Research in Astronomy and Astrophysics, 24:105010 (11pp), 2024 October https://doi.org/10.1088/1674-4527/ad76eb

Physics-informed Neural Network for Force-free Magnetic Field Extrapolation

Yao Zhang[1,2], Long Xu[3], and Yihua Yan[1,2]
[1] State Key Laboratory of Space Weather, National Space Science Center, Chinese Academy of Sciences, Beijing 100190, China
[2] University of Chinese Academy of Sciences, Beijing 101408, China
[3] Faculty of Information Science and Engineering, Ningbo University, Ningbo 315211, China; lxu@nao.cas.cn
Received 2024 June 22; revised 2024 August 4; accepted 2024 August 16; published 2024 October 9

Abstract

In this paper, we propose a physics-informed neural network extrapolation method that leverages machine learning techniques to reconstruct coronal magnetic fields. We enhance the classical neural network structure by introducing the concept of a quasi-output layer to address the challenge of preserving physical constraints during the neural network extrapolation process. Furthermore, we employ second-order optimization methods for training the neural network, which are more efficient compared to the first-order optimization methods commonly used in classical machine learning. Our approach is evaluated on the widely recognized semi-analytical model proposed by Low and Lou. The results demonstrate that the deep learning method achieves high accuracy in reconstructing the semi-analytical model across multiple evaluation metrics. In addition, we validate the effectiveness of our method on the observed magnetogram of active region.

Key words: Sun: magnetic fields – Sun: corona – magnetohydrodynamics (MHD)

1. Introduction

The force-free field model, as discussed by Neukirch (2005), Schrijver et al. (2006), Wiegelmann (2008), and Wiegelmann & Sakurai (2021), is a crucial theoretical framework for studying the magnetic field of the solar corona. This model is essential for understanding solar burst events in the corona, such as coronal mass ejections, filament eruptions, and flares. Magnetic field extrapolation using the force-free field involves reconstructing the coronal magnetic field by using the photospheric magnetic field as a boundary condition and solving the partial differential equations (PDEs) associated with the force-free field model.

The force-free model satisfies the following equations:

$$(\nabla \times \boldsymbol{B}) \times \boldsymbol{B} = \boldsymbol{0}, \quad \text{in } \Omega, \tag{1}$$

where $\boldsymbol{B} = \boldsymbol{B}(x, y, z) = (B_x, B_y, B_z)$ is the field, and Ω is the open space above the Sun.

$$\nabla \cdot \boldsymbol{B} = 0, \quad \text{in } \Omega, \tag{2}$$

and

$$\boldsymbol{B} = \boldsymbol{B_0}, \quad \text{on } \partial\Omega, \tag{3}$$

where $\partial\Omega$ is the boundary of Ω.

Equation (1) describes that the Lorentz force is zero, and it also implies that

$$\nabla \times \boldsymbol{B} = \alpha \boldsymbol{B}, \quad \text{in } \Omega, \tag{4}$$

where $\alpha = \alpha(x, y, z)$, i.e., α is a function of spatial location (x, y, z). This equation describes various types of magnetic fields: the potential field (PF) when $\alpha = 0$, the linear force-free field (LFFF) when α is constant, and the nonlinear force-free field (NLFFF) when α is variable.

Combining Equations (2), (3), and (4), we obtain that

$$\nabla \times \boldsymbol{B} = \alpha \boldsymbol{B} \text{ in } \Omega, \ \nabla \cdot \boldsymbol{B} = 0 \text{ in } \Omega, \ \boldsymbol{B} = \boldsymbol{B_0} \text{ on } \partial\Omega. \tag{5}$$

The force-free model is succinctly and accurately formulated by Equation (5). The PF and LFFF models can be solved analytically using the Green's function method or the Fourier method, as discussed by Wiegelmann & Sakurai (2021). However, it remains unclear whether a solution to the NLFFF model exists or is unique (Priest 2014). Despite this uncertainty, various numerical methods have been proposed to solve the NLFFF model. Generally, most existing numerical methods for the NLFFF model fall into six main categories: the vertical integration method (Nakagawa 1974; Amari et al. 1997), the magnetohydrodynamics (MHD) evolutionary method (Mikić & McClymont 1994; Jiang et al. 2013), the Grad–Rubin method (Grad & Rubin 1958; Régnier et al. 2002), the optimization method (Wheatland et al. 2000; Wiegelmann et al. 2006), the boundary element method (Yan & Sakurai 2000; Yan & Li 2006), and the magneto-frictional method (Yang et al. 1986; Guo et al. 2016). The numerical algorithms mentioned above are grid-based methods due to them all involving numerical differentiation or integration. In this paper, we propose a grid-free method based on deep learning using automatic differentiation (Baydin et al. 2018).

In recent years, deep learning has experienced rapid development and has become closely intertwined with

scientific computing. The combination of machine learning and scientific computing has provided new insights and methodologies for tackling complex scientific and engineering problems. This integration has propelled advancements in scientific research and has offered powerful tools and theoretical foundations for data-driven scientific simulations, predictions, and optimizations.

The remainder of this paper is organized as follows: Section 2 provides the development of our proposed force-free neural network model. Section 3 develops the corresponding optimization algorithm. The test and results are shown in Section 4. Finally, the conclusion is given in Section 5.

2. Force-free Neural Network Model

The force-free model can be written as the following:

$$\begin{aligned}&\text{Differential Equation 1: } \nabla \times \boldsymbol{B}(x, y, z) \times \boldsymbol{B}(x, y, z) = \mathbf{0},\ (x, y, z) \in \Omega,\\ &\text{Differential Equation 2: } \nabla \cdot \boldsymbol{B}(x, y, z) = 0,\ (x, y, z) \in \Omega,\\ &\text{Boundary Condition: } \boldsymbol{B}|_{\partial\Omega} = \boldsymbol{B}(x', y', z'),\ (x', y', z') \in \partial\Omega.\end{aligned} \tag{6}$$

The solution to Equation (6) can be approximated using a neural network, leveraging the universal approximation theorem (Cybenko 1989; Hornik 1991; Zhou 2020). Let $\boldsymbol{B}^{NN}(x, y, z)$ represent the solution obtained from the neural network. This solution satisfies the following mathematical model

$$\begin{aligned}\lambda_0 \left\|\boldsymbol{B}^{NN}(x', y', z') - \boldsymbol{B}(x', y', z')\right\|_2^2 + \lambda_1 \left\|\nabla \cdot \boldsymbol{B}^{NN}(x, y, z)\right\|_2^2\\ \cdots + \lambda_2 \frac{\left\|\nabla \times \boldsymbol{B}^{NN}(x, y, z) \times \boldsymbol{B}^{NN}(x, y, z)\right\|_2^2}{\left\|\nabla \times \boldsymbol{B}^{NN}(x, y, z)\right\|_2^2 \cdot \left\|\boldsymbol{B}^{NN}(x, y, z)\right\|_2^2} = 0,\end{aligned} \tag{7}$$

where λ_0, λ_1, and λ_2 are arbitrary positive real numbers. This model is known as the physics-informed neural network (PINN) model (Raissi et al. 2019). The term physics-informed in this context signifies being force-free in our research. Hence, we aptly label this neural network as the force-free neural network.

