the training dataset during the advanced stages of learning. However, this improvement does not translate equivalently to the test dataset, where ViT incurs a higher error rate than TFNO. Such a discrepancy is indicative of overfitting within the ViT model, a condition where a model's excessive complexity captures not only the underlying data distribution but also the idiosyncratic noise within the training set. This results in a degradation of performance on previously unseen data. Conversely, TFNO's consistent test performance, despite higher training errors, suggests a more robust generalization capacity. This pattern highlights the potential advantages of TFNO's architectural or training strategies that may inherently provide regularization, thus mitigating overfitting. While CNN-RNN and CNN-LSTM show comparable learning patterns, CNN-RNN slightly outperforms CNN-LSTM in test error rates. But both models have much lower performance levels than TFNO and ViT.

Table 3.1 Comparative Performance Metrics of Neural Network Models

Model	Number of model parameters	Average training time for one epoch	MSE	R^2	RE	MAE
TFNO	103,989,400	**31.96s**	**0.0502**	**0.9498**	**0.2205**	**0.0831**
ViT	811,791,152	39.25s	0.1240	0.8759	0.3498	0.1417
CNN-RNN	66,046,528	32.93s	0.2514	0.7485	0.4995	0.2230
CNN-LSTM	66,178,112	37.10s	0.3257	0.6743	0.5686	0.2940

Table 3.1 consolidates additional experimental details. The TFNO exhibits an excellent balance between training efficiency and predictive accuracy. Despite having fewer parameters than ViT, TFNO achieves superior test accuracy, as indicated by the lowest MSE among the models, while maintaining competitive training times. While the ViT demonstrating the lowest training error reveals a suboptimal generalization capability, marked by a higher test MSE. This disparity between training and test performance is indicative of an overfitting tendency, exacerbated potentially by its parameter-rich architecture that encompasses the largest number of model parameters among the evaluated models. CNN-based recurrent architectures, the CNN-RNN and CNN-LSTM, show intermediate behavior. These models possess a lower parameter count relative to TFNO but exhibit longer training times, their test errors suggest a moderate capacity for generalization. However, these architectures do not reach the predictive performance of TFNO, nor do they offer computational advantages over ViT, as reflected in their middle-range test MSE, R-square (R^2), Relative Error (RE), and Mean Absolute Error (MAE) scores.

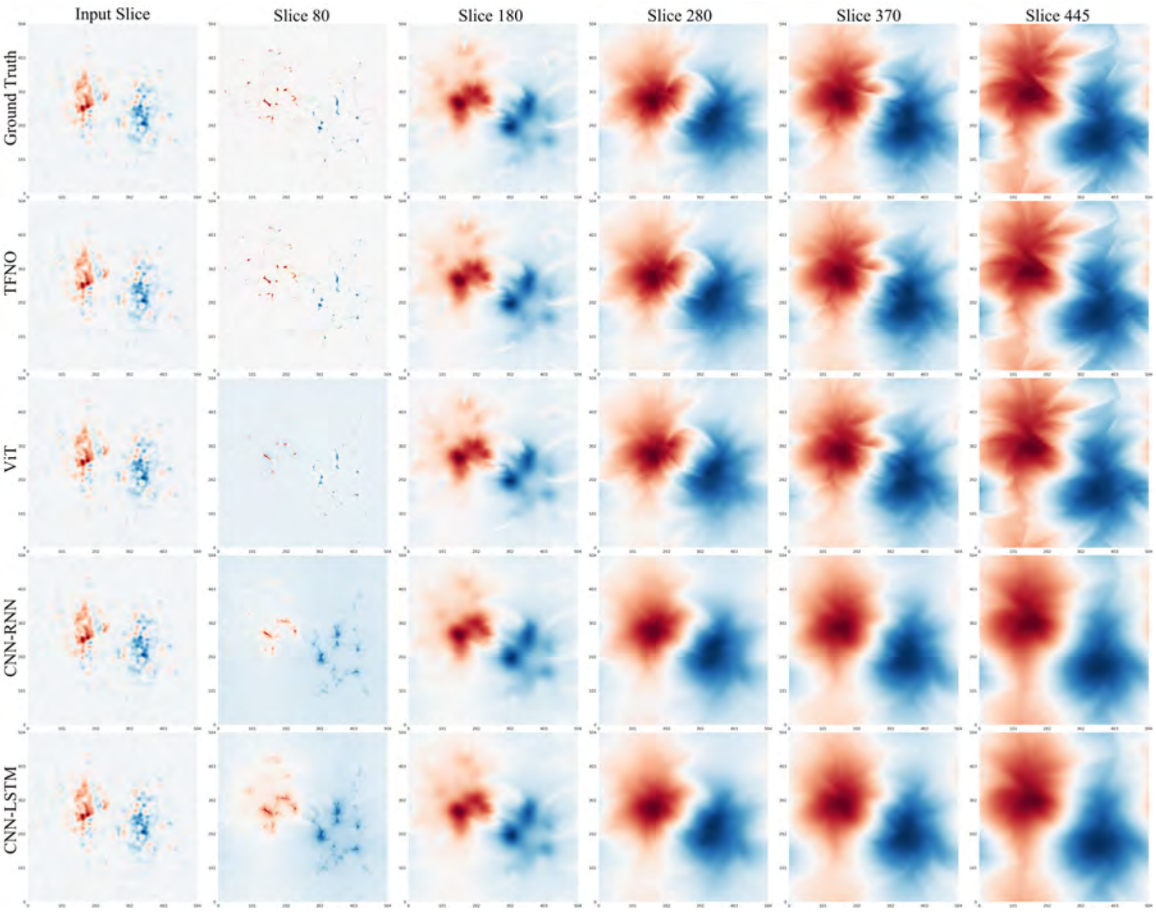

Figure 3.2 Comparative visualization of the B_z predictive performance of various neural network models.

Figure 3.2 provides a comparative visualization of the B_z predictive performance of various neural network models. The figure shows selected slices, captured at different heights of the cube. The first column is the input data of the model. 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 then represents the “ground truth”, which serves as a benchmark for evaluating the subsequent rows - TFNO, ViT, CNN-RNN, and CNN-LSTM.

