Data-driven Discovery of Partial Differential Equations for Multiple-Physics Electromagnetic Problem

Bing Xiong, Haiyang Fu, Feng Xu, Yaqiu Jin

Abstract—Deriving governing equations in Electromagnetic (EM) environment based on first principles can be quite tough when there are some unknown sources of noise and other uncertainties in the system. For nonlinear multiple-physics electromagnetic systems, deep learning to solve these problems can achieve high efficiency and accuracy. In this paper, we propose a deep learning neutral network in combination with sparse regression to solve the hidden governing equations in multiple-physics EM problem. Pareto analysis is also adopted to preserve inversion as precise and simple as possible. This proposed network architecture can discover a set of governing partial differential equations (PDEs) based on few temporal-spatial samples. The data-driven discovery method for partial differential equations (PDEs) in electromagnetic field may also contribute to solve more sophisticated problem which may not be solved by first principles.

Index Terms—deep learning, convolutional neural network, partial differential equations, multiple electromagnetic physics;

I. INTRODUCTION

With comprehensive application of electromagnetic wave spectrum in radar, communication and navigation and increased computing power and storage, the amount of electromagnetic data comes to a huge number. Less well understood is how to distill the underlying electromagnetic physical laws from electromagnetic data in complex systems. The big electromagnetic data provides new opportunities for the data-driven discovery of new physics laws or making complex system modelling computationally feasible. Traditional derivation of governing equations relies on fundamental laws and theoretical solution can be obtained with analytic and computational methods. However, the realistic scene in application is complex to tackle, which involves multiple-physics, multiple-scale and nonlinear. For a multiple-physics system with complex interaction mechanisms, there is no exact quantitative analytic solution and computational cost is high to solve a set of partial differential equations (PDEs). Therefore, the key question is how to use sparse data given to discover the principle of a complex system to make predictions of physical quantities outside the range of the measurement data. If the model can be learned from sparse data and then perform prediction, it is significant for complex electromagnetic multiple-physics system and beyond.

The data-driven discovery methods have been developing quickly during the past decades. It can be mainly divided into three categories, including symbolic regression, sparse regression and deep learning. One of earlier research on data-driven discovery for free-form natural laws is proposed by Bongard and Lipson (2007) and Schmidt and Lipson (2009) [1]. The main idea is to calculate numerical differentiation of experimental data first and use symbolic regression based on evolutionary algorithm to compare with analytical derivative solutions. Later, sparse regression method was raised by Brunton et al. (2016) [3], Schaeffer (2017) [4], and Rudy et al. (2017) [5]. A candidate function dictionary needs to be established which consists of simple and derivatives terms and then use sparse regression to select the proper terms as part of the equations. Recently, Raissi and Karniadakis (2017) used full connection deep neural network to discover underlying physics of nonlinear PDEs [6], [7] with less data. The key idea is to set up a universal neural network to approximate the solution of the PDE by minimizing the loss function and the derivatives can then be calculated by automation differentiation based on neural network [7]. However, the explicit form of the PDEs is assumed to be known. Recently, Long et al. (2018) [7] utilized the connection between differentiation and convolution to discover nonlinear equations with minor knowledge on the equation form. The wavelet frame filters in convolutional neural network (CNN) with a training kernel is adopted to approximate spatial differentiations. Long et al. (2018) [8] upgraded their network by imposing appropriate constraints on filters and using a newly designed symbolic neural network to express the analytical form of function clearly.

In sum, the data-driven discovery method of the underlying physics of PDEs are in progress. However, they still have some issues, such as symbolic regression with high computation cost and difficulty for dealing with large scale problem. Sparse regression needs to set numerical differentiations beforehand which costs large storage and may generate unrelated terms. The PDE-net for discovering unknown equations by Long et al. (2018) [6], [7] requires no knowledge on the differential operators and associated discrete approximations, which has been applied into 2-dimensional linear variable-coefficient convection-diffusion equation.

However, it is still challenging for the data-driven method in order to solve multiple-physics electromagnetic physics problem. There are two main characteristics for data-driven methods for multiple physics electromagnetic application. First, the electromagnetic scatter field is usually varying fast with time. The characteristic of the EM field needs to be extracted in the time series of data. Secondly, multiple-physics electromagnetic problems involve a set of coupled different partial differential equations, which may inherit time-varying
in heterogeneous coefficients. It requires one unified algorithm to retrieve multiple coefficients of the PDE equation set simultaneously. Therefore, the data-driven discovery method for complex electromagnetic system will be specially designed as a framework.

