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

Master's thesis (Tech.)
UNIVERSITY OF TURKU
Department of Future Technologies
Embedded Electronics
2020
Bing Xiong

Supervisors:
Prof. Ossi Kalevo
Dr. Haiyang Fu

UNIVERSITY OF TURKU
Department of Future Technologies

Bing Xiong: Data-driven Discovery of Multiple-Physics Electromagnetic Partial Differential Equations

Master of Science in Technology Thesis, 71p.
Embedded Electronics
January, 2020

The subject of data-driven discovery for equations has developed rapidly in recent years, especially in the field of finding equations of unknown forms, which provides new ideas for the study of complex systems. When there are unknown noise sources and other uncertain factors in the system, it is quite difficult to directly derive the system governing equation, because the equation is complicated and the calculation cost is large. But if we try to find the equation directly from the data, it will be helpful to improve these problems.

For the data in nonlinear multi-physics electromagnetic system, the deep learning method can be used to find the equation, which can obtain the governing equation form accurately and has high time efficiency and parameter precision. This thesis studies the algorithm of data-driven discovery equations in electromagnetic multiple physics problems and realizes the inversion of Maxwell's multiple physics equations.

Firstly, three methods of data-driven equation discovery are introduced, including symbol regression, sparse regression and neural network. Secondly, an algorithm based on sparse regression and convolutional neural network is proposed for multiple physics equations of Maxwell equations. This algorithm uses Euler method to approximate time differentiation and convolution kernel to compute space differentiation. At the same time, in the training process, the pareto analysis method was used to remove the redundancy.
Then, the model algorithm is applied to the multi-physics coupling simulation data of electromagnetic plasma, and the homogeneous and non-homogeneous equations of electromagnetic propagation are realized by using less time and space observation field samples, which has certain anti-noise performance. For the problem of propagation in uniform medium, the influence of spatial and temporal sampling method on the inversion precision of equation coefficients is studied. Under the condition of inhomogeneous media propagation, this thesis finds the changing law of inhomogeneous coefficient by changing the weight scale of neural network, aiming at the problem that the equation coefficient varies with the spatial scale. By using the properties of trigonometric series and some prior knowledge, the expression of the coefficient of inhomogeneous terms is approximated, and satisfactory results are obtained.

Finally, the thesis summarizes the proposed method and its main conclusion. In both homogeneous and inhomogeneous media, the model has good performance. Meanwhile, the author discusses the possible improvement methods for other problems and the idea that the structure of the model can be adjusted in a small range in the future and applied to the high-dimensional space and the problems with high-order spatial differentiation in the governing equations.

Data-driven discovery of partial differential equations in electromagnetic field problems can help solve complex problems, reduce computational complexity and improve computational speed. In the future study of complex system problems, data-driven discovery of governing equations will play an important role.

Keywords: deep learning, convolutional neural network, partial differential equations, multiple electromagnetic physics

TABLE OF CONTENTS

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Background and Significance of Research

Data-driven discovery is become more and more popular in recent years. As early as the beginning of the last century, scientists explored ways to discover physical laws through data [1]. With the explosive growth of data in various fields, the use of machines to extract rules from all kinds of complex data is bound to become a hot topic.

With comprehensive application of electromagnetic wave spectrum in radar, communication, navigation, computational power and storage, the amount of electromagnetic big data comes to a huge number. The thing that needs to be studied further is that how to distill the underlying electromagnetic physical laws in such a complex system. 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 solution can be obtained with analytic and computational methods. However, the realistic scene is sophisticated to tackle, which involves multiple-physics, multiple-scale and nonlinearity.

For a multiple physics system with complex interaction mechanisms, there is no exact quantitative analytic solution and the computational cost is high to solve a set of partial differential equations (PDEs). Therefore, the key question is how to use sparse given data to discover the principle of a complex system. If the model can be learned from spare data and then perform prediction, it is significant for complex electromagnetic multiple-physics system and beyond.

The data-driven discovery method is developing quickly during the past decades. It can be mainly divided into three categories, including symbolic regression, sparse regression and deep learning.

1.2 Typical Methods

1.2.1 Symbolic Regression

Earlier researches on data-driven discovery for free-form natural laws are based on symbolic regression. Two of recent researches are proposed by Bongard and Lipson (2007) [2] and Schmidt and Lipson (2009) [3]. The main idea is to calculate numerical differentiations of experimental data firstly and then use symbolic regression based on evolutionary algorithm to compare numerical differentiations with analytical derivative solutions. Unlike traditional linear regression methods, which fit the parameters of an equation with the given form, symbolic regression has its strength and novel use, it searches not only the form of the parameters but also the equation. The initial expression is composed of a random combination of algebraic operators (+, -, ×, ÷), analytic functions (such as some trigonometric functions), constants, state variables and other building blocks of mathematics. The new equations are formed by combining the algebraic operators, analytic function and so on with parameters of each term. Pareto analysis [4][5] is mentioned for reducing the complexity of underlying expressions. With the increasing of terms in expression, the prediction ability (which is accuracy when doing prediction) rises while it is apparent that if the expression of a physical law is too long, it is probably a wrong expression with the theory of Occam's razor principle [6]. Thus, in their research, they choose the model with the tradeoff of complexity and accuracy.

This method worked well in some traditional and simple physical laws' discovery such as a double-pendulum. When dealing with sophisticated physical model like EM problem, it may not be stable enough. What's more, the requirement of computation cost is large when the order of numerical differentiations rise, at the same time, the calculation of numerical differentiations will always cause error because the property of the numerical differentiation method cause this problem.

1.2.2 Sparse Regression

In addition to symbolic regression, other kinds of regression algorithms are commonly used in the field of machine learning, such as ridge regression, the least absolute shrinkage and selection operator (LASSO) regression [7], etc. Sparse regression is mainly proposed based on the problem that too many features make the results easy to over-fit and it is also used in compressive sensing to recover

information from a randomly sample signal [8][9]. Its advantage is that it can constrain the number of result features well, which on the one hand makes the model more compact, and on the other hand makes the result more reasonable and accurate. Sparse regression also has another application for dictionary learning which is also called sparse representation [10][11].

In recent years, sparse regression method for data-driven discovery on underlying governing equations was raised by Brunton et al. (2017) [12], Schaeffer (2017) [13], and Rudy et al. (2017) [14] (2019) [15]. As most underlying governing equations of physics process are PDEs having a universal expression. We assume there is a function $u(x_1, x_2, \dots, x_n)$ and it has at most quadratic nonlinearity, it has a PDE having the following term:

$$f(x_1, \dots, x_n; u, \frac{\partial u}{\partial x_1}, \dots, \frac{\partial u}{\partial x_n}; \frac{\partial^2 u}{\partial x_1 \partial x_1}, \dots, \frac{\partial^2 u}{\partial x_1 \partial x_n}) \tag{1}$$

In most physics motion, time t as an independent variable is indispensable and probably has only first-order differentiation. Therefore, Equation (1) can be equal to the following expression:

$$\frac{\partial u}{\partial t} = [x_1, \dots, x_n, u, \frac{\partial u}{\partial x_1}, \dots, \frac{\partial u}{\partial x_n}, \frac{\partial^2 u}{\partial x_1 \partial x_1}, \dots, \frac{\partial^2 u}{\partial x_1 \partial x_n}] \cdot \boldsymbol{V} \tag{2}$$

where vector $\boldsymbol{V}$ indicates the parameter of each candidate term at the right-hand-side of Equation (2). With this vector, all possible terms of Equation (2) can be selected by non-zero parameters.

To use sparse regression, a candidate function dictionary needs to be established which consists of simple and derivatives terms shown in Equation (2), and the candidate function dictionary can be defined as $\widetilde{F}(t)$:

$$\tilde{F}(t) = [x_1, \ldots, x_n, u, \frac{\partial u}{\partial x_1}, \ldots, \frac{\partial u}{\partial x_n}, \frac{\partial^2 u}{\partial x_1 \partial x_1}, \ldots, \frac{\partial^2 u}{\partial x_1 \partial x_n}]$$

(3)

$\tilde{F}(t) \cdot \boldsymbol{V}$ is the prediction form while the true form of equation can be marked as $F(t)$. Then sparse regression method is used to select the proper terms as part of the equations. There are several kinds of sparse regression:

L_2 norm sparse regression:

$$\min_{\boldsymbol{V}} \sum_{i=1}^{N} \| F(t_i) - \tilde{F}(t_i) \|_2 + \lambda \| \boldsymbol{V} \|_2$$

(4)

L_1 norm sparse regression:

$$\min_{\boldsymbol{V}} \sum_{i=1}^{N} \| F(t_i) - \tilde{F}(t_i) \|_2 + \lambda \| \boldsymbol{V} \|_1$$

(5)

Actually L_0 sparse regression is the best chose for sparse regression, because L_0 norm can directly select parameters as zero or non-zero. However, L_0 norm is hard to calculate so L_1 and L_2 norm sparse regression are better choices for calculation. Then, L_1 norm sparse regression is easier to get sparse solution then those with L_2 norm, so we often chose L_1 norm sparse regression to select correct terms from candidate function dictionary. In Schaeffer's research [13], he chose L_1 norm sparse regression for optimization.

In Rudy's research [14], they think L_1 norm sparse regression performs poorly when data are highly correlated, thus, they applied a new algorithm modified from ridge regression with tough thresholding, the main idea is to iteratively optimize the tolerance of sparse regression which is λ in Equation (5) and (6) then finally find the best solution.

The training data for sparse regression is generated from various ways including finite difference, spectral method and so on. For the calculation of numerical differentiations, most works choose finite difference method which may introduce large error, Rudy's method [14] is quite inspiring. They use polynomial interpolation to generate spatial differentiations from points near the sampling location. This method is good at using less spatial points which means it need less spatial points and sampling locations. This characteristic has a very big advantage in practical problem processing, especially for those cases with very few spatial sampling points, such as radar signals. Rudy' s method uses modified finite difference technology for noiseless data, which use polynomial interpolation to estimate partial differentials and filter noise by singular value decomposition. It is applicable to various standardized PDE equations. Rudy et al. tested seven different equations: KdV equation, Burgers equation, nonlinear Schrodinger equation, NLS equation, KS equation, reaction diffusion equation, and Navier Stokes equation.

Then in 2019, Rudy et al. raised a new method [15] for deal with data changing depending on time or space. In their up to date research, parametric PDEs are studied. They define a new kind of sparse regression called group sparse regression while the traditional sparse regression only minimized the non-zero term for one time, in other word for one vector to optimized. The group sparse regression considers vectors to optimized within a time series. There is a well-studied method for solving group sparsity called group LASSO, they compared GLASSO with Sequential Thresholding Ridge Regression (STRG) and got the conclusion that STRG worker better than GLASSO.

However, there are still some problems need to be optimized in sparse regression for data discovery area. Sparse regression method cannot find coefficients which appear as the parameter of each term, if the term is inside the function like sin ax, it is hard for sparse regression to find its correct form. For noise dealing, it is too complicated because it involves singular value decomposition which may make the whole system looks not so automatic. Even though sparse regression can use better strategy to calculate spatial differentiations like polynomial interpolation which needs less spatial points, it needs enough time points to ensure the result. It will

cause the increase of the computation cost of the whole system exponentially as the dimension of the candidate function dictionary increases.

Sparse regression method is of great significance in the field of data-driven discovery on equations. After fully absorbing the experience of predecessors, researchers used the idea of sparse recognition based on modern machine learning methods, combined with the advantages of symbolic regression, and optimized the computational performance and saved computational resources. At the same time, the idea of introducing sparse vector as punishment term is also an implementation of pareto analysis. From all aspects, this method is in line with human beings to explore the laws of physics and more efficient than traditional physics discovery routine. Their researches have provided treasurable value for the advancement of future generations.

1.2.3 Deep Neural Network

In recent years, thanks to the continuation of Moore's law, the computing power of computers has been in a state of rapid growth. As it is found that graphic process unit can provide acceleration for matrix operation, the comprehensive ability of neural network has achieved a qualitative breakthrough, which makes neural network become the leading role in this era of artificial intelligence. Various types of neural networks have been successively improved in efficiency.

Fully connected deep neural network [16] has very high complexity, which all the characteristics of different characteristics of decomposition is very sufficient for making it to the study of complex data usually has a good performance, but because of the result too mad, once processing image problem, must bring the explosive growth of the amount of data, it is not only to calculate the force is a challenge, also is a kind of the waste of resources, at the same time, along with the network, the gradient diffusion and gradient explosion problem will always threaten the training of the network.

Deep fully connected neural network in addition to not easy to deal with high dimension problems, also can't help with input data sequence having good

recognition effect. Recurrent neural network (RNN) [17] [18] arises at the historic moment, for a long-time signal sequence of training will introduce gradient dispersion and explosion. The influence of gradient it also contributed to the formation of long short-term memory (LSTM) network [19] [20]. LSTM is a neural network based on time circulation, which is specially constructed to solve the long-term dependence problem existing in normal RNN, because all RNN have a chain of repeating modules, and such repeated chain structure can cause serious gradient dispersion or gradient explosion. LSTM is a type of neural network that contains LSTM blocks or other kinds of neural network. In literature or other materials, LSTM blocks may also be described as a kind of intelligent network unit, because it can remember values of indefinite length of time. There is a gate in the block that can determine whether the input is important enough to be remembered and output. LSTM has many versions, one of which is GRU (Gated Recurrent Unit). According to the test of Google, the most important one in LSTM is Forget gate, followed by Input gate and Output gate.

If the input data to process is not of time series but of high dimension, convolution neural network (CNN) is the best choice. The attractive characteristic of the CNN that it shares parameters makes it stable and trainable when the layer number is big. When dealing with the problem of high dimension, it can keep gradient and optimization quality, while at the same time the researchers can choose features that they want according to their own requirements manually from the training in order to more easily get the classification results. There several kinds of CNNs which indicate the development of it, they are LeNet [21], AlexNet [22], VGG [23], GoogleNet [24], ResNet [25].

ResNet is almost the most wildly used CNN now for feature extraction. VGG tried to find out how deep the deep learning network could be so that it could continuously improve classification accuracy. In our daily impression, the deep learning model should be more expressive with the increase of depth (complexity and more parameters). Based on this basic knowledge, CNN classification network developed from seven layers in AlexNet to sixteen or nineteen layers in VGG, and later to twenty-two layers in GoogleNet. However, with the deepening of the research, then we found

that the depth of CNN network after reaching a certain depth, if we blindly increase the layer number not bring further classification performance improvement, it will cause the network convergence becomes slower, the test dataset classification accuracy is also worse, in other words the depth deepening makes network recognition rate significantly decreased. After eliminating the problem of over-fitting the model caused by the small data set, we found that the classification accuracy was still decreased in the deep network (compared with the shallow network). Due to the lack of understanding of the principle, VGG network reached 19 layers and then increased the number of layers began to lead to a decline in classification performance. The author of ResNet thought of the concept of residual representation commonly used in the field of conventional computer vision and further applied it to the construction of CNN model, so there was the basic residual learning block. It uses multiple parameter layers to learn the representation of residuals between input and output, instead of using parameter layers to directly. It does not try to learn the function relation between the input and output as what general CNN networks do (such as AlexNet/VGG, etc.). Experiments show that it is much easier (faster convergence) and more efficient (higher classification accuracy can be achieved by using more layers) to directly learn resists than to directly learn the mapping between inputs and outputs.