In practice, the determination of $\boldsymbol{B}^{NN}(x, y, z)$ involves minimizing the cost function $f(\boldsymbol{p})$. The cost function is defined as follows

$$\begin{aligned}f(\boldsymbol{p}) = \lambda_0 \frac{1}{m}\sum_{i=1}^{m} \left\|\boldsymbol{B}^{NN}(x_i', y_i', z_i'; \boldsymbol{p}) - \boldsymbol{B}(x_i', y_i', z_i')\right\|_2^2\\ \cdots + \lambda_1 \frac{1}{n}\sum_{j=1}^{n} \left\|\nabla \cdot \boldsymbol{B}^{NN}(x_j, y_j, z_j; \boldsymbol{p})\right\|_2^2 \quad + \quad \cdots,\\ \lambda_2 \frac{1}{n}\sum_{j=1}^{n} \frac{\left\|\nabla \times \boldsymbol{B}^{NN}(x_j, y_j, z_j; \boldsymbol{p}) \times \boldsymbol{B}^{NN}(x_j, y_j, z_j; \boldsymbol{p})\right\|_2^2}{\left\|\nabla \times \boldsymbol{B}^{NN}(x_j, y_j, z_j; \boldsymbol{p})\right\|_2^2 \cdot \left\|\boldsymbol{B}^{NN}(x_j, y_j, z_j; \boldsymbol{p})\right\|_2^2}\end{aligned} \tag{8}$$

where $(x_i', y_i', z_i') \in \partial\Omega$ and $(x_j, y_j, z_j) \in \Omega$ represent randomly sampled training sets. The sample size on $\partial\Omega$ is denoted by m, while the sample size in Ω is denoted by n. By adjusting the parameter $\boldsymbol{p}$ and iteratively updating $\boldsymbol{p}$ in the neural network, we can minimize the cost function to obtain the desired magnetic field $\boldsymbol{B}(x, y, z)$.

Remark 1. It is obvious that if $f(\boldsymbol{p})$ in Equation (8) is reduced to zero, then this implies Equations (1), (2), and (3) or (6) hold immediately.

Remark 2. Keep in mind that the learnable parameter $\boldsymbol{p}$ is a vector collecting all weights and biases of the force-free neural network.

To address this optimization problem, which minimizes $f(\boldsymbol{p})$ in Equation (8) to 0, we propose a deep learning approach using a multilayer perceptron (MLP). The architecture of the MLP, displayed in Figure 1, consists of $(L+2)$ fully connected layers, each with N units. The activation function we used is the hyperbolic tangent sigmoid (tanh) elementwise function, defined as $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, where x is a real number. Both the input and output layers are fully connected layers with a size of 3. The architecture can be formalized as follows:

1. The first layer:

$$[u_1^{(1)}, u_2^{(1)}, \cdots y_N^{(1)}]^T = \tanh\left(W^{(1)}[x, y, z]^T + \boldsymbol{b}^{(1)}\right),$$

where $W^{(1)} \in \mathbb{R}^{N\times 3}$, $\boldsymbol{b}^{(1)} \in \mathbb{R}^N$.

2. The hidden layers:

$$[u_1^{(i)}, u_2^{(i)}, \cdots u_N^{(i)}]^T = \tanh\left(W^{(i)}[u_1^{(i-1)}, u_2^{(i-1)}, \cdots u_N^{(i-1)}]^T + \boldsymbol{b}^{(i)}\right),$$

where $W^{(i)} \in \mathbb{R}^{N\times N}$, $\boldsymbol{b}^{(i)} \in \mathbb{R}^N$, $2 \leqslant i \leqslant L$.

3. The quasi-output layer:

$$[B_x', B_y', B_z']^T = \left(W^{(L+1)}[u_1^{(L)}, u_2^{(L)}, \cdots u_N^{(L)}]^T + \boldsymbol{b}^{(L+1)}\right),$$

where $W^{(L+1)} \in \mathbb{R}^{3\times N}$, $\boldsymbol{b}^{(L+1)} \in \mathbb{R}^3$.

4. The final-output layer:

$$[B_x, B_y, B_z]^T = \begin{bmatrix} w_{11} & 0 & 0 \\ 0 & w_{22} & 0 \\ 0 & 0 & w_{33} \end{bmatrix} [B_x', B_y', B_z']^T + [b_1, b_2, b_3]^T \tag{9}$$

where $w_{ii}, b_i \in \mathbb{R}$ for $1 \leqslant i \leqslant 3$.

It is crucial to include a quasi-output layer in the architecture of the force-free neural network. Without this layer, the loss function $f(\boldsymbol{p})$ may not converge to zero during the training process. Moreover, in some cases, it can even diverge and become significantly larger than zero, as depicted in Figure 2.

It should be noted that for a set of PDEs with a unique solution, different boundary conditions will yield different solutions. A PINN minimizes the residuals of the equations and boundary data by adjusting its learnable parameters. When these residuals are sufficiently small, the PINN is considered to provide a nonlinear representation of the PDEs' solutions that

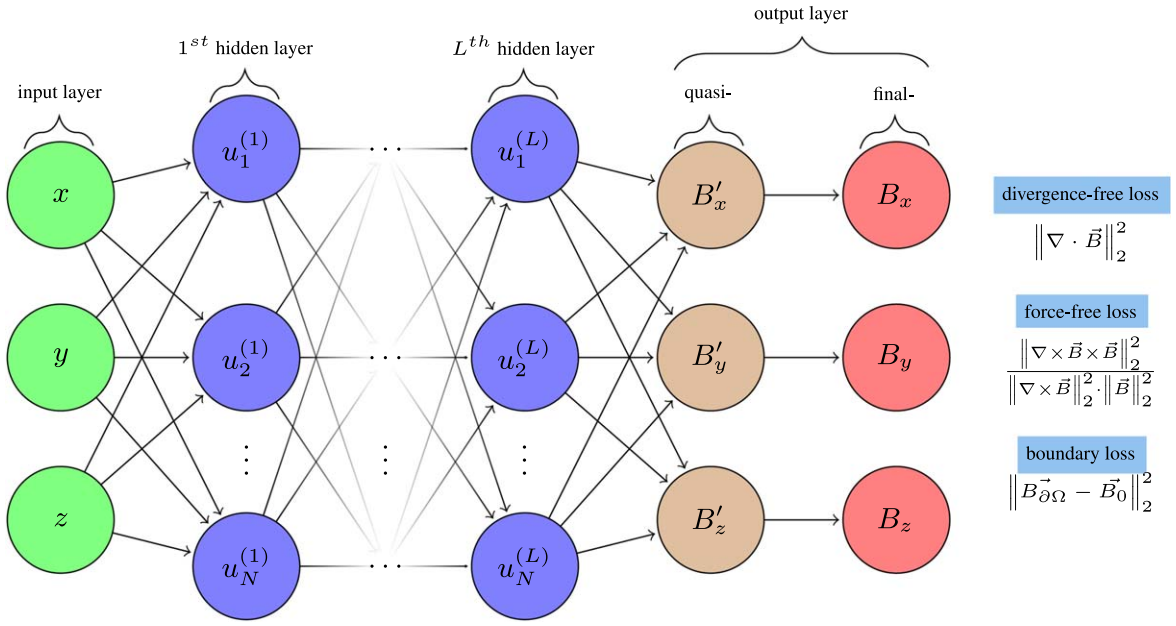
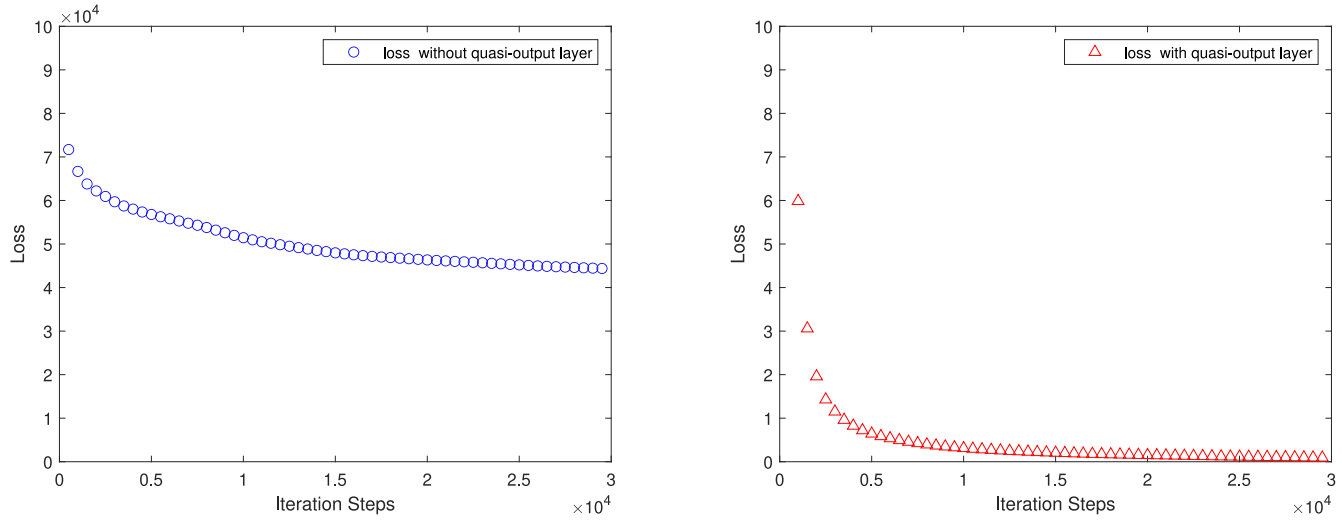