This paper aims to design a data-driven network architecture to discover a set of nonlinear PDE equations for multiphysics EM problem. A unified neural network with CNN is proposed in combination with sparse regression and pareto analysis. This proposed network is applied for a electromagnetic wave and plasma interaction system, which can discover the coefficient of the PDE set with relative good accuracy. Also, we investigate the possibility of inferring inhomogeneous coefficients with little prior knowledge. The data-driven methods of the PDEs set will pay the way for deriving heterogeneous coefficients. It requires one unified algorithm to retrieve multiple coefficients of the PDE equation set simultaneously. Therefore, the data-driven discovery method for complex electromagnetic system will be specially designed as a framework.

II. Methodology

For simplicity, we assume there are two physical variable \(u(x,y,t), v(x,y,t)\) in a 2-dimensional space \((x,y)\) varying with time \(t\). The universal expression of their governing equations can be expressed as following

\[
\begin{align*}
\frac{\partial u}{\partial t} &= f(x,y,u,v, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial y^2}, \ldots) \\
\frac{\partial v}{\partial t} &= g(x,y,u,v, \frac{\partial u}{\partial x}, \frac{\partial v}{\partial y}, \frac{\partial u}{\partial y}, \frac{\partial v}{\partial x}, \frac{\partial^2 v}{\partial x^2}, \frac{\partial^2 v}{\partial y^2}, \ldots)
\end{align*}
\]

(1)

where \(f\) and \(g\) are nonlinear function of all possible terms including \(u\) and \(v\), their first and second order derivatives and other parameters, respectively.

A. Time derivative

We define \(u(t_i, \cdot)\) or \(v(t_i, \cdot)\) as all spatial value of function \(u\) or \(v\) at \(t = t_i\). For time derivative, according to the forward euler method, we obtain the following formula

\[
\begin{align*}
\hat{u}(t_{i+1}, \cdot) &= u(t_i, \cdot) + \Delta t \cdot \frac{\partial u}{\partial t} \\
\hat{v}(t_{i+1}, \cdot) &= v(t_i, \cdot) + \Delta t \cdot \frac{\partial v}{\partial t}
\end{align*}
\]

(2)

where \(\hat{u}(t_{i+1}, \cdot)\) or \(\hat{v}(t_{i+1}, \cdot)\) is the approximation value at \(t_{i+1}\). Then, we obtain expression

\[
\begin{align*}
\hat{u}(t_{i+1}, \cdot) &= u(t_i, \cdot) + \Delta t \cdot f \\
\hat{v}(t_{i+1}, \cdot) &= v(t_i, \cdot) + \Delta t \cdot g
\end{align*}
\]

(3)

B. Spatial derivative

For spatial derivative calculation, the connection between convolutions and differentiations was studies by Cai et al. (2012) [10] and by Dong et al. (2017) [8]. Here, we demonstrate one simple example of their work [8] [10] to show how to express differentiation by convolution. Consider that the 2-dimensional Haar wavelet filter banks containing one low-pass filter \(h_{00}\) and three high-pass filters \(h_{10}, h_{01}\) and \(h_{11}\) have forms:

\[
\begin{align*}
h_{00} &= \frac{1}{4} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix},
h_{10} = \frac{1}{4} \begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix},
h_{01} = \frac{1}{4} \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix},
h_{11} = \frac{1}{4} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}.
\end{align*}
\]

(4)

Then, the circular convolution (labeled as \(\otimes\)) between filters and function \(u\) for instance is expressed as:

\[
\begin{align*}
h_{00} \otimes u &\approx u, \\
h_{10} \otimes u &\approx \frac{1}{2} \delta_x \frac{\partial u}{\partial x}, \\
h_{01} \otimes u &\approx \frac{1}{2} \delta_y \frac{\partial u}{\partial y}, \\
h_{11} \otimes u &\approx \frac{1}{4} \delta_x \delta_y \frac{\partial^2 u}{\partial x \partial y}.
\end{align*}
\]

where \(\delta_x\) and \(\delta_y\) are the spatial size of \(x\) and \(y\) direction of function \(u\) or \(v\), respectively. Similarly for \(v\),

\[
\begin{align*}
h_{00} \otimes v &\approx v, \\
h_{10} \otimes v &\approx \frac{1}{2} \delta_x \frac{\partial v}{\partial x}, \\
h_{01} \otimes v &\approx \frac{1}{2} \delta_y \frac{\partial v}{\partial y}, \\
h_{11} \otimes v &\approx \frac{1}{4} \delta_x \delta_y \frac{\partial^2 v}{\partial x \partial y}.
\end{align*}
\]

By multiplying constant coefficients, convolution can represent differentiation in neural network effectively. The calculation of the second order differentiation is also mentioned in [7], it is ignored since our current work does not involve it as will be discussed in future.

C. \(\Delta t\) block

Therefore, we obtain a partial neural network structure from time derivative calculation based on forward Euler’s method. For simplicity, the partial neural network architecture will be shown only for variable \(u\) and target function \(f\), similarly for variable \(v\) and target function \(g\).

