PHYSICS-BASED DATA-DRIVEN MODELING OF COMPOSITE MATERIALS AND STRUCTURES THROUGH MACHINE LEARNING

by

Fei Tao

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

School of Aeronautics and Astronautics

West Lafayette, Indiana

May 2022

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Wenbin Yu, Chair

School of Aeronautics and Astronautics

Dr. Vikas Tomar

School of Aeronautics and Astronautics

Dr. Antonio Bobet

Lyles School of Civil Engineering

Dr. Xin Liu

Department of Industrial, Manufacturing, and Systems Engineering, UT Arlington

Dr. Kunming Mao

Dassault Systèmes Simulia Corp.

Approved by:

Dr. Gregory A. Blaisdell

To my parents, Jianhua Tao, and Suqin Chen

ACKNOWLEDGMENTS

Completing this Ph.D. dissertation is a milestone in my life. It is impossible to complete this dissertation without the help of several people.

I would like to express my deep gratitude and appreciation to my supervisor, Prof. Wenbin Yu for his continuous trust, support, and guidance throughout this Ph.D. I still remember how depressed I was when I was looking for Ph.D. opportunities. It was he who gave me a chance and support to continue pursuing my dream. During my Ph.D. study, he has been a mentor, a motivational force, and a supporter of my growth as a researcher and in other aspects of my professional development. His insightful thoughts, profound knowledge, and professional values (truth, unity, balance, and humble boldness) will always inspire me. I feel very blessed to have him as my advisor.

I would also like to thank Prof. Xin Liu for helping me identify the topic of this dissertation. I have been always appreciated his help when I decided to change the research topic during the second year of my Ph.D. I am also thankful for his support and guidance on programming coding and paper writing. He is the one who made all this possible. He is a great mentor and a caring friend.

I want to sincerely thank my esteemed committee members, Prof. Vikas Tomar and Dr. Kunming Mao for their time and constructive suggestions in improving my dissertation, Prof. Antonio Bobet for providing priceless feedback on my doctoral research and patient supervision on my master dissertation.

Funding throughout my study was provided by Composites Design and Manufacturing Hub (cdmHUB), U.S. Army Research Laboratory, and Purdue University. I was also benefited from the internship at Dassault Systems in Fall 2018. I deeply appreciate the support and encouragement from Bob Durbin, Kunming Mao, Yan Xu, and Zhi Yuan.

I would like to acknowledge my lab mates past and present: Fang Jiang, Bo Peng, Hamsasew M. Sertse, Ning Liu, Zhenyuan Gao, Ernesto Camarena, Xin Liu, Orzuri Rique Garaizar, Ankit Deo, Su Tian, Banghua Zhao, Haodong Du, Rong Chiu for their constant encouragement and support. I would specifically like to thank Xin Liu, Su Tian, and Haodong

Du for the productive collaboration, Rong Chiu, Banghua Zhao, Orzuri Rique Garaizar, and Ankit Deo for the insightful discussions.

I want to thank my friends, Amy Getchell, Yuchung Lin, Wan-chih Jiang, and Xunachi Li for having the lunch group with me and discussing new ideas and personal matters, Chen Ma and Yaxiong Chen for their encouragement and accompaniment, especially for the constantly walk and talk along the Celery Bog during the COVID-19 pandemic, Linna Duan and Zhenglyu Lyu for including me in their traveling plans, Yao Sun for having me test his new dishes and taking care of the maintenance of my car. Their accompany and encouragement made my journey at Purdue very pleasant and memorable.

Many thanks to my friends from Church, pastor Michael Lin, Nancy Lin, Mingbo Li, Edward Morrison, Bei Morrison, Shengwei An, Jingyuan Wan, Desheng Xu, Terry Tang, Wenxuan Li, Yujun Leng, Yongxi Wang, Yichen Li, Xinxin Zhang, Edie Yao, Linda Yang, Haozheng Qu, Shiming Yin, and Yi Zhang. Their prayers enlighted my life during the most difficult time in my Ph.D.

I must also thank my significant other, Yuanxin Zhang. Although the pandemic has separated us for almost two years, her love, support, and encouragement have pulled me out of hopelessness and enlightened my path forward. Last but not least, I want to acknowledge my parents Jianhua Tao and Suqin Chen, and my elder brothers Zhaolin Tao, Zhaoqing Tao, Yuanqing Tao, and Zhaoping Tao for their endless family support and unconditional love. I was born in a big family, with five boys in the home. I cannot imagine how challenging it can be to raise so many children. Although the family is big, my parents respect my choice and encourage me to pursue my dream. I always appreciated that they have offered me the best they could provide.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Composite materials have been successfully applied in various industries, such as aerospace, automobile, and wind turbines, etc. Although the material properties of composites are desirable, the behaviors of composites are complicated. Many efforts have been made to model the constitutive behavior and failure of composites, but a complete and validated methodology has not been completely achieved yet. Recently, machine learning techniques have attracted many researchers from the mechanics field, who are seeking to construct surrogate models with machine learning, such as deep neural networks (DNN), to improve the computational speed or employ machine learning to discover unknown governing laws to improve the accuracy. Currently, the majority of studies mainly focus on improving computational speed. Few works focus on applying machine learning to discover unknown governing laws from experimental data. In this study, we will demonstrate the implementation of machine learning to discover unknown governing laws of composites. Additionally, we will also present an application of machine learning to accelerate the design optimization of a composite rotor blade.

To enable the machine learning model to discover constitutive laws directly from experimental data, we proposed a framework to couple finite element (FE) with DNN to form a fully coupled mechanics system FE-DNN. The proposed framework enables data communication between FE and DNN, which takes advantage of the powerful learning ability of DNN and the versatile problem-solving ability of FE. To implement the framework to composites, we introduced positive definite deep neural network (PDNN) to the framework to form FE-PDNN, which solves the convergence robustness issue of learning the constitutive law of a severely damaged material. In addition, the lamination theory is introduced to the FE-PDNN mechanics system to enable FE-PDNN to discover the lamina constitutive law based on the structural level responses.

We also developed a framework that combines sparse regression with compressed sensing, which leveraging advances in sparsity techniques and machine learning, to discover the failure criterion of composites from experimental data. One advantage of the proposed approach is that this framework does not need Bigdata to train the model. This feature satisfies

the current failure data size constraint. Unlike the traditional curve fitting techniques, which results in a solution with nonzero coefficients in all the candidate functions. This framework can identify the most significant features that govern the dataset. Besides, we have conducted a comparison between sparse regression and DNN to show the superiority of sparse regression under limited dataset. Additionally, we used an optimization approach to enforce a constraint to the discovered criterion so that the predicted data to be more conservative than the experimental data. This modification can yield a conservative failure criterion to satisfy the design needs.

Finally, we demonstrated employing machine learning to accelerate the planform design of a composite rotor blade with strength consideration. The composite rotor blade planform design focuses on optimizing planform parameters to achieve higher performance. However, the strength of the material is rarely considered in the planform design, as the physic-based strength analysis is expensive since millions of load cases can be accumulated during the optimization. Ignoring strength analysis may result in the blade working in an unsafe or low safety factor region since composite materials are anisotropic and susceptible to failure. To reduce the computational cost of the blade cross-section strength analysis, we proposed to construct a surrogate model using the artificial neural network (ANN) for beam level failure criterion to replace the physics-based strength analysis. The surrogate model is constructed based on the Timoshenko beam model, where the mapping is between blade loads and the strength ratios of the cross-section. The results showed that the surrogate model constraint using machine learning can achieve the same accuracy as the physics-based simulation while the computing time is significantly reduced.

1. INTRODUCTION

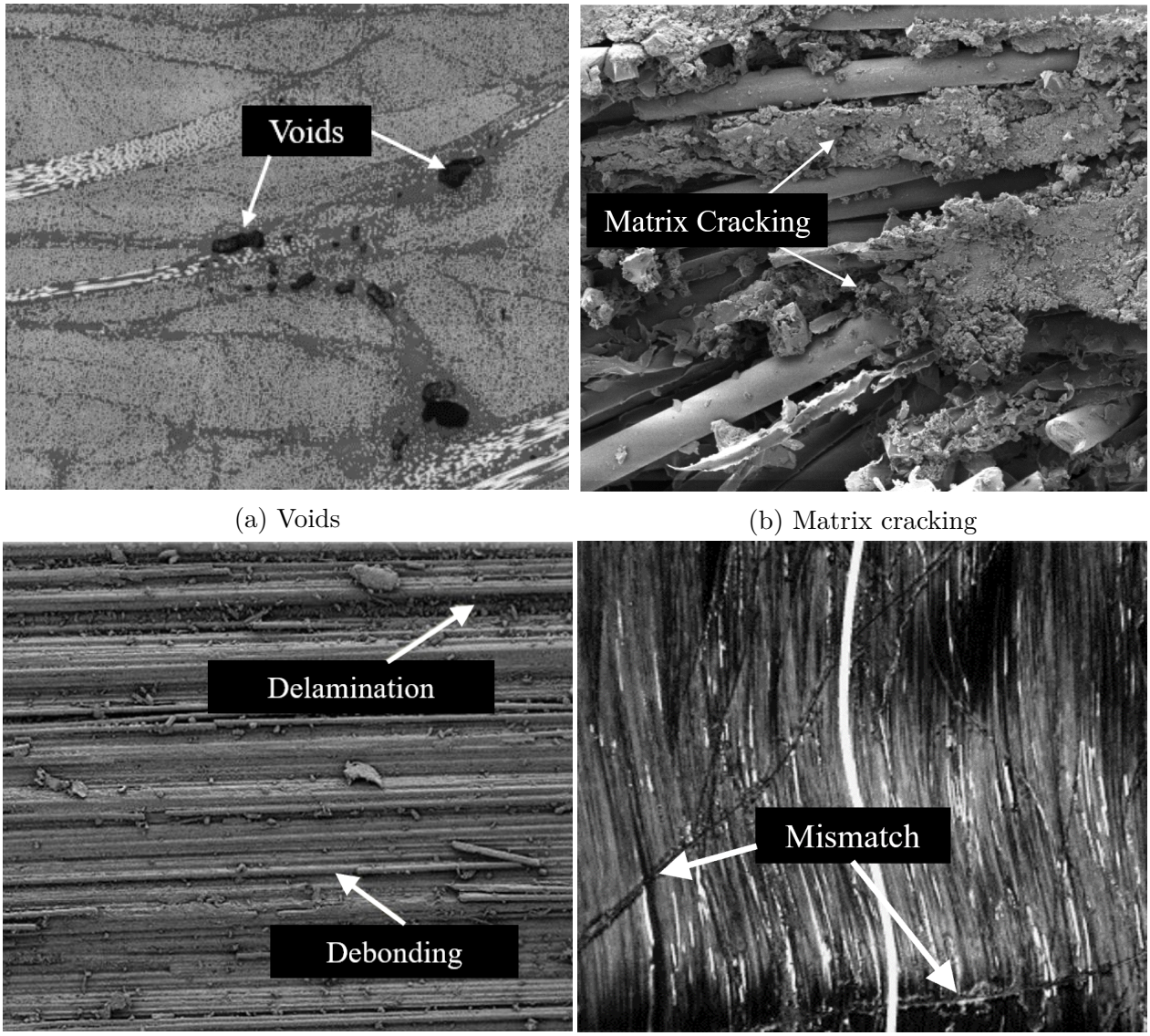
1.1 Background and Motivation

Composite materials have been widely used in various industries, such as aerospace, automobile, and wind turbines, etc due to their high strength/stiffness-to-weight ratio, excellent fatigue life, and great design flexibility. For example, Figure 1.1 shows the Boeing 787 Dreamliner jet. The material content of Boeing 787 is 80% composites by volume [1].