Figure 1. The architecture for our force-free neural network, where $\boldsymbol{B} = \boldsymbol{B}^{NN}(x, y, z)$, $\boldsymbol{B}_{\partial\Omega} = \boldsymbol{B}^{NN}(x', y', z')$, $\boldsymbol{B}_0 = \boldsymbol{B}(x', y', z')$, $(x, y, z) \in \Omega$, $(x', y', z') \in \partial\Omega$. The output is the solution of (6) with respect to any input $(x, y, z) \in \Omega \cup \partial\Omega$.



(a) Without the quasi-output layer (b) With the quasi-output layer

Figure 2. Training loss vs. iterations of the force-free neural network in an example. Without the quasi-output layer, it cannot converge to near zero, while with the quasi-output layer it can.

satisfies the boundary conditions. Jarolim et al. (2023) implements specific treatment of the data residual term as follows

$$L_{B_0}^{\text{New}} = \left\| \max\left\{ \text{abs}\left(\frac{\boldsymbol{B}}{B_0} - \frac{\boldsymbol{B_0}}{B_0} \right) - \frac{\boldsymbol{B}_{\text{error}}}{B_0}, 0 \right\} \right\|^2 = \frac{1}{B_0^2} L_{B_0},$$

where the term $\boldsymbol{B}_{\text{error}}$ represents the error map and $B_0 = 2500$ G. Mathematically, this operation reduces the optimization difficulty of the neural network. However, this data residual is artificially reduced by a factor of 6250,000. Numerically, after introducing $\boldsymbol{B}_{\text{error}}$, which relaxes the deviation of the boundary data, the difference between the output of the PINN

and the observations at the boundary will affect the accuracy of the numerical solution of the PDEs.

We do not use normalization or denormalization operations; instead, we propose the concept of a quasi-output layer. This approach is chosen partly to facilitate the training of the neural network and partly to avoid amplifying or diminishing the residuals of the equations and boundary data terms. This method helps ensure that the numerical solutions of the PDEs, which satisfy the boundary conditions, are found without disrupting the physical laws inherent in the data.

3. Optimization

To minimize the cost function $f(\boldsymbol{p})$, we adopt an iterative scheme using the update rule:

$$\boldsymbol{p}_{k+1} = \boldsymbol{p}_k + \alpha_k \boldsymbol{d}_k, \tag{10}$$

where $\boldsymbol{p}_k$ represents the parameter vector at iteration k, α_k is the step size, and $\boldsymbol{d}_k$ is the search direction. The goal is to find $\boldsymbol{p}$ such that $f(\boldsymbol{p}_k) \leqslant f(\boldsymbol{p}_{k+1})$, approaching a value close to 0 as k approaches infinity.

Motivated by the alternating variable method, we first fix $\boldsymbol{d}_k$ and seek α_k. Then, after fixing α_k, we update $\boldsymbol{d}_k$. Next, we will discuss the detailed updating process for α_k in Section 3.1, and the updating process for $\boldsymbol{d}_k$ in Section 3.2.

3.1. Update α_k

In this step, we construct a cubic function $\varphi(\alpha)$ defined as

$$\varphi(\alpha) = f(\boldsymbol{p}_k + \alpha \boldsymbol{d}_k),$$

which interpolates $\varphi(0)$, $\varphi'(0)$, $\varphi(a_0)$, $\varphi(a_1)$ (taking suitable a_0, $a_1 > 0$). Then we determine α_k as

$$\alpha_k = \underset{\alpha}{\operatorname{argmin}}\, \varphi(\alpha).$$

This approach allows us to determine the optimal step size α_k for each iteration, enhancing the convergence of the optimization process.

3.2. Update $\boldsymbol{d}_k$

In this step, we consider the dimension of $\nabla f(\boldsymbol{p})$, which is $\nabla f(\boldsymbol{p}) \in \mathbb{R}^{110,706}$ when $L = 10$ and $N = 100$. This dimension is too large to store or compute directly in the numerical algorithm.

To address this issue, we use a limited memory quasi-Newton method called the limited memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm (Liu & Nocedal 1989). Jarolim et al. employ a first-order optimization algorithm. According to Watt et al. (2020), second-order methods, like L-BFGS, typically require fewer iterations to converge to an optimal solution compared to first-order methods. Moreover, Nocedal & Wright, in the book *Numerical Optimization*, highlight that first-order optimization algorithms, such as the ADAM optimizer used by Jarolim et al., can perform poorly without proper input data normalization. In contrast, quasi-Newton methods, like the one we use, are less sensitive to this issue. Jarolim et al. apply normalization and denormalization steps outside the neural network when using a vanilla PINN, whereas we do not use such operations.

In this method, we update $\boldsymbol{d}_k$ as

$$\boldsymbol{d}_k = -C_k \nabla f(\boldsymbol{p}_k),$$

where C_k is updated based on the previous iterations using a recursive formula. Specifically, for $1 \leqslant k \leqslant m$:

$$C_k = E_{k-1} C_{k-1} E_{k-1}^T + g_{k-1} \vec{h}_{k-1} \vec{h}_{k-1}^{\,T}, \ (1 \leqslant k \leqslant m),$$

and for $k > m$:

$$\begin{aligned} C_k = {} & (E_{k-1}^T \cdots E_{k-m}^T) C_k^0 (E_{k-m} \cdots E_{k-1}) \\ & + g_{k-m} (E_{k-1}^T \cdots E_{k-m+1}^T) h_{k-m} h_{k-m}^{\,T} (E_{k-m+1} \cdots E_{k-1}), \\ & + \cdots + g_{k-1} h_{k-1} h_{k-1}^{\,T}, (k > m) \end{aligned}$$

where E_k is a matrix, g_k is a scalar, and C_k^0 is a matrix. That is,

$$E_k = I - g_k i_k h_k^{\,T}, \ g_k = \frac{1}{i_k^{\,T} h_k}, \ C_k^0 = \frac{h_{k-1}^{\,T} i_{k-1}}{i_{k-1}^{\,T} i_{k-1}} I.$$

C_0 is settled by the finite differences at $\boldsymbol{p}_0$ (Nocedal & Wright 2006). The vectors $\boldsymbol{h}_k$ and $\boldsymbol{i}_k$ are defined as

$$h_k = p_{k+1} - p_k, \ i_k = \nabla f_{k+1} - \nabla f_k.$$

The L-BFGS algorithm allows us to update $\boldsymbol{d}_k$ efficiently and effectively in high-dimensional problems, improving the convergence of the optimization process.

4. Test Cases

Low & Lou (1990) provided a set of separable and semi-analytical solutions for Equations (1) and (2) with two parameters (n_{LL} and m_{LL}) in spherical coordinate systems. They then introduced two additional parameters (l and Φ) to obtain the realistic-looking magnetic fields in Cartesian coordinate systems. In this paper, we choose two specific parameter configurations, which are identical to those used in Schrijver et al. (2006) and Valori et al. (2007):

1. Case 1: $n_{LL} = 1$, $m_{LL} = 1$, $l = 0.3$, $\Phi = \frac{\pi}{4}$,
2. Case 2: $n_{LL} = 3$, $m_{LL} = 1$, $l = 0.3$, $\Phi = \frac{4\pi}{5}$.

to evaluate our method.