At present, ResNet has replaced VGG as the basic feature extraction network in general computer vision problems. The FPN network proposed by Facebook that can effectively generate multi-scale feature expression can also obtain an optimal combination of CNN features by taking ResNet as the basic network to give full play to its capabilities.

On the other hand, the convolutional neural network needs more experience of the researcher in parameter adjustment, and the physical meaning of the convolution kernel is still unclear. Therefore, while enjoying the high efficiency brought by the convolutional neural network, we also must bear the trouble caused by its poor performance to the researcher.

In the field of data-driven discovery equations, neural networks have been widely used in recent years. As for the problem of finding equations, the input data of the

network is probably neither pictures nor time series, so we need to make highly customized adjustments to the structure and use method of various neural networks.

Initially, using a deep learning to accurately identify nonlinear relation from these input and output data couples is, at best, too simple for some input and output data which is potentially high-dimensional. Fortunately, in many cases involving the modeling of physical and aerodynamic systems, there is a great gap between the prior knowledge being used and being not used that by modern machine learning cases. Make the knowledge principle controlled by the laws of physics change over time of the dynamic system, or some practical validation rules or other special knowledge, also it can make them as a regular item for the change of the system to a certain degree of constraint (for example, in incompressible fluid dynamic problems, those who abandon any solution has no practical significance, is likely to be in violation of the principle of conservation of mass). As a summary, if apply this structured knowledge coding method into a learning algorithm, can enlarge algorithm can obtain the information content, making it able to quickly adjust yourself, let oneself to the correct solution as soon as possible, in the case of only a few of the available training sample can still play a role.

In recent years, many scholars explored on neural network's utility in the data-driven discovery equations issues. As previously mentioned, how to better use of the neural network complexity and power of expression, at the same time adding certain constraints, is to apply neural networks to the important issue in data modeling problem, also because of these methods to the choice of network architecture.

Raissi and Karniadakis (2017) used fully connected deep neural network to discover underlying physics of nonlinear PDEs [26], [27] with less data required for training. 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 [28]. In their research, automation differentiation's application is achieved by TensorFlow, currently one of the hottest open source libraries for deep learning computations. The TensorFlow.gradients() function helps a lot when dealing with differentiation calculation within the same graph. They did researches in different cases. Take

Burger's Equation as an example. Firstly, they use 100 points randomly spread on the boundary of space and time for training, while in this case, the form of equation is known already. After this, they did experiment on 10,000 points randomly spread on both boundary and inside the function and this time without knowing the form of function. Both results are good for discover the underlying physics motion. However, the explicit form of the PDEs is assumed to be known in the first case and in the second case the final result is not an exact equation but only prediction on time and space domain. Their method provided valuable reference for later research in many aspects, for example, they abandoned the traditional numerical differential calculation method in favor of the more advanced and less error automatic differential calculation. In terms of the use of neural network, they tend to use the fully connected network structure conservatively. Although the structure is complex enough to make the network have better representativeness, the calculation cost of training is also very large because the network parameters are too miscellaneous. Therefore, if it is necessary to increase the complexity of the network to improve the recognition rate of the model to the physical equation, the fully connected network is not necessarily a very good choice.

More recently, Long et al. (2018) [29] raised PDE-NET, he utilized the connection between differentiation and convolution to discover nonlinear equations with minor knowledge on the equation form [30] [31]. The wavelet frame filters in convolutional neural network (CNN) with a training kernel is adopted to approximate spatial differentiations. In Long's work, he gives an example for demonstrating the use of convolution to approximate differentiation. Assume that we have the 2D Haar wavelet filters as following:

$$h_{00} = \frac{1}{4}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, h_{10} = \frac{1}{4}\begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix}, h_{01} = \frac{1}{4}\begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix}, h_{11} = \frac{1}{4}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}. \tag{6}$$

where h_{00} is low-pass filter and the rest three are high-pass filter. And we define the wavelet transformation **W** on a 2D function u as following:

$$\mathbf{W}(u) = \{h_{ij} \otimes u: 0 \le i,j \le 1\}$$

(7)

Based on this transformation, we have the following relation between the differentiation and convolution:

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

(8)

where δ_x and δ_y indicates spatial grid on the two directions. Therefore, the approximation relation is shown clearly. The proof of this relation can be obtained by referring to the Taylor expansion.

$$\begin{aligned} &\sum_{k_1,k_2=-\frac{1}{2}}^{\frac{1}{2}} h[k_1,k_2]u(x+k_1\delta x, y+k_2\delta y) \\ &= \sum_{k_1,k_2=-\frac{1}{2}}^{\frac{1}{2}} h[k_1,k_2] \sum_{i,j=0}^{1} \frac{\partial^{i+j}u}{\partial^i x\, \partial^j y}|_{x,y} \frac{k_1^i k_2^j}{i!\, j!} \delta x^i \delta y^j + o(|\delta x|^1 + |\delta y|^1) \\ &= \sum_{i,j=0}^{1} m_{i,j} \delta x^i \delta y^j \cdot \frac{\partial^{i+j}u}{\partial^i x\, \partial^j y}|_{x,y} + o(|\delta x|^1 + |\delta y|^1) \end{aligned}$$

(9)

in which $[k_1, k_2]$ indicates the subscript of filter h, and $m_{i,j}$ can be expressed as

$$m_{i,j} = \frac{1}{i!\,j!} \sum_{k_1,k_2=-\frac{1}{2}}^{\frac{1}{2}} k_1^i k_2^j h[k_1, k_2]\,, i, j = 0,1$$

(10)

here $m_{i,j}$ is a quite important variable in PDE-NET, which is the trainable in network.

Long et al. (2019) [32] 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.

The popularity of neural network method in various fields in recent years is mainly attributed to the rapid development of computing power of modern computers. And the black box's characteristics, on the one hand, confuse people about its fundamental principles, on the other hand, make it extremely robust, which is an advantage over any other traditional machine learning methods, and also makes it more like the human brain, after all, people still cannot understand how the human brain works. In the field of data-driven discovery equations, we borrow the excellent feature that is, the differentiation capability of neural network, and hope to reduce the ambiguity in the principle part as much as possible through structural improvement. The present study is only a preliminary exploration of the use value of neural network in this field, there is still a long way to go in the future.

In sum, the data-driven discovery method of the underlying physics of PDEs are in progress. However, they still have some problems that is, symbolic regression requires high computation cost and has trouble 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) [29][32] requires no knowledge on the differential operators and associated discrete approximations, which has been applied for 2-dimensional linear variable-coefficient convection-diffusion equation.

1.3 Novelty of the Thesis

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. Secondly, the multiple-physics electromagnetic problem involves a set of different partial differential equations. It requires one unified algorithm to retrieve the multiple coefficient of the PDE equation set simultaneously. Therefore, the data-driven discovery method for complex electromagnetic system will be specially designed as a framework. That's what this thesis wants to provide solutions specifically.

1.3.1 Unified Neural Network

In this thesis, we aim to design a data-driven network architecture to discover a set of nonlinear PDE equations. A unified neutral network with convolution is proposed in combination with spare regression and pareto analysis. By using the sparse regression method, we add the sparse regression elements in some specific locations in the network as punishment terms to constrain the complexity of the found equation, which is based on Occam's razor principle. On the other hand, we further apply the pareto analysis criteria, and we will compare the final loss function values after each training, so as to clearly conclude whether the candidates we have screened out really benefit from making the discovered equation more accurate.

1.3.2 Application on Multiple-Physics EM Problems

This proposed network is applied for an electromagnetic wave and plasma interaction system, which can discover the coefficient of the PDE set with relatively good accuracy. As is known to all, EM field has always been a thorny problem in many physical problems, which is mainly due to the abstractness of EM problems, the high uncertainty of the environment, and various complicated interference factors in practical problems. However, among the many researchers mentioned above, there are few studies involving the discovery of EM field equations, so there are very few

theoretical or experimental attempts in this field. Although many physical equations have many similarities in form, such as the derivative of time usually for the first order, but the EM field problems involved in a variety of physical process, and the influence of nonlinear effect is larger. There are not deeply studied by previous researchers, this part also shows that why they did not choose to EM field equation as a subject in the study. This is a new field that we want to explore. We hope to propose a more appropriate solution to EM field problems through an algorithm that combines the valuable experience of predecessors and more innovations.

1.3.3 Exploration on Inhomogeneous Problem

No matter the equations studied by previous researcher or EM equations, there are not many relevant researches on the problems of inhomogeneous classes. The problem of inhomogeneous can be divided into two categories: one is the change of inhomogeneous in time, the other is the change of inhomogeneous in space. The difficulty in dealing with inhomogeneous problems is mainly that the equation in non-uniform problems is changing. It may be that the form of the equation has changed, or the coefficients of some terms of the equation have changed, so that no definite form of the equation can be found. In this thesis, we also explore the problem of inhomogeneous. In this case, we firstly identified the variation of the coefficients in a very small space, and then further used the properties of trigonometric series to express the expression of the variation rule implied by the coefficients through trigonometric series. During this process, sparse regression is also used to realize the selection of trigonometric series, and finally good experimental results are obtained.

1.4 Structure of Thesis

In this thesis, the following content can be divided into three parts.

In Chapter 2, the main methodology and architecture of system are introduced. The structure of convolutional neural network is demonstrated firstly with the explanation of how to calculate time and space differentiations. And training method including sparse regression and Pareto analysis together with the architecture of system are introduced later.

In Chapter 3, simulations are carried out on homogeneous and inhomogeneous problems respectively. Preprocessing is introduced firstly including data collection and feature scaling. Then, in homogeneous cases, the simulations are carried out on one universal sampling and several personalized sampling strategies for each target equation. In inhomogeneous cases, the modification of network and new method to extract the expression of inhomogeneous parameter are introduced.

In Chapter 4, discussion and conclusion on the simulation results are given, the problems that need to be improved are analyzed and some possible solutions are given. And the future work for data-driven discovery in EM problem is introduced, it is mainly to focus on arbitrary inclination angle, high order of differentiation. What's more, based on the performance of the previous two works, application on real data is also an important direction.

The data-driven methods of the PDEs set will pay the way for deriving equations for complex partially known and unknown systems including nonlinear, multiple physics, EM equations and beyond.

Chapter 2 Methodology

In this chapter, it introduces the basic algorithms and basic principles adopted in this thesis. The methods for how to calculate time and space differentiations are introduced and after which is the whole structure of convolutional neural network. And training method including sparse regression and Pareto analysis together with the architecture of system are introduced later. The method introduced in this chapter will be utilized in Chapter 3, while for clarity, electromagnetic equations are not referred in this chapter, all equations as examples are of universal form.

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

$$\begin{aligned}\frac{\partial u}{\partial t} &= f(x, y, u, v, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y}, \frac{\partial^2 u}{\partial x\,\partial y}, \frac{\partial^2 v}{\partial x\,\partial y}\dots)\\ \frac{\partial v}{\partial t} &= g(x, y, u, v, \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y}, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial^2 v}{\partial x\,\partial y}, \frac{\partial^2 u}{\partial x\,\partial y}\dots)\end{aligned} \tag{11}$$

where f and g is the nonlinear function of all possible terms including nonlinear functions of u and v, their derivatives and other parameters, respectively.

2.1 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 calculation, according to the forward Euler method, we have the following formula

$$\begin{aligned}\tilde{u}(t_{i+1},\cdot) &= u(t_i,\cdot) + \Delta t \cdot \frac{\partial u}{\partial t}\\ \tilde{v}(t_{i+1},\cdot) &= v(t_i,\cdot) + \Delta t \cdot \frac{\partial v}{\partial t}\end{aligned} \tag{12}$$

where $\tilde{u}(t_{i+1},\cdot)$ or $\tilde{v}(t_{i+1},\cdot)$ is the approximation value at $t = t_i$. Then, we have this expression:

$$\begin{aligned}\tilde{u}(t_{i+1},\cdot) &= u(t_i,\cdot) + \Delta t \cdot f \\ \tilde{v}(t_{i+1},\cdot) &= v(t_i,\cdot) + \Delta t \cdot g\end{aligned} \tag{13}$$

There are various choices for time derivative calculation like finite differentiation, Runge-Kutta method [33] and so on. The reason for choosing forward Euler method is that the structure of network is based on the calculation method, thus, to keep the neural network's structure simple enough for training, we select the simplest method for time derivation calculation. If there is higher requirement for accuracy of time derivation, the network should be adjusted together with the method for time derivative.

2.2 Spatial Derivative

For spatial derivative calculation, the connection between convolutions and differentiations was studies by Cai et al. (2012) [30] and by Dong et al. (2017) [31]. Here, we demonstrate one simple example of their work [30][31] to show how to express the relation between differentiation and convolution. Assume that we have the 2-dimensional Haar wavelet filters includes one low-pass filter h_{00} and three high-pass filters h_{10}, h_{01} and h_{11}:

$$h_{00} = \frac{1}{4}\begin{pmatrix}1 & 1\\ 1 & 1\end{pmatrix}, h_{10} = \frac{1}{4}\begin{pmatrix}1 & -1\\ 1 & -1\end{pmatrix}, h_{01} = \frac{1}{4}\begin{pmatrix}1 & 1\\ -1 & -1\end{pmatrix}, h_{11} = \frac{1}{4}\begin{pmatrix}1 & -1\\ -1 & 1\end{pmatrix}. \tag{14}$$

Then, the circular convolution operator is labeled as $\otimes$ and we define the wavelet transformation **W** on a 2D function u as following:

$$\mathbf{W}(u) = \{h_{ij} \otimes u: 0 \le i, j \le 1\}$$

(15)

Based on this transformation, we have the following relation between the differentiation and convolution:

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

(16)

where δ_x and δ_y are the grids of x and y direction of function u or v, respectively.

Similarly, for v,

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

(17)

The proof of Equation (16) can be obtained by the Taylor expansion.

$$\sum_{k_1,k_2=-\frac{1}{2}}^{\frac{1}{2}} h[k_1,k_2]u(x+k_1\delta x, y+k_2\delta y)$$

$$= \sum_{k_1,k_2=-\frac{1}{2}}^{\frac{1}{2}} h[k_1,k_2] \sum_{i,j=0}^{1} \frac{\partial^{i+j}u}{\partial^i x\,\partial^j y}|_{x,y} \frac{k_1^i k_2^j}{i!\,j!}\delta x^i \delta y^j + o(|\delta x|^1 + |\delta y|^1)$$

$$= \sum_{i,j=0}^{1} m_{i,j}\delta x^i \delta y^j \cdot \frac{\partial^{i+j}u}{\partial^i x\,\partial^j y}|_{x,y} + o(|\delta x|^1 + |\delta y|^1)$$

(18)

in which k_1, k_2 indicates the subscript of filter h and $m_{i,j}$ can be expressed as

$$m_{i,j} = \frac{1}{i!\,j!} \sum_{k_1,k_2=-\frac{1}{2}}^{\frac{1}{2}} k_1^i k_2^j h[k_1,k_2]\,, i,j = 0,1$$

(19)

here $m_{i,j}$ is a shorthand for a coefficient. In our work, the coefficient is simplified to a trainable weight which can save computation power and performs well.