Figure 1.1. Boeing 787 Dreamliner [1]

A composite material is manufactured from two or more individual materials. The individual materials are combined to produce a material with improved properties. These individual materials are usually called constituents of composites. Most composites have two constituents, one is the matrix and the other is the reinforcement [2], [3]. The matrix binds the reinforcement and holds the relative positions of the constituents. Besides, when the reinforcements are broken, the matrix also provides a critical inelastic response so that the stress concentrations are reduced dramatically, and internal stresses are redistributed [4]. The reinforcement receives support from the matrix and provides excellent strength and stiffness to the composites. Some reinforcing materials, such as graphite fibers, nanotubes, SiC fibers, and alumina fibers, may also enable composites to have great thermal and electrical conductivity, controlled thermal expansion, and wear resistance in addition to superior mechanical properties [5]–[7].

Although the material properties of composites are desirable, composites could have the manufacturing defects and is susceptible to damages [8]. The manufacturing defects include voids, matrix cracks fiber breakages, delamination, fiber misalignment, residual stresses, curing and processing tolerances, etc [9]. Figure 1.2 shows the typical manufacturing defects in composites. These defects are related to the manufacturing quality controls and can



(a) Voids

(b) Matrix cracking

(c) Delamination and debonding

(d) Fiber mismatch

Figure 1.2. The typical manufacturing defects of composites [10], [11]

be tackled during the design phase using a safety factor, which is a ratio of a structure's strength to the actual applied load [12]. The damages are produced by accidental or in-

service loads. One of the most dangerous aspects of damage is that a composite structure may accumulate barely visible damage under in-service loads, which could cause the abrupt collapse of a structure. To overcome this issue, some composites structure may take a very conservative design. Usually, some designs only use the elastic region as the design space. Such a design employs failure criteria to determine the safety of a structure under various loading conditions. A number of failure criteria have been proposed to predict the failure of composites. These failure criteria can be classified into microscopic and macroscopic criteria. The microscopic criteria are based on the analysis of the microstructure of a composite. The macroscopic criteria use macroscopic variables, such as averaged stresses or strains, to predict the failure.

In the last couple of decades, many efforts have been made to model the constitutive behavior of composites, i.e. multiscale modeling has been proposed to compute the effective constitutive relation, the continuum damage mechanics (CDM) is developed to describe the damage behavior of composites. The advances of constitutive models enable the design using a smaller safety factor or expanding the design space to the post-failure region, which ultimately could save materials and reduce the cost of design. Although much progress has been made, a complete and validated methodology for modeling the behavior of composite has not been completely achieved yet. This is mainly due to the complex nature of composites, as the performance of a composite structure is dependent on a range of parameters including the geometry, material, lay-up, loading conditions, load history, and failure modes [13].

Recently, the data-driven based machine learning models has been successfully applied in various engineering fields, such as image processing, language translation, and speech recognition [14], [15]. There are several attractions to applying machine learning to mechanics. Firstly, machine learning models, such as deep neural networks (DNN), can be used to construct the mapping between the inputs and outputs of the constitutive relation, which can avoid the computationally expensive iterations and improve the computational speed. Secondly, DNN enables the constitutive law to be learned directly from experimental measurable data. The constitutive law can be constructed in a form-free manner and avoids the possible inaccuracies associated with the presumed functions in the constitutive laws [16]. Thirdly, some machine learning models, such as sparse regression, can be used to discover

governing equations from data, which has the potential to improve the accuracy of existing models [17]. Lastly, machine learning can be used to replace the computationally expensive processes in an analysis to reduce the computational cost.

Currently, the study of machine learning in mechanics mainly focuses on improving the computational speed. There is less research focus on employing machine learning to discover the unknown governing laws. For example, in the constitutive modeling, many researchers used direct data, such as strain-stress pairs or strain-energy pairs, to train a DNN model as a surrogate model to reduce the computational cost. The DNN models trained by these studies are in good agreement with the training data. However, these approaches cannot be used to discover unknown constitutive laws. As the training data for these approaches are generated by numerical simulations, which requires a known constitutive law with certain assumptions [18]. These DNN can only learn the constitutive laws with predefined assumptions. Another issue that prevents researchers to apply machine learning to discover governing laws is related to data size constraint. DNN is powerful to approximate complicated nonlinear functions and being very efficient after being trained. But the training of DNN requires a huge amount of data. Unfortunately, the data size of composites mechanical test is still small compared to Bigdata. This constraint requires researchers to come up with new machine learning methods to deal with the data size issue.

In this study, we will demonstrate applying machine learning to discover unknown governing laws of composites. We will design a framework to couple finite element (FE) with DNN to form FE-DNN to discover constitutive laws based on experimental measurable data. In addition, to solve the data size constraint, we combined sparse regression with compressed sensing, which can be viewed as a machine learning technique, to discover failure criteria of composites from experimental data. Finally, we will show the application of machine learning to accelerate the analysis of composite structures. For demonstration, an example of design optimization of a composite rotor blade was presented.

1.2 Literature Review

1.2.1 Failure criterion of composite materials

A reliable design of a composite structure needs to consider the failure of the composites. A number of failure criteria have been proposed to predict the failure of composites. These failure criteria can be classified into microscopic and macroscopic criteria. The microscopic criteria are based on the analysis of the microstructure of a composite, which requires computing the stresses and failure at the micro-level to predict the macroscopic failure [19], [20]. The macroscopic criteria, which are more popular in engineering practices, assumed that the failure can be described by the macroscopic variables, such as averaged stresses or strains [21].

The macroscopic models can be categorized into three groups. The first group predicts the failure by directly comparing stresses or strains with the corresponding strength. The representative failure criteria are maximum stress or strain failure criteria. However, this approach does not take the interaction between stresses or strains into account, which could result in an inaccurate prediction of the failure for a certain stress state [22]. The second group tries to account for the six stress or strain components in a single function, i.e. Tsai-Hill criterion and Tsai-Wu criterion [23]. The main criticism is that these approaches combine distinct failure mechanisms into a single function, which is non-phenomenological for composite materials [24], [25]. The third group tried to take the distinct failure mechanisms into a failure criterion to form a phenomenological model, which usually separates the fiber and matrix failure mechanisms and has the potential to yield a more accurate prediction of composites failure. Various phenomenological models, such as those proposed by Hashin [25], [26], Puck [27], Cuntze [28], Pinho [29], belong to the third group.

Hashin failure criterion is one of the most popular phenomenological models in engineering practice due to its simplicity of application. It divides failure into four failure modes, namely, tensile and compressive fiber modes, tensile and compressive matrix modes. Although remarkable success has been achieved using the Hashin failure criterion, it does not always fit the experimental results very well. For example, the Hashin failure criterion cannot accurately describe the compressive fiber failure mode of composites. Besides, it should

be noted that the Hashin failure criterion is constructed in a quadratic form. The linear terms, some stresses coupling terms, and higher-order terms are neglected for simplicity. These simplifications are needed when only limited data are available. However, over the past few decades, quite a few experimental failure data have been collected for several composite materials. For example, the two series of Worldwide Failure Exercises (WWFE) have accumulated hundreds of failure data points for some composite materials [30]. Although the data size is still small compared to Bigdata, it would be of interest to leverage the existing data to improve the prediction of composite failure criteria.

Brunton et al. employed a framework that combined sparse regression (SR) with compressed sensing for discovering the governing equations of several nonlinear dynamical systems from data [17]. Rudy further applied sparse regression to discover partial differential equations from time series measurements in the spatial domain [31]. This approach shows promising performance in discovering unknown functions with limited data and exhibits great convergence robustness. However, the approach has not been applied to study the failure behaviors of composites, especially for the real experiential data yet. This inspired us to employ a new data-driven approach to discover the failure criteria of composites.

The failure criteria mentioned above are constructed based on a single composite material. For a composites structure, the multiscale approach is usually taken to analyze the failure of the structure [32]. In the multiscale analysis, a micromechanics model will calculate the effective properties. The structure analysis will take the effective properties and calculate the global response at the upper level. Then, a dehomogenization will compute the local 3D stress fields and analyze the failure at each material point. This procedure is inefficient as a multiscale failure analysis needs to be performed whenever there is a new load case [33]. For a composites rotor blade under various flight conditions, numerous load cases can be accumulated. Unlike the failure criteria of a single material, the analytical form of the failure criterion at the structural level is typically not available. This is because a structure can have various components and complicated geometries. It would be difficult to find an analytical model to predict the failure indicator.