Our method is tested by striving to reconstruct Case 1 and Case 2 in the 64^3 pixel box defined by $[-1, 1] \times [-1, 1] \times [0, 2]$.

For Case 1, the NLFFF on all six boundaries of the box is provided. We use a 7-layer MLP with ($L = 5$) and a width $N = 128$. The parameters $\lambda_0 = 1$, $\lambda_1 = 1$ and $\lambda_2 = 0.95$ are set, and $m_0 = 10$. The parameter vector $\boldsymbol{p}$ is initialized using Kaiming initialization (He et al. 2015). We randomly select 23,816 points from the boundary $\partial\Omega$ and 360,000 points from the interior Ω to train the MLP.

For Case 2, only the NLFFF on the bottom boundary of the box is given. In this case, we set $L = 7$, $N = 156$, $\lambda_0 = 1.25$, $\lambda_1 = 1$, $\lambda_2 = 1$ and $m_0 = 50$. The parameter vector $\boldsymbol{p}$ is initialized in the same way as in Case 1. We randomly select 4096 points from the bottom boundary of the box and 360,000 points from the interior Ω to train the MLP. To enforce lateral and top boundary conditions, we choose a PF on these boundaries and select 20,482 points from them.

After training the force-free neural network MLP, we obtain the optimized parameter vector

$$\boldsymbol{p}^* = \underset{\boldsymbol{p}}{\operatorname{argmin}} f(\boldsymbol{p}).$$

For the evaluation of the reference NLFFF, we introduce several common metrics as outlined by Schrijver et al. (2006). These metrics include the correlation based on vector inner product ($C_{\rm vec}$), the correlation based on the Cauchy–Schwarz inequality ($C_{\rm cs}$), the normalized vector error ($E_{\rm n}$), the mean vector error ($E_{\rm m}$), the total magnetic energy normalized to the reference field ($\mathcal{E}$), and the ratio between the magnetic energy of the reconstructed and PFs ($\mathcal{E}_{\rm P}$).

The $C_{\rm vec}$ is defined as

$$C_{\rm vec} \equiv \frac{\sum_{i=1}^{M} \boldsymbol{B}_i^{\rm Ref} \cdot \boldsymbol{B}_i^{\rm Rec}}{\left(\sum_{i=1}^{M} \|\boldsymbol{B}_i^{\rm Ref}\|_2^2 \sum_{i=1}^{M} \|\boldsymbol{B}_i^{\rm Rec}\|_2^2\right)^{\frac{1}{2}}},$$

where $\boldsymbol{B}^{\rm Ref}$ is the reference field , and $\boldsymbol{B}^{\rm Rec}$ is the reconstruction of the $\boldsymbol{B}^{\rm Ref}$ (similarly hereinafter). It is clear that if the reference field is exactly reconstructable, then $C_{\rm vec} = 1$.

The $C_{\rm cs}$ is a measure of the angular differences between the reference and the reconstructed fields which are defined as

$$C_{\rm cs} \equiv \frac{1}{M} \sum_{i=1}^{M} \frac{\boldsymbol{B}_i^{\rm Ref} \cdot \boldsymbol{B}_i^{\rm Rec}}{\|\boldsymbol{B}_i^{\rm Ref}\|_2 \|\boldsymbol{B}_i^{\rm Rec}\|_2},$$

where M is the number of the selected vectors in the whole space.

In the two metrics mentioned above, the larger value achieved implies better performance.

The $E_{\rm n}$ is defined as

$$E_{\rm n} \equiv \frac{\sum_{i=1}^{M} \|\boldsymbol{B}_i^{\rm Ref} - \boldsymbol{B}_i^{\rm Rec}\|_2}{\sum_{i=1}^{M} \|\boldsymbol{B}_i^{\rm Ref}\|_2}.$$

The $E_{\rm m}$ is defined as

$$E_{\rm m} \equiv \frac{1}{M} \sum_{i=1}^{M} \frac{\|\boldsymbol{B}_i^{\rm Ref} - \boldsymbol{B}_i^{\rm Rec}\|_2}{\|\boldsymbol{B}_i^{\rm Ref}\|_2}.$$

Unlike the first two metrics $E_{\rm n}$ and $E_{\rm m}$, the smaller value achieved implies the better performance. Therefore, $E_{\rm n}' = 1 - E_{\rm n}$ and $E_{\rm m}' = 1 - E_{\rm m}$ are usually adopted as the metric.

Table 1
Comparison with the Other Methods and Performance Boost with Different Metrics on Case 1

Model	$C_{\rm vec}$	$C_{\rm cs}$	$E_{\rm n}'$	$E_{\rm m}'$	$\mathcal{E}$	$\mathcal{E}_P$
Low	1	1	1	1	1	1.29
Ours	1.00	1.00	0.98	0.98	0.99	1.28
Wiegelmann	1.00	1.00	0.98	0.98	1.02	1.31
McTiernan	1.00	0.99	0.92	0.87	1.00	1.30
Valori	0.99	0.68	0.71	0.33	0.98	1.21
Wheatland	0.98	0.83	0.64	0.42	0.90	1.17
Regnier	0.93	0.49	0.41	0.09	0.80	1.04
LFFF	0.88	0.90	0.50	0.42	0.77	1.00
PF	0.85	0.82	0.45	0.35	0.77	1

Table 2
Comparison with the Other Methods and Performance Boost with Different Metrics on Case 2

Model	$C_{\rm vec}$	$C_{\rm cs}$	$E_{\rm n}'$	$E_{\rm m}'$	$\mathcal{E}$	$\mathcal{E}_{\rm P}$
Low	1	1	1	1	1	1.10
Ours	1.00	0.43	0.79	0.01	0.99	1.09
Wiegelmann	1.00	0.57	0.86	−0.25	1.04	1.14
McTiernan	1.00	0.51	0.84	−0.38	1.04	1.14
Valori	0.99	0.55	0.75	−0.15	1.02	1.12
Wheatland	0.99	0.58	0.69	0.13	0.96	1.05
Regnier	0.94	0.28	0.49	−1.7	0.74	0.82
LFFF	0.93	0.08	−0.80	−37	1.04	1.15
PF	0.92	0.35	0.47	−0.63	0.91	1

The $\mathcal{E}$ is defined as

$$\mathcal{E} \equiv \frac{\sum_{i=1}^{M} \|\boldsymbol{B}_i^{\rm Rec}\|_2^2}{\sum_{i=1}^{M} \|\boldsymbol{B}_i^{\rm Ref}\|_2^2}.$$

$\mathcal{E}$ shows how well the reconstructable energy is contained in the reference field. The closer the value comes to 1, the better the performance is.

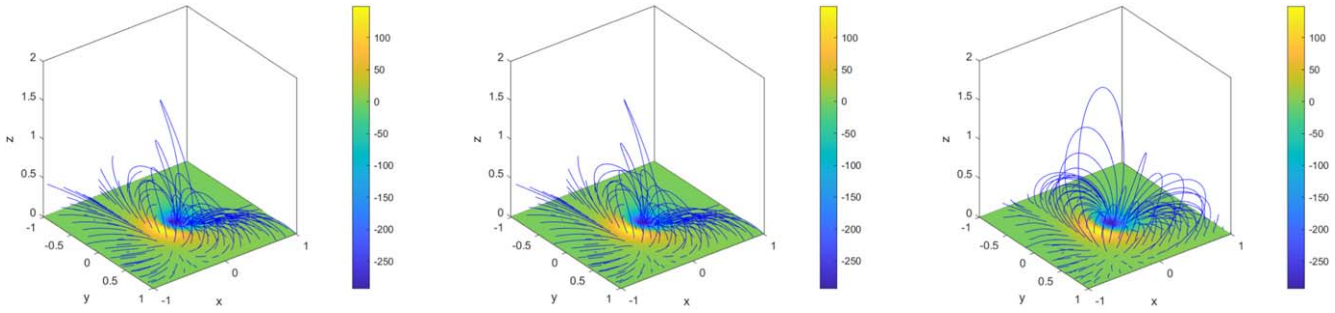
The $\mathcal{E}_{\rm P}$ is defined as

$$\mathcal{E}_{\rm P} \equiv \frac{\sum_{i=1}^{M} \|\boldsymbol{B}_i^{\rm Rec}\|_2^2}{\sum_{i=1}^{M} \|\boldsymbol{B}_i^{\rm P}\|_2^2}.$$

where $\boldsymbol{B}_i^{\rm P}$ is the PF. The larger the value achieved, the more free energy there is.