Figure 3.3 shows the error map corresponding to Figure 3.2, which is obtained by subtracting the predicted results from the ground truth and then taking the absolute value. The blue hues represent areas of lower error, indicating that the predicted

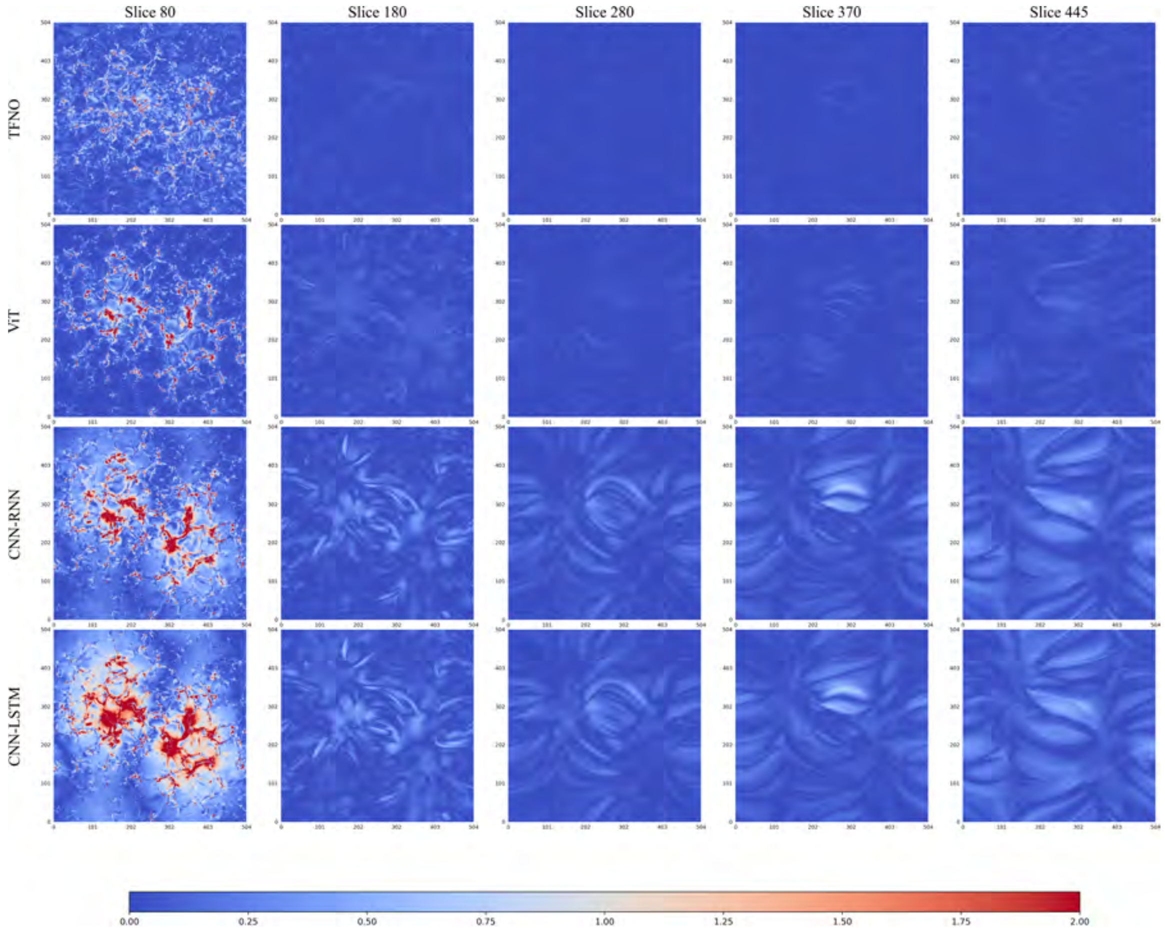

Figure 3.3 B_z Error map for four models.

values are closer to the actual values. These regions can be interpreted as areas where the model has reached a high level of accuracy, successfully capturing the underlying patterns and dynamics of the data. Conversely, the red hues represent areas of higher error, indicating a deviation between the model output and the ground truth. The intensity of the red color is proportional to the severity of the error, with brighter hues indicating more pronounced discrepancies. The presence of red indicates a flaw or potential over-fitting of the model, in which case the model may not be able to generalize the essential features of the data or may have learned noise and anomalies that affect its prediction performance for unknown data.

Cross-referencing Figures 3.2 and 3.3 reveals that around slice 80, a marked concentration of red in the error maps is evident, signifying that all models exhibit certain predictive inaccuracies around a height of approximately 0 Mm. In this case, TFNO and ViT still perform well among these four methods. This observation can be attributed to several potential factors, including the inherent complexity of the data and the presence of noise. Examination of the data slice at 0 Mm in Figure 3.2 reveals that the features are not distinctly apparent and the data presents a considerable degree of complexity, interspersed with significant areas of blank space. Such complexity necessitates a heightened level of abstraction and feature extraction capability from the models.

At other heights, models develop a more refined representation of the data, and a corresponding reduction in prediction errors is observed, as reflected in the error maps of subsequent slices. The reduction in error rates in later slices, marked by the proliferation of blue areas, indicates an enhancement in learning and adaptive capabilities. The TFNO model, in particular, demonstrates a more uniform pattern with diminished errors, suggesting more effective learning strategies or inherent robustness within the model's architecture.

While it may be challenging to discern differences between TFNO and ViT in Figure 3.2, Figure 3.3 allows us to observe that in later slices, ViT exhibits serpentine, thin-line errors, indicating a deficiency in capturing fine details as compared to TFNO. The CNN-RNN and CNN-LSTM, although they too exhibit learning capabilities, present significantly larger errors than TFNO and ViT, which are not only quantitatively substantial but also lack meaningful physical interpretation.