Figure 1 illustrates a \(\Delta t\) block [8] as a layer of neural network to advance variables based on equations in time. In principle, the structure of the \(\Delta t\) block is an interpretation of Equation (3). The two left terms in Fig.1 represent \(\Delta t \cdot f\), that is a candidate terms database in the form of neural network. Thus, the input of this network is clear, that’s all possible functions with \(u\) and \(v\) in this example. Here, it is noted that for variable \(v\) and its derivatives are also included for advancing \(u\) in the candidate database because we assume that there is coupling between \(u\) and \(v\). The prior knowledge will help reduce the number of possible candidate functions associated.
Fig. 1: The architecture of a $\Delta t$ block [6] to advance function in time. The two left parts represent the target function $f$ in Equation (3), that includes a candidate term database in the form of neural network as in Equation (5-6). The input of this network is all possibly related functions which are $u$ and $v$ in this example. It is noted that function $v$ and its derivatives are also included in the candidate database because we assume that there exist coupling between $u$ and $v$ in PDEs.

with PDEs. The output of $\Delta t$ block is the predicted value at the next time stamp $\tilde{u}(t_{i+1}, \cdot)$.

In general, the whole network consists of several $\Delta t$ blocks by continuously connecting each block one by one. The number of $\Delta t$ blocks is determined by different convergence criteria.

In Fig. 1, the coefficients $w_1, w_2, w_3...$ are neural network weights for each candidate term, respectively. The network weights will be shared in each layer and become the output vector of this model. The non-zero weight represents the term of the target equation, where $n$ means the number of candidate terms. We define it $W$ as

$$W = [w_1, w_2, w_3, w_4, w_5, ..., w_n], \quad (7)$$

D. Sparse regression

Sparse regression is a common method in data-discovery area. When there are many terms to discover with spare data, it means that the system may face with over-fitting problems. Regularization method need to be adopted in training process and $L_1$ norm regularization is commonly used as $L_1$ is easier to calculate than $L_0$ norm regularization.

In our approach, when the training of network is finished, the output $W$ will be limited by regularization method in the final layer of network.

The output of $\Delta t$ block is the predicted value at the next time step $\tilde{u}(t_{i+1}, \cdot)$. Therefore, the loss function of a single block is defined as

$$l_i = \| \tilde{u}(t_{i+1}, \cdot) - u(t_{i+1}, \cdot) \|_2 \quad (8)$$

Consider sparse regression to optimize the output $W$ at the end of the complete network, we define that loss function of the entire network is sum of each block’s loss according to $L_1$ norm of $W$. The accumulated loss function can make prediction reliable within certain time step numbers $N$. The expression for final loss $L$ is

$$L = \sum_{i=1}^{N} \| \tilde{u}(t_i, \cdot) - u(t_i, \cdot) \|_2 + \lambda \| W \|_1 \quad (9)$$

where $\lambda$ is the regularization parameter to determine the weight of regularization term.

E. Pareto analysis

Pareto analysis is a technique that can be useful in situations where many possible action processes are competing for attention. In essence, the problem-solver evaluates the benefits of each action and selects some of the most effective actions that bring the total benefits reasonably close to the maximum possible benefits. While it is common to refer to pareto as the "80/20" rule, in all cases, 20% of the reasons for 80% of the problems are assumed to be a convenient rule of thumb, not and should not be considered an immutable law of nature. This technique helps identify the top priority part of the problem that needs to be addressed.

The application of Pareto analysis adopted in [6] is the final guarantee of the discovery result of the system. By cutting the term with the minimum parameter out and training again, we compare the later loss with the previous one. If loss reduces, then we repeat the process until loss rises.

Figure 2 illustrates the architecture of the whole neural network to deal with a given PDE equation set. This process includes
Fig. 2: The architecture of discovering the PDE equation set. The procedure steps include 1) Check the number of PDEs equations. 2) Determine the time derivative of active terms in the PDE. 3) Activate multiple equation network corresponding to each equations. 4) Train network with sparse regression. 5) Pareto analysis helps to cut out unrelated terms with minimum parameters.

1) Find out the number of possible equations to discover, e.g., $f$ or $g$.
2) Determine time derivatives, e.g., $\dot{u}(t_{i+1}, \cdot)$ or $\dot{v}(t_{i+1}, \cdot)$.
3) Decide the number of networks to activate.
4) Utilize spare regression to train network weights $W$.
5) Cut out terms based on Pareto analysis.
6) Iterate and train again if not converged.