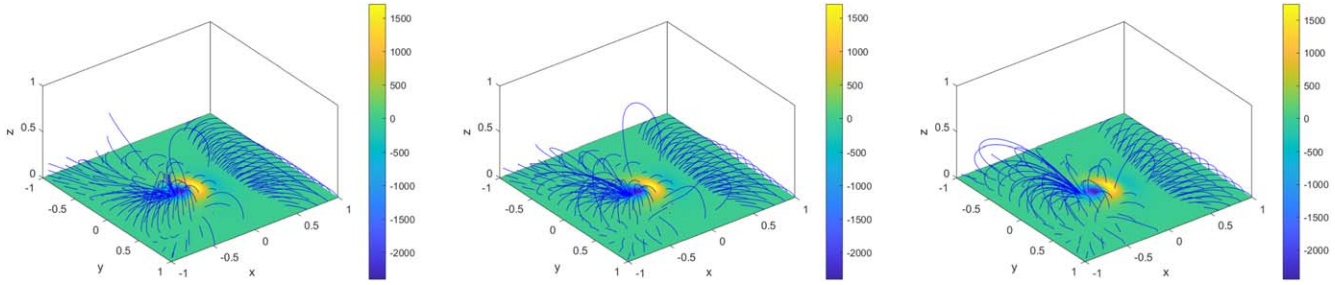
We compare our force-free neural network model with Low & Lou (1990), Wiegelmann (2004), optimization method implemented by McTiernan; Valori et al. (2005), Wheatland (2004), Régnier et al. (2002), LFFF and PF. Tables 1 and 2 show evaluation metric results based on the test cases with $64 \times 64 \times 64$ uniform grids. Note that our proposed model consistently achieves the best performance.

It is well known that magnetic field lines are a visual tool used to represent the magnetic field. A magnetic field line is defined such that the tangent at any point is in the direction of the field $\boldsymbol{B}$ (Priest 2014). Figures 3 and 4 illustrate the



(a) Reference field generated by Low & Lou (b) Reconstructed field by the force-free neural network (c) Reconstructed field by the PF model

Figure 3. The magnetic field lines all start at $z = 0$ and have starting x, y positions ranging from -1 to 1 and increments by 0.125 for Case 1.



(a) Reference field generated by Low & Lou (b) Reconstructed field by the force-free neural network (c) Reconstructed field by the PF Model

Figure 4. The magnetic field lines all start at $z = 0$ and have starting x, y positions ranging from -1 to 1 and increments by 0.125 for Case 2.

visualization of the three-dimensional vector magnetic field with magnetic field lines. These magnetic field lines all start at $z = 0$ and have initial x, y positions ranging from -1 to 1 with increments of 0.125.

Figures 3(a) and 4(a) illustrate the field lines of the exact solution and the corresponding field lines resulting from our reconstruction procedure in Figures 3(b) and 4(b). The reconstructed solution and the reference solution agree in most of the space $\Omega \cup \partial\Omega$.

It is of utmost importance to maintain the integrity of the lateral and top boundary conditions. Neglecting these conditions would significantly impair the performance of field reconstruction, as clearly demonstrated in Figures 4(a) and 5.

Furthermore, it is important to highlight that the force-free neural network method distinguishes itself from other approaches, including the upward integration method, MHD relaxation method, optimization approach, and boundary element method, as discussed in a comprehensive review article Wiegelmann & Sakurai (2021). Unlike these methods, which typically employ fixed boundary conditions throughout the computation, our approach emphasizes learning the magnetic field distribution along the entire boundary using partial boundary data. During the training process, the boundary data continually adapt until they become consistent with the sampled data on the boundary, ensuring accuracy and coherence.

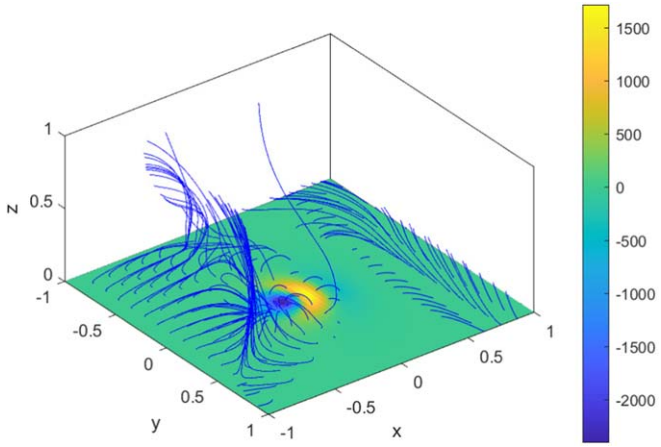
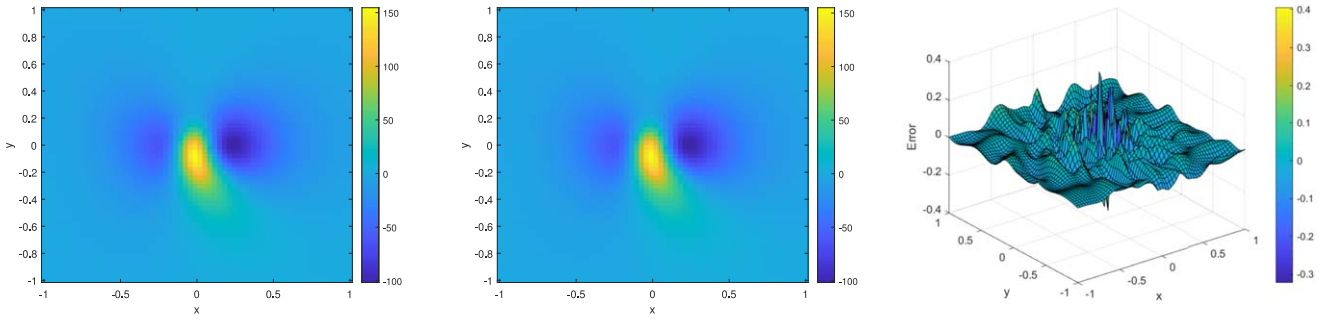
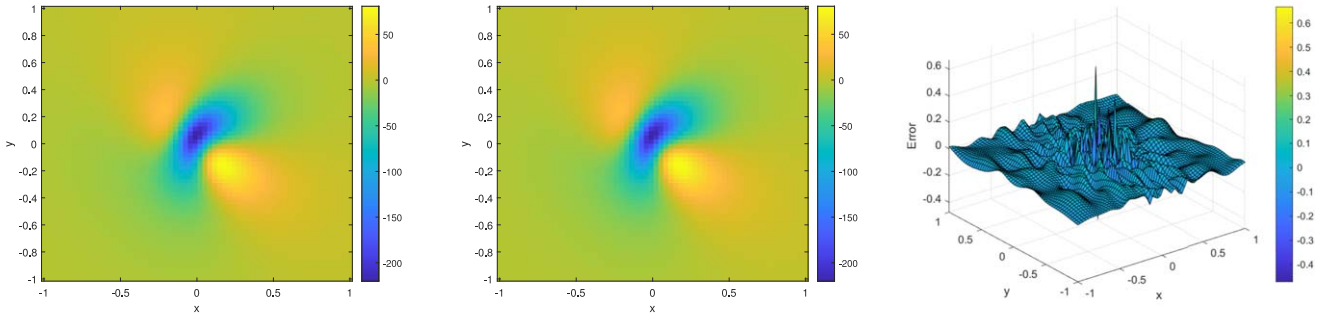