1.2.2 Constitutive modeling of composite materials

Generally speaking, there are two approaches to model the constitutive behavior of composite materials. The first approach is to approximate the heterogeneous composites by an imaginary homogeneous material with effective material properties through multiscale modeling [34], [35]. Numerous analytical micromechanics models have been proposed for the composites constitutive modeling. For example, Voigt and Reuss presented the Voigt and Reuss rules of mixtures, which provide the upper and lower bounds of effective material stiffness respectively [36]. Hashin and Shtrikman developed a stricter bounds for the elastic moduli for various composite materials [37]. Hill presented the self-consistent model, which predicts the effective properties based on the assumption that a single particle inclusion is embedded in the effective medium [38]. Further, the Mori-Tanaka method was proposed to correlate averaged stresses and averaged strains of fiber with the matrix in a composite based on Eshelby elasticity solution for heterogeneous material [39]. The self-consistent model and Mori-Tanaka model are two of the most widely accepted micromechanics models, numerous analytical models were developed later based on them. In addition to analytical methods, a number of numerical methods have been developed to provide a more accurate and universal approach for composites constitutive modeling. For example, the representative volume element (RVE) method which is based on finite element (FE) analysis [40], [41] are widely used in multiscale analysis. A RVE is a material volume that can represent the whole material and should be chosen so that the effective properties will not depend on the boundary conditions. The mathematical homogenization theory is another popular micromechanics method, which works differently than the RVE analysis as it applied the formal asymptotic method through a two-scale formulation [42], [43]. Besides, the mechanics of the structure genome (MSG) method, which is formulated based on the principle of minimum information loss was proposed to unify the analysis of different types of composite material structures [44]–[46]. This method simplifies the multiscale analysis due to the flexibility in applying boundary conditions and superior computational efficiency.

The other approach is to construct a function to approximate the homogenized material behavior with several unknown parameters which will be determined from experiments. The

damage in a composite material causes the degradation of material stiffness. The CDM is developed to describe damage initiation and evolution [47]–[52]. The CDM assumed that the degradation of stiffness can be considered as the macroscopic representation of the micro-defects. The material degradation caused by these micro-defects can be represented by a damage tensor, which is related to different failure mechanisms [53]. Some CDM models were proposed for fiber-reinforced composites by Talreja [54], Matzenmiller [55], Maimi [56], [57], Lapcyzk [58], and Jiang [59]. Although many efforts have been made, it is still difficult and expensive to model the damage in composite material. This is mainly due to the complexity of damage mechanisms of composite materials, as damages in a composite material are presented in multiple scales, from the fiber level through the lamina level to the structural level [60]. In addition, it is impossible to produce identical specimens with well-characterized microstructural features [61].

The application of deep neural networks (DNN) has attracted great interest in constitutive modeling. One of the attractions of this approach is that it can establish a complex nonlinear relationship in a formless manner, which avoids the possible errors associated with the pre-assumed function in the constitutive law [16]. DNN has the potential to reduce the computational cost and discover the unknown constitutive law. At present, the research of DNN in the constitutive modeling mainly focuses on reducing the computational cost [62]–[69]. These studies use direct data, such as strain-stress pairs or strain-energy pairs, to train surrogate models to replace expensive numerical simulations. However, such a model cannot be used to discover unknown constitutive laws, since the data is generated by numerical simulations using assumed constitutive laws.

A few studies have been attempted to use DNN to discover unknown constitutive laws. These studies used an inverse method to solve for the strain-stress pairs based on structural level data, such as forces and displacements which can be directly measured from an experiment. For example, Huang et al. [70] proposed a method to embed DNN in finite element (FE) model to enable DNN to learn the constitutive law based on the experimental measurement. Xu. et al. [71] further introduced positive definite neural networks into the system and improved the robustness of the method. However, these two methods require a full-field observation of the forces and displacements while only limited observed data is

usually available in most experiments. The full field observation requirement prevents the application of this method to complex problems, such as a three-dimensional (3D) solid problem. To enable DNN to discover constitutive laws with limited measurements, Jahromi et al. [72] presented a brute force approach that is based on a trial and error basis to find the constitutive law. Flaschel et al. [73] proposed an unsupervised method to automate the discovery of hyperelastic constitutive law with displacement and global force data.

1.2.3 Composites rotor blade planform design

A rotorcraft, with the rotor powered by engines throughout the flight, allows it to take off and land vertically. Over the past few decades, rotorcraft have demonstrated their versatility in both civilian and military applications [74]. The material and structure of the rotorcraft have a significant impact on the performance. The composite materials are being used more and more extensively in helicopter rotor blades due to their superior mechanical properties, lightweight, and tailorable compared with metals [75]. However, due to the huge design space of the materials and wide variety of flight conditions at which the main rotor is operated throughout the flight envelope. It is difficult to design the rotorcraft components with optimal characteristics [76].

Many works have been done to integrate a multidisciplinary design framework to optimize the design of rotorcraft to improve performance. For example, Leon et al. [77], [78] performed a multi-objective aerodynamic optimization for an ERATO rotor with a CONMIN/elsA/HOST computational workflow via coupling Dakota optimizer with Nash Game algorithm. They studied the influence of blade twist, chord, and sweep on the rotor performance in a forward flight. Wang et al. [79], [80] applied the sequential quadratic programming optimizer SNOPT and conducted a constrained multi-objective optimization of the UH-60A main rotor concerning the design of blade twist, airfoil thickness, and camber. Lim [81] presented a study that fixed the cross-sectional design and conducted parametric studies of the dynamic performance of rotor blades with several planform parameters .

The rotor blade design rarely considers the strength of the materials [32]. One issue is that the failure analysis of the rotor blade is usually performed via a multiscale analysis.

In this analysis, the blade properties will be computed using a finite element analysis over the blade section, followed by the global beam analysis to calculate the global response at the system level to obtain blade loads. Then, the local 3D stress fields will be computed and failure will be evaluated at each material point [82]. This procedure is inefficient in an optimization framework, as the failure analysis needs to be performed each time whenever there is a new structural load. However, the ignoring of strength consideration may result in the blade working in an unsafe or low safety factor region, as composite materials are anisotropic and susceptible to failure.

1.3 Objectives and Outline

The design of composites needs to predict the behavior and evaluate the failure under certain loading conditions. These analyses require failure criterion and constitutive law. The computational cost of these analyses can be expensive. Machine learning has the potential to reduce the computational cost and discover the unknown governing law of composites.

To discover constitutive laws of composite materials and structures based on experiment data, the existing methods need to write a FE code based on an automatic differentiation package, such as Tensorflow or PyTorch. Due to the limitation of the current automatic differentiation packages, the FE model assembling and solving process is significantly time-consuming. Besides, these methods require writing in-house codes to generate FE model geometry and mesh which makes it hard to apply to complex nonlinear FE problems. Another challenge is that the current methods are not stable for severely damaged composite materials. Because these methods did not apply any constraints to the material stiffness matrix predicted by DNN. The diagonal elements of the stiffness matrix might become negative during the training, which will result in non-positive strain energy and thus terminate the training process. In addition, it is expensive to directly apply these methods to learn the lamina constitutive law based on the structural response of laminate. As a composite laminate is typically heterogeneous and manufactured to have multi-directional layups. Learning the lamina constitutive law of a multi-directional laminate with these methods requires creating all the layers explicitly in the FE model. The training of such a model is

similar to perform optimization of a direct numerical simulation. The cost of the training can be extremely expensive.

To address the above issues, we integrated Abaqus with DNN to enables data communication between Abaqus and DNN and makes it possible to use partially observed data to train DNN. This approach leverages the versatile FE analysis ability of Abaqus and the powerful machine learning using DNN. In addition, to solve the training issue of severely damaged material, we coupled FE with Cholesky decomposed positive definite deep neural networks (PDNN) to form FE-PDNN. This modification imposes the positive definite constraint to the stiffness matrix predicted by DNN and thus improves the convergence stability of the training process. To solve the computational cost issue, the lamination theory is introduced to the FE-PDNN mechanics system, which enables FE-PDNN to learn the lamina constitutive law based on the laminate structural response.

For the failure criterion of composites, it is attractive to apply DNN to discover failure criterion from data, as DNN is powerful to approximate complicated nonlinear functions and being very efficient after being trained. However, DNN requires a huge amount of training data. Unfortunately, the data size of composites failure data is still small compared to Bigdata. Besides, DNN also requires the data set of failure index under one, which can only be approximated under certain failure criteria. Therefore, an alternative approach that can accurately capture the failure criterion without the harsh requirement in the training data size is greatly valuable. To discover failure criteria of composites from experimental data, we proposed to combine sparse regression with compressed sensing, which can be viewed as a machine learning technique to discover failure criterion from data. This framework does not need Bigdata to train the model, which satisfies the current failure data size constraint. Besides, unlike traditional curve fitting, which results in a solution with nonzero coefficients in all the candidate functions, this framework can identify the most important functions of a failure criterion from the data [17]

To reduce the cost of the the failure analysis of composite structures, machine learning technique can be a useful tool to construct the structured level failure criterion to reduce computational cost. For this procedure, conducting simulation to generate the training data is one of the most important steps. In this study, we will use VABS [83] to perform the

multiscale failure analysis to generate the training database. The process to generate data is very similar to conduct a regular multiscale failure analysis, except that we will use a high-dimensional surface sampling technique to generate various combinations of the structural loads.

In summary, in this study, we proposed to

1. Implement FE-DNN via coupling Pytorch and the commercial FE code Abaqus to learn nonlinear constitutive law based on limited experimental data

2. Apply positive definite deep neural network (PDNN) to the proposed mechanics system to form FE-PDNN so that the robustness of the mechanics system can be improved.

3. Couple the classical lamination theory with DNN to learn the constitutive law of lamina based on laminate surface observations.

4. Combine sparse regression with compressed sensing to discover failure criterion from experiment data.

5. Construct structure level failure criterion to accelerate the failure analysis of composite structures.

The remainder of this thesis is organized as follows. Chapter 2 introduces coupling FE with DNN to discover the constitutive law of composites. In Chapter 3, we will implement a framework to combine sparse regression with compressed sensing to discover failure criteria of composites from experimental data. Next, in Chapter 4, we will demonstrate the application of machine learning model to construct a beam-level failure criterion surrogate model, which will be integrated with Dakota to achieve rotor blade optimization with strength consideration. Chapter 5 highlights the main outcomes of this thesis and outlines the future works of this study.

2. FINITE ELEMENT COUPLED DEEP NEURAL NETWORK MECHANICS SYSTEM

This chapter presents a finite element (FE) coupled deep neural networks mechanics system (FE-DNN). The proposed approach enables neural networks to learn the constitutive law based on the structural level response, such as forces and displacements. To leverage the versatile modeling capability of commercial FE codes, we integrated the commercial FE code Abaqus with DNN, which enables data communication between Abaqus and DNN. To solve the convergence robustness issue of learning the constitutive law of a severely damaged material, we applied a positive-definite constraint to neural networks to form FE-PDNN. In addition, the lamination theory is introduced to the FE-PDNN mechanics system to enable FE-PDNN to discover the lamina constitutive law based on the structural level responses.