By multiplying constant coefficients, convolution can represent differentiation in neural network effectively. The calculation of the second order differentiation is also mentioned in [30], it is ignored since our current work does not involve it as will be discussed in future.

2.3 Δt 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 Δt block [29] as a layer of neural network to advance variables based on equations in time. In principle, the structure of the Δt block is an interpretation of Equation (13). The two left terms in Figure. 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

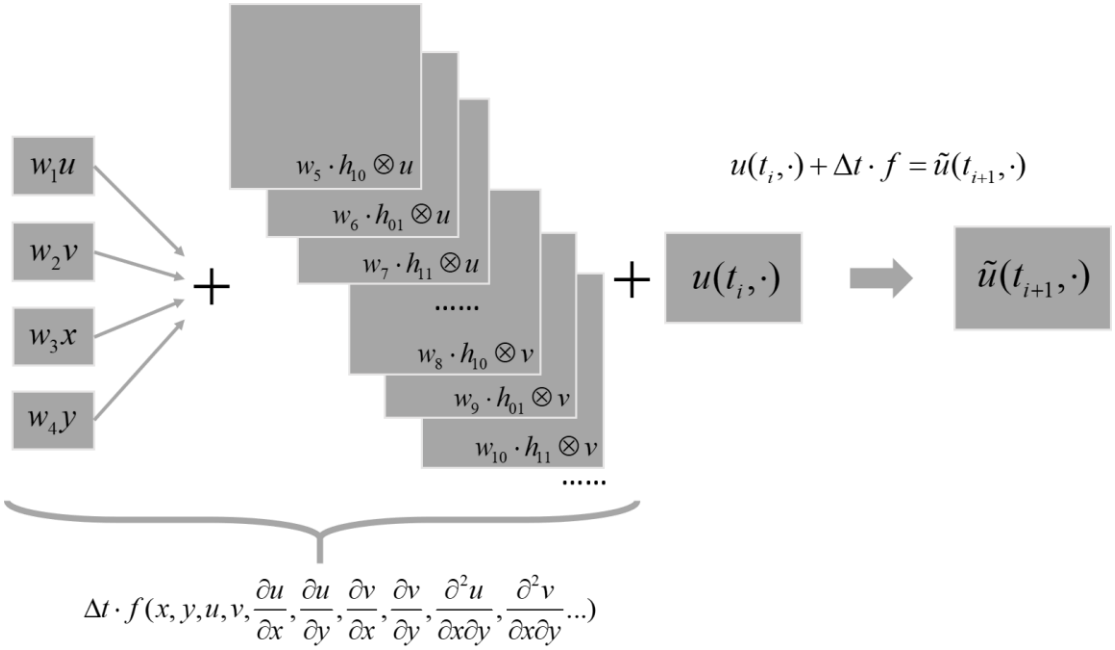

Figure 1. The architecture of a Δt block [29] to advance function in time. The two left parts represent the target function f in Equation (13), that includes a candidate term database in the form of neural network as in Equation (5-6). The input of this network is all possibly related functions which 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 is coupling between u and v in PDEs set.

prior knowledge will help reduce the number of possible candidate functions associated with PDEs. The output of Δt block is the predicted value at the next time stamp $\tilde{u}(t_{i+1},\cdot)$.

In Figure. 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

$$\boldsymbol{W} = [w_1, w_2, w_3, w_4, w_5, \ldots, w_n] \tag{20}$$

In general, the whole network consists of several Δt blocks by continuously connecting each block one by one. For the number of Δt blocks, it is determined by convergence criteria we choose.

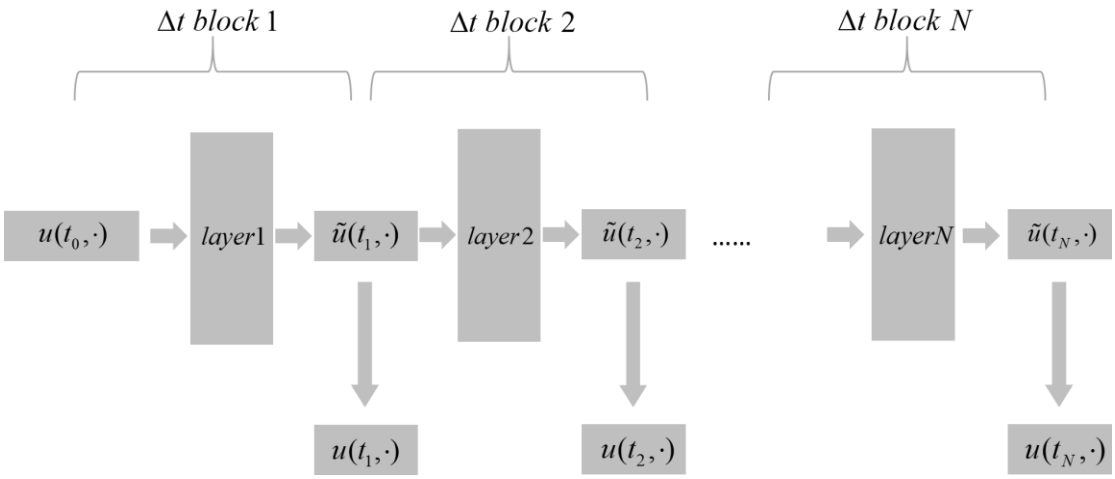

Figure 2. Δt blocks are connected one by one [29], and the parameters like weights are shared in each block, this method saves the number of parameters to train and keep the form of equation at the same stage as time going step by step.

2.4 Sparse Regression for Regularization

Sparse regression is a common method in data-discovery area. When there are too many terms discovered with small parameters, in our case, it means that the system may face with over-fitting problems. Regularization method need to be adopted in training process to avoid over-fitting. For the type of regularization, even though L_0

is the most direct method and effective method for regularization, it is still hard to calculate and in deep learning framework, it is hard to calculate gradients. Thus, L_1 norm regularization is more commonly used as it is easier to calculate than L_0 norm regularization and more powerful than L_2 norm regularization.

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

The output of Δt 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

$$l_i = \| \tilde{u}(t_{i+1},\cdot) - u(t_{i+1},\cdot) \|_2 \tag{21}$$

Consider sparse regression to optimize the output $\boldsymbol{W}$ 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 $\boldsymbol{W}$. The accumulated loss function can make prediction reliable within certain time step numbers N. The expression for final loss L is

$$L = \sum_{i=1}^{N} \| \tilde{u}(t_i,\cdot) - u(t_i,\cdot) \|_2 + \lambda \| \boldsymbol{W} \|_1 \tag{22}$$

where λ is the regularization parameter to decide the weight of regularization terms.

2.5 Pareto Analysis

Pareto analysis [4] is a method that can be effective in scenarios where many potential action processes are competing to become the principal component. In short, the problem-solver makes decision through evaluating the benefits of each action and then selects a small number of the most effective actions that bring the

total benefits to approximate the maximum possible benefits [4]. While it is common to call pareto as the "80/20" rule, in all cases, 20% of the reasons for 80% of the problems are assumed to be a rule of thumb for remembering, not and should not be considered an absolute principle in nature. This method helps to identify the main component part of the problem that needs to be notices and located.

The application of Pareto analysis that has been adopted 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.

2.6 System Architecture

Figure 3. illustrates the architecture of the whole neutral network to deal with a given equation set. This process includes:

(1) Find out the number of possible equations to discover, e. g., f or g .

(2) Determine time derivatives, e. g., $\widetilde{u}(t_{i+1},\cdot)$ or $\widetilde{v}(t_{i+1},\cdot)$.

(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.

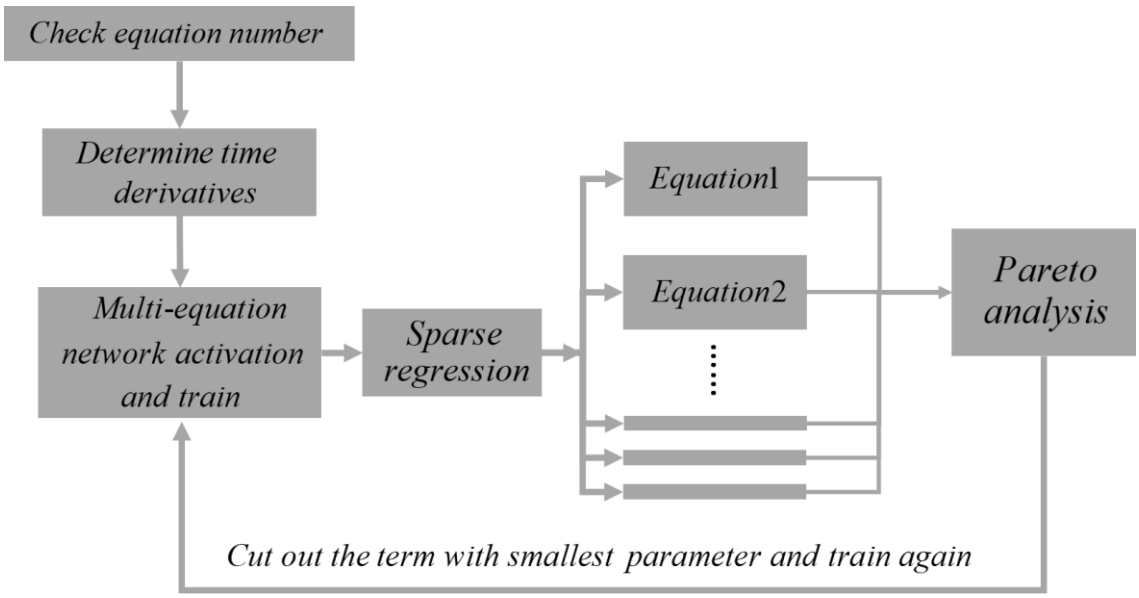

Figure 3. The architecture of discovering the PDE equation set. The procedure steps include 1) Check the number of PDEs equations. 2) Determine the time derivative of active terms in the PDE. 3) Activate multiple equation network corresponding to each equation. 4) Train network with sparse regression regularization. 5) Pareto analysis helps to cut out unrelated terms with minimum parameters.

Thus, the whole system is shown as above. There are some things need to be explained more. First is that, the first step checking number of equations is quite important in our method, to ensure covering all possible equations (as the number and the form of equation are both unknown before training) we will preprocess the network into all possible structures by calculating the time derivatives. In our simulation, all possible time derivatives are indicating one equation respectively. Second is that the activation of neural network is done manually currently, we will optimize the code in the future to make all process into automation. Third is that, Pareto analysis is implemented after training, which means it analyzes the result of

discovering for equation, so this step is different from sparse regression and cannot be replaced.

2.7 Summary

In this chapter, the main methodology used in this thesis is introduced. The whole neural network's structure is based on the calculation of time derivative which is the process of forward Euler method and the convolutional layers are consist of spatial differentiation operators which are approximated by convolutional kernel. These two methods ensure that the network can express a universal form of PDE. What's more, sparse regression and Pareto analysis are introduced as main training methods to keep the form of equation clean and simple enough. The whole system architecture includes all methods mentioned above and works for a set of equations.

Chapter 3 Simulation Model

3.1 Forward Model

In the ionosphere, we usually have the following deformation equations for Maxwell's equations. For the anisotropic magnetized cold plasma, assume that the angle between the background magnetic field **B** and the $+z$ axis of the electromagnetic wave propagating along the direction of $+z$ is theta, and in the yoz plane, ignore the movement of ions.

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 their constitutive relations are given as

$$\nabla \times \mathbf{H} = \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t} + \mathbf{J}$$
$$\nabla \times \mathbf{E} = -\mu_0 \frac{\partial \mathbf{H}}{\partial t}$$
$$\frac{\partial \mathbf{J}}{\partial t} + \nu_c \mathbf{J} = \varepsilon_0 \omega_p^2 \mathbf{E} + \omega_{ce} \times \mathbf{J} \tag{23}$$

where $\mathbf{H}$, $\mathbf{E}$ and $\mathbf{J}$ are vectors of magnetic intensity, electric field intensity, and polarized current density, respectively. μ_0, ε_0 are the magnetic permeability and vacuum permittivity respectively. Here, ν_c , ω_p and ω_{ce} are electron collision frequency, plasma frequency, and electron cyclotron 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 θ with respect to the $\mathbf{z}$ axis. Equation (23) is only for the situation that ion motion is neglected and cold plasmas is assumed.

Figure 4. shows the propagation model of EM wave in plasmas with arbitrary magnetic inclination angle θ with respect to the $\boldsymbol{z}$ axis. The electromagnetic wave is propagating along the $\boldsymbol{z}$ axis direction. The scattered field for a 1D magnetized plasma slab are solved based on the current density convolution finite-difference time domain (JEC-FDTD) [11].

To verify our method, we calculate the scattering field $\mathbf{E}_s$ for a one-dimensional magnetized plasma slab shown in Figure. 4. The computational space takes up 800 grids along the $\boldsymbol{z}$ directions, among which plasma takes up 200-600 grids with uniform distribution and a thickness of $d = 3cm$. The JEC-FDTD method is utilized to get the solution of Equation (23). The algorithm of JEC-FDTD can be found in [34]. For uniform plasma distribution, parameters in the simulation are set as following in Table I.

Table I. System Parameters

Symbol	Value
Space iteration step size δz	75 μm
Time iteration step size δt	0.125 ps
Electron collision frequency ν_c	20 Ghz
Electron cyclotron frequency ω_{ce}	30 Ghz
Uniform plasma frequency ω_p	40 Ghz
Permittivity of vacuum ε_0	8.85×10^{-12}
Magnetic inclination angle θ	90°

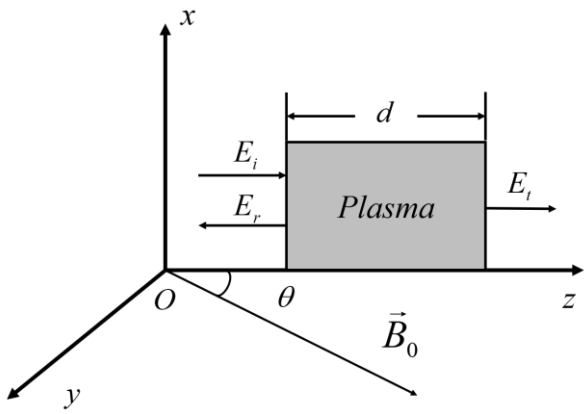

Figure 4. Multiple physics EM wave interaction with plasma with arbitrary magnetic inclination $\theta = 90°$. The spatial domain is in the $\boldsymbol{z}$ direction.