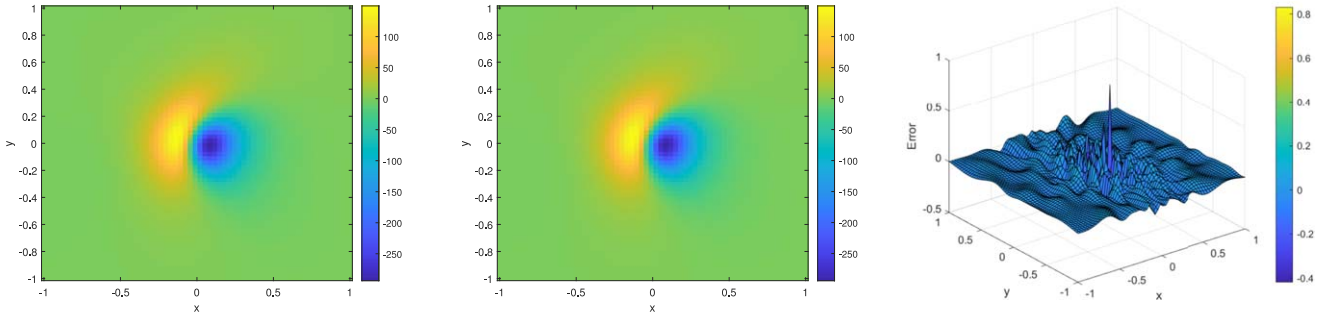
Figure 5. In Case 2, without the lateral and top boundary conditions, the magnetic field lines originate from $z = 0$ and have initial x, y positions ranging from -1 to 1 with increments of 0.125.



(a) B_x^{Ref} (b) B_x^{Rec} (c) Difference in B_x



(d) B_y^{Ref} (e) B_y^{Rec} (f) Difference in B_y



(g) B_z^{Ref} (h) B_z^{Rec} (i) Difference in B_z

Figure 6. The reference and reconstructed magnetograms, as well as the differences between them, are shown for Case 1.

Therefore, the reference magnetograms and the reconstructed magnetograms are depicted in the left two columns of Figures 6 and 7, while the differences between the reference and reconstructed magnetograms are displayed in the right column of the same figures. A uniform grid is generated with $x_i, y_i = -1 + \frac{2}{63}(i-1)$ $(1 \leqslant i \leqslant 64)$ in the $z = 0$ plane. The reconstructed contour plot abides by the reference contour plot in the great majority of the bottom boundary layer.

The above results demonstrate that our method successfully reconstructs the NLFFFs and achieves good agreement with the reference fields in most of the space $\Omega \cup \partial\Omega$. In addition, we calculated the divergence-free and force-free factors from the lower boundary layer to different heights using the following formulas

$$\frac{1}{I}\sum_{i=1}^{I}\frac{|\nabla \cdot \boldsymbol{B}_i|}{\|\boldsymbol{B}_i\|_2}, \qquad \frac{1}{I}\sum_{i=1}^{I}\frac{\|\nabla \times \boldsymbol{B}_i \times \boldsymbol{B}_i\|_2}{\|\nabla \times \boldsymbol{B}_i\|_2 \|\boldsymbol{B}_i\|_2}, \tag{11}$$

where I is the number of magnetic field samples from the photosphere to different heights. It should be noted that when

$$\frac{1}{I}\sum_{i=1}^{I}\frac{\|\nabla \times \boldsymbol{B}_i \times \boldsymbol{B}_i\|_2}{\|\nabla \times \boldsymbol{B}_i\|_2 \|\boldsymbol{B}_i\|_2} \leqslant 0.5,$$

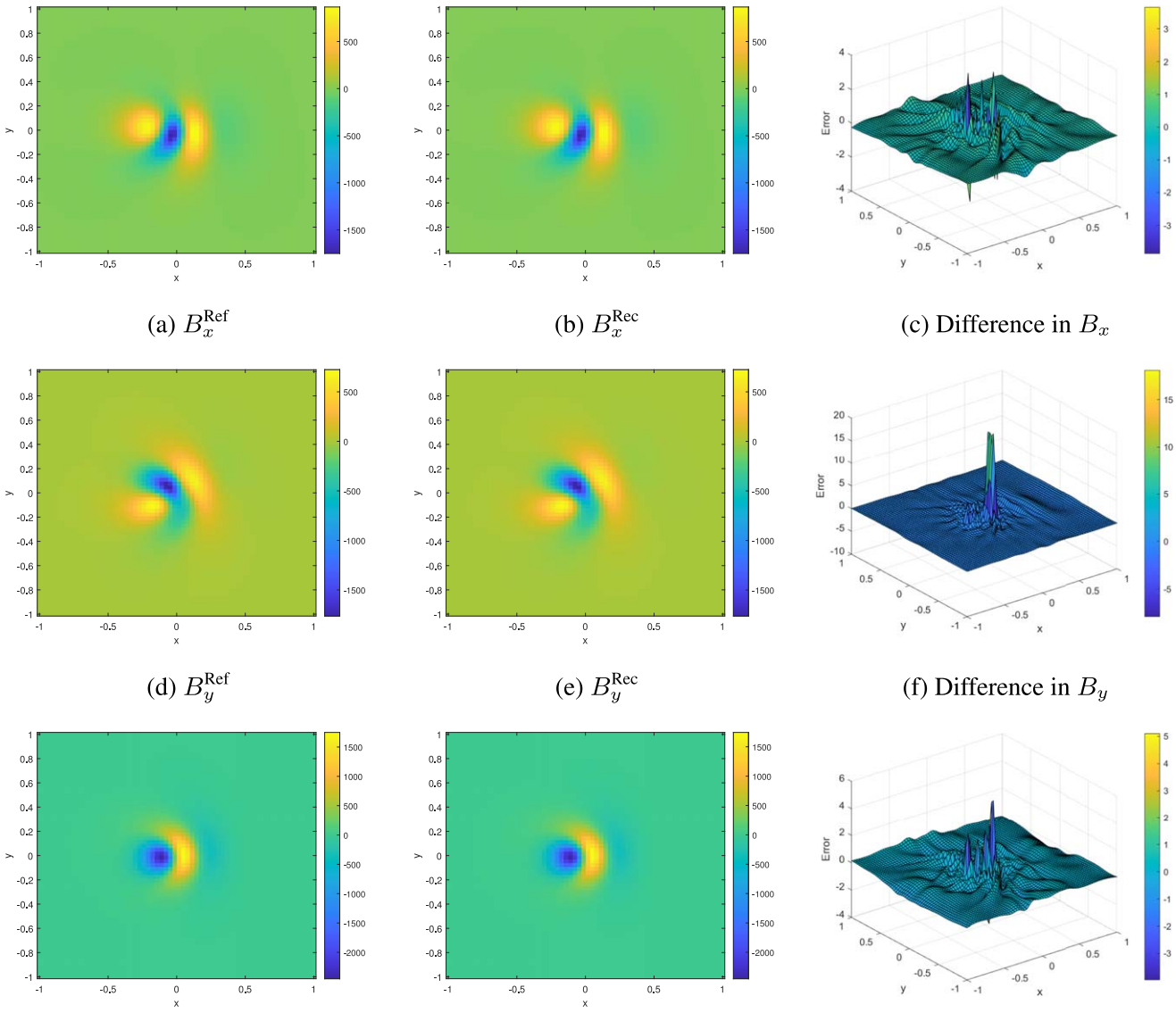
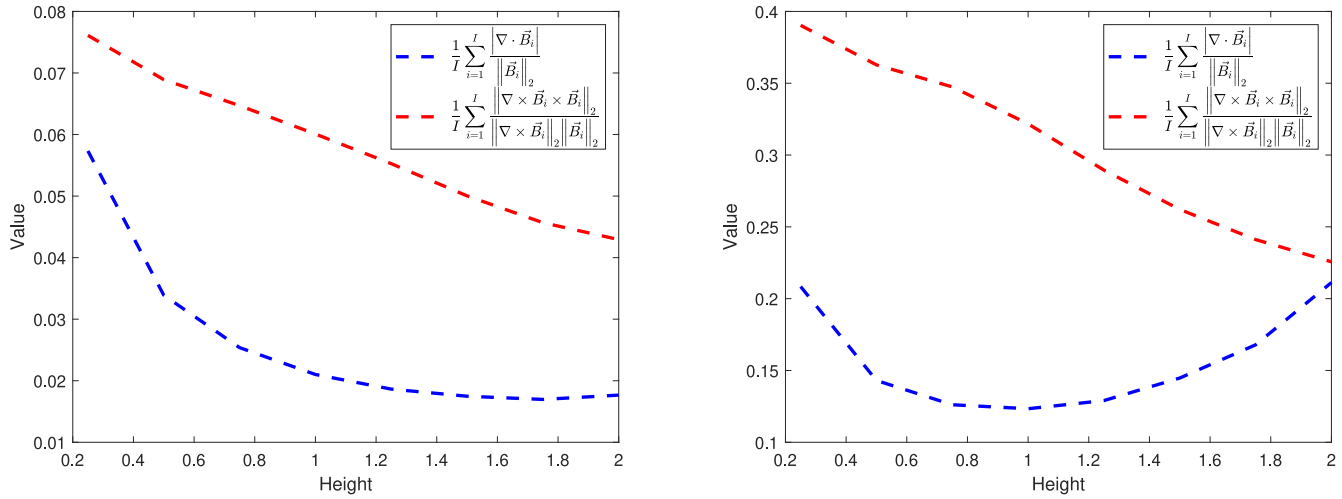