III. SIMULATION

A. Forward model

Here, we consider the interaction between electromagnetic waves and magnetized plasmas with collisions. For such multiple physic complex system, the Maxwell’s partial differential equations and constitutive relations are given as

$$\nabla \times \mathbf{H} = \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t} + \mathbf{J}$$

$$\nabla \times \mathbf{E} = -\mu_0 \frac{\partial \mathbf{H}}{\partial t}$$

$$\frac{\partial \mathbf{J}}{\partial t} + \nu_e \mathbf{J} = \varepsilon_0 \omega_p^2 \mathbf{E} + \omega_{ce} \times \mathbf{J}$$

(10)

where $\mathbf{H}$, $\mathbf{E}$ and $\mathbf{J}$ are magnetic intensity vector, electric field intensity vector, and polarized current density vector respectively. $\varepsilon_0$, $\mu_0$ are the vacuum permittivity and magnetic permeability respectively. Here, $\omega_p$, $\omega_{ce}$ and $\nu_e$ are plasma frequency, electron cyclotron frequency and electron collision frequency, respectively. The background magnetic field is $\mathbf{B}_0 = B_y \hat{y} + B_z \hat{z}$. The wave propagates with magnetic inclination angle $\theta$ with respect to the $z$ axis. Equation (3) is only for cold plasmas and the ion motion is neglected.

To verify our method, we calculate the scattering field $\mathbf{E}_s$ for a one-dimensional magnetized plasma slab shown in Fig. 3. The computational space takes up 800 grid along the $z$ directions, among which plasma takes up 200-600 grids with uniform distribution and a thickness of $d = 3$ cm. The JEC-FDTD method is applied to obtain the solution of Equation (9). The algorithm can be found in details [11]. For uniform plasma distribution, parameters in the simulation are set as following in Table I.

**TABLE I: System Parameter**

| Symbol                              | Value  |
|-------------------------------------|--------|
| Electron collision frequency $\nu_e$| 20 GHz |
| Electron cyclotron frequency $\omega_{ce}$ | 30 GHz |
| Uniform plasma frequency $\omega_p$ | 40 GHz |
| Magnetic inclination angle $\theta$ | 90°    |

Fig. 3: Multiple-physics system: EM wave interaction with plasmas with arbitrary magnetic inclination $\theta$. The spatial domain is in the $z$ direction.

For $\theta = \pi/2$, Equations (10) are expressed as a scalar differential equation as:

$$\frac{\partial E_x}{\partial z} = -\mu_0 \frac{\partial H_y}{\partial t}$$

(11)
\[
- \frac{\partial H_y}{\partial z} = \varepsilon_0 \frac{\partial E_x}{\partial t} + J_x \quad (12)
\]
\[
\frac{\partial J_x}{\partial t} + \nu_0 J_x = \varepsilon_0 \omega_p^2 E_x + \omega_{ce} J_z \quad (13)
\]
\[
\frac{\partial J_z}{\partial t} + \nu_0 J_z = \varepsilon_0 \omega_p^2 E_z - \omega_{ce} J_x \quad (14)
\]
\[
J_z = -\varepsilon_0 \frac{\partial E_z}{\partial t} \quad (15)
\]

Here, there are 5 Equations to discover with five unknown parameters \( \mu_0, \varepsilon_0, \varepsilon_0 \omega_p^2, \omega_{ce}, \nu_0 \). This set of PDE equations is our first goal for equation discovery. This set of PDE equations is for EM wave propagation in anisotropic plasmas, however, it can effective reveal the application for our proposed network.

**B. Prepossessing**

The whole process of inverse model is shown in Fig. 4. First of all, the physical input data are \( E_x, J_x, E_z, J_z, H_y \) calculated from JEC-FDTD algorithm. Then, sampling is conducted for the input data \( E_x, J_x, E_z, J_z, H_y \) for training, respectively.

Next, normalization is executed directly after data sampling. As we use the same input for different equations, we need to do normalization for each equations because the order of magnitudes of input data will cause big error for data training and lead to wrong results. Specifically, we multiply each input data with a designed parameter respectively to keep them into the same order of magnitude. Afterwards, each input data should be compared with time derivative \( \frac{\partial E_x}{\partial t}, \frac{\partial J_x}{\partial t}, \frac{\partial E_z}{\partial t}, \frac{\partial J_z}{\partial t}, \frac{\partial H_y}{\partial t} \), which are included by the target equation to keep in the same magnitude.

For collecting training data we select three physical quantity of the scattered field \( E_x, E_z, H_y \) and two currents \( J_x, J_z \) at spatial grids \( z(\Delta z) = 190 \sim 210 \) including plasma-vacuum boundary (boundary grid 200). For temporal series of physical quantities, we choose few samples at the time period \( t(\Delta t) = 500 \sim 510 \), which includes the maximum value of the scattered field from plasmas.