For $\theta = 90°$, Equations (23) are expressed as a scalar differential equation as:

$$\frac{\partial E_x}{\partial z} = -\mu_0 \frac{\partial H_y}{\partial t} \tag{24}$$

$$-\frac{\partial H_y}{\partial z} = \varepsilon_0 \frac{\partial E_x}{\partial t} + J_x \tag{25}$$

$$\frac{\partial J_x}{\partial t} + \nu_c J_x = \varepsilon_0 \omega_p^2 E_x + \omega_{ce} J_z \tag{26}$$

$$\frac{\partial J_z}{\partial t} + \nu_c J_z = \varepsilon_0 \omega_p^2 E_z - \omega_{ce} J_x \tag{27}$$

$$J_z = -\varepsilon_0 \frac{\partial E_z}{\partial t} \tag{28}$$

Here, there are 5 Equations (24-28) to discover with five unknown parameters including $\mu_0, \varepsilon_0, \varepsilon_0\omega_p^2, \omega_{ce}, \nu_c$. This set of PDE equations is our first goal for equation discovery. This set of PDE equations is simply for EM wave propagation in anisotropic plasmas, however, it can effective reveal the application for our proposed network.

3.2 Preprocessing

The whole process of inverse model is shown in Figure. 5. The physical quantity input data are E_x, J_x, E_z, J_z, H_y calculated from JEC-FDTD. Sampling is conducted for the input data E_x, J_x, E_z, J_z, H_y, respectively.

3.2.1 Feature Scaling

For preprocessing the data, normalization is also an important part, because each quantity is of different orders of magnitude, if there is no normalization, the parameters learning of each term will have large difference with each other. This would cause big wrong for sparse regression also for optimization of neural network.

For normalization, the goal is to modified each input data into the same order of magnitude by comparing with the time derivatives. Taking Equation (24) as example, we have the relations as following:

$$\frac{\partial E_x}{\partial z} = \frac{\delta E_x}{\delta z} \tag{29}$$

$$\frac{\partial H_y}{\partial t} = \frac{\delta H_y}{\delta t} \tag{30}$$

In which, δt and δz are temporal and spatial grids respectively shown in Table I. δE_x and δH_y are differences respectively. Thus, the partial differentials can be expressed in terms of differences. Then, we have

$$\frac{\delta E_x}{\delta z} = -\mu_0 \frac{\delta H_y}{\delta t} \tag{31}$$

$$\delta H_y = -\frac{\delta t}{\mu_0 \delta z}\delta E_x \tag{32}$$

The normalization coefficient is $-\frac{\delta t}{\mu_0 \delta z}$ for E_x of this equation.

Similarly, Equation (25) has:

$$-\frac{\delta H_y}{\delta z} = \varepsilon_0 \frac{\delta E_x}{\delta t} + J_x \tag{33}$$

$$\varepsilon_0 \frac{\delta E_x}{\delta t} = -\frac{\delta H_y}{\delta z} - J_x \tag{34}$$

$$\delta E_x = -\frac{\delta t}{\varepsilon_0 \delta z}\delta H_y - \frac{\delta t}{\varepsilon_0} J_x \tag{35}$$

The normalization coefficient are $-\frac{\delta t}{\varepsilon_0 \delta z}$, $-\frac{\delta t}{\varepsilon_0}$ for H_y and J_x respectively of this equation.

Equation (26) has:

$$\frac{\delta J_x}{\delta t} + \nu_c J_x = \varepsilon_0 \omega_p^2 E_x + \omega_{ce} J_z \tag{36}$$

$$\delta J_x = -\delta t \nu_c J_x + \delta t \varepsilon_0 \omega_p^2 E_x + \delta t \omega_{ce} J_z \tag{37}$$

The normalization coefficient is $-\delta t \nu_c$, $\delta t \varepsilon_0 \omega_p^2$, $\delta t \omega_{ce}$for J_x, E_x and J_z respectively of this equation.

Equation (27) has:

$$\frac{\delta J_z}{\delta t} + \nu_c J_z = \varepsilon_0 \omega_p^2 E_z - \omega_{ce} J_x \tag{38}$$

$$\delta J_z = -\delta t \nu_c J_z + \delta t \varepsilon_0 \omega_p^2 E_z - \delta t \omega_{ce} J_x \tag{39}$$

The normalization coefficient is $-\delta t \nu_c$, $\delta t \varepsilon_0 \omega_p^2$, $-\delta t \omega_{ce}$ for J_z , E_z and J_x respectively of this equation.

Equation (28) has:

$$J_z = -\varepsilon_0 \frac{\delta E_z}{\delta t} \tag{40}$$

$$\delta E_z = -\frac{\delta t}{\varepsilon_0} J_z \tag{41}$$

The normalization coefficient is $-\frac{\delta t}{\varepsilon_0}$ for J_z of this equation.

The normalization is the key process step during the overall process of the inverse model for a multiple-physics EM problem as shown in the following steps:

Step 1: Data collected from a solution of PDE sets Equations (24-28).

Step 2: Data spare sampling and normalization.

Step 3: Check the total number of equations and determine the time derivative of the input data.

Step 4: Multi-Equation network activation and training as shown in Figure 3.

Step 5: Sparse regression is set at the end of training and the detailed form of equations will be shown later.

Step 6: After Pareto analysis, distortion terms will be deleted.

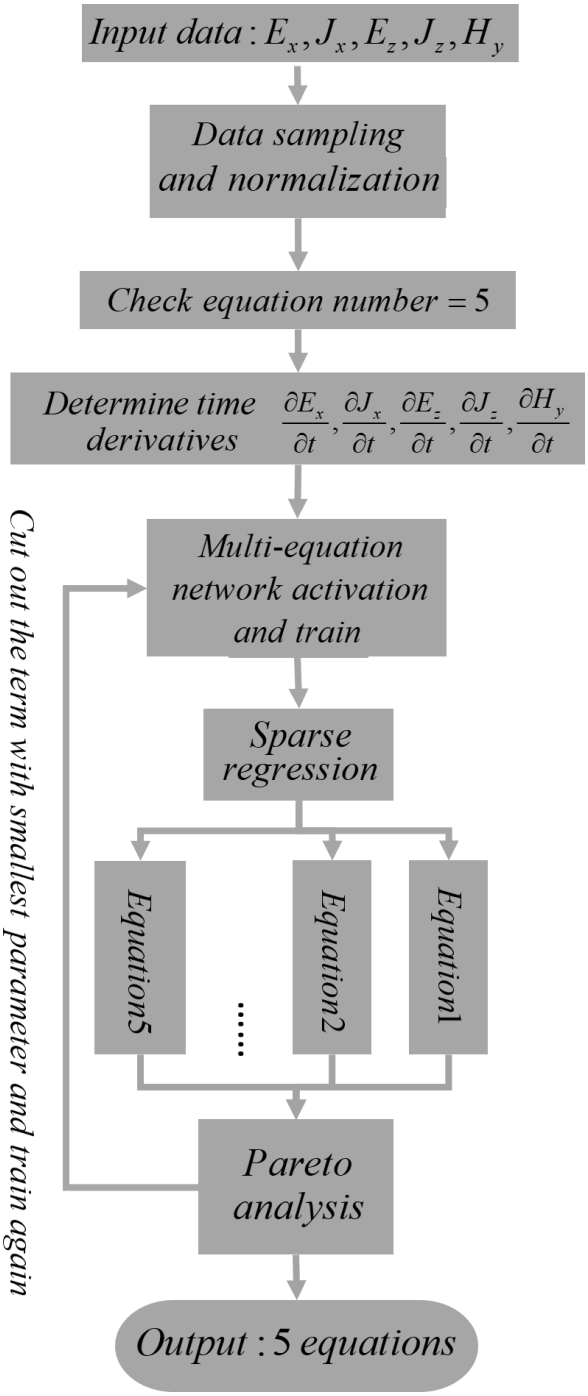

Figure 5. The overall process of the inverse model for a multiple-physics EM problem. Step 1: Data collected from a solution of PDE sets Equations (24-28). Step 2: Data spare sampling and normalization. Step 3: Check the total number of equations and determine the time derivative of the input data. Step 4: Multi-Equation network activation and training as shown in Figure 3. Step 5: Sparse regression is set at the end of training and the detailed form of equations will be shown later. Step 6: After Pareto analysis, distortion terms will be deleted.

Normalization is executed directly after data sampling. As we use the same input for different equations, we need to do normalization for each equation when the data is set up as candidate function in this term because the order of magnitudes of input data will cause big error for data training and lead to wrong results.

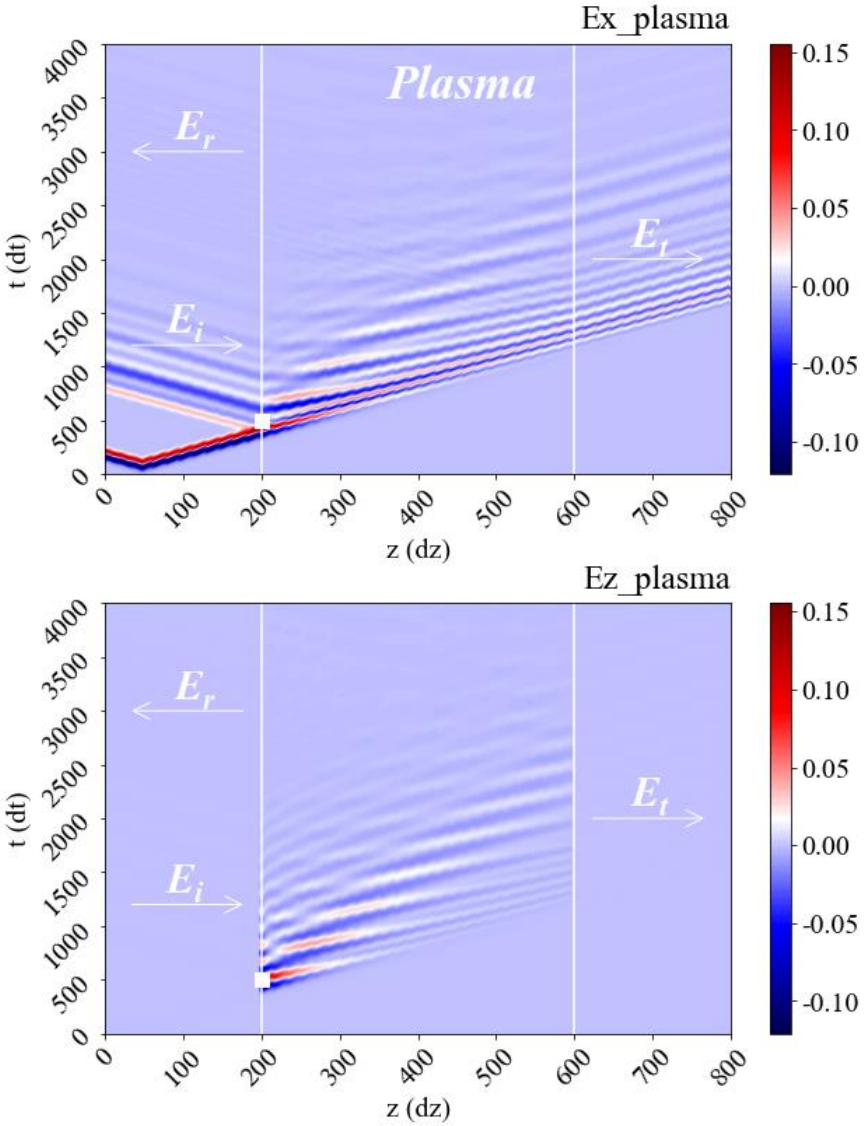

Figure 6. Data collection location for E_x, E_z. For upper part, this is a x-direction electric field intensity E_x picture. The two white lines means the incident location and exit location. The small white square represents the collection area. For lower part, it is the same for E_z.

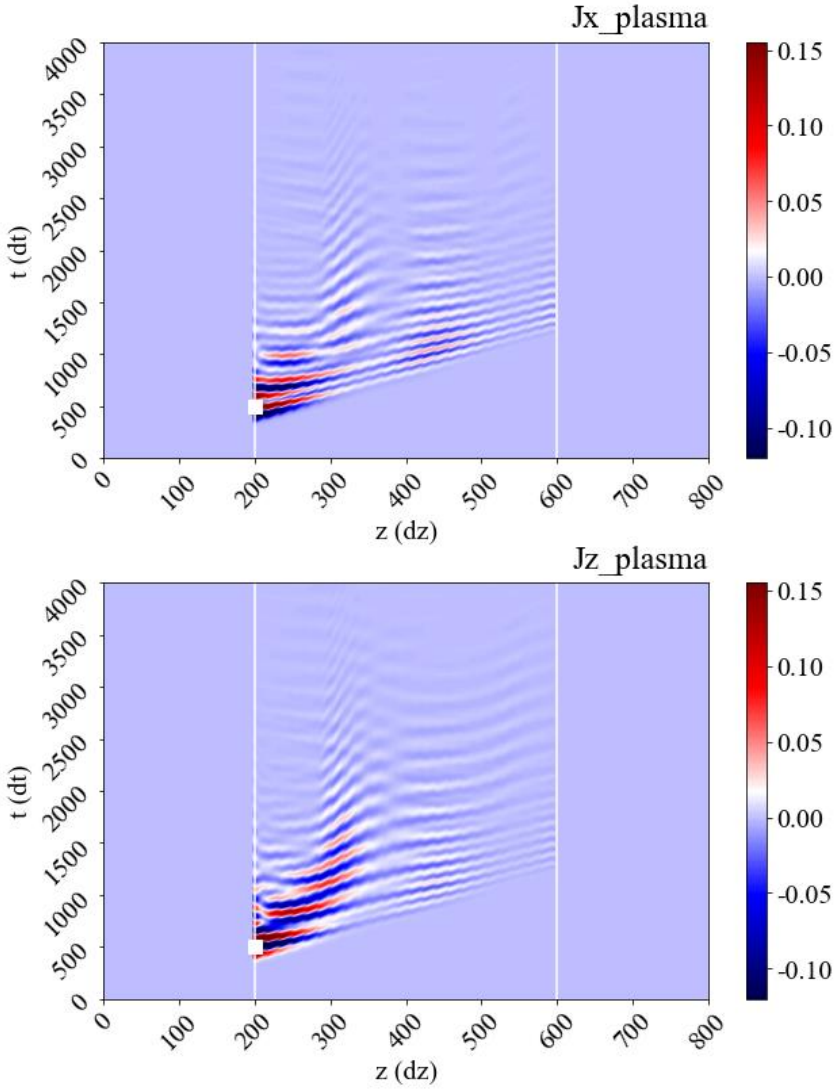

Figure 7. Data collection location for J_x,J_z. The pictures for J_x,J_z, the collection locations are the same while the white box indicates the sampling area. It can be observed that we sampled the data from the location with as much information as possible.

Thus, 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 and then multiply a parameter to keep all candidate functions as same magnitude with the time derivative.