Figure 7. The reference and reconstructed magnetograms, as well as the differences between them, are shown for Case 2.

the average angle between the current density and the magnetic field is less than 30° or greater than 150°. We calculate divergence-free and force-free factors at different heights using Equation (11) for Cases 1 and 2. The above two factors are shown in Figure 8, indicating a general decline with increasing altitude, except for the divergence-free factor in Case 2. Moreover, Case 1 outperforms Case 2 significantly, primarily because all six boundaries of the box are provided for Case 1, whereas only the bottom boundary is specified for Case 2. These findings align with the physical principles of the force-free field, thereby affirming the effectiveness and reliability of our method.

Finally, we provide a test case of the observational active region (AR). The magnetic field region to be calculated, $\bar{\Omega}$, is defined as $\bar{\Omega} = [-\max(x_i), \max(x_i)] \times [-\max(y_j), \max(y_j)] \times [0, \max(z_k)]$. To ensure $\bar{\Omega} \subseteq [-1, 1] \times [-1, 1] \times [0, 2]$, we define $x_0 = \frac{x}{l_0}$, $y_0 = \frac{y}{l_0}$, $z_0 = \frac{z}{l_0}$, with $\nabla_0 = \left[\frac{\partial}{\partial x_0}, \frac{\partial}{\partial y_0}, \frac{\partial}{\partial z_0}\right]^T$, where $l_0 = \max(\max(x_i), \max(y_j), \max(z_k))$. Therefore, for observational



(a) For case 1. (b) For case 2.

Figure 8. Divergence-free and force-free factors at different heights for Cases 1 and 2.

data of the AR, the loss function $f(\boldsymbol{p})$ of the force-free neural network can be equivalently expressed as

$$
\begin{aligned}
f(\boldsymbol{p}) \;=\; & \lambda_0 \frac{1}{m}\sum_{i=1}^{m} \|\boldsymbol{B}^{NN}(x_{0_i}^{'}, y_{0_i}^{'}, z_{0_i}^{'}; \boldsymbol{p}) - \boldsymbol{B}(x_{0_i}^{'}, y_{0_i}^{'}, z_{0_i}^{'})\|_2^2 \\
& + \lambda_1 \frac{1}{n}\sum_{j=1}^{n} \left\| \frac{1}{l_0}\nabla_0 \cdot \boldsymbol{B}^{NN}(x_{0_j}, y_{0_j}, z_{0_j}; \boldsymbol{p})\right\|_2^2 \\
+ \lambda_2 \frac{1}{n}\sum_{j=1}^{n} & \frac{\|\nabla_0 \times \boldsymbol{B}^{NN}(x_{0_j}, y_{0_j}, z_{0_j}; \boldsymbol{p}) \times \boldsymbol{B}^{NN}(x_{0_j}, y_{0_j}, z_{0_j}; \boldsymbol{p})\|_2^2}{\|\nabla_0 \times \boldsymbol{B}^{NN}(x_{0_j}, y_{0_j}, z_{0_j}; \boldsymbol{p})\|_2^2 \cdot \|\boldsymbol{B}^{NN}(x_{0_j}, y_{0_j}, z_{0_j}; \boldsymbol{p})\|_2^2},
\end{aligned}
$$

where λ_0, λ_1, and λ_2 are the weight coefficients in the neural network's loss function.

For NOAA AR 11158 on 2011 February 14, the photospheric magnetic field measurements are used as the boundary condition for the bottom surface, while the extrapolation results of the PF model are used as boundary conditions for the other five surfaces. We employ a neural network with 9 layers ($L = 7$), each containing 200 neurons. The model's hyperparameters are set as $\lambda_0 = 1$, $\lambda_1 = 1$, and $\lambda_2 = 3$, with m set to 100. We uniformly select 46,683 (247×189) points on the bottom boundary $\partial\Omega$ and randomly select 360,000 points within the space Ω for training the neural network. In addition, 23,499 points are sampled on the other five surfaces as part of the boundary conditions.

Figure 9(b) displays the projection of the local magnetic field lines of our proposed method on the xoy plane. It is very consistent with the observation in Figure 9(a) in terms of the layout of the magnetic field, the distribution and contours of the magnetic field lines, and the positions of the magnetic loop's footpoints. Figure 9(a) shows the 193 Å imaging of NOAA AR 11158 from the Atmospheric Imaging Assembly (AIA) Active Region Patches data set (Dissauer et al. 2023), covering a larger area than the corresponding SHARPs (Bobra et al. 2014). Figure 9(c) features the projection result from the NLFFF extrapolation data set from Kusano et al. (2021), using the MHD relaxation method, while Figure 9(d) presents the projection result from the PF model extrapolation method. They both fail to match the observations in Figure 9(a) closely in terms of magnetic field configuration and contours of magnetic field lines.