2.1 Theory and Fundamentals

2.1.1 Fundamentals of deep neural networks

The basic idea of neural networks is to use multiple neurons to describe a complex system. An artificial neural networks model is called a deep neural networks (DNN) if the number of hidden layers is larger than two. As shown in Figure 2.1, neural networks usually consist of an input layer, hidden layer(s), and an output layer. The input layer is the first layer of the neural networks, it accepts the input values and passes them to the next layer. The hidden layer(s) perform nonlinear transformations of the inputs that entered the networks. The number and sizes of hidden layer(s) are hyperparameters, which will be determined based on the complexity of the problem. The output layer is the last layer of the network and gives the predicted values of the model [84].

The trainable parameters, weights (w_{ij}) and biases (b_j), are used to establish the connections between neurons. The weight decides how much influence the input will have on the output. The bias, which is an additional input into the next layer, is used to tune the activa-

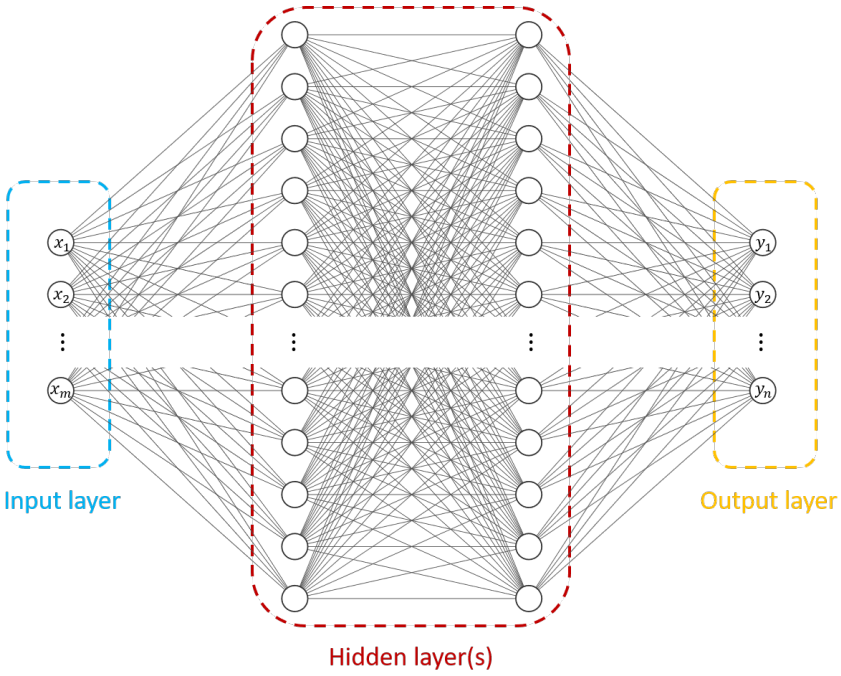


Figure 2.1. The structure of artificial neural networks

tion function [85]. The outputs from preceding neurons (a_i) will multiply the corresponding weights. Summing these values and adding the bias of that neuron gives

$$z_j = \sum_{i=1}^{n} a_i w_{ij} + b_j \tag{2.1}$$

where n is the number of neurons in the previous layer, the z_j will be passed onto the activation function, which is a nonlinear function and results in a scalar value for the neuron. The scalar will be the input of the following layer. In this work, the choice of the activation function is the rectified linear unit function (ReLU), which has the form:

$$g(z) = \max(z, 0) \tag{2.2}$$

ReLU is plotted in Figure 3.10.

Besides, the loss function is expressed in the mean square error (MSE) form

$$L = \frac{1}{2n} \sum_{i=1}^{n} \left(y^{(i)} - \hat{y}^{(i)} \right)^2 \tag{2.3}$$

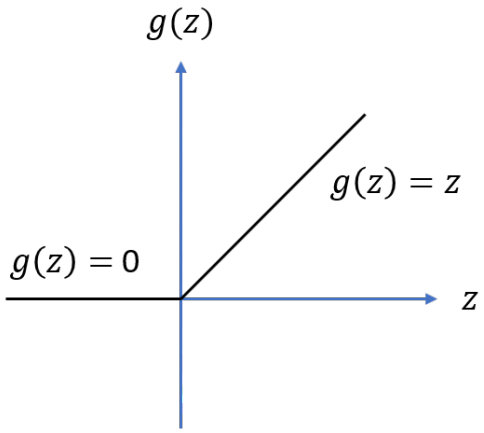


Figure 2.2. The plot of ReLU function

where n is the number of training samples, $y^{(i)}$ is the predicted result by the neural networks model and $\hat{y}^{(i)}$ is the exact result of the training samples. To find the minimum of the loss function, the weights and biases can be updated as

$$\begin{aligned} w_{ij}^{'} &= w_{ij} - \eta \frac{\partial L}{\partial w_{ij}} \\ b_{j}^{'} &= b_{j} - \eta \frac{\partial L}{\partial b_{j}} \end{aligned} \tag{2.4}$$

where $w_{ij}^{'}$ and $b_{j}^{'}$ are the corresponding new weights and bias, w_{ij} and b_j are the old weights and bias, η is the learning rate which is a hyperparameter that dominates the updated amount in response for estimated error in each optimization step [86]. The loss gradient in Eq. (2.4) can be determined with the backward propagation equations. The backward propagation equations are [85]

$$\begin{aligned} \boldsymbol{\delta}^{L} &= \left(\boldsymbol{a}^{L} - \boldsymbol{y}\right) \odot \boldsymbol{g}'\left(\boldsymbol{z}^{L}\right) \\ \boldsymbol{\delta}^{l} &= \left(\left(\boldsymbol{w}^{l+1}\right)^{T} \boldsymbol{\delta}^{l+1}\right) \odot \boldsymbol{g}'\left(\boldsymbol{z}^{l}\right) \\ \frac{\partial L}{\partial b_{j}^{l}} &= \delta_{j}^{l} \\ \frac{\partial L}{\partial w_{jk}^{l}} &= a_{k}^{l-1} \delta_{j}^{l} \end{aligned} \tag{2.5}$$

where δ is the error of the layer, the superscript L denotes the output layer and l denotes the hidden layers, $\boldsymbol{g}(\cdot)$ is the activation function. The bold symbols in Eq. (2.5) indicate the variables in the matrix form. The $\odot$ represents the element-wise product of two vectors.

2.1.2 Framework of FE-DNN mechanics system

DNN is powerful to learn the intrinsic laws from existing data. However, there are several issues that limit the application of DNN to learn constitutive law. First, the constitutive law of a material is typically expressed in terms of stress and strain. But the experiment can only provide the load-deflection curve. The inhomogeneous stress and strain data are not directly measurable from experiment, which indicates the direct input and output of DNN are not available from experiment data. Second, the training of DNN is an unconstrained optimization process, which causes the issue that the constitutive law learned by DNN cannot be guaranteed to comply with the physics laws. For example, when applying DNN to construct the relationship between strain and stress, the learned stress might not follow the equilibrium equations.

The FE was proposed to couple with DNN to form a mechanics system to solve these issues. The proposed system can be described using Figure 2.3. The $\boldsymbol{I}^s$ and $\boldsymbol{O}^s$ represent the system input and output. The system input and output can be directly measured from experiments, such as the displacement and force in a tensile test. The FE in Figure 2.3 refers to the process of solving the FE model. The Subsystem in the figure can be an analytical equation, a filter, or can be ignored for some scenarios. The choice of the Subsystem is problem dependent. The direct input and output of the DNN model are represented by the i_k and o_l. The input i_k needs to be derived from Subsystem or FE or both FE and Subsystem. There are two ways to transfer the information from DNN to FE. One method is to use DNN to predict the material stiffness matrix, the predicted stiffness matrix will be used to form the global stiffness matrix in FE. The other method is to use DNN to predict the strain and stress relationship, then the Jacobian of the stress and strain will be computed via automatic differentiation method and transferred to FE. FE will solve the model and

output the system output $\boldsymbol{O}^s$ which will be used to construct the loss function. The loss function is the objective function and will be minimized through an optimizer.

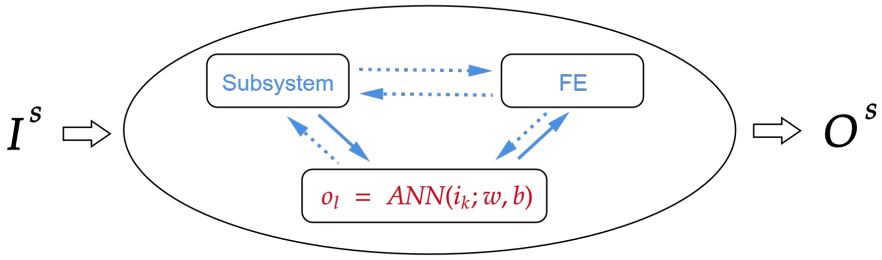


Figure 2.3. The framework of the FE-DNN mechanics system

For this mechanics system, the simplest scenario is an uncoupled problem represented by the solid blue line in the figure. The input data (i_k) of the DNN model can be obtained directly from the system input ($\boldsymbol{I}^s$) through the Subsystem. The outputs of the DNN model (o_l) are sent to FE to compute the system output ($\boldsymbol{O}^s$) . For example, using this approach to learn the linear elastic constants of general anisotropic materials. If the output of the DNN is the engineering constants, this system becomes an uncoupled system. This is because the DNN output is constant for this scenario, as the engineering constants (o_l) is independent of DNN input (i_k).

A more complicated and interesting scenario for this system is to deal with a fully coupled problem. For example, for nonlinear problems, the constitutive law is expressed as the mapping between strains (input) and stresses (output) via DNN. The strains have to be derived using the system input with Subsystem, FE, and DNN, as solving the strains requires using the constitutive equations that are represented by the DNN model. Thus, the determining of DNN input also depends on the output of DNN. The DNN model is fully coupled with the entire mechanics system.

The proposed system has several benefits. Firstly, this system enables DNN to learn a constitutive law based on limited indirect data, which can be easily measured from experiments. Secondly, the constitutive law is learned in a form-free way that avoids the accuracy loss caused by the presumed function forms of constitutive laws. Thirdly, since FE is coupled

to compute the DNN outputs, the learned outputs automatically satisfy equilibrium and the kinematics equations, which guaranteed that the learned result follows physical laws. Finally, it is possible to replace the FE a with commercial code, such as Abaqus. This change avoids rewriting the FE solver based on the automatic differentiation package and facilitates the creating of the FE model by using powerful commercial codes.