Figures 3.4 and 3.6 present a comparative visualization of various neural network models' performance on the x and y-axis data, which exhibit a higher complexity with a more abundant presence of features within the images. The CNN-RNN and CNN-LSTM models struggle to effectively manage this type of data. In the realm of convolutional neural networks, filters are predominantly localized and adept at capturing local patterns such as edges and shapes. However, when considering the

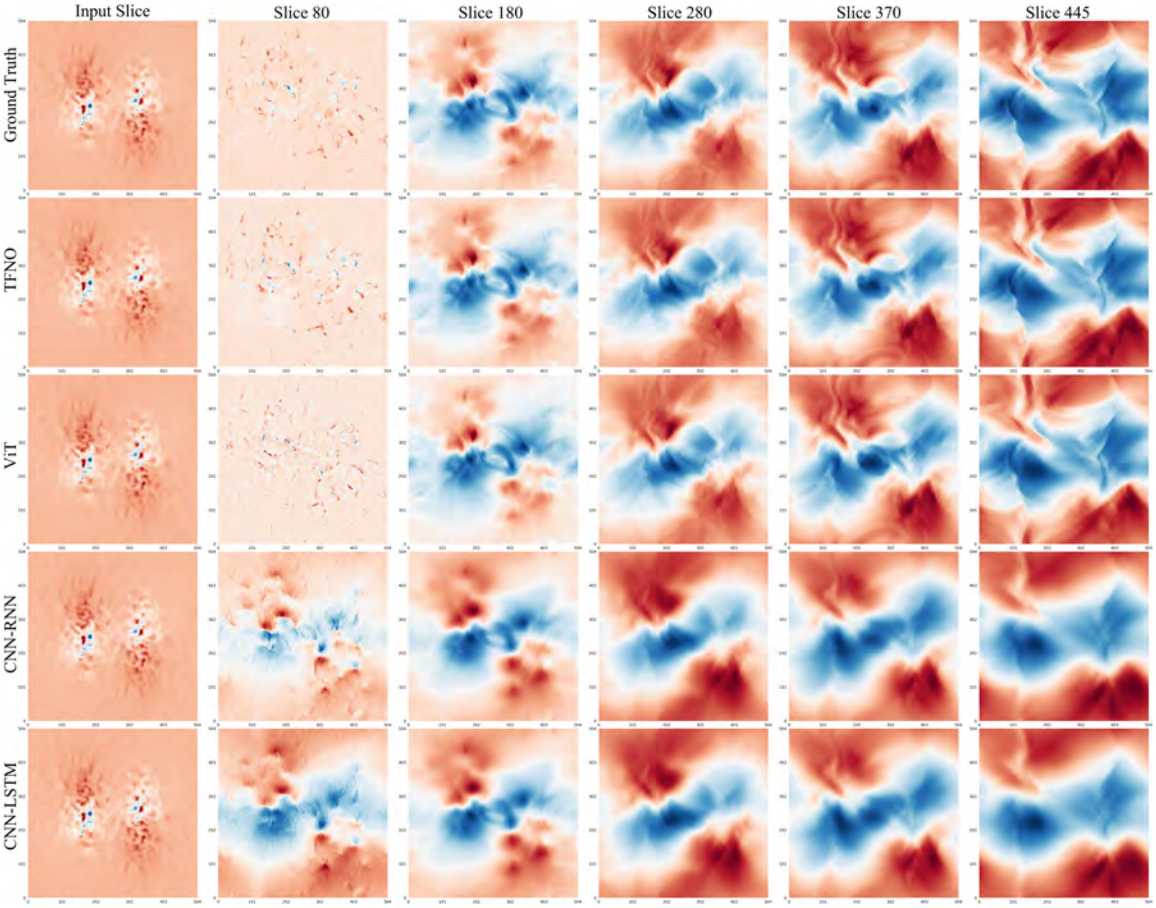

Figure 3.4 Comparative visualization of the B_y predictive performance of various neural network models.

MHD equations, a significant degree of global correlation is present. Events occurring within one region have the potential to impact distant areas, contributing to the notion that CNN-RNN and CNN-LSTM models falter in adapting to the x and y-axis data due to their limited capacity to address these global dependencies. The localized nature of convolutional filters, while beneficial for identifying immediate, discernible patterns, falls short in capturing the extensive, interconnected dynamics characteristic of MHD phenomena. This shortfall underscores the challenge posed by the inherent global correlations within the data, where the actions within one sector resonate across the broader field, necessitating an analytical approach that encompasses both local and distant interactions.