Figure 5 shows examples for the total electric field \( E_x \) and \( E_z \) in perpendicular to the magnetic field. The sampling scheme is the same for all five quantities. In Figure 5, the two white lines represent the boundary of plasma. The small white square represents the data collection area from both inside and outside the plasma. For time sampling, we choose 10 time steps after time 500, because the gradient of curve at this time is obvious so that the function is easy to be recognized by neural network.

The sampling area for training network is shown in Figure 6. In the upper panel, the black square shows the data sampling area for \( E_x \) and \( E_z \) in time domain. In the bottom panel, the black horizontal line indicates the border of plasma and our collection area begins right on the incident border. It shows that only a small number of time and spatial information are continuously sampled for training.

**C. Network Construction**

From forward Euler’s method, we have
\[
\tilde{E}_x(t_{i+1}, \cdot) = E_x(t_i, \cdot) + \Delta t \cdot \frac{\partial E_x}{\partial t} \quad (16)
\]
\[
\tilde{J}_x(t_{i+1}, \cdot) = J_x(t_i, \cdot) + \Delta t \cdot \frac{\partial J_x}{\partial t} \quad (18)
\]
\[
\tilde{J}_z(t_{i+1}, \cdot) = J_z(t_i, \cdot) + \Delta t \cdot \frac{\partial J_z}{\partial t} \quad (19)
\]
\[
\tilde{E}_z(t_{i+1}, \cdot) = E_z(t_i, \cdot) + \Delta t \cdot \frac{\partial E_z}{\partial t} \quad (20)
\]

For one-dimension problem, the spatial direction is only \( z \), so we use only one filter to get one kind of spatial differentiation. Take \( E_x \) for example:
\[
h_{10} \otimes E_x \approx \frac{1}{2} \delta_z \cdot \frac{\partial E_x}{\partial z} \quad (21)
\]

Based on this, we get the adjusted neural network structure. To fulfill the function of Equation (21), we use a function called

---

**Fig. 4:** The overall process of the inverse model for a multiphysics EM problem. Step 1: Data collected from a solution of PDE Equations (11-15). Step 2: Data sparse sampling and normalization. Step 3: Check the number of equations and determine the time derivative of the input data. Step 4: Multi-Equation network activation and training shown in Fig.1. Step 5: Sparse regression is set at the end of training and the detailed form of equations are shown. Step 6: After Pareto analysis, distortion terms will be deleted.
conv2d in TensorFlow to generate convolution calculation. In our case, since it is one dimensional 1D problem, we obtain the shape of convolution kernel of $2 \times 1$, so that this function still works.

By screening different weights of network, the correct equation term and its corresponding coefficient can be obtained. This is one single time block for Equations (11-15).

As one single block $\delta_t$ given above, we have 10 $\delta_t$ time block to build up the whole network. By using the parameter sharing characteristic of neural network, the training coefficient results become accurate if the number of time blocks is increased. Consider that all redundant terms may lead to the problem of complicated neural network structure, we reduce the number of candidates based on certain prior physical knowledge, thus making the network training more efficient.

**D. Result and Analysis**

For the system of PDEs set, we build a neural network based on sparse temporal-spatial sample data for training and predict the coefficients of PDEs without interfering each other.

In this process, we need to fully normalize the data, taking into account that the magnitude of various field quantities of electromagnetic field data with huge differences. By evaluating the order of magnitude of each candidate, we set a reasonable normalization coefficient for each candidate with certain prior knowledge to ensure that the disparity of magnitude order will not cause great impact on training results of the network.

1) Homogeneous simulation: Wave propagation in homogeneous medium is the most common problem for EM applications. Therefore, we select a homogeneous case as our first attempt to validate methodology proposed above.

Table II summarizes inversion results of proposed methodology applied to a multiple-physics EM interaction system of Equations (11-15) with all coefficients homogeneous. All data shown in Table II are sampled from $z = 200$ to $z = 209$, and $t = 500$ to $t = 509$. In total, 10 spatial points and 10 time points are selected for comparison. The five PDEs are obtained simultaneously with five coefficients $\mu_0, \varepsilon_0, \nu_c, \varepsilon_0\omega_p^2, \omega_{ce}$. It is noted that $\nu_c, \omega_p, \omega_{ce}$ are three important parameters of collisional magnetized plasmas. For simplicity, we combine $\varepsilon_0\omega_p^2$ together as one parameter. Most of the relative errors are relatively low approximately less than 5%. We add noise into the input data with signal to noise ratio $\text{SNR} = 40 \text{ dB}$. The equations with some noise $\text{SNR} = 40 \text{ dB}$ can also be identified with increased errors. In general, the inversion error increases but can still discover physical system even if significantly sub-sampled spatially and temporally.
TABLE II: Inversion Result for PDEs with 10 spatial points for each equation