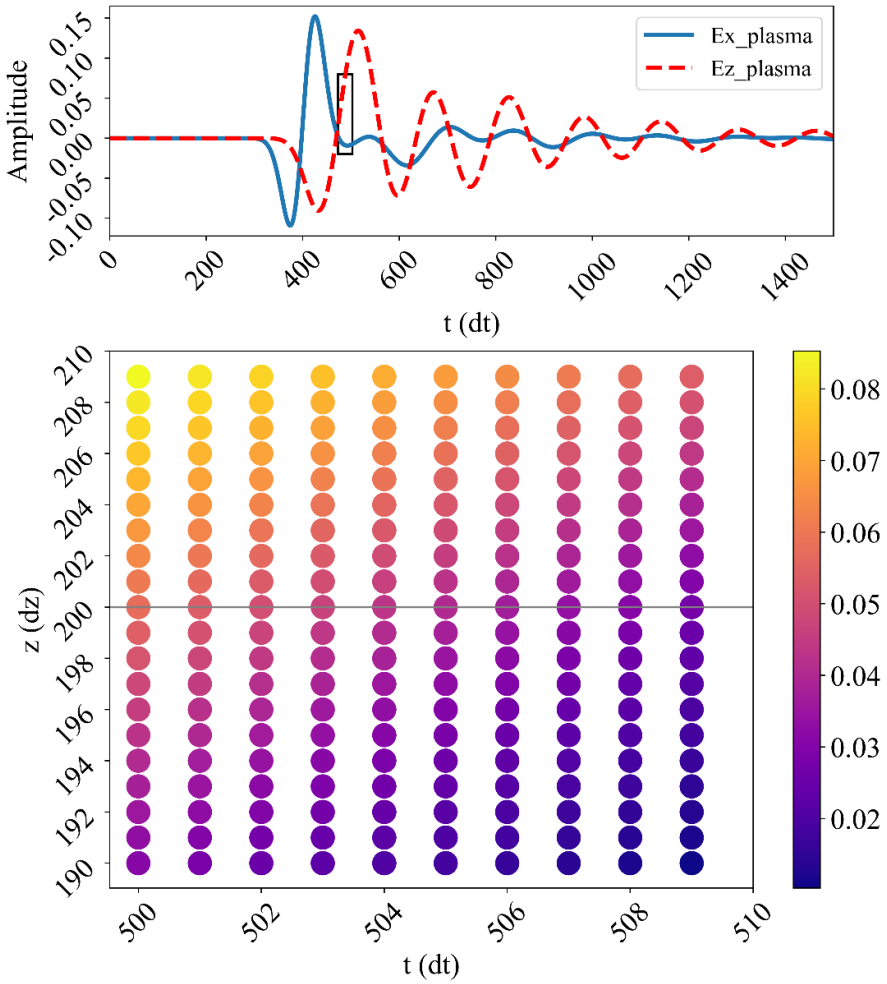

Figure 8. Sampling area in detailed. In the upper panel, the black square shows the data sampling area for E_x and E_z in time domain. In the bottom panel, the black horizontal line indicates the border of plasma and our collection area begins right on the incident border.

3.2.2 Data Collection

For training data collecting, we select three physical quantities of the scattered field E_x, E_z, H_y and two currents J_x, J_z at spatial grids 200 to 219 close to plasma-vacuum boundary (grid 200). For temporal series of physical quantities, we choose few samples at the time period 500-509, which includes the maximum value of the scattered field.

Figure 6. and Figure 7. 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 6. 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 collection location on time axis, we choose 10 steps from 500-time step, because at this location the gradient of curve is obvious so that the function is easy to be recognized by neural network.

The sampling area for training network is shown in Figure 8. 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 is clear that only a small number of spatial information and time are sampled.

3.3 Network Construction

From forward Euler's method, we have

$$\tilde{E_x}(t_{i+1},\cdot) = E_x(t_i,\cdot) + \Delta t \cdot \frac{\partial E_x}{\partial t} \tag{42}$$

$$\tilde{H_y}(t_{i+1},\cdot) = H_y(t_i,\cdot) + \Delta t \cdot \frac{\partial H_y}{\partial t} \tag{43}$$

$$\tilde{J_x}(t_{i+1},\cdot) = J_x(t_i,\cdot) + \Delta t \cdot \frac{\partial J_x}{\partial t} \tag{44}$$

$$\tilde{J_z}(t_{i+1},\cdot) = J_z(t_i,\cdot) + \Delta t \cdot \frac{\partial J_z}{\partial t} \tag{45}$$

$$\tilde{E_z}(t_{i+1},\cdot) = E_z(t_i,\cdot) + \Delta t \cdot \frac{\partial E_z}{\partial t} \tag{46}$$

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:

$$h_{10} \otimes E_x \approx \frac{1}{2}\delta_z \frac{\partial E_x}{\partial z}$$

(47)

From this, we can get the adjusted neural network structure. To fulfil the function of Equation (47), we use a function called conv2d in TensorFlow to generate convolution calculation. In our case this is a 1D problem, so we set the shape of convolution kernel to 2×1, so that this function still works for it.

Based on the detailed situation of physical quantities, we adjust the Δt block's structure into a detailed one. The design principle is not changed at all, while the candidate function layer is compressed for simplicity in calculation.

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 (24-28).

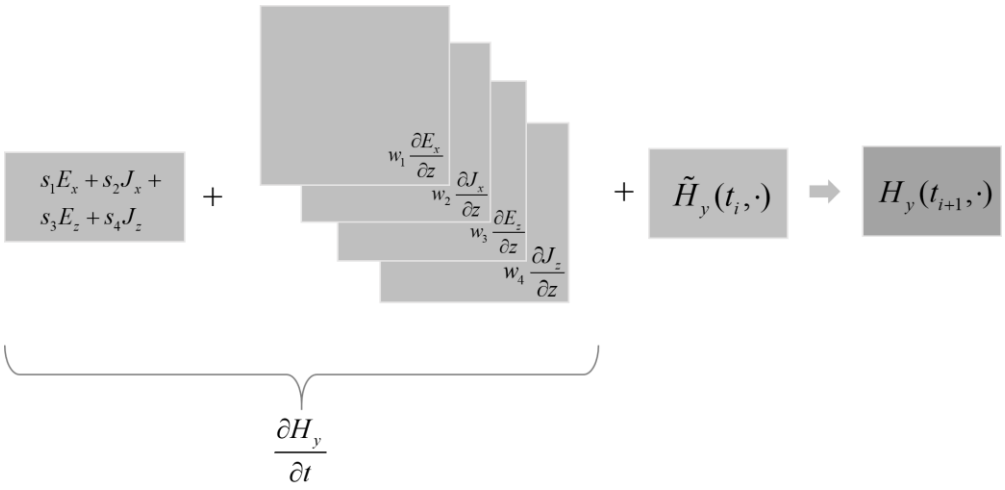

Figure 9. The adjusted neural network structure - function h contains all possibilities of this equation and as the spatial direction is only z, so we use only one filter to get one kind of spatial differentiation.

As one single block given above, we have 10 Δt blocks to build up the whole network. By using the parameter sharing characteristic of neural network, the training coefficient results becomes 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.

Apparently, only a small amount of information is needed and the distance of each spatial point is very close. This character is quite meaningful in radar communication when ground detection location is limited.

3.4 Result and Analysis

More space points can make the prediction results more accurate, but in practical application, that is, in the process of radar communication, it is difficult for us to collect data in a wide range of space, most of which come from a few points in the center. Therefore, we train this network for different cases of different spatial points number.

For the whole system of equations, we build a comprehensive neural network, which can use the same set of training data to train and predict the coefficients of different equations without interfering with 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 will show 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.

3.4.1 Homogenous Simulation

More space points can make the prediction results more accurate, but in practical application, that is, in the process of radar communication, it is difficult for us to collect data in a wide range of space, most of which come from a few points in the center. Therefore, we train this network for different cases of different spatial points number.

For the whole system of equations, we build a comprehensive neural network, which can use the same set of training data to train and predict the coefficients of different equations without interfering with 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 will show 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.

Table II. Inversion Result with 10 spatial points for each equation from t=500 to 509

Equation	Error without noise	Error with noise $SNR = 65$db
$\frac{\partial E_x}{\partial z} = -\mu_0 \frac{\partial H_y}{\partial t}$	μ_0: 4.32%	μ_0: 4.61%
$-\frac{\partial H_y}{\partial z} = \varepsilon_0 \frac{\partial E_x}{\partial t} + J_x$	ε_0: 14.11%	ε_0: 14.56%
$\frac{\partial J_x}{\partial t} + \nu_c J_x = \varepsilon_0 \omega_p^2 E_x + \omega_{ce} J_z$	ν_c: 40.69%, $\varepsilon_0\omega_p^2$: 3.85%, ω_{ce}: 0.98%	ν_c: 27.15%, $\varepsilon_0\omega_p^2$: 12.9%, ω_{ce}: 0.67%
$\frac{\partial J_z}{\partial t} + \nu_c J_z = \varepsilon_0 \omega_p^2 E_z - \omega_{ce} J_x$	ν_c: 8.70%, $\varepsilon_0\omega_p^2$: 0.20%, ω_{ce}: 1.53%	ν_c: 25.43%, $\varepsilon_0\omega_p^2$: 3.85%, ω_{ce}: 18.10%
$J_z = -\varepsilon_0 \frac{\partial E_z}{\partial t}$	ε_0: 2.50%	ε_0: 2.82%

Table II applies the methodology proposed to a multiple-physics EM interaction system of Equations (24-28) with all coefficients homogeneous. The five PDEs can be obtained simultaneously with five coefficients. For simplicity, we combine $\epsilon_0\omega_p^2$ together as one parameter. In Figure 10, it shows the loss value and accuracy with the training step going. For clearly, the process for recording loss and accuracy are all collected during the last round of Pareto analysis which means in this round, there is no distorting terms at all.

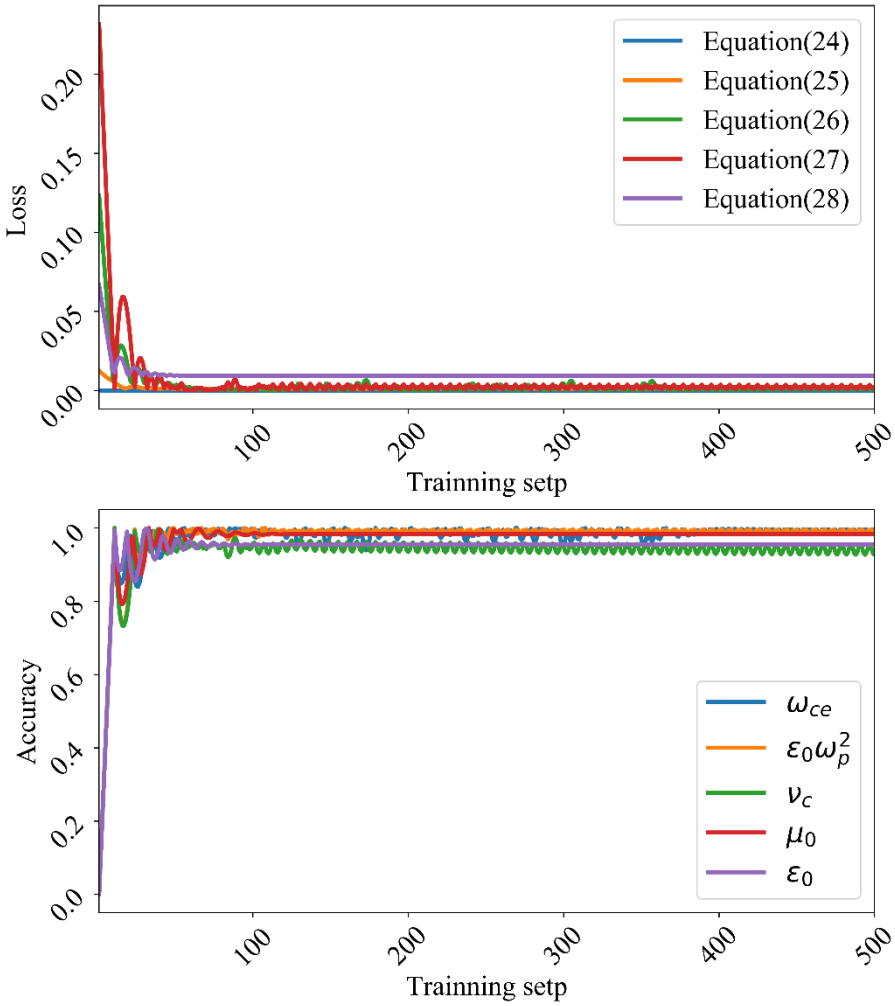

Figure 10. Loss function and accuracy converge with the number of training steps in last round of training. Upper panel shows the trend of loss function for each equation and the lower panel shows the accuracies of each coefficient. As there are some coefficients appears repeatedly in several equations, we select the best results for demonstration. It can be seen that in the last round of training, it converges fast and steadily.

For most coefficients, the relative error is relatively low less than 5%. The equation is also identified with the addition of Gaussian white noise to data set. We add noise with $SNR = 65$db, even though the accuracy reduced, this method can still recognize the effective term for equations and keep a relatively high accuracy at the same time. The method can discover each physical equation even the data was slightly subsampled spatially and temporally.

There are still two coefficients having big error even without noise. The main reason for this problem is that, the universe sampling strategy which is to sample all data at the same spatial location and same time stamp for each equation. If we want to get better accuracy for them, each equation should have the better sampling strategy. Here we analyzed this problem from both spatial and temporal sights.

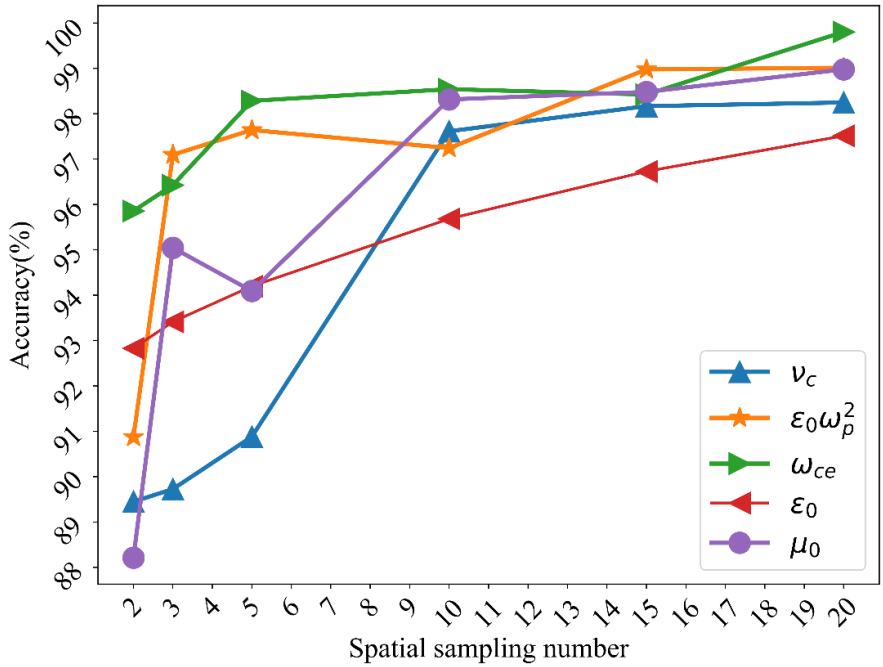

Figure 11. Inversion accuracy dependent on spatial sampling numbers. It is noted that each point indicates the accuracy of normalized parameters. The spatial sampling points are 2, 3, 5, 10, 15, 20, respectively and the beginning location is z = 200. It is noted that inversion accuracy reaches to 95% only for three samples but reduces substantially for two samples.