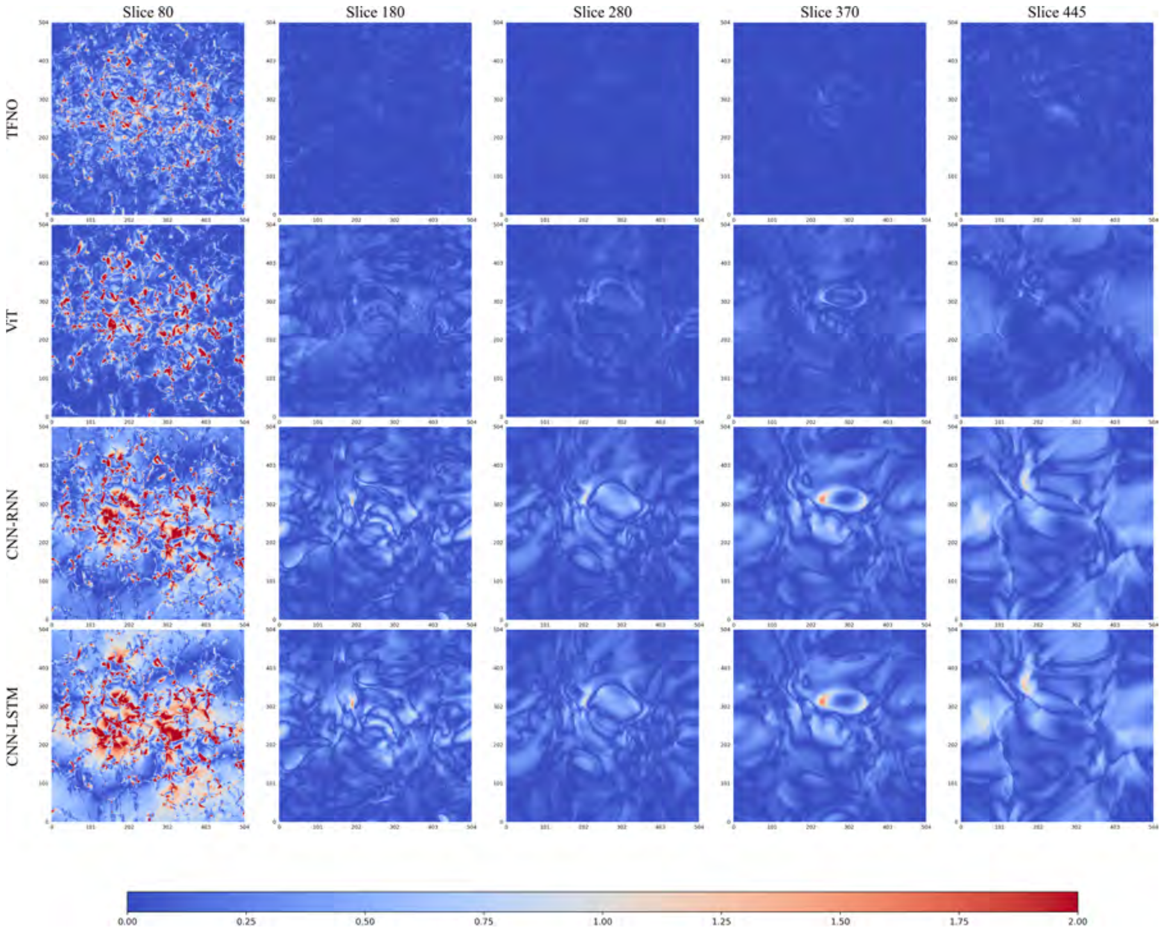

Figure 3.5 B_y Error map for four models.

Contrastingly, the TFNO and ViT demonstrate a pronounced advantage in processing global data. The TFNO, through its innovative architecture, bridges the gap between local and global data analysis by efficiently encoding the comprehensive scope of interactions within its operational framework. Similarly, ViT, leveraging the self-attention mechanism inherent to Transformers, excels in identifying and integrating distant relational data, facilitating a holistic understanding of the image content. The capacity of TFNO and ViT to adeptly manage complex datasets, especially those imbued with significant global interrelations as seen in the MHD equations, positions these models as superior alternatives for tasks requiring an extensive synthesis of both local and widespread data characteristics. This distinction highlights the limitations of traditional CNN-based approaches in tackling tasks with inherent global complexities.

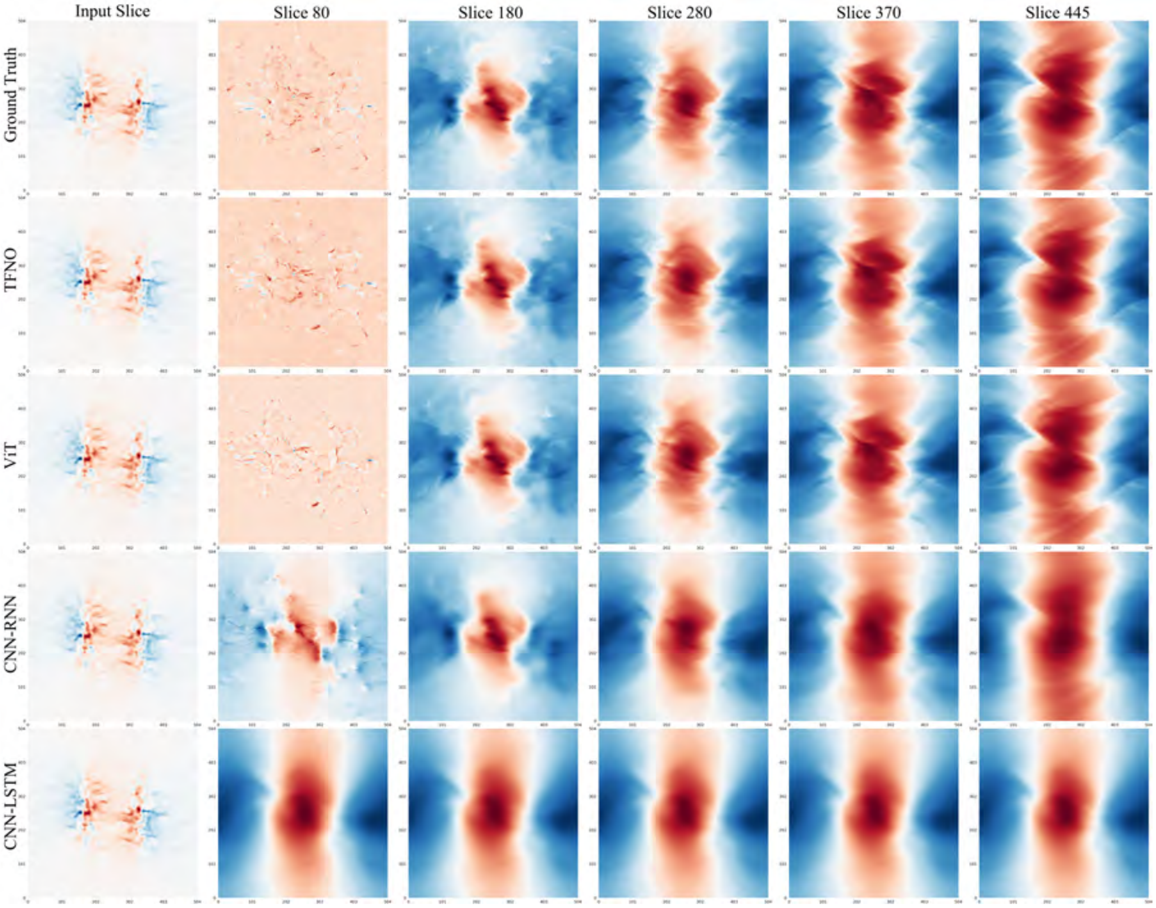

Figure 3.6 Comparative visualization of the B_x predictive performance of various neural network models.

Similarly, by comparing the error maps in Figures 3.4 and 3.6, one can discern that the prediction results of TFNO are nearly indistinguishable from the ground truth, whereas ViT reveals remarkable discrepancies in finer details when juxtaposed with the ground truth. In the context of accelerating coronal magnetic field computations, achieving heightened precision is invariably the paramount priority.