| Equation | Error without noise | Error with noise SNR = 40dB |
|----------|---------------------|-----------------------------|
| \( \frac{\partial E_x}{\partial z} = -\mu_0 \frac{\partial H_y}{\partial t} \) | \( \mu_0 : 1.68\% \) | \( \mu_0 : 10.39\% \) |
| \( -\frac{\partial H_y}{\partial z} = \varepsilon_0 \frac{\partial E_x}{\partial t} + J_x \) | \( \varepsilon_0 : 10.14\% \) (the average of two terms) | \( \varepsilon_0 : 12.92\% \) (the average of two terms) |
| \( \frac{\partial J_x}{\partial t} + \nu_c J_x = \varepsilon_0 \omega_p^2 E_x + \omega_{ce} J_x \) | \( \nu_c : 9.12\% \), \( \varepsilon_0 \omega_p^2 : 0.53\% \), \( \omega_{ce} : 1.06\% \) | \( \nu_c : 23.38\% \), \( \varepsilon_0 \omega_p^2 : 6.84\% \), \( \omega_{ce} : 3.07\% \) |
| \( \frac{\partial J_z}{\partial t} + \nu_c J_z = \varepsilon_0 \omega_p^2 E_z - \omega_{ce} J_x \) | \( \nu_c : 0.39\% \), \( \varepsilon_0 \omega_p^2 : 1.02\% \), \( \omega_{ce} : 1.46\% \) | \( \nu_c : 9.19\% \), \( \varepsilon_0 \omega_p^2 : 5.73\% \), \( \omega_{ce} : 35.14\% \) |
| \( J_z = -\varepsilon_0 \frac{\partial E_z}{\partial t} \) | \( \varepsilon_0 : 4.32\% \) | \( \varepsilon_0 : 5.33\% \) |

Fig. 7: Inversion accuracy dependent on spatial sampling numbers. It is noted that each point indicates the accuracy of normalized parameters. The spatial sampling points are 2, 3, 5, 10, 15, 20, respectively and the beginning location is \( z = 200 \). It is noted that inversion accuracy reaches to 95% only for three samples but reduces substantially for two samples.

The sampling scheme of physical quantities in spatial and time domain is the most important parameter for calculating derivatives to ensure inversion accuracy. On one hand, the dependence of inversion accuracy on spatial sampling numbers is depicted for coefficients \( \nu_c \), \( \varepsilon_0 \omega_p^2 \), \( \omega_{ce} \), \( \varepsilon_0 \), \( \mu_0 \) in PDEs in Figure 7. The spatial sampling points are 2, 3, 5, 10, 15, 20, respectively and the beginning location is \( z = 200 \) at plasma-vacuum boundary. It is noted that inversion accuracy reaches to 95% for three spatial sampling position but reduces substantially for just two spatial samples. Thus, for considering the spatial sampling limitation, we need to make trade-off between accuracy and spatial sampling scale.

On the other hand, temporal sampling scenario of the field sequence data plays an important role in inversion accuracy as well. Figure 8 depicts the dependence of accuracy \( \varepsilon_0 \) on the gradient of \( E_z \) based on Equation (15) as an example. We select three typical time sampling blocks corresponding to low (red), middle (blue), and high gradient (green) in the upper panel of Figure 8. In each block, there are 10 temporal samples. It is noted that the accuracy reaches 99% for high gradient (green) and reduces to 90% for low gradient (red). It is noted that the inversion accuracy correlates with the time derivative of the sampled data. With data varying rapidly with time, the neutral network can catch data characteristics easily. For complete PDE equation set, it will require a common time block for all quantities with high gradients within reasonably large amplitude.

Fig. 8: Inversion accuracy dependent on temporal sampling scenario. Three time sampling blocks are marked for low (red), middle (blue), and high gradient (green) with 10 samples in each block. Here is an example of Equation (15) with the time derivative of \( E_z \) at spatial location \( z = 200 \). For low gradient (red block in upper panel), the accuracy is low (red in lower panel). It is noted that the inversion accuracy correlates with the time derivative of sampled data block.

2) Inhomogeneous simulation: For all simulation above, we assume homogeneous coefficient for discovering the PDEs set.
To verify algorithm further, we investigate the inhomogeneous coefficient further which often occurs for a wide range of applications.

Network adjustment. The structure of neural network needs to be adjusted to solve inhomogeneous problems. First, as the coefficients of inhomogeneous terms are varying with spatial position, we change parameters of inhomogeneous terms from a single number to a length-adaptive tensor. We call this length-adaptive tensor inhomogeneous parameter tensor. This tensor will be trainable in neural network. The length of inhomogeneous parameter tensor is equal to the spatial sampling point number. Secondly, we assume that prior knowledge is known which candidate terms have inhomogeneous parameters so that not all candidate terms should have inhomogeneous parameter tensors. Finally, the predicted parameter is no longer a single number but a tensor which indicates the varying parameters within spatial domain. After training, the underlying equations with inhomogeneous coefficients can be distilled from the inhomogeneous parameter tensor by the gradient descent method.