A. Spatial Sampling Analysis

The data sampling location on time and spatial domains are also one of important infects in training process. By adopting more spatial points we can get better parameters prediction accuracies relatively. As is shown in Figure 11. each point indicates the accuracy (which is $1 - relative\ error\%$) of normalized parameter, we can see that with the increasing of spatial point number the accuracy rises. However,

by considering the spatial sampling limitation, we need to make trade-off between accuracy and spatial sampling scale.

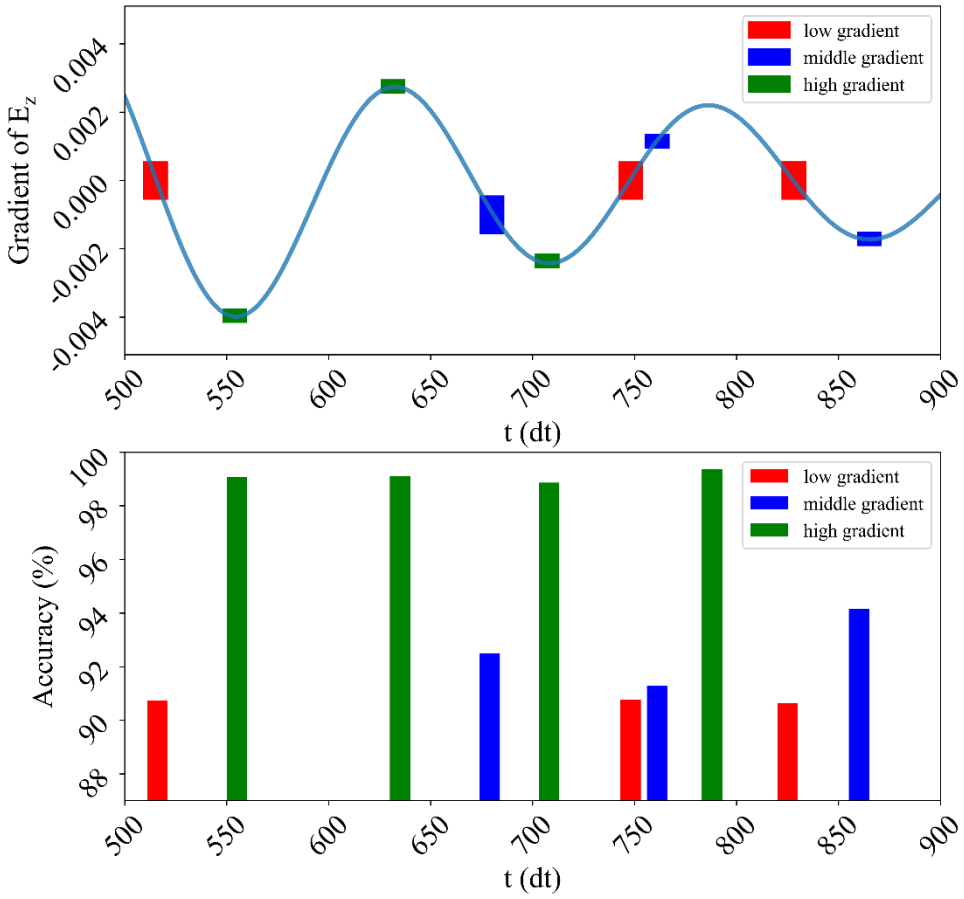

Figure 12. Inversion accuracy dependent on temporal sampling scenario. Three-time sampling blocks are marked for low (red), middle (blue), and high gradient (green) with 10 samples in each block. Here is an example of Equation (28) with the time derivative of E_z at spatial location z = 200. For low 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.

B. Temporal Sampling Analysis

On the other hand, temporal sampling scenario of the field sequence data plays an important role as well. Figure 12. depicts the dependence of accuracy of ε_0 on the gradient of E_z based on Equation (28) 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 12.

In each block, there are 10 temporal samples. It is noted that the accuracy reaches 99% for high gradient (green) and reduces to approximately 90% for low gradient (red). With data varying rapidly with time, the neutral network can catch data characteristics easily. For the PDE equation set, it will require a common time block during which all quantities have high gradients with relatively large amplitude.

C. Improved Sampling Strategy

In Table II, all equations are discovered from a universal sampling strategy, which is relatively good for result generation for all cases. However, if the sampling location for time and space can be customized personalized well for each equation based on the analysis in section A and B shown above, the result can be optimized as the following table.

Here, we keep the size of sampling area and change the sampling location. We try to find better sampling strategy for those equations with bad results.

By adopting the analysis shown above, we find that no matter the equation includes the spatial differentiation, if the temporal sampling area can be chosen at the location with high time gradient, the result turns to be better than before. For equations that need to be optimized, we discuss them one by one individually.

Firstly, we discuss the equation without spatial differentiations as shown in Table III. It can be seen that the optimization of time sampling area can help to get better accuracies. And the space sampling area is already suitable for equations that without spatial differentiations.

Table III. Inversion Result by using improved sampling strategy without noise for Equation (26)(27)(28)

Equation	Before Optimization	After Optimization
$\frac{\partial J_x}{\partial t} + \nu_c J_x = \varepsilon_0 \omega_p^2 E_x + \omega_{ce} J_z$	ν_c: 40.69%, $\varepsilon_0\omega_p^2$: 3.85%, ω_{ce}: 0.98% (z=200-209, t=500-509)	ν_c: 14.43%, $\varepsilon_0\omega_p^2$: 0.32%, ω_{ce}: 0.83% (z=305-314, t=700-709)
$\frac{\partial J_z}{\partial t} + \nu_c J_z = \varepsilon_0 \omega_p^2 E_z - \omega_{ce} J_x$	ν_c: 8.70%, $\varepsilon_0\omega_p^2$: 0.20%, ω_{ce}: 1.53% (z=200-209, t=500-509)	ν_c: 8.83%, $\varepsilon_0\omega_p^2$: 1.79%, ω_{ce}: 0.22% (z=202-211, t=507-516)
$J_z = -\varepsilon_0 \frac{\partial E_z}{\partial t}$	ε_0: 2.50% (z=200-209, t=500-509)	ε_0: 0.90% (z=200-209, t=550-559)

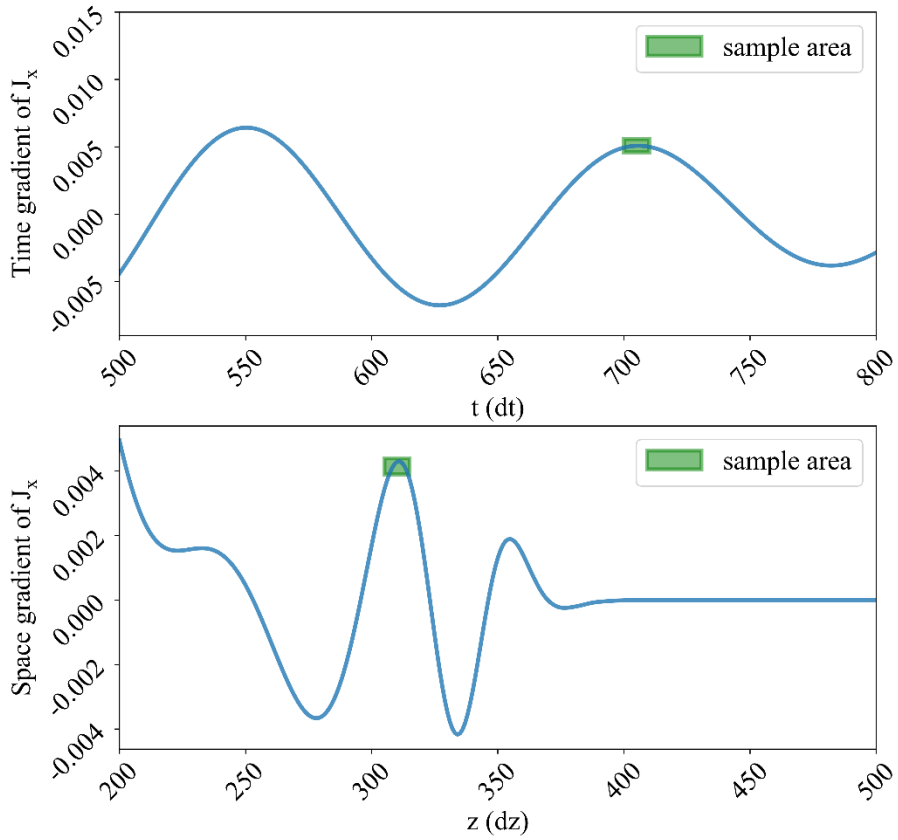

Figure 13. Optimized sample area for Equation (26) where z=305-314, t=700-709.

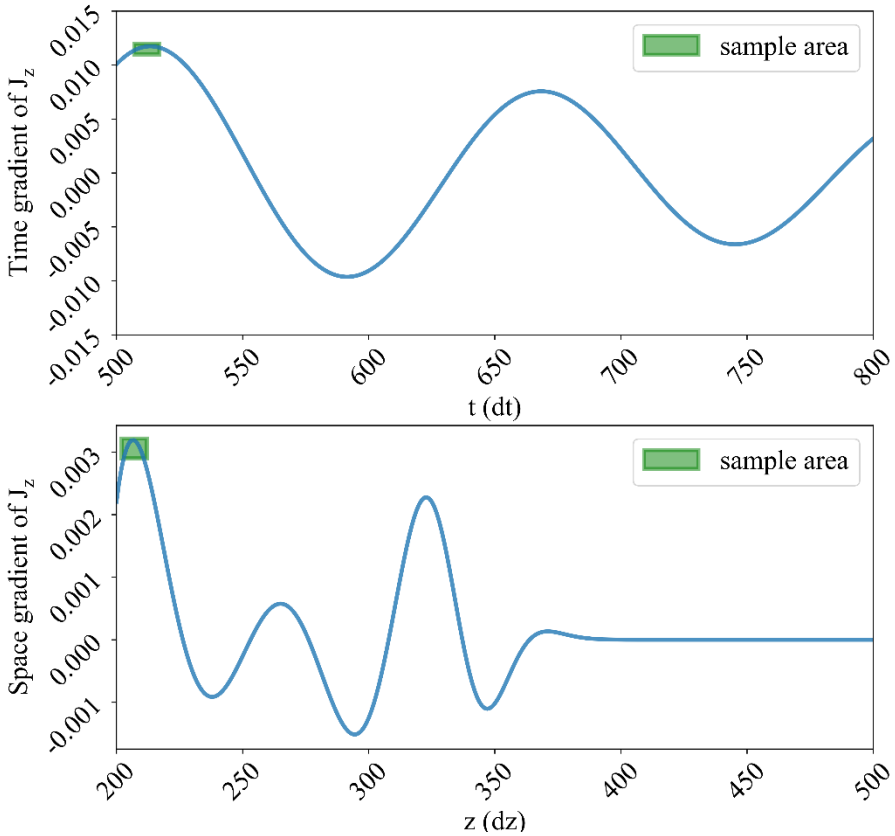

Figure 15. Optimized sample area for Equation (27) where z=202-211, t=507-516.

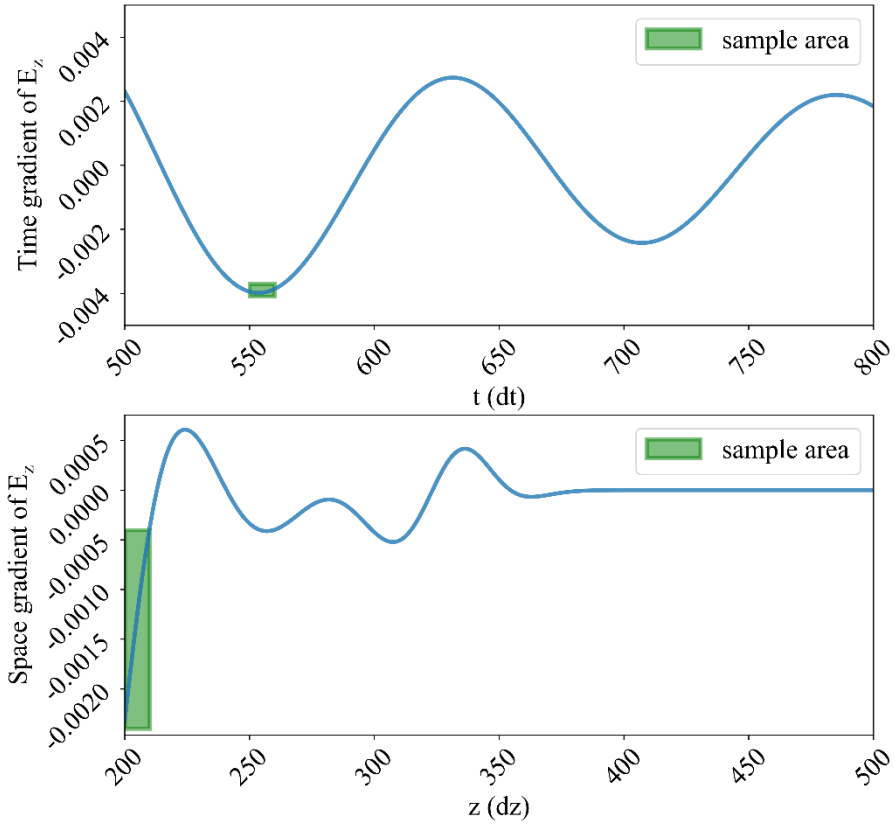

Figure 14. Optimized sample area for Equation (28) where z=200-209, t=550-559.

Table IV. Inversion Result by using improved sampling strategy without noise for Equation (24)(25)

Equation	Before Optimization	After Optimization
$\frac{\partial E_x}{\partial z} = -\mu_0 \frac{\partial H_y}{\partial t}$	μ_0: 4.32% (z=200-209, t=500-509)	μ_0: 2.08% (z=318-327, t=577-586)
$-\frac{\partial H_y}{\partial z} = \varepsilon_0 \frac{\partial E_x}{\partial t} + J_x$	ε_0: 14.11% (z=200-209, t=500-509)	ε_0: 3.31% (z=338-347, t=650-659)

In Table IV, it shows the optimized sample strategy for equations with spatial differentiations. In future work, this kind of equations are more common, so the sampling area selection are quite important for equation like them. By collecting space points around high spatial gradient area, optimized sampling strategy can help to get better accuracy significantly.