For objectives centered around the acceleration of coronal magnetic field computations, where higher accuracy not only enhances the reliability of simulations but also contributes to more effective predictive models of space weather phenomena, the choice of computational tools becomes critical. The comparison between TFNO

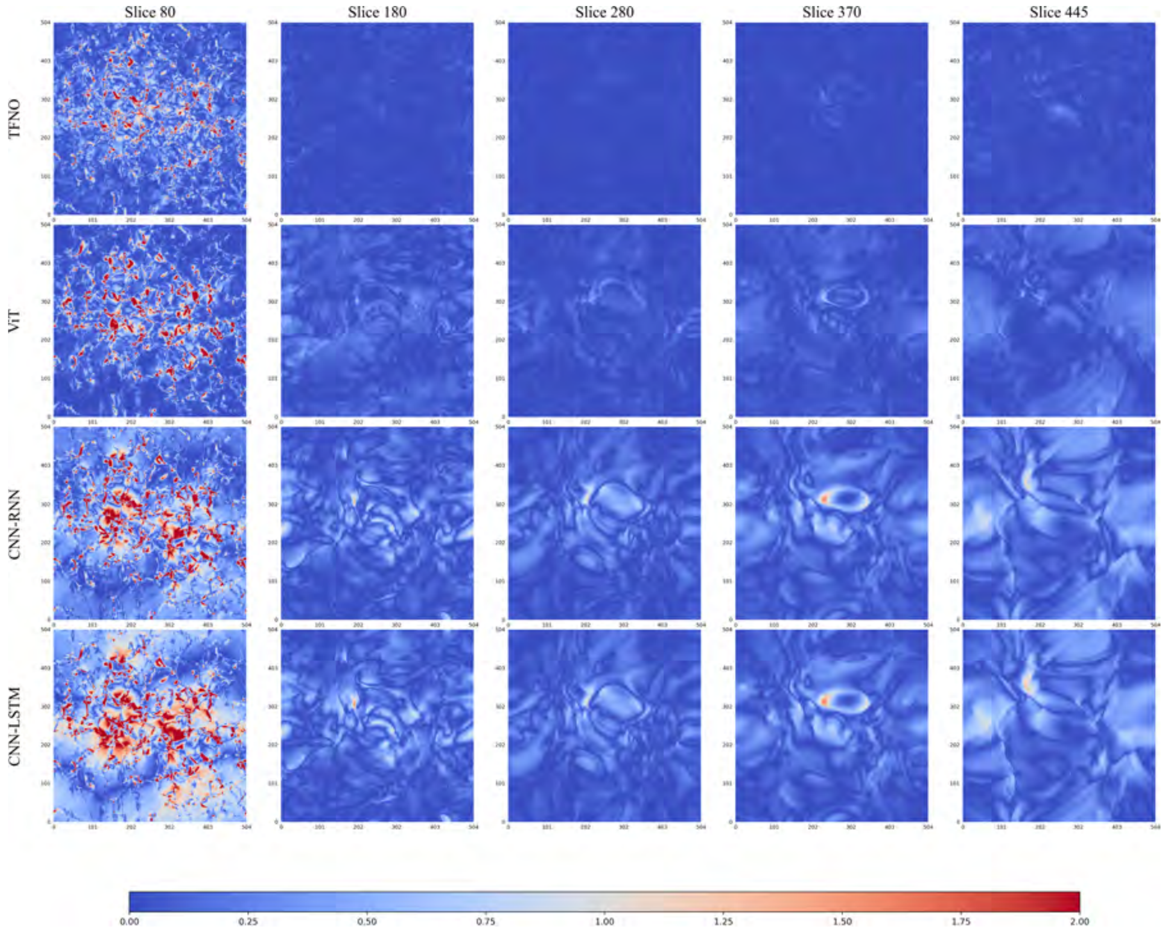

Figure 3.7 B_x Error map for four models.

and ViT, as evident from their respective error maps, distinctly highlights the superior performance of TFNO in meeting the rigorous demands of precision. This observation advocates for a nuanced approach in selecting neural network models tailored to the specific requirements of accuracy and detail-oriented tasks within the domain of computational physics and beyond.

In conclusion, the TFNO stands out for its robustness and ability to capture detailed features with lower errors, 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 detail-specific predictions. The recurrent architectures, CNN-RNN and CNN-LSTM, demonstrate a moderate performance that might benefit from further optimization to reach the efficacy of TFNO and ViT.

3.3.2 Physics Analysis

When assessing the performance of various ML models in reproducing the Reynolds-averaged Magnetohydrodynamics (RMHD) outputs, it is vital to check whether the models can successfully replicate the comprehensive physical quantities found in the synthetic 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 models' capabilities in reflecting the magnetic field intensity 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 models' competencies not only in emulating the RMHD 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 3.8 shows the 2D histograms that compare predictions from different ML models (TFNO, ViT, CNN-LSTM, and CNN-RNN) to the ground truth across various height = 2.0 Mm, 7.0 Mm, and 11.0 Mm. Each model 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, closer to the surface, the TFNO and ViT models 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 and 11.0, TFNO and ViT consistently keep points near the diagonal, showcasing sustained prediction accuracy. On the contrary, CNN-LSTM and CNN-RNN models exhibit a wider spread of points, hinting at a decline in predictive precision.