2.1.3 Theory to couple FE with deep neural networks

Given a reference domain $\Omega \in \mathbb{R}$, with boundary Γ_u and Γ_t, and subject to static load. Assuming negligible body forces, known traction boundary condition, and virtual displacement $\delta\mathbf{u}^{\mathrm{T}}$ vanishing on Γ_u. The virtual work expression of the equation of equilibrium is given by [87]

$$\int_{\Omega} \delta\boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\sigma}\mathrm{d}\Omega - \int_{\Gamma_t} \delta\mathbf{u}^{\mathrm{T}}\bar{\mathbf{t}}\mathrm{d}\Gamma_t = 0 \tag{2.6}$$

where $\delta\boldsymbol{\varepsilon}^{\mathrm{T}}$ is the virtual strain, $\boldsymbol{\sigma}$ is stress, and $\bar{\mathbf{t}}$ is traction. Defining all integrals to be summed over individual elements, we can approximate the weak form as

$$\sum_{e} \int_{\Omega^e} \delta\boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\sigma}\mathrm{d}\Omega - \sum_{e} \int_{\Gamma_t^e} \delta\mathbf{u}^{\mathrm{T}}\bar{\mathbf{t}}\mathrm{d}\Gamma = 0 \tag{2.7}$$

where Ω^e and Γ_t^e represent element domains and parts of boundaries of elements where tractions are specified, e is the total number of elements. The approximation in Eq. (2.7) is associated with the fact that the the sum of element domains Ω_e is not always identical to Ω. With the finite element shape function N_a, the displacements can be expressed as

$$\boldsymbol{u} = \sum_{a} N_a \tilde{\mathbf{u}}_a \tag{2.8}$$

where a is the number of nodes for each element, $\tilde{\mathbf{u}}_a$ is the nodal values of the displacement. The strain can be approximated as

$$\boldsymbol{\varepsilon} = \boldsymbol{\mathcal{S}}\boldsymbol{u} = \sum_{a} \mathbf{B}_a \tilde{\mathbf{u}}_a \tag{2.9}$$

where $\boldsymbol{\mathcal{S}}$ is a differential operator, $\mathbf{B}_a$ is the deformation matrix. With the above approximations, Eq. (2.7) can be expressed as

$$\sum_e \delta\tilde{\mathbf{u}}_a^{\mathrm{T}} \left[\int_{\Omega^e} \mathbf{B}_a^{\mathrm{T}} \boldsymbol{\sigma} \mathrm{d}\Omega - \int_{\Gamma_t^e} N_a \bar{\mathbf{t}} \mathrm{d}\Gamma \right] = 0 \tag{2.10}$$

Next, we need to define the constitutive law in terms of neural networks models to solve Eq. (2.10). We can express the constitutive model as:

$$\mathcal{M} = \mathrm{NN}(\boldsymbol{\varepsilon}; \boldsymbol{\theta}) \tag{2.11}$$

where NN represents deep neural networks, $\boldsymbol{\varepsilon}$ is the strain, $\boldsymbol{\theta}$ represents weights and biases of neural networks. DNN is trained to build the mapping between strain and the stiffness matrix $\mathcal{M}$. Clearly, $\mathcal{M}$ is a function of $\boldsymbol{\varepsilon}$ and $\boldsymbol{\theta}$. The constitutive relation can be expressed as

$$\boldsymbol{\sigma} = \mathcal{M}(\boldsymbol{\varepsilon}; \boldsymbol{\theta}) \boldsymbol{\varepsilon} \tag{2.12}$$

Substituting Eq. (2.12) into Eq. (2.10) yields

$$\sum_e \delta\tilde{\mathbf{u}}_a^{\mathrm{T}} \left[\int_{\Omega^e} \mathbf{B}_a^{\mathrm{T}} \mathcal{M}(\boldsymbol{\varepsilon}; \boldsymbol{\theta}) \boldsymbol{\varepsilon} \mathrm{d}\Omega - \int_{\Gamma_t^e} N_a \bar{\mathbf{t}} \mathrm{d}\Gamma \right] = 0 \tag{2.13}$$

Plugining Eq. (2.8) into Eq. (2.13) results in

$$\sum_e \delta\tilde{\mathbf{u}}_a^{\mathrm{T}} \left[\int_{\Omega^e} \mathbf{B}_a^{\mathrm{T}} \mathcal{M}(\boldsymbol{\varepsilon}; \boldsymbol{\theta}) \mathbf{B}_a \tilde{\mathbf{u}}_a \mathrm{d}\Omega - \int_{\Gamma_t^e} N_a \bar{\mathbf{t}} \mathrm{d}\Gamma \right] = 0 \tag{2.14}$$

After summing the element integrals and noting that $\delta\tilde{\mathbf{u}}_a$ is arbitrary, then in each increment step, Eq.(2.14) can be regarded as a system of linear equation

$$\mathbf{K}\mathbf{u} - \mathbf{f} = \mathbf{0} \tag{2.15}$$

with

$$\mathbf{K} = \sum_e \int_{\Omega^e} \mathbf{B}_a^{\mathrm{T}} \mathcal{M}(\boldsymbol{\varepsilon}; \boldsymbol{\theta}) \mathbf{B}_a \, \mathrm{d}\Omega \quad \text{and} \quad \mathbf{f} = \sum_e \int_{\Gamma_t^e} N_a \bar{\mathbf{t}} \mathrm{d}\Gamma \tag{2.16}$$

The unknown displacement $\mathbf{u}$ can be solved

$$\mathbf{u} = \mathbf{K}^{-1}\mathbf{f} \tag{2.17}$$

Then the loss function can be formed as

$$\boldsymbol{L}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i}^{N}\left(\mathbf{u}_i\left(\boldsymbol{F}_i, \mathcal{M}(\boldsymbol{\varepsilon};\boldsymbol{\theta})\right) - \hat{\mathbf{u}}_i\right)^2 \tag{2.18}$$

where $\boldsymbol{L}$ is the system loss function, $\mathbf{u}_i(\cdot)$ and $\hat{\mathbf{u}}_i$ are the computed displacements and observed displacements respectively, N is the total number of measurable datasets, $\boldsymbol{F}_i$ is the external force applied to the FE model at the ith step.

2.1.4 Training scheme of FE-DNN mechanics system

Since the measurable data from experiments are forces and displacements $(\boldsymbol{F}_i, \hat{\mathbf{u}}_i)$. Then, the training of the DNN can be mathematically formulated as a PDE-constrained optimization problem:

$$\begin{aligned}
&\text{find} \quad \boldsymbol{\theta} \\
&\text{minimize} \quad \boldsymbol{L}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i}^{N}\left(\mathbf{u}_i\left(\boldsymbol{F}_i, \mathcal{M}(\boldsymbol{\varepsilon};\boldsymbol{\theta})\right) - \hat{\mathbf{u}}_i\right)^2 \\
&\text{subject to} \quad \mathcal{N}(\boldsymbol{\theta}, \mathbf{u}_i(\boldsymbol{\theta})) = 0
\end{aligned}$$

where $\mathcal{N}(\cdot)$ represents the PDE constraint. The data flow of this framework can be summarized in Figure 2.4.

The optimization of the loss function $\boldsymbol{L}$ is to find a set of weights and biases $\boldsymbol{\theta}$ of DNN that minimizes the loss function and satisfies the PDE constraint. To update weights and biases of DNN in the system, the backward propagation equations need to be modified. With the data flow shown in Figure 2.4, the parameters of the DNN model can be updated as

$$\frac{\partial \boldsymbol{L}}{\partial \boldsymbol{\theta}} = \frac{1}{N}\sum_{i=1}^{N} 2\left(\mathbf{u}_i - \hat{\mathbf{u}}_i\right)\left(\sum_{m=1}^{M}\frac{\partial \mathbf{u}_i}{\partial \mathcal{M}_m}\frac{\partial \mathcal{M}_m}{\partial \boldsymbol{L}_c^{(m)}}\frac{\partial \boldsymbol{L}_c^{(m)}}{\partial \boldsymbol{\theta}}\right) \tag{2.19}$$

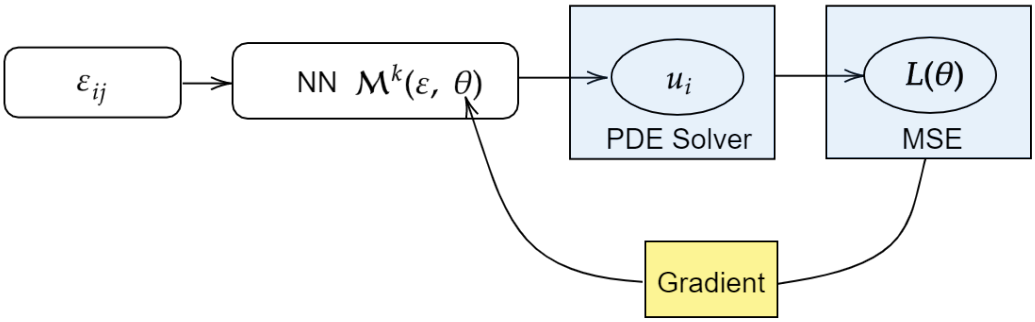


Figure 2.4. The data flow of the FE-DNN mechanics system

where M is the total number of outputs of DNN that contributed to system output, $\boldsymbol{L}_c^{(m)}$ is the mth DNN output, and $\mathcal{M}_m$ is the mth output of the stiffness matrix. In Eq. (2.19), the $(\mathbf{u}_i - \hat{\mathbf{u}}_\mathbf{i})$ can be computed analytically, the $\partial \mathbf{u}_i / \partial \mathcal{M}$ involves solving FE model, it can be computed via automatic differentiation or finite difference, and $\frac{\partial \mathcal{M}_m}{\partial \boldsymbol{L}_c^{(m)}} \frac{\partial \boldsymbol{L}_c^{(m)}}{\partial \boldsymbol{\theta}}$ can be handled by the automatic differentiation package.

For an FE analysis, the computational complexity is $O(NW^2)$, where N is the number of nodes, W is the bandwidth of a banded stiffness matrix [88]. Since we need to compute the gradient with respect to stiffness matrix entries, the computational complexity is $O\left(N^2W^2\right)$. For a neural networks model, the backward propagation complexity is estimated as $O(\omega m)$, where ω is the number of weights, m is the number of inputs [89]. Moreover, for each time step, the computational complexity of LSTM per weight is O(1). Thus, the overall complexity per step is $O\left(\omega m + \ N^2W^2\right)$.