The input data of the scattered field for inhomogeneous density is obtained based on the JEC-FDTD algorithm. Here, we assume that \( \omega_0^2 \) obeys a sinusoidal envelope, positively proportional to plasma density for inversion. A normal expression of inhomogeneous parameter is

\[
\omega_p^2 \sim A \sin(kz + \varphi) + C
\]  

(22)

Here, \( A, k, \varphi, C \) is amplitude, wave number, phase and constant, respectively. Plasma parameter \( \omega_p^2 \) varies as Equation (22) from \( z = 200 \) to \( z = 600 \) with two wavelength inside plasmas correspondingly.

In our simulation, the exact form and normal expression of inhomogeneous parameter are unknown. In general, we assume that unknown inhomogeneous coefficients can be expressed as a kind of trigonometric series in the form of

\[
\frac{1}{2} A_0 + \sum_{n=1}^{N} (A_n \sin(nx) + B_n \cos(nx))
\]  

(23)

where \( n \) is the order of the series, \( A_n \) and \( B_n \) are coefficients, respectively.

We assume that \( N = 3 \) is sufficient to represent the expression of inhomogeneous parameter tensor. Thus, the estimated function of inhomogeneous parameters \( \tilde{h}(z) \) can be expressed as

\[
\tilde{h}(z) = A_1 \sin(kz) + B_2 \cos(kz) + A_2 \sin(2kz) + B_2 \cos(2kz) + A_3 \sin(3kz) + B_3 \cos(3kz) + C
\]  

(24)

where \( A_1, A_2, A_3 \) and \( B_1, B_2, B_3 \) are unknown coefficients to discover, respectively.

All unknown coefficients are determined in Equation (24) by sparse regression. We use adaptive moment optimizer (Adam optimizer) to solve the following equation

\[
\arg \min_{S} \sum \| \tilde{h}(z) - \hat{h}(z) \|_2 + \lambda \| S \|_1
\]  

(25)

For spatial sampling, we choose 20 spatial position collected from \( z = 240 - 260 \) for training and then predict inhomogeneous coefficient in the whole spatial domain. The identified coefficients of \( S \) are \( A_1 = 0.5036, B_1 = -0.0011, A_2 = -0.0020, B_2 = 0.0002 \) and \( A_3 = 0.0005, B_3 = 0.0018 \), respectively. The identified constant number is \( C = 0.9951 \) with true value 1. The dominant term \( A_1 \sin(kz) \) is successfully selected among all candidates and other terms are small enough to be ignored. The true and identified models of inhomogeneous coefficient are given by

\[
\omega_p^2 \sim 0.5 \sin(3.142z) + 1 \quad (\text{True})
\]  

\[
\omega_p^2 \sim 0.5036 \sin(3.123z) + 0.9951 \quad (\text{Identified})
\]  

(26) (27)

Figure 9 shows the inversion result of inhomogeneous coefficient in comparison with theoretical value by Equation (26). The spatial sampling position is shown by the black box from \( z = 240 - 260 \) for training. Although sampling only at few spatial grids, inhomogeneous coefficients can be obtained in the sampling regime and then make prediction outside measurement regime according to discovered equations. Basically, the inversion result agrees well with theoretical value after iteration with only 20 spatial and 10 temporal samples. For a Gaussian white noise of SNR = 40dB, the identified model with noise is

\[
\omega_p^2 \sim 0.5897 \sin(3.360z) + 1.082 \quad (\text{Identified})
\]  

(28)

However, there are some errors of prediction with noise. One possible source of error comes from the neural network training of inhomogeneous parameter tensor. Another source of error may arise from processes during distilling the underlying equation with inhomogeneous parameter tensor. Even though, this method still discover and predict the inhomogeneous parameter.

![Inhomogeneous parameter prediction without noise. The blue solid line is theoretical value, and the dotted line is predicted value. The black box indicates spatial sampling position from z = 240 - 260 inside plasmas for training.](image)

**Fig. 9: Inhomogeneous parameter prediction without noise.** The blue solid line is theoretical value, and the dotted line is predicted value. The black box indicates spatial sampling position from \( z = 240 - 260 \) inside plasmas for training.

**IV. CONCLUSION AND DISCUSSION**

In this paper, we have presented a data-driven network architecture to discover the hidden nonlinear PDE equation set, where first-principles derivation may be intractable. The
architecture extends the idea from the PDE-net with compressive sensing, which approximates differential operations by convolutions with properly constrained filters and approximate the nonlinear response by deep neural networks. We add a coefficient regression module to the system and Pareto analysis to improve the identification accuracy. The architecture can contribute construct a model which characterizes the observed dynamics and generalizes to unsampled regime of parameter space.