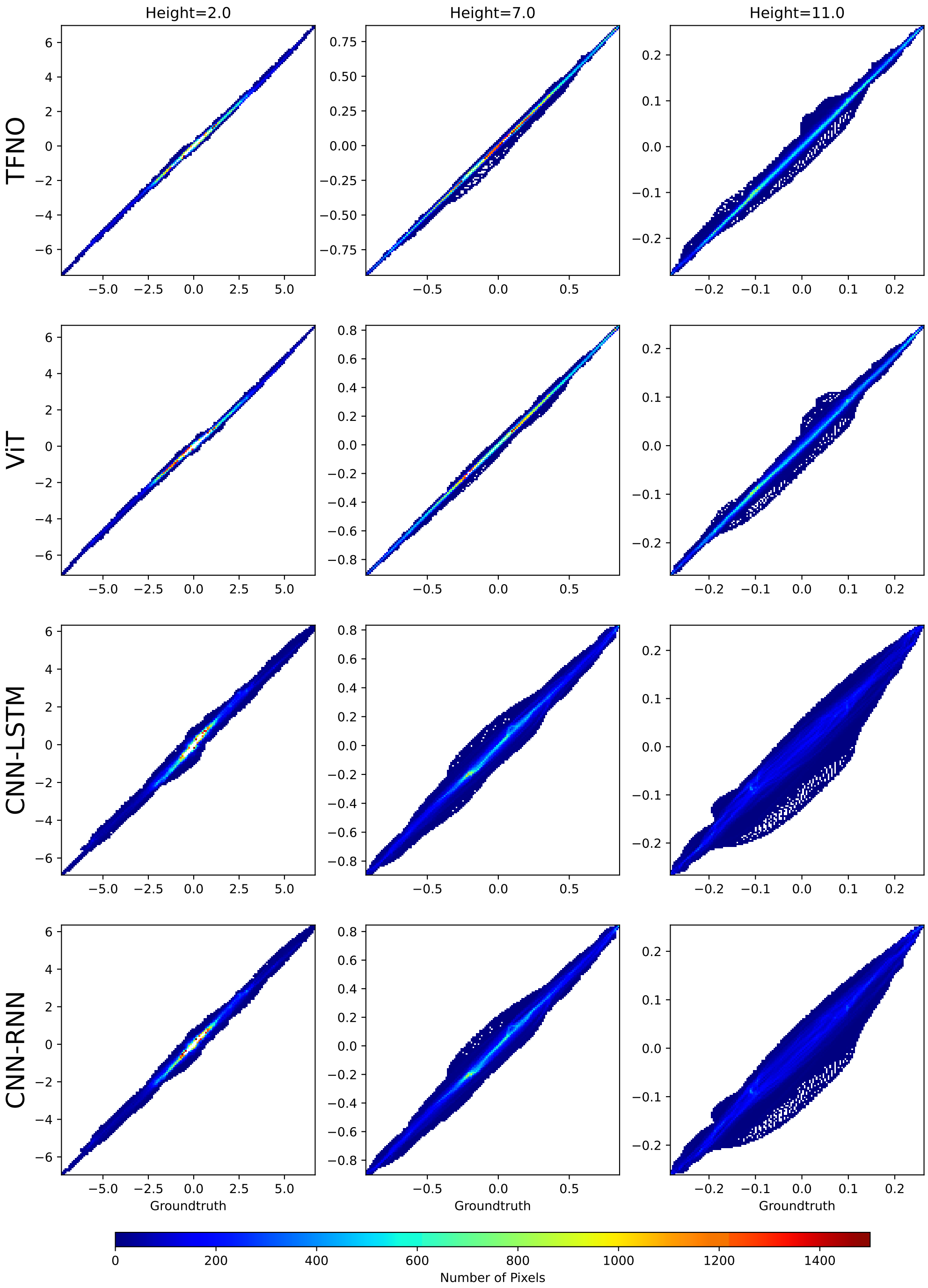

Figure 3.8 2D histogram of TFNO, ViT, CNN-LSTM, and CNN-RNN to the ground truth, at height = 2.0 Mm, 7.0 Mm, and 11.0 Mm. Color gradient denotes the number of data points.

Moreover, we have evaluated physics-based measurements varying with height to assess each model's ability to reproduce the RMHD model outputs. Figure 3.9

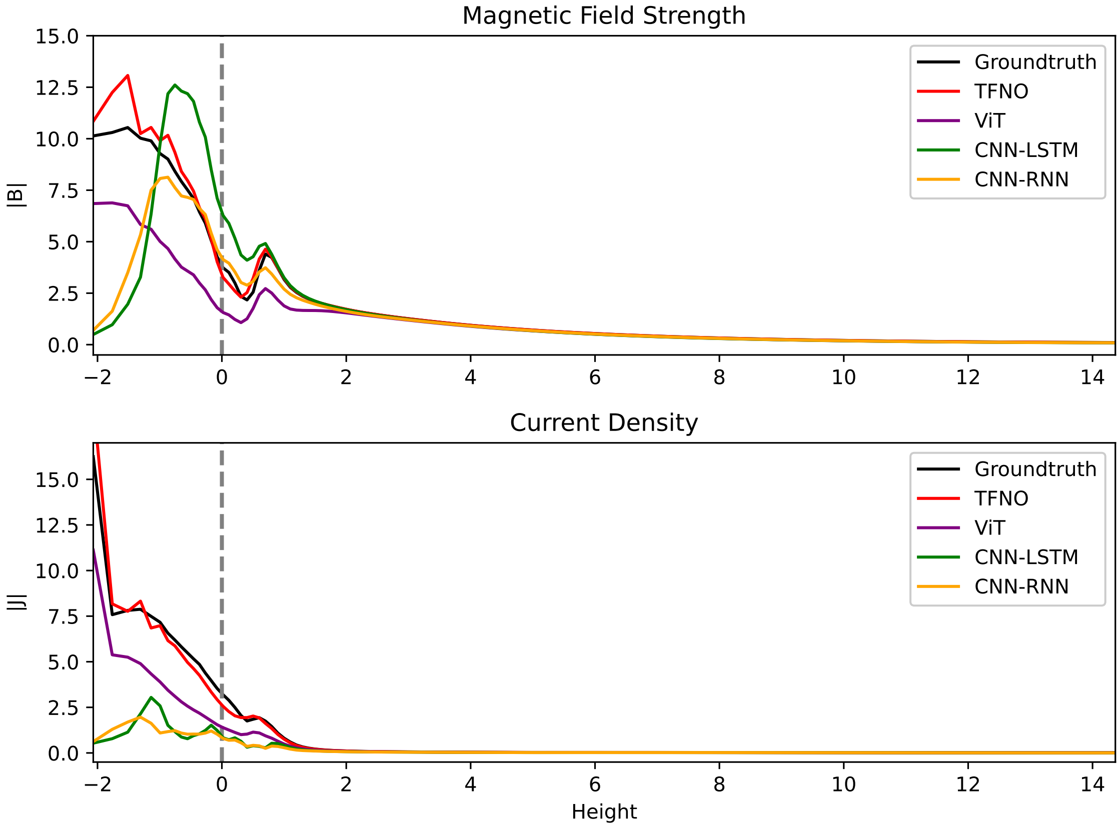

Figure 3.9 Comparison of physics quantities derived from TFNO, ViT, CNN-LSTM, and CNN-RNN models. Upper panel: Variation of magnetic field strength as a function of height. Lower panel: Variation of current density as a function of height.