The previous research has shown that the second-order/quasi-second-order optimizer works better for the FE coupled neural networks problems [70], [71], [90]. Thus, in this work, the Limited-memory BFGS (L-BFGS) optimizer [91], which is a quasi-second-order optimizer, was used to train the FE-DNN model. A brief introduction of L-BFGS optimizer is presented in Appendix A.1.

A two-round training scheme was proposed to train the FE-DNN mechanics system. The training process is shown in Figure 2.5. The purpose of the first-round training is to deter-

mine accurate inputs and outputs from indirect experiment data. The first-round training will be performed incrementally. The experimental data, i.e. the load and corresponding displacements, will be divided into n sets, with n as the number of the total training steps of the first-round. The choice of n needs to consider convergence and performance of the entire system. As a too small n will make the training inefficient, a too large n may cause the neural network to lose convergence. At each step, the initialized weights, biases, and neural network inputs are obtained from the previous step. The load and DNN predicted constitutive law will be transferred into the FE model. FE solver will solve the FE model and compute the gradient of the displacement with respect to the constitutive law parameters. Next, the difference between the observed displacements and computed displacements will be evaluated. If the difference does not reach the termination criterion, the weights and biases will be updated through an optimizer. Otherwise, the DNN inputs, outputs, and model parameters will be saved for the second-round training and transferred to the next step as initial parameters. By repeating the first-round training for all the datasets, the complete sets of inputs and outputs of DNN can be obtained. After the completion of the first-round training, the second-round training will be performed to approximate the overall relationship between the accurate inputs and outputs.

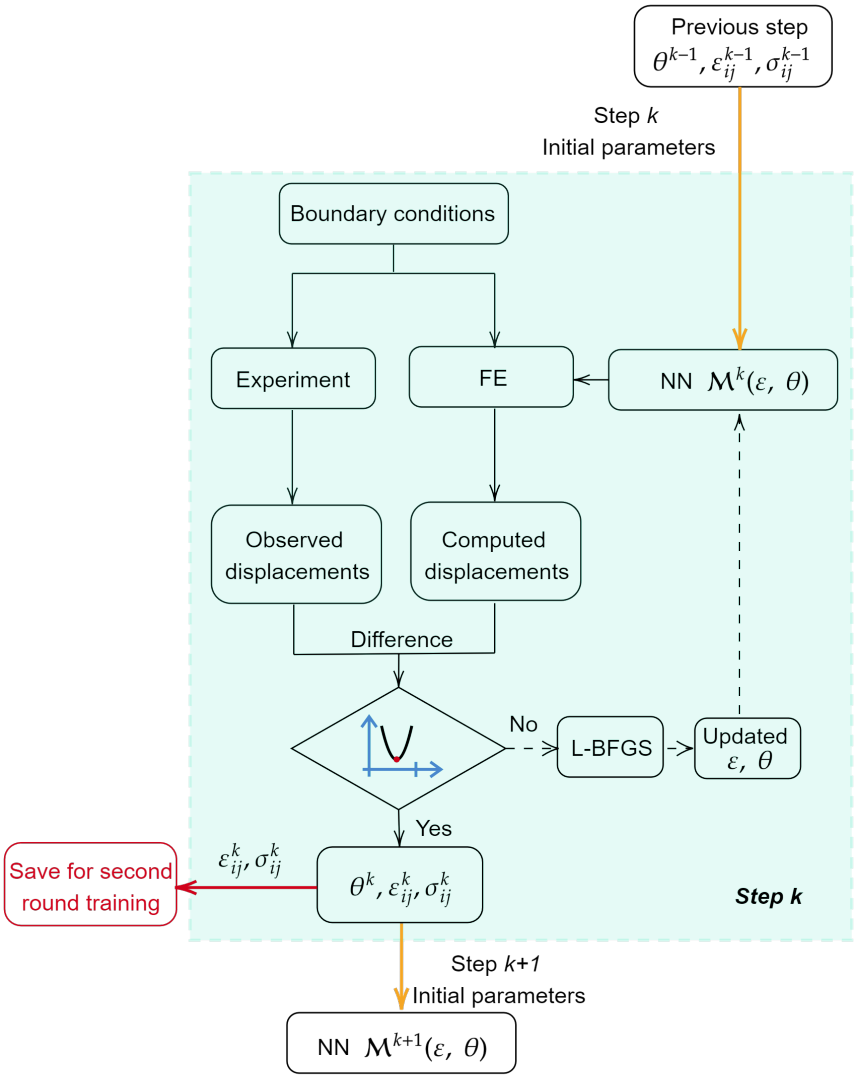


Figure 2.5. Flowchart of the training of the FE-DNN mechanics system

2.2 Integrate Abaqus with FE-DNN

The commercial FE code Abaqus is integrated with DNN to improve the efficieny of FE-DNN. This method enables data communication between Abaqus and DNN, which facilitates the creating of the FE model by using a commercial code and avoids rewriting the FE code based on the automatic differentiation package. Besides, this integration can couple with different theories in Abaqus and avoids rewriting these theories based on automatic

differentiation package. For demonstration, we applied this method to learn linear elastic constants of constituents of a fiber-reinforced composite.

2.2.1 Implementation of FE-DNN with Abaqus

For a FE code written by an automatic differentiation package, the training only needs to focus on the forward propagation. The backward propagation can be handled by the automatic differentiation package. However, for this integration, the computation of backward propagation of DNN and FE has to be done separately. Thus, the training process is required to be modified.

The modified training process is shown in Figure 2.6. As shown in the figure, at each step, the initialized weights, biases, and neural network inputs are obtained from the previous step. Then, the load and DNN predicted constitutive law will be transferred into the Abaqus input file. A script is used to change the input file to Abaqus design sensitivity analysis (DSA) input file. Abaqus will solve the FE model and compute the gradient of the displacement with respect to the constitutive law parameters via DSA [92]. Next, an Abaqus python script will be used to read the displacement and gradient with respect to the constitutive law parameters from Abaqus output file. The displacement and gradient are used to rebuild the backpropagation equation (Eq. 2.19). The difference between the observed displacements and computed displacements will be evaluated. If the difference does not reach the termination criterion, the weights and biases will be updated through an optimizer. Otherwise, the DNN inputs, outputs, and model parameters will be saved for the second-round training and transferred to the next step as initial parameters. By repeating doing that for all the datasets, the complete sets of inputs and outputs of DNN can be obtained. After the completion of the first-round training, the second-round training will be performed to approximate the overall relationship between the accurate inputs and outputs.

For the implementation of DNN, we leveraged PyTorch to construct a neural network. PyTorch is an open-source symbolic tensor manipulation software library developed by Facebook [93]. PyTorch uses automatic differentiation in its library, which can be used for the

forward and backward propagation. The PyTorch sequential function was used to build a multi-layered neural network.

The key to couple Abaqus with DNN is to form Eq. (2.19) correctly. In the Abaqus-DNN frame, $(\mathbf{u}_i - \hat{\mathbf{u}}_i)$ can be computed analytically, $\partial \mathbf{u}_i / \partial \mathcal{M}_m$ can be evaluated using the Abaqus design sensitivity analysis capability, and $\partial \mathcal{M}_m / \partial \boldsymbol{\theta}$ can be handled by the Pytorch. During the training of Abaqus-DNN, an Abaqus python script was written to access the Abaqus analysis result (odb file). Unfortunately, Abaqus python was compiled based on python 2, which does not support the PyTorch package. Therefore, we have to separate the Abaqus python script and neural network optimization script. The data communication between Abaqus and neural network was achieved by saving the observed nodes' displacements, strains, and the corresponding gradients to disk using python NumPy savez function. Then, the saved file was read in another script, which will normalize the displacements to the range $[0, 1]$ and follow the explicit backpropagation equation to update weights and biases.

The training followed the process described in Figure 2.6. The training would be terminated after the loss reached the accuracy or the training step reached the maximum steps. The training datasets for both problems were relatively small. Thus, a quad-core computer was used to do the training. The linear problem was completed within 20 minutes, the damage problem used 10 hours to complete the training.