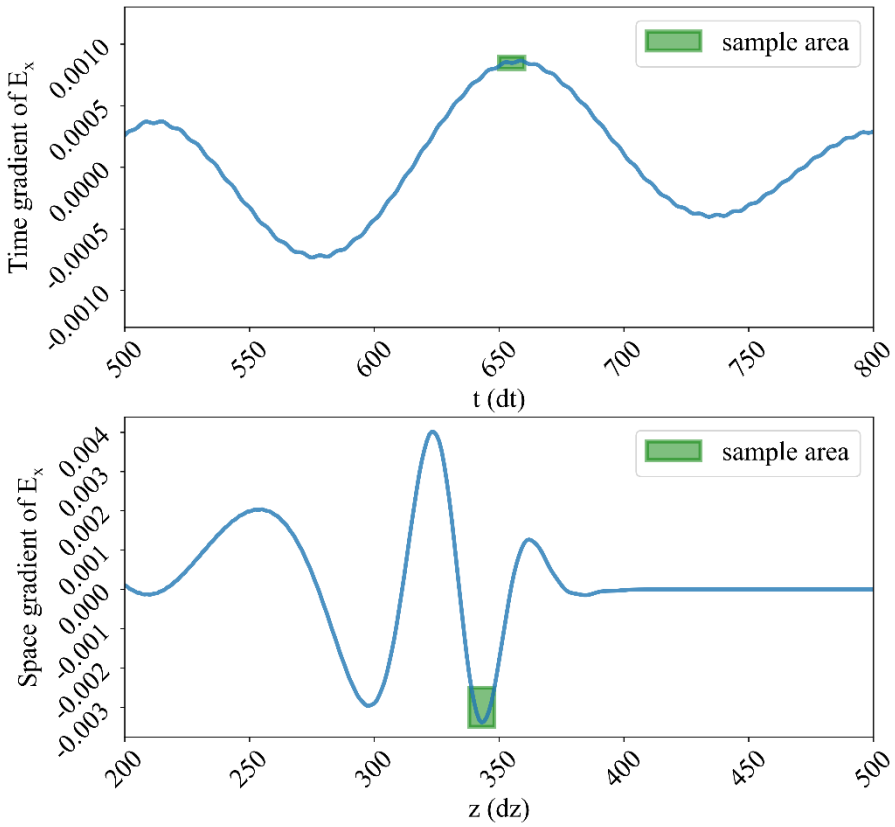

Figure 16. Optimized sample area for Equation (24) where z=318-327, t=577-586.

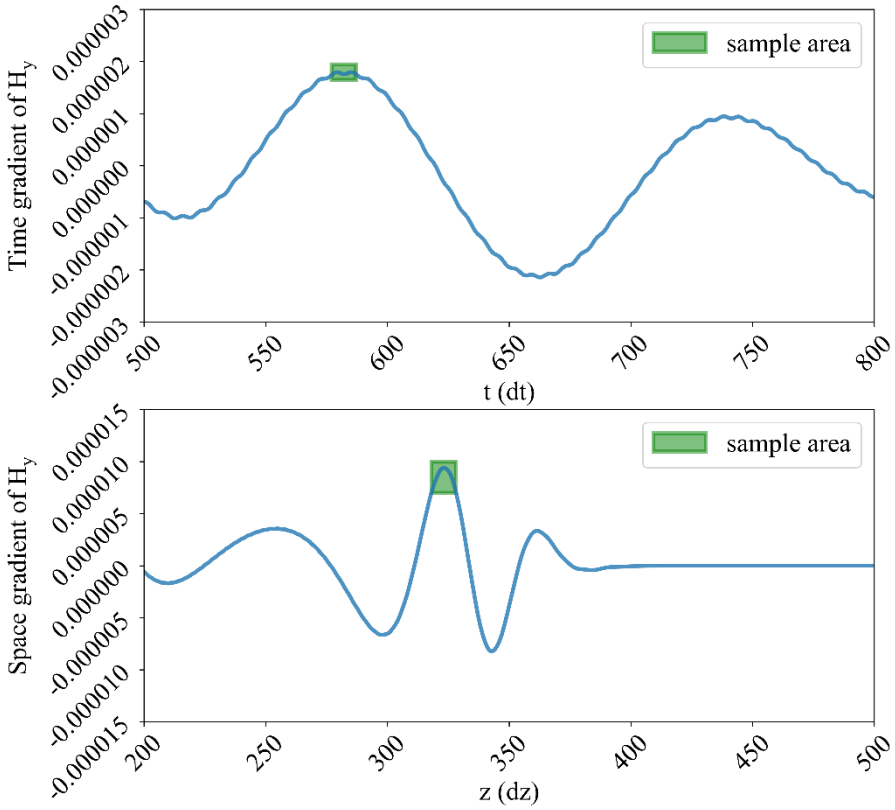

Figure 17. Optimized sample area for Equation (25) where z=338-347, t=650-659.

By using optimized sampling strategy, all equations can obtain better results as following. The cost for this strategy is that the sampling area should be personalized for each equation respectively.

In application problem, a universal sampling strategy can be used as first trail to get to know the basic form of equation and then detailed sampling strategy can be used to get more precise results.

3.4.2 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. Input data of the scattered field for inhomogeneous density is generated by the JEC-FDTD algorithm.

A. Network Adjustment

The network needs to be adjusted. 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 ω_p^2 obeys a sinusoidal envelope, which is positively proportional to plasma density. A normal expression of varying parameter can be written as

$$\omega_p^2 \sim A\sin(kz + \varphi) + C \tag{48}$$

Here, A, k, φ, C is amplitude, wave number, phase and constant, respectively. Plasma parameter ω_p^2 varies from $z = 200$ to $z = 600$ with two wavelengths inside plasmas correspondingly.

B. Data Collection for Inhomogeneous Case

In our case, ω_p^2 is positively correlated with the sine function and this is the prior knowledge we have. Therefore, we can get a normal expression of the changing parameter and Equation (26) containing ω_p^2 can be taken for example.

And ω_p^2 is changed along inside plasma which is z from 200 to 600. We collected 20 spatial points from $z = 240$ to $z = 259$ for training and got the whole predicted distribution inside the plasma. The reason for choosing in this space is that because the prior knowledge shows us that the peak of wave locates here while in more normal cases, the sampling area should be spread in the space of media. For this case, the simulation needs to be done several times and then select the best result, as time limitation, we will try next time.

C. Equation Extraction From Inhomogeneous Parameter

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 [36] in the form of

$$\frac{1}{2}A_0 + \sum_{n=1}^{N}(A_n \sin(nx) + B_n \cos(nx)) \tag{49}$$

in which n is the order of the series, A_n and B_n are coefficients respectively.

As not all modes contain the main power of signal, so we choose that $n = 1,2,3$ is enough to represent the expression of inhomogeneous parameter tensor in our case. Thus, the estimated function of inhomogeneous parameters $\tilde{h}(z)$ is written as

$$\begin{aligned}\tilde{h}(z) = A_1 \sin(kz) + B_1 \cos(kz) + A_2 \sin(2kz) + \\ B_2 \cos(2kz) + A_3 \sin(3kz) + B_3 \cos(3kz) + C\end{aligned} \tag{50}$$

To solve the expression of $\tilde{h}(z)$, we adopt adaptive moments optimizer (Adam) [35] to solve the following sparse regression formula:

$$\arg\min_{\mathbf{S}} \sum \| h(z) - \tilde{h}(z) \|_2 + \lambda \| \mathbf{S} \|_1$$
$$\mathbf{S} = [A_1, B_1, A_2, B_2, A_3, B_3, C] \tag{51}$$

where $A_1, B_1, A_2, B_2, A_3, B_3$ are unknown coefficients to discover, respectively and λ is the sparse coefficient here in this case is set as 0.0075 .

20 spatial points are collected from $z = 240 - 260$ for training and then predict inhomogeneous coefficient in the whole spatial domain. The identified coefficients of $\mathbf{S}$ 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 is 1. The dominant term $A_1 \sin(kz)$ is successfully selected and other terms are small enough to be ignored.

The result is shown in Figure 18. The theoretical value of coefficients in Equations (48) are $A = 0.5, k = \pi, \varphi = 0, C = 1$. And the predicted expression function is shown as following:

$$\omega_p^2 \sim 0.5\sin(3.142z) + 1(\text{True})$$
$$\omega_p^2 \sim 0.5036\sin(3.123z) + 0.9951(\text{Identified}) \tag{52}$$

The result is reliable.

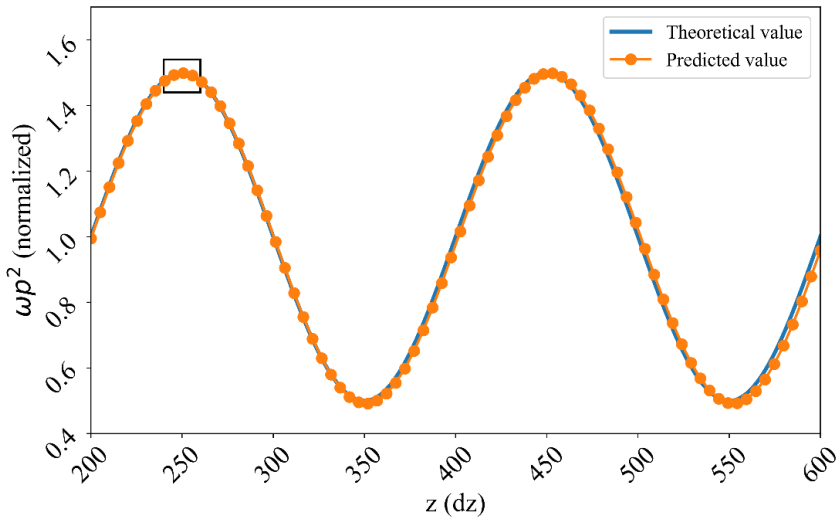

Figure 18. Inhomogeneous parameter prediction result, the black box shows the sampling area, the prediction result is quite good without noise.

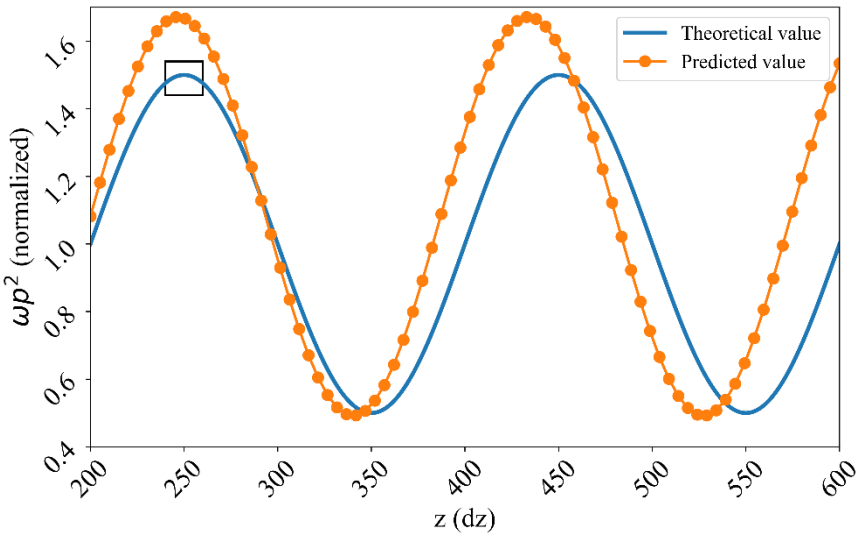

Figure 19. Inhomogeneous parameter prediction with $40db$ noise, it shows that when noise is added, the prediction result turn to worse than cases without noise.

In Figure 18. it shows the inversion result in comparison with theoretical true value by Equation (52). The Sample area is shown by the black box. Although sampling only within few spatial grids, the inhomogeneous coefficient can be obtained in the sampling regime. Basically, the inversion result agrees well with theoretical value after iteration with only 20 spatial and 10 temporal samples. And we also deliver the inversion result with a Gaussian white noise of $40db$ for each input value respectively. The identified model with noise is given by

$$\omega_p^2 \sim 0.5897 \sin(3.360z) + 1.082 (\text{Identified}) \tag{53}$$

Often our training results will have many redundant terms, that is, other candidates also have a weak coefficient. By using Pareto analysis, we can remove these distortion terms effectively.

However, there are still some errors for long distance prediction. There are two possible reasons. One source of error comes from the neural network training of inhomogeneous parameter tensor. Another source of error comes from process during distilling the underlying equation from inhomogeneous parameter tensor.

Even though, this method still gives a prediction of trend the inhomogeneous parameter.

3.5 Summary and Analysis

In this chapter, the forward model for simulation is introduced firstly, the key algorithm to calculate the forward model to get training set is JEC-FDTD. Then, the preprocessing including feature scaling, data collection and network construction are introduced. For data collection, the following simulation sections take a universal sampling strategy (which is $z = 200\ to\ 209,\ t = 500\ to\ 509$) for better demonstration and comparison. Finally, simulations are done in the environment of one-dimensional electromagnetic multi-physics system, the model is simulated and verified for many times according to the different media propagation conditions by using less time and space observation field samples. The results are as follows:

First, under the condition of homogeneous medium propagation, the model can not only discover the governing equations of the system with 10 time and space samples, respectively, but also realize the inversion of the coefficients of the equations with higher accuracy. Similarly, under the condition of inhomogeneous medium propagation, the model also shows excellent inversion performance. In addition, the model is tested for its anti-noise performance under two media propagation conditions.

Second, on the premise that the propagation condition of homogenous medium remains unchanged, the inversion performance of the network structure at different sampling locations is obtained by changing the sampling locations of time and space for several times, and the possible reasons affecting the inversion accuracy of the equation are analyzed, so as to improve and optimize the experimental results. Based on the experimental results obtained in this part, this thesis also discusses the underlying physical causes.

Thirdly, under the condition of inhomogeneous medium propagation, this thesis finds the changing shape of inhomogeneous coefficient by changing the weight scale of neural network, aiming at the problem that the equation coefficient varies with the

spatial scale. By using the properties of trigonometric series and some prior knowledge, the expression of the coefficient of inhomogeneous term is approximated, and satisfactory results are obtained.

Chapter 4 Conclusion and Discussion

4.1 Conclusion

In this thesis, we have presented a data-driven network architecture to discover the hidden nonlinear PDE equation set. The architecture extends the idea from the PDE-net [29] with sparse regression, which approximates differential operations by convolutions with properly constrained filters and approximate the nonlinear response by deep neural networks. By using Pareto analysis during training, we make the equations found as simple as possible with no redundant terms.

Through our experiments and attempts, the practical application of CNN method in the inversion of EM field equations is preliminarily realized. 1D cases on both homogeneous and inhomogeneous problem sets are tested and the results are satisfying. To raise the quality of results, we make simulations on different spatial and temporal sampling area to explore the best sampling strategies for each equation. The results are refined while the analysis is still need to be studied in the future. This analysis will be widely different in other physical systems which is based on the attribute of system itself.

To further verify our method, we generate the problem on inhomogeneous cases. The structure of neural network is adjusted, the weight of inhomogeneous term is extended to certain length which can be equal to the number of spatial sampling points. This method works well on inhomogeneous problem when the expression of coefficient is unknown.

This thesis provides a certain contribution and reference for the development of data-driven modelling from theory to practice. At the same time, the application of this technology in the field of EM field can bring more computing convenience and the expansion of practical application scenarios for radar communication technology.