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 models follow a similar trajectory. Overall, TFNO delivers more consistent and accurate predictions of these physical quantities, particularly close to the surface where the magnetic settings are complex.

Figure 3.10 illustrates the variation of the horizontally averaged inclination angle θ and azimuthal angle ϕ as a function of height, derived from different ML models

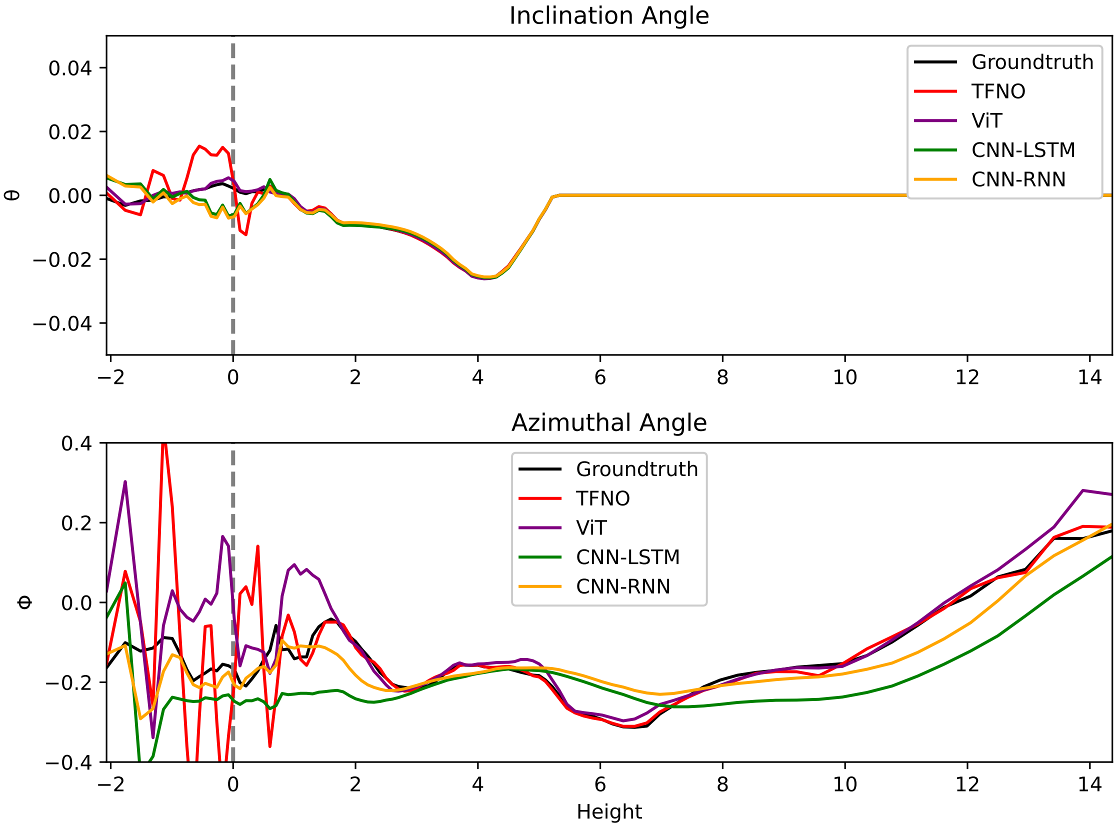

Figure 3.10 Comparison of field orientation derived from TFNO, ViT, CNN-LSTM, and CNN-RNN models. Upper panel: Variation of inclination angle as a function of height. Lower panel: Variation of azimuthal angle as a function of height.

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 models 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 models 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 sooner than ViT. Although CNN-LSTM and CNN-RNN models 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 model with the ground truth suggests it may provide a more physically accurate magnetic field orientation with height.

CHAPTER 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. The TFNO model, with its tensorized approach, has demonstrated the capability to simulate the complex magnetic solar outer atmosphere with reduced computational demands compared to traditional physics-driven models like Bifrost. By leveraging the advantages of the Fourier Neural Operator, the TFNO exhibits the potential to capture periodic phenomena, crucial for the accurate representation of space weather dynamics.

The findings from the comparative study reveal that the TFNO model not only achieves superior prediction accuracy but also maintains computational efficiency, as evidenced by the lowest MSE among the models evaluated and competitive training times. This balance is particularly outstanding given the fewer parameters within the TFNO architecture. In contrast, models like the ViT, despite showing promise in training error minimization, indicate a tendency toward overfitting, which is manifested in a higher test MSE. This discrepancy underscores the need to carefully consider model complexity and its impact on generalization capabilities.

In the post physics analysis, the TFNO stands out for its precision in predicting magnetic field strength and current density, crucially maintaining this accuracy across varying atmospheric heights. The TFNO model exhibits a remarkable adherence to ground truth data, particularly below a height of 2Mm, a region characterized by complex magnetic interactions. Its performance is consistently superior to that of other machine learning models, such as the ViT, CNN-LSTM, and CNN-RNN, 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.

Looking forward, we do not have high-resolution ground truth data so we can not validate TFNO's super-resolution capabilities, which will be a future challenge. These limitations present opportunities for future research, to enhance the predictive accuracy of such neural operators and extend their applications to higher-resolution datasets. This will require innovations in data preprocessing and neural network modeling, potentially leading to more nuanced simulations that can contribute more substantially to our understanding of solar phenomena and their effects on Earth.

For future work, it is essential to address the current limitations and explore the entire data cube in training processes. Additionally, validating the TFNO's super-resolution capabilities will be a crucial step. Furthermore, we will try to apply TFNO to other datasets such as 3D solar magnetic field datasets of active regions built by using the nonlinear force-free magnetic field (NLFFF) extrapolation [32].