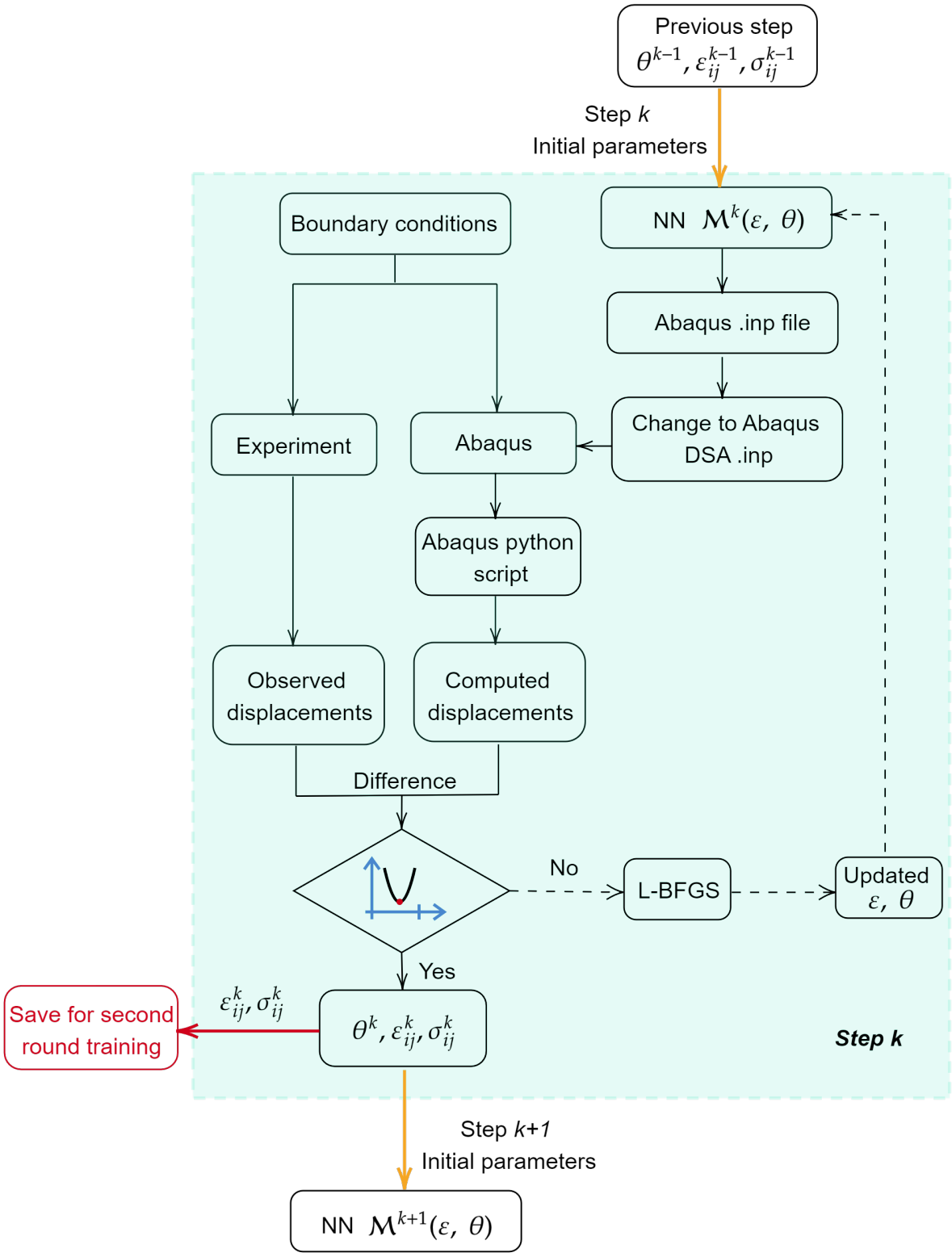


Figure 2.6. Flowchart of the training of the FE-DNN with Abaqus

2.2.2 Example 1: Learn linear elastic constants of constituents of a fiber-reinforced composite

Example 1 problem description

A 3D square packed model is used to model the linear elastic behavior of a fiber-reinforced composite as shown in Figure 2.7. The model has $L \times W \times H = 1 \times 1 \times 1$ mm and is fixed on the $x = 0$ surface. The surface pressures are applied at $x = 1$ and $z = 1$ planes with magnitude $p = -10$ MPa respectively. This model has two constituents, which are fiber and matrix. The fiber volume fraction is 0.6. The fiber and matrix are assumed to be transversely isotropic and isotropic respectively. The engineering constants of these two materials are listed in Table 2.1.

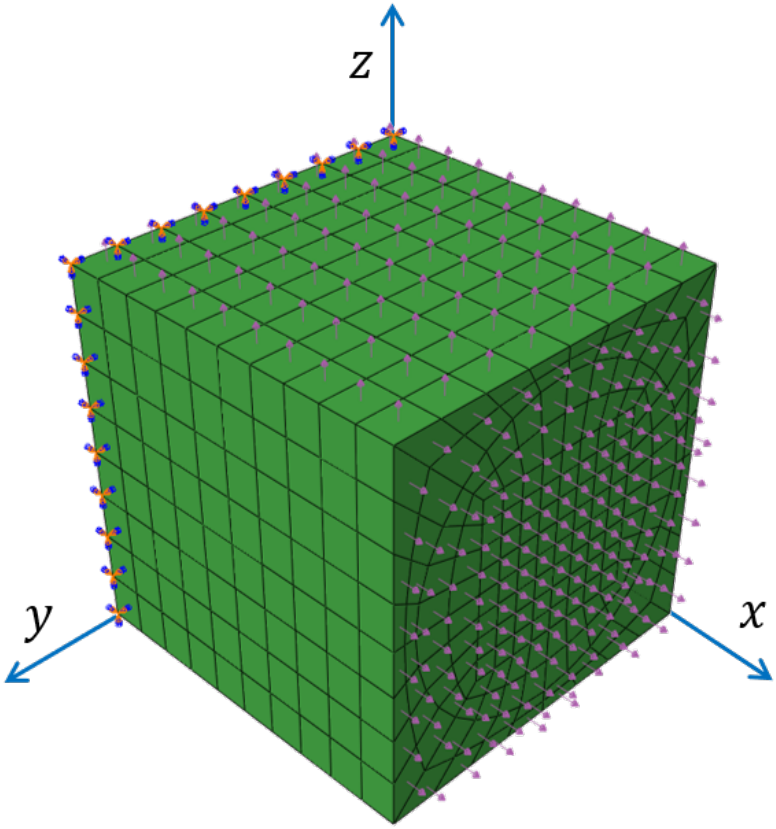


Figure 2.7. A squared-packed composites model

Table 2.1. Fiber and matrix material properties

	E_1(GPa)	E_2(GPa)	v_{12}	G_{12}(GPa)	G_{23}(GPa)
Fiber	276.00	19.50	0.28	70.00	7.62
	E(GPa)			v	
Matrix	4.76			0.36	

For a linear elastic material, the constitutive law is

$$\boldsymbol{\sigma} = \boldsymbol{C}\boldsymbol{\varepsilon} \tag{2.20}$$

where $\boldsymbol{\sigma}$ is the stress, $\boldsymbol{\varepsilon}$ is the strain, $\boldsymbol{C}$ is the stiffness matrix. Eq. (2.20) can be replaced by a DNN as shown in Eq. (2.12). For a FE analysis, the Jacobian of $\boldsymbol{\sigma}(\boldsymbol{\varepsilon})$ is required to form the global stiffness matrix to solve for the unknown variables. Although DNN can form an accurate mapping between input and output, it cannot get an accurate gradient of input to output at the same time. Therefore, instead of using the DNN model to directly learn the mapping between stresses and strains, the DNN model is used to predict the Jacobian of $\boldsymbol{\sigma}(\boldsymbol{\varepsilon})$. The general form of the Jacobian of $\boldsymbol{\sigma}(\boldsymbol{\varepsilon})$ is

$$\mathcal{M}(\boldsymbol{\varepsilon}, \boldsymbol{\theta}) = \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}} = \begin{bmatrix} \frac{\partial \sigma_{11}}{\partial \varepsilon_{11}} & \frac{\partial \sigma_{11}}{\partial \varepsilon_{22}} & \frac{\partial \sigma_{11}}{\partial \varepsilon_{33}} & \frac{\partial \sigma_{11}}{\partial \gamma_{23}} & \frac{\partial \sigma_{11}}{\partial \gamma_{13}} & \frac{\partial \sigma_{11}}{\partial \gamma_{12}} \\ & \frac{\partial \sigma_{22}}{\partial \varepsilon_{22}} & \frac{\partial \sigma_{22}}{\partial \varepsilon_{33}} & \frac{\partial \sigma_{22}}{\partial \gamma_{23}} & \frac{\partial \sigma_{22}}{\partial \gamma_{13}} & \frac{\partial \sigma_{22}}{\partial \gamma_{12}} \\ & & \frac{\partial \sigma_{33}}{\partial \varepsilon_{33}} & \frac{\partial \sigma_{33}}{\partial \gamma_{23}} & \frac{\partial \sigma_{33}}{\partial \gamma_{13}} & \frac{\partial \sigma_{33}}{\partial \gamma_{12}} \\ & & & \frac{\partial \sigma_{23}}{\partial \gamma_{23}} & \frac{\partial \sigma_{23}}{\partial \gamma_{13}} & \frac{\partial \sigma_{23}}{\partial \gamma_{12}} \\ & \text{symmetric} & & & \frac{\partial \sigma_{13}}{\partial \gamma_{13}} & \frac{\partial \sigma_{13}}{\partial \gamma_{12}} \\ & & & & & \frac{\partial \sigma_{12}}{\partial \gamma_{12}} \end{bmatrix} \tag{2.21}$$

For a linear elastic problem, $\mathcal{M}$ can be regarded as the linear elastic stiffness matrix $\boldsymbol{C}$. Clearly, $\boldsymbol{C}$ is not a function of $\boldsymbol{\varepsilon}$, as material properties of linear elastic material are constant. Thus, $\boldsymbol{C}(\boldsymbol{\varepsilon}, \boldsymbol{\theta})$ can be reduced to $\boldsymbol{C}(\boldsymbol{\theta})$. This simplification makes the problem become an uncoupled problem as represented by the solid blue line in Figure 2.3. Thus, the input of

DNN can be set as a fixed single value. Besides, since the fiber and matrix are assumed to be transversely isotropic and isotropic respectively, Eq. (2.21) reduces to

$$\begin{aligned}\mathcal{M}(\boldsymbol{\theta}) &= \boldsymbol{C}(\boldsymbol{\theta}) \\ &= \begin{bmatrix} C_{11}(\boldsymbol{\theta}) & C_{12}(\boldsymbol{\theta}) & C_{12}(\boldsymbol{\theta}) & 0 & 0 & 0 \\ & C_{22}(\boldsymbol{\theta}) & C_{23}(\boldsymbol{\theta}) & 0 & 0 & 0 \\ & & C_{22}(\boldsymbol{\theta}) & 0 & 0 & 0 \\ & & & \frac{C_{22}(\boldsymbol{\theta})-C_{23}(\boldsymbol{\theta})}{2} & 0 & 0 \\ & \text{symmetric} & & & C_{55}(\boldsymbol{\theta}) & 0 \\ & & & & & C_{55}(\boldsymbol{\theta}) \end{bmatrix}\end{aligned} \tag{2.22}$$

After discretizing the computation domain and assembling the global stiffness matrix, we can express the unknown displacements as [94]

$$\mathbf{v}^h = \mathbf{K}(\boldsymbol{C})^{-1}\mathbf{F} \tag{2.23}$$

where $\mathbf{v}^h$ is the unknown displacement, $\mathbf{K}$ is the global stiffness matrix, $\mathbf{F}$ is the external force. Eq. (2.23) shows that the unknown displacement $\mathbf{v}^h$ is a function of $\boldsymbol{C}(\boldsymbol{\theta})$. Since $\boldsymbol{C}$ is a function of neural network weights and biases, clearly, $\mathbf{K}$ is also a function of neural network weights and biases. Therefore, the structure of the Abaqus-DNN system can be described in Figure 2.8. As shown in the figure, both fiber and matrix neural networks have one input. As mentioned earlier, the value of the input was set to be one. The fiber neural network has five outputs which account for the five independent engineering constants of a transversely isotropic material. The outputs of the matrix neural network are two which represent the independent engineering constants of an isotropic material.