4.1.1 Results Summary

A. Homogeneous Cases

We firstly add a coefficient sparse regression module to the system to improve the identification accuracy, which is like the role of Pareto analysis. In order to verify the network, we applied this network architecture for a set of PDEs equations for an electromagnetic wave and plasma interaction system.

With 10 temporal points and 10 spatial points sampled, the accuracy of each term's parameter reaches higher than 80% for every target equation. And the system is also robust for noise with SNR= $65db$. Results show that our method can discover unknown equations with remarkably reduced measurements in a stable manner.

To further study the sample area's influence on the inversion accuracy, we design more simulations for each equation. By adopting optimized sampling strategy, which is to find high time gradient and space gradient area as sample area, we raise the inversion accuracy for each equation as following:

1) ν_c: 40.69% → ν_c: 14.43%

2) ε_0: 2.50% → ε_0: 0.90%

3) $\varepsilon_0\omega_p^2$: 3.85% → $\varepsilon_0\omega_p^2$: 0.32%

4) μ_0: 4.32% → μ_0: 2.08%

5) ω_{ce}: 1.53% → ω_{ce}: 0.22%

The results show that by adopting more suitable sample area, the inversion accuracy will be improved, this gives instruction for future applications that multiple sampling strategy is necessary in some way.

B. Inhomogeneous Cases

For the cases of transaction in inhomogeneous media, this method is verified to have the ability for solving the problem well with some prior knowledge. The trigonometric series are applied to approximate the true form of expression of inhomogeneous parameters and sparse regression is adopted to solve the problem.

By collecting 20 spatial points and 10 temporal points the result is quite good shown by figure comparing the predicted value with true value. This method can provide the expression of inhomogeneous parameter which can further express the change of physical coefficient in space. The algorithm is robust to noise with SNR=40db. If totally no prior knowledge in the future, we plan to combine a new candidate function dictionary for function discovery.

4.1.3 To Be Improved

A. Anti-noise Performance

The anti-noise performance of the system still has room for improvement. The following ideas can be referred to.

Firstly, we can improve the sampling position and adopt a variety of sampling schemes for training at the same time. It can not only ensure that the correct form of the equation is screened out, but also ensure that the accuracy of the equation coefficient is high.

Secondly, we can optimize and upgrade the network model, and use high-precision time-difference structure to build the network.

Thirdly, a nonlinear function can be added in candidate function database to improve the fault tolerance of instability. Even though it will introduce a redundant term into the form of equation, it is relatively stable and can be neglected when it works well for keeping the robustness of the system.

B. No Prior Knowledge for Inhomogeneous Problems

For inhomogeneous problem, the prior knowledge is that which term is inhomogeneous term is unknown for us. In reality problem, the prior knowledge is hard to get. The basic solution is to set all weights to the same length with the length of spatial sampling, this method must increase the pressure of computation. Furthermore, a more complex and complete candidate function database can be set for searching the form of expression for inhomogeneous terms.

4.2 Future Work

In the future, we would like to make some progress listed as following.

4.2.1 Arbitrary Inclination Angle and High Order of Differentiation

A. Arbitrary Inclination Angle

We will consider arbitrary inclination angle θ which means that the space is not one dimension at all. To solve this, there is need to add the representation module of curl and high order differentiation operators in the neural network, which is also a very important research direction. The network and algorithm will be upgrade based on new problem setting.

As has been mentioned in Chapter I, for a 2D case, there are relations as following:

$$h_{00}=\frac{1}{4}\begin{pmatrix}1 & 1\\ 1 & 1\end{pmatrix}, h_{10}=\frac{1}{4}\begin{pmatrix}1 & -1\\ 1 & -1\end{pmatrix}, h_{01}=\frac{1}{4}\begin{pmatrix}1 & 1\\ -1 & -1\end{pmatrix}, h_{11}=\frac{1}{4}\begin{pmatrix}1 & -1\\ -1 & 1\end{pmatrix} \quad (54)$$

$$\begin{aligned} h_{00}\otimes u &\approx u,\\ h_{10}\otimes u &\approx \frac{1}{2}\delta_x\frac{\partial u}{\partial x},\\ h_{01}\otimes u &\approx \frac{1}{2}\delta_y\frac{\partial u}{\partial y},\\ h_{11}\otimes u &\approx \frac{1}{4}\delta_x\delta_y\frac{\partial^2 u}{\partial x\partial y}. \end{aligned} \quad (55)$$

where δ_x and δ_y indicates spatial grid on the two directions.

So that in 2D problem, the differentiation operators in convolutional layers can be expressed as above. However, no matter in 1D or higher dimension problems, there may be some high order differentiations like $\frac{\partial^2 u}{\partial x\partial y}$, $\frac{\partial^2 u}{\partial x^2}$ and so on.

Thus, more candidate functions should be considered.

B. High Order of Differentiation

To discover terms like $\frac{\partial^2 u}{\partial x \partial y}$, $\frac{\partial^2 u}{\partial x^2}$ is inevitable in future's research. We need to take more consideration in the assumption for the highest order of spatial differential terms of the equation, mainly by increasing the calculation number of spatial derivatives to determine how many terms of equations need to be found. The network structure is also more complex because more candidate functions need to be considered for screening. Thus, the $\frac{\partial^2 u}{\partial x^2}$ term in 1D problem can be expressed as following

$$h_1 = \frac{1}{2}(-1,2,-1) \tag{56}$$

$$h_1 \otimes u \approx \frac{1}{2}(\delta x)^2 \frac{\partial^2 u}{\partial x^2} \tag{57}$$

For terms like $\frac{\partial^4 u}{\partial x^2 \, \partial y^2}$ in 2D problem, there is relation as following:

$$h_{22} = \frac{1}{16}\begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix} \tag{58}$$

$$h_{22} \otimes u \approx \frac{1}{16}(\delta x)^2(\delta y)^2 \frac{\partial^4 u}{\partial x^2 \, \partial y^2} \tag{59}$$

where δ_x and δ_y indicates spatial grid on the two directions.

Therefore, the convolutional kernel can be modified according to the assumption of the highest order included by the target equation we made.

4.2.2 Real Data Set

Also, it is valuable to try the proposed framework on real data set of nonlinear EM wave and ionospheric plasma interaction experiments [37][38][39]. In the application of practical problems, this topic has a very important application value, whether in the case of too few time and space sampling points, or in the case of inhomogeneous media, or a variety of mixed physics problems environment. There are already some researches on modeling of electromagnetic wave [40], our experiment has a certain reference significance.

4.2.3 Physical Law Discovery System

The original intention of data-driven discovery equation is to replace human beings with machines to discover the laws of nature. Furthermore, with today's increasingly complex data, it is difficult for us to discover the laws of physics through manual calculation and searching for rules like physicists in Newton's time, and the method of first principle derivation is also difficult to be applied in complex physical environment. On the other hand, in the face of some questions have no fixed patterns to follow, for example, financial area, climate prediction, they usually contain large amount of data, there is no fixed patterns, let alone the governing equations. Modeling from data can largely reduce the cost of analyzing these data, can turn the need of large amount of calculation into simple and quick work. We believe that in the future, we will have an AI physicist [41] who has both rigorous thinking, enduring patience and strong memory, making great contributions to human development.

Reference

1. Einstein, A., Podolsky, B., & Rosen, N. (1935). Can quantum-mechanical description of physical reality be considered complete?. Physical review, 47(10), 777.
2. Bongard, J., & Lipson, H. (2007). Automated reverse engineering of nonlinear dynamical systems. Proceedings of the National Academy of Sciences, 104(24), 9943-9948.
3. Schmidt, M., & Lipson, H. (2009). Distilling free-form natural laws from experimental data. science, 324(5923), 81-85.
4. Hochman, H. M., & Rodgers, J. D. (1969). Pareto optimal redistribution. The American economic review, 59(4), 542-557.
5. Smits, G. F., & Kotanchek, M. (2005). Pareto-front exploitation in symbolic regression. In Genetic programming theory and practice II (pp. 283-299). Springer, Boston, MA.
6. Blumer, A., Ehrenfeucht, A., Haussler, D., & Warmuth, M. K. (1987). Occam's razor. Information processing letters, 24(6), 377-380.
7. Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267-288.
8. Candes, E., & Romberg, J. (2007). Sparsity and incoherence in compressive sampling. Inverse problems, 23(3), 969.
9. Baraniuk, R. G. (2007). Compressive sensing. IEEE signal processing magazine, 24(4).
10. Elad, M., & Aharon, M. (2006). Image denoising via sparse and redundant representations over learned dictionaries. IEEE Transactions on Image processing, 15(12), 3736-3745.
11. Mairal, J., Bach, F., Ponce, J., & Sapiro, G. (2009, June). Online dictionary learning for sparse coding. In Proceedings of the 26th annual international conference on machine learning (pp. 689-696). ACM.
12. Brunton, S. L., Brunton, B. W., Proctor, J. L., Kaiser, E., & Kutz, J. N. (2017). Chaos as an intermittently forced linear system. Nature communications, 8(1), 19.

13. Schaeffer, H. (2017). Learning partial differential equations via data discovery and sparse optimization. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 473(2197), 20160446.
14. Rudy, S. H., Brunton, S. L., Proctor, J. L., & Kutz, J. N. (2017). Data-driven discovery of partial differential equations. Science Advances, 3(4), e1602614.
15. Rudy, S., Alla, A., Brunton, S. L., & Kutz, J. N. (2019). Data-driven identification of parametric partial differential equations. SIAM Journal on Applied Dynamical Systems, 18(2), 643-660.
16. Lin, M., Chen, Q., & Yan, S. (2013). Network in network. arXiv preprint arXiv:1312.4400.
17. Mikolov, T., Karafiát, M., Burget, L., Černocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In Eleventh annual conference of the international speech communication association.
18. Zaremba, W., Sutskever, I., & Vinyals, O. (2014). Recurrent neural network regularization. arXiv preprint arXiv:1409.2329.
19. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.
20. Tai, K. S., Socher, R., & Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. arXiv preprint arXiv:1503.00075.
21. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.
22. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).
23. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
24. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).
25. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

26. Raissi, M., & Karniadakis, G. E. (2018). Hidden physics models: Machine learning of nonlinear partial differential equations. Journal of Computational Physics, 357, 125-141.
27. Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2017). Inferring solutions of differential equations using noisy multi-fidelity data. Journal of Computational Physics, 335, 736-746.
28. Baydin, A. G., Pearlmutter, B. A., Radul, A. A., & Siskind, J. M. (2018). Automatic differentiation in machine learning: a survey. Journal of machine learning research, 18(153).
29. Long, Z., Lu, Y., Ma, X., & Dong, B. (2017). PDE-net: Learning PDEs from data. arXiv preprint arXiv:1710.09668.
30. Cai, J. F., Dong, B., Osher, S., & Shen, Z. (2012). Image restoration: total variation, wavelet frames, and beyond. Journal of the American Mathematical Society, 25(4), 1033-1089.
31. Dong, B., Jiang, Q., & Shen, Z. (2017). Image restoration: Wavelet frame shrinkage, nonlinear evolution pdes, and beyond. Multiscale Modeling & Simulation, 15(1), 606-660.
32. Long, Z., Lu, Y., & Dong, B. (2019). PDE-Net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network. Journal of Computational Physics, 399, 108925.
33. Gottlieb, S., & Shu, C. W. (1998). Total variation diminishing Runge-Kutta schemes. Mathematics of computation of the American Mathematical Society, 67(221), 73-85.
34. Zhang, J., Fu, H., & Scales, W. (2018). FDTD analysis of propagation and absorption in nonuniform anisotropic magnetized plasma slab. IEEE Transactions on Plasma Science, 46(6), 2146-2153.
35. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
36. Zygmund, A. (2002). Trigonometric series (Vol. 1). Cambridge university press.
37. Fu, H. Y., Scales, W. A., Bernhardt, P. A., Briczinski, S. J., Kosch, M. J., Senior, A., ... & Ruohoniemi, J. M. (2015, August). Stimulated Brillouin scattering during electron gyro-harmonic heating at EISCAT. In Annales Geophysicae (Vol. 33, No. 8, pp. 983-990). Copernicus GmbH.

38. Fu, H. Y., Scales, W. A., Bernhardt, P. A., Jin, Y. Q., & Briczinski, S. J. (2018). Asymmetry in stimulated emission polarization and irregularity evolution during ionospheric electron gyroharmonic heating. Geophysical Research Letters, 45(18), 9363-9371.
39. Fu, H., & Scales, W. A. (2018, December). Kinetic modeling of stimulated electromagnetic emissions during ionospheric heating experiment. In 2018 12th International Symposium on Antennas, Propagation and EM Theory (ISAPE) (pp. 1-3). IEEE.
40. Zhang, Y., Fu, H., & Xiong, B. (2018, December). Reduced-order modeling methods of electromagnetic wave propagation in magnetized plasmas. In 2018 12th International Symposium on Antennas, Propagation and EM Theory (ISAPE) (pp. 1-4). IEEE.
41. Wu, T., & Tegmark, M. (2018). Toward an AI physicist for unsupervised learning. arXiv preprint arXiv:1810.10525.

Acknowledge

To finish this thesis there are many people offered me treasurable support. First of all, I would like to thank my supervisor Prof. Fu. During the past two years, it was her taught me from a student knowing nothing about machine learning and programming to one equipped with various skills now. Besides, I learned how to process data on different platforms, how to efficiently read international paper and quickly learn from them, how to summarize my opinion into mathematical expression scrupulously. Without her, I cannot manage to do such a tough research almost having no Chinese reference and no predecessor researcher's help, I also have no way to maintain continuous execution to complete such a task with a long time span. In my graduate career, it was her kept encouraging me to move forward when I was frustrated about my experiment result and it was her giving me tolerance and patience when I am lazy and negative from time to time. It can be said that without her there is no my achievements, not to mention this graduation thesis. I want to express my sincere gratitude to her for her efforts.

Then, I want to thank my Finnish supervisor, Professor Ossi Kalevo. When I was studying in Finland, Prof. Kalevo gave me a lot of attention in my life. He was always enthusiastic about my project and dissertation. Without Prof. Kalevo's help, my essay writing style and the completeness of my simulation would be greatly compromised.

And I would also like to thank all the teachers who have given me courses and guidance in the past two years. Without their hard work, my professional level could not be improved and improved. I would also like to thank them for bringing their academic knowledge and methods of academic research to me in and out of the classroom. Their education benefited me all my life.

In the past three years, my parents gave me great support in my life, they supported me from the postgraduate entrance examination to the present graduation. No matter how much trouble I have, or in the face of sadness, they always give me the greatest comfort and encouragement, let me move forward bravely in the face of adversity.

Here I would also like to thank my junior Zhang Yangyang, Gao Yuwen, Yang Ming. With their help, I saved a lot of time for background knowledge and

experimental data preparation, and they also gave me a lot of encouragement and recognition.

Finally, I'd like to thank my girlfriend, Ji Yuehua. In the past, it was she who fought side by side with me, helping me find my way when I was lost, cheering me on when I was depressed, and patiently comforting me when I was bored. She helped me a lot when I met problems doing this research.

I want to thank all the people who have helped me during my graduate years.