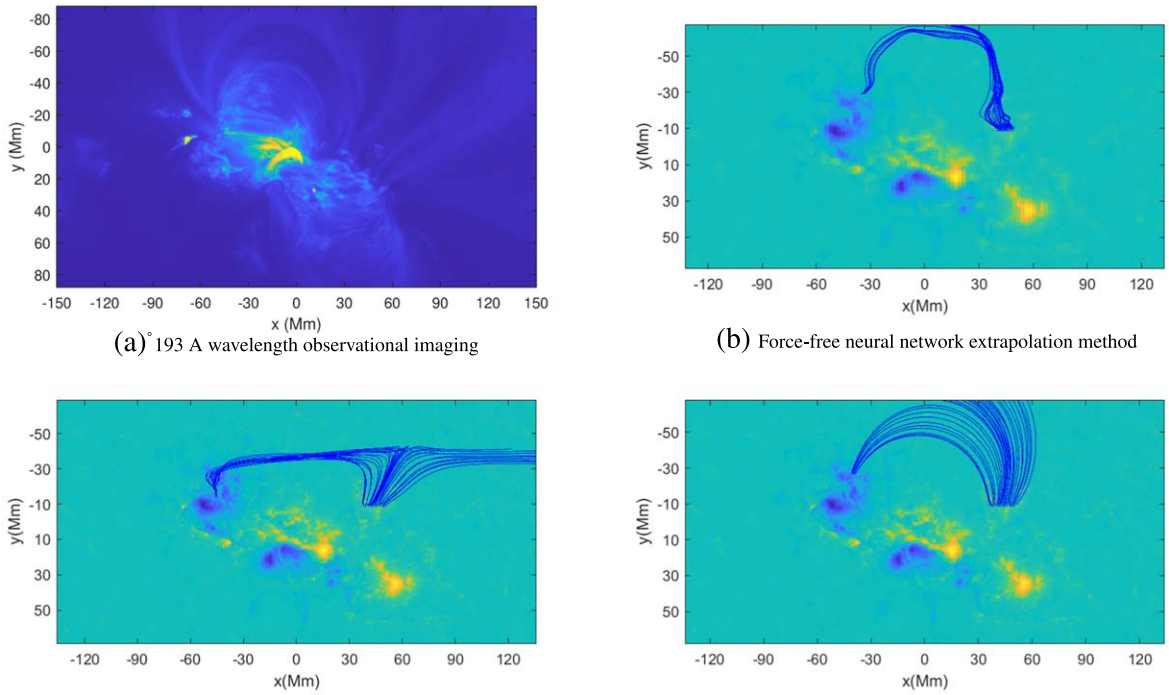
In addition, the divergence-free and force-free factors from the photosphere to various heights are computed using Equation (11) for the extrapolation result obtained by our method. These factors are plotted in Figure 10 as a curve that decreases with increasing height. This trend aligns with the physical principles governing the coronal magnetic field, demonstrating the effectiveness and reliability of our method.

It is worth noting that the implementation of these cases is conducted using MATLAB R2022a on a laptop with an Intel Core i7 2.30 GHz processor and 32 GB of RAM.

5. Conclusion

This paper presents a deep learning approach, the force-free neural network, for exploring the force-free magnetic field. The numerical algorithm is established to solve the proposed model under the optimization framework. The test cases show that the proposed model can outperform most state-of-the-art models.

An interesting direction for future work is to explore methods for accelerating the training of the MLP used in the force-free neural network. In addition, further study on the adaptive selection of trade-off parameters could improve the efficiency of the implementation.



(a) 193 A wavelength observational imaging

(b) Force-free neural network extrapolation method

(c) MHD relaxation method

(d) PF model extrapolation method

Figure 9. Comparison of local magnetic field line projections on the xoy plane with observational imaging data. It is important to note that Figure 9(a) is directly rendered using MATLAB, without utilizing the AIA color table. The origin of the images is at the top left corner rather than at the bottom left corner, with the center coordinates at (0, 0).

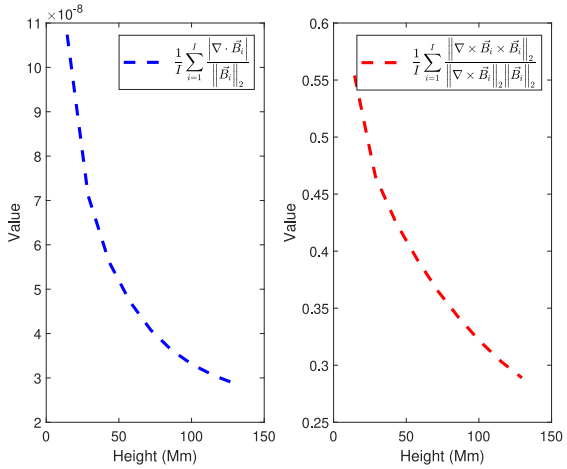


Figure 10. Divergence-free and force-free factors at different heights for NOAA AR 11158.

In practical applications, it is often the case that only the bottom boundary condition (photosphere layer of the Sun) is available, while lateral and top boundary conditions are unavailable in the NLFFF problem described by Equation (6). In future research, we will investigate approaches to handle the incomplete boundary condition problem under certain physical assumptions.

Acknowledgments

This work was supported by the National Key R&D Program of China (Nos. 2021YFA1600504, 2022YFE0133700, 2022 YFF0503900), and the National Natural Science Foundation of China (NSFC, Grant Nos. 11790305 and 11973058). Prof. Xiaoshuai Zhu of the National Space Science Center of the Chinese Academy of Sciences is acknowledged for discussions. We also thank the anonymous reviewer and the scientific editor for their valuable and constructive feedback, which have significantly improved this paper.

References

Amari, T., Aly, J., Luciani, J., Boulmezaoud, T., & Mikic, Z. 1997, SoPh, 17, 129
Baydin, A., Pearlmutter, B., Radul, A., & Siskind, J. 2018, JMLR, 18, 1
Bobra, M., Sun, X., Hoeksema, J., et al. 2014, SoPh, 289, 3549
Cybenko, G. 1989, MCSS, 2, 303
Dissauer, K., Leka, K., & Wagner, E. 2023, ApJ, 942, 1
Grad, H., & Rubin, H. 1958, in Proc. Second United Nations Int. Conf. on the Peaceful Uses of Atomic Energy, 31, Theoretical and Experimental Aspects of Controlled Nuclear Fusion, 190
Guo, Y., Xia, C., Keppens, R., & Valori, G. 2016, ApJ, 828, 82

He, K., Zhang, X., Ren, S., & Sun, J. 2015, in 2015 IEEE Int. Conf. on Computer Vision (Piscataway, NJ: IEEE), 1026
Hornik, K. 1991, NN, 4, 251
Jarolim, R., Thalmann, J., Veronig, A., & Podladchikova, T. 2023, NatAs, 7, 1171
Jiang, C., Feng, X., Wu, S., & Hu, Q. 2013, ApJL, 771, L30
Kusano, K., Iijima, H., Kaneko, T., et al. 2021, ISEE Database for Nonlinear Force-Free Field of Solar Active Regions, doi:10.34515/DATA.HSC-00000
Liu, D., & Nocedal, J. 1989, MatPr, 45, 503
Low, B., & Lou, Y. 1990, ApJ, 352, 343
Mikić, Z., & McClymont, A. 1994, in ASP Conf. Ser. 68, Solar Active Region Evolution: Comparing Models with Observations, ed. K. S. Balasubramaniam & G. W. Simon (San Francisco, CA: ASP), 225
Nakagawa, Y. 1974, ApJ, 190, 437
Neukirch, T. 2005, in Proc. GLOBECOM 15, Proc. Int. Scientific Conf. on Chromospheric and Coronal Magnetic Fields (ESA SP-596)
Nocedal, J., & Wright, S. 2006, Numerical Optimization (New York: Springer)
Priest, E. 2014, Magnetohydrodynamics of the Sun (Cambridge: Cambridge Univ. Press)
Raissi, M., Perdikaris, P., & Karniadakis, G. 2019, JCP, 378, 686
Régnier, S., Amari, T., & Kersalé, E. 2002, A&A, 392, 1119
Schrijver, C., Derosa, M., Metcalf, T., et al. 2006, SoPh, 235, 161
Valori, G., Kliem, B., & Fuhrmann, M. 2007, SoPh, 245, 263
Valori, G., Kliem, B., & Keppens, R. 2005, A&A, 433, 335
Watt, J., Borhani, R., & Katsaggelos, A. 2020, Machine Learning Refined: Foundations, Algorithms, and Applications (Cambridge: Cambridge Univ. Press)
Wheatland, M. 2004, SoPh, 222, 247
Wheatland, M., Sturrock, P., & Roumeliotis, G. 2000, ApJ, 540, 1150
Wiegelmann, T. 2004, SoPh, 219, 87
Wiegelmann, T. 2008, JGRA, 113, 1
Wiegelmann, T., Inhester, B., & Sakurai, T. 2006, SoPh, 233, 215
Wiegelmann, T., & Sakurai, T. 2021, LRSP, 18, 1
Yan, Y., & Li, Z. 2006, ApJ, 638, 1162
Yan, Y., & Sakurai, T. 2000, SoPh, 195, 89
Yang, W., Sturrock, P., & Antiochos, S. 1986, ApJ, 309, 383
Zhou, D. 2020, ACHA, 48, 787