REFERENCES

[1] J. E. Borovsky, G. L. Delzanno, P. J. Erickson, A. Halford, B. Lavraud, S. Savage *et al.*, "The future of space physics 2022," *Frontiers in Astronomy and Space Sciences*, vol. 11, p. 1403148.

[2] P. Judge and H. Peter, "The structure of the chromosphere: Properties pertaining to element fractionation," in *Solar Composition and its Evolution—from Core to Corona: Proceedings of an ISSI Workshop 26–30 January 1998, Bern, Switzerland.* Springer, 1998, pp. 187–202.

[3] S. Braginskii, "Transport processes in a plasma," *Reviews of plasma physics*, vol. 1, p. 205, 1965.

[4] V. Shafranov, "On magnetohydrodynamical equilibrium configurations," *Soviet Physics JETP*, vol. 6, no. 3, p. 1013, 1958.

[5] L. Woltjer, "A theorem on force-free magnetic fields," *Proceedings of the National Academy of Sciences*, vol. 44, no. 6, pp. 489–491, 1958.

[6] Y.-M. Wang and N. Sheeley Jr, "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.*, vol. 392, pp. 310–319, 1992.

[7] J. Leenaarts, M. Carlsson, V. Hansteen, and R. Rutten, "Non-equilibrium hydrogen ionization in 2d simulations of the solar atmosphere," *Astronomy & Astrophysics*, vol. 473, no. 2, pp. 625–632, 2007.

[8] M. Den, T. Tanaka, Y. Kubo, and S. Watari, "Three-dimensional mhd simulation of the solar wind from the solar surface to 400 solar radius using reppu (reproduce plasma universe) code," in *Proceedings of the 34Th International Cosmic Ray Conference (ICRC2015). 30 July-6 August 2015. The Hague, The Netherlands*, 2015.

[9] M. Carlsson, V. H. Hansteen, B. V. Gudiksen, J. Leenaarts, and B. De Pontieu, "A publicly available simulation of an enhanced network region of the sun," *Astronomy & Astrophysics*, vol. 585, p. A4, 2016.

[10] J. Berg and K. Nyström, "A unified deep artificial neural network approach to partial differential equations in complex geometries," *Neurocomputing*, vol. 317, pp. 28–41, 2018.

[11] J. Han, A. Jentzen, and W. E, "Solving high-dimensional partial differential equations using deep learning," *Proceedings of the National Academy of Sciences*, vol. 115, no. 34, pp. 8505–8510, 2018.

[12] Z. Long, Y. Lu, X. Ma, and B. Dong, “Pde-net: Learning pdes from data,” in *International conference on machine learning.* PMLR, 2018, pp. 3208–3216.

[13] A. Malek and R. S. Beidokhti, “Numerical solution for high order differential equations using a hybrid neural network—optimization method,” *Applied Mathematics and Computation*, vol. 183, no. 1, pp. 260–271, 2006.

[14] Y. Yang and F. Shen, “Modeling the global distribution of solar wind parameters on the source surface using multiple observations and the artificial neural network technique,” *Solar Physics*, vol. 294, no. 8, p. 111, 2019.

[15] R. Jarolim, J. Thalmann, A. Veronig, and T. Podladchikova, “Probing the solar coronal magnetic field with physics-informed neural networks,” *Nature Astronomy*, vol. 7, no. 10, pp. 1171–1179, 2023.

[16] H. Baty and V. Vigon, “Modelling solar coronal magnetic fields with physics-informed neural networks,” *Monthly Notices of the Royal Astronomical Society*, vol. 527, no. 2, pp. 2575–2584, 2024.

[17] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, “Fourier neural operator for parametric partial differential equations,” *arXiv preprint arXiv:2010.08895*, 2020.

[18] T. Falk, D. Mai, R. Bensch, Ö. Çiçek, A. Abdulkadir, Y. Marrakchi, A. Böhm, J. Deubner, Z. Jäckel, K. Seiwald *et al.*, “U-net: deep learning for cell counting, detection, and morphometry,” *Nature methods*, vol. 16, no. 1, pp. 67–70, 2019.

[19] J. Kossaifi, N. Kovachki, K. Azizzadenesheli, and A. Anandkumar, “Multi-grid tensorized fourier neural operator for high-resolution pdes,” *arXiv preprint arXiv:2310.00120*, 2023.

[20] Å. Nordlund and K. Galsgaard, “A 3d mhd code for parallel computers,” 1995.

[21] K. Galsgaard and Å. Nordlund, “Heating and activity of the solar corona: 1. boundary shearing of an initially homogeneous magnetic field,” *Journal of Geophysical Research: Space Physics*, vol. 101, no. A6, pp. 13 445–13 460, 1996.

[22] B. V. Gudiksen, M. Carlsson, V. H. Hansteen, W. Hayek, J. Leenaarts, and J. Martínez-Sykora, “The stellar atmosphere simulation code bifrost-code description and validation,” *Astronomy & Astrophysics*, vol. 531, p. A154, 2011.

[23] X. Zhao, Y. Sun, T. Zhang, and B. Xu, “Local convolution enhanced global fourier neural operator for multiscale dynamic spaces prediction,” *arXiv preprint arXiv:2311.12902*, 2023.

[24] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, “Neural operator: Graph kernel network for partial differential equations,” *arXiv preprint arXiv:2003.03485*, 2020.

[25] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.

[26] J. Sun, Z. Niu, K. A. Innanen, J. Li, and D. O. Trad, "A theory-guided deep-learning formulation and optimization of seismic waveform inversion," *Geophysics*, vol. 85, no. 2, pp. R87–R99, 2020.

[27] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[28] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "Cnn-rnn: A unified framework for multi-label image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2285–2294.

[29] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?" in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 6546–6555.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[31] J. M. Taylor, D. Pardo, and I. Muga, "A deep fourier residual method for solving pdes using neural networks," *Computer Methods in Applied Mechanics and Engineering*, vol. 405, p. 115850, 2023.

[32] Z. Zhao, L. Xu, X. Zhu, X. Zhang, S. Liu, X. Huang, Z. Ren, and Y. Tian, "A large-scale dataset of three-dimensional solar magnetic fields extrapolated by nonlinear force-free method," *Scientific Data*, vol. 10, no. 1, p. 178, 2023.