We apply this network architecture for Maxwells equations for a multiple-physics EM system. The discovery of PDEs are complex in terms of multiple-physics, non-linearity, inhomogeneity, and anisotropy. We can discover Maxwell's equations coupled with plasma current conservation equation with low error on the coefficient of PDEs. The homogeneous coefficients of PDEs have been constructed based on sparse temporal-spatial data. Also, our attempt demonstrates the possibility of inferring inhomogeneous coefficients of physical parameters with prior knowledge. By assuming that the expression of coefficient of inhomogeneous parameters can be equal to some order of trigonometric series, we use sparse regression to discover the equation underlying in the inhomogeneous coefficients with little prior knowledge. The identified equation requires a batch of adjacent temporal-spatial samples to evaluate derivatives of physical variables. When the discretized solution contains measurement noise, the numerical derivatives becomes challenging to evaluate that will need further work.

Finally, it is noted that current network has been tested for the 1-D PDEs for the extraordinary wave $\theta = 90^\circ$ with only 5 equations coupled. Considering arbitrary angle $\theta$ in 3-D spatial domain, we need to add the representation module of curl operator in the neural network. Secondly, this method has been simplified to identify models with some prior knowledge for inhomogeneous parameter. If no prior knowledge is available, we will need a new candidate function database for identifying unknown terms similary as homogenous network. Thirdly, kinetic effects of plasmas are ingnored in the forward model, and we make assumptions for cold plasmas as well. Finnaly, it is valuable to try the proposed framework on real data of nonlinear EM wave and space plasma interaction experiments [12]–[14] and other potential applications.

ACKNOWLEDGEMENT

This work was supported by the National Key Research and Development Program of China under Grant 2017YFB0502703 and also supported by the Shanghai Science Foundation under Grant 19ZR1403900 and Grant 19511100600.

REFERENCES

[1] J. Bongard and H. Lipson “Automated reverse engineering of nonlinear dynamical systems,” Phys. Rev., vol. 104, no. 24, pp.99439948, 2007.
[2] M. Schmidt and H. Lipson “Distilling Free-Form Natural Laws from Experimental Data,” Science., vol. 324, no. 5923, pp.81-85, 2009.
[3] S. L. Brunton, J. L. Proctor and J. N. Kutz, “Discovering governing equations from data by sparse identification of nonlinear dynamical systems,” Proceedings of the National Academy of Sciences., vol. 113, no. 15, pp.3932-3937, 2016.
[4] H. Schaeffer, “Learning partial differential equations via data discovery and sparse optimization,” Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences., vol. 473, no. 2197, pp.20160446, 2017.
[5] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, “Data-driven discovery of partial differential equations,” Science Advances, vol. 5, no. 4, pp.e1602614, 2017.
[6] M. Raissi and G. E. Karniadakis, “Hidden physics models: Machine learning of nonlinear partial differential equations,” Journal of Computational Physics, vol.357, pp.125-141, 2018.
[7] M. Raissi, P. Perdikaris, G. E. Karniadakis, “Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations,” arXiv preprint arXiv:1711.10566, 2017.
[8] Z. Long, Y. Lu, X. Ma, and B. Dong, “PDE-net: Learning PDEs from data,” arXiv preprint arXiv:1710.09668, 2017.
[9] Z. Long, Y. Lu, B. Dong , “PDE-Net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network,” arXiv preprint arXiv:1812.04426, 2018.
[10] J. F. Car, B. Dong, S. Osher and Z.W. Shen “Image restoration: total variation, wavelet frames, and beyond.” Journal of the American Mathematical Society., vol. 25, no. 4, pp.10331089, 2012.
[11] J. Zhang, H.Y. Fu, W. A. Scales, “FDTD analysis of propagation and absorption in nonuniform anisotropic magnetized plasma slab.” IEEE Transactions on Plasma Science., vol. 46, no. 6, pp.21462153, 2018.
[12] H. Y. Fu et al., “Stimulated Brillouin scattering during electron gyroharmonic heating at EISCAT,”Annales Geophysicae, vol. 33, no. 8, pp. 983990, 2015.
[13] H. Y. Fu, W. A.Scales, P. A. Bernhardt, Y. Q. Jin, S. J. Briczinski, “Asymbmetry in stimulated emission polarization and irregularity evolution during ionospheric electron gyroharmonic heating”, vol. 45, no. 18, pp. 9363-9371, 2018.
[14] H. Y. Fu and W. A. Scales, “Kinetic modeling of stimulated electromagnetic emissions during ionospheric heating experiment,” 2018 12th International Symposium on Antennas, Propagation and EM Theory (ISAPE), Hangzhou, China, 2018, pp. 1-3. doi: 10.1109/ISAPE.2018.8634190