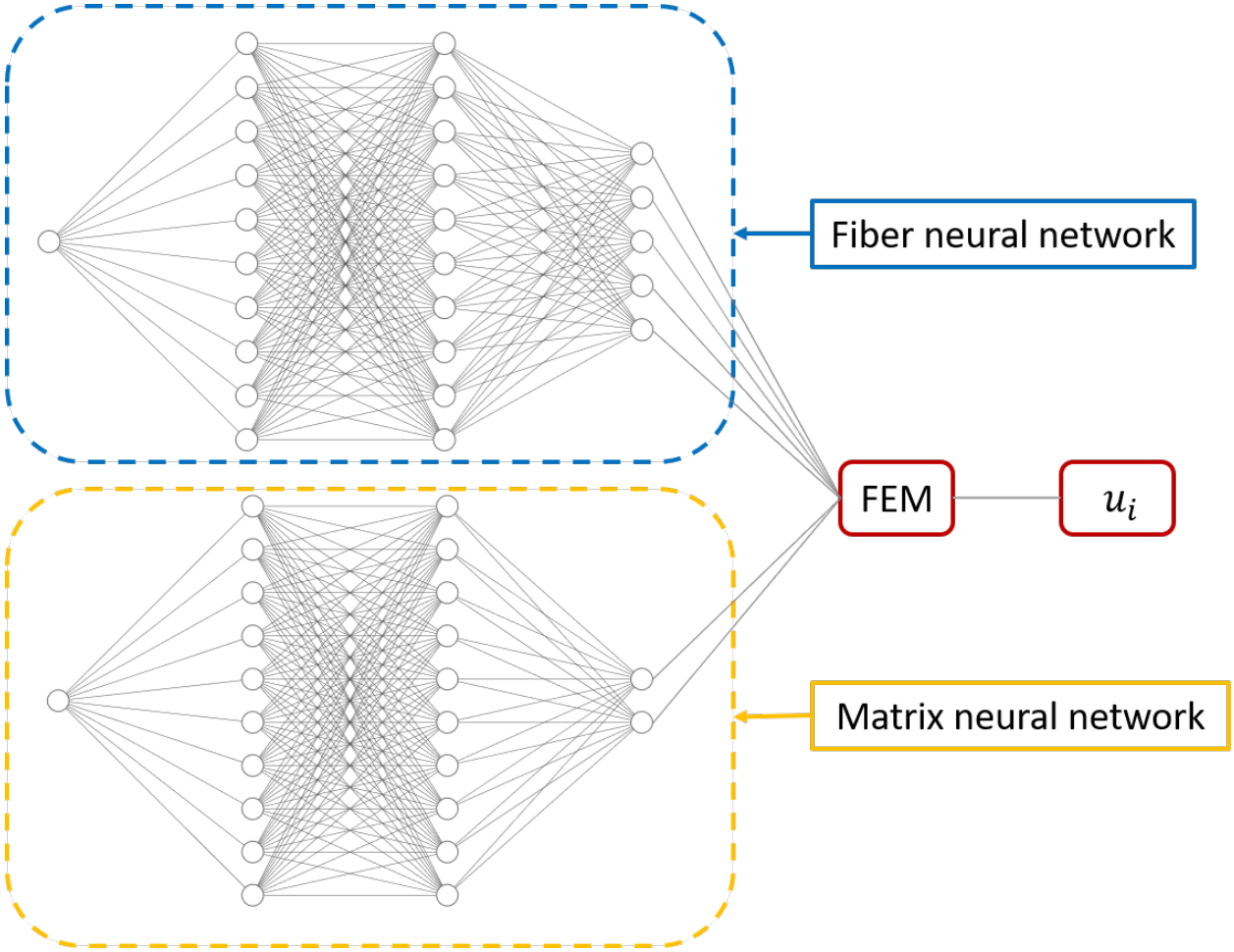


Figure 2.8. The structure of FE-DNN for a linear elastic problem

The backward propagation equation can be explicitly expressed as

$$\begin{aligned}
\frac{\partial \boldsymbol{L}}{\partial \boldsymbol{\theta}} =& \frac{1}{N}\sum_{i=1}^{N} 2\left(\mathbf{u}_i - \hat{\mathbf{u}}_i\right)\left(\sum_{i=1}^{M}\frac{\partial \mathbf{u}_i}{\partial \mathcal{M}_m}\frac{\partial \mathcal{M}_m}{\partial \boldsymbol{\theta}}\right) \\
=& \frac{2}{N}\left[\left(u_1-\hat{u}_1\right)\left(\frac{\partial u_1}{\partial y_{11}}\frac{\partial y_{11}}{\partial \boldsymbol{\theta}}\right)+\left(u_1-\hat{u}_1\right)\left(\frac{\partial u_1}{\partial y_{12}}\frac{\partial y_{12}}{\partial \boldsymbol{\theta}}\right)+\cdots+\left(u_1-\hat{u}_1\right)\left(\frac{\partial u_1}{\partial y_{22}}\frac{\partial y_{22}}{\partial \boldsymbol{\theta}}\right)\right]+ \\
& \frac{2}{N}\left[\left(u_2-\hat{u}_2\right)\left(\frac{\partial u_2}{\partial y_{11}}\frac{\partial y_{11}}{\partial \boldsymbol{\theta}}\right)+\left(u_2-\hat{u}_2\right)\left(\frac{\partial u_2}{\partial y_{12}}\frac{\partial y_{12}}{\partial \boldsymbol{\theta}}\right)+\cdots+\left(u_2-\hat{u}_2\right)\left(\frac{\partial u_2}{\partial y_{22}}\frac{\partial y_{22}}{\partial \boldsymbol{\theta}}\right)\right]+ \\
& \vdots \\
& \frac{2}{N}\left[\left(u_N-\hat{u}_N\right)\left(\frac{\partial u_N}{\partial y_{11}}\frac{\partial y_{11}}{\partial \boldsymbol{\theta}}\right)+\left(u_N-\hat{u}_N\right)\left(\frac{\partial u_N}{\partial y_{12}}\frac{\partial y_{12}}{\partial \boldsymbol{\theta}}\right)+\cdots+\left(u_N-\hat{u}_N\right)\left(\frac{\partial u_N}{\partial y_{22}}\frac{\partial y_{22}}{\partial \boldsymbol{\theta}}\right)\right]
\end{aligned} \tag{2.24}$$

where $y_{1p}, p = 1, \ldots, 5$ and $y_{2q}, q = 1, 2$ are the outputs of the fiber neural network and matrix neural network respectively.

This model used 1920 C3D8R elements after a convergence study. After trial-and-error tests, the structure of the neural network for fiber was identified as $[1, 10, 10, 5]$, which indicates that the neural network has one input, five outputs, and two hidden layers with each hidden layer have ten neurons. The activation function was chosen to be the ReLU function. For the matrix, the neural network was chosen to be $[1, 10, 10, 2]$. The final step's load and the corresponding displacements were selected for training. Therefore, the training can be completed within one step as shown in Figure 2.6. The observation nodes used for training are shown along the red line in Figure 2.9. Note that the training only used the nodes on the exterior surface. This indicates that the proposed Abaqus-DNN model can be completed using only the surface displacements, which are easily measured from an experiment.

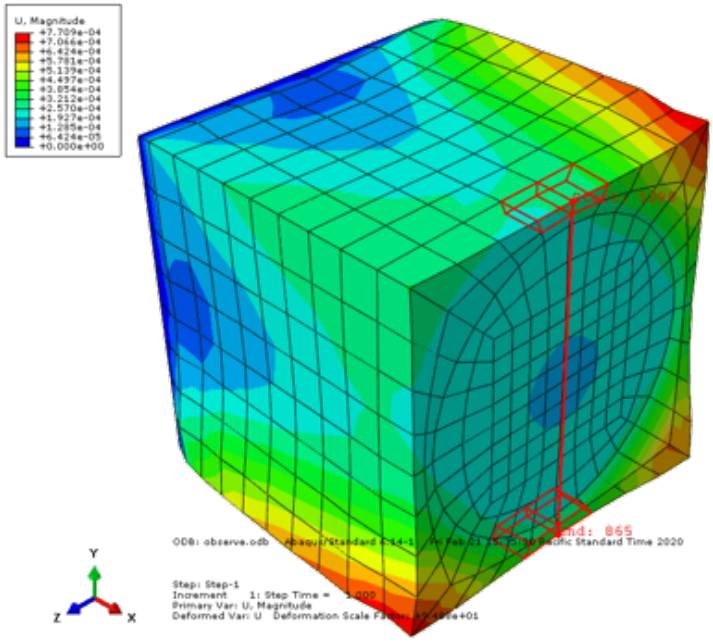


Figure 2.9. The observation nodes for training

Example 1 results and discussion

Figure 2.10 shows that the system loss of training data with respect to the number of iterations. As can be seen in the figure, the loss decreased fast with the increase of iterations. The loss function Eq. (2.18) reached an optimal value of 10^{-14} within 100 iterations.

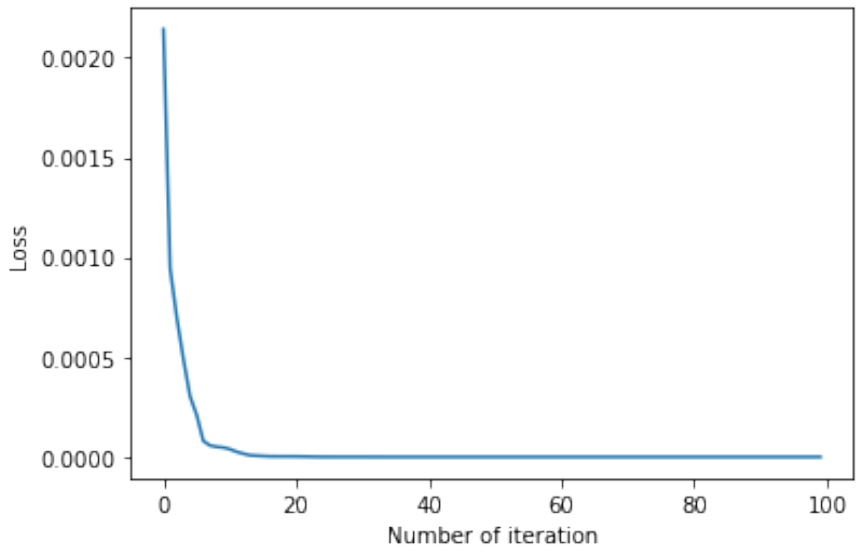


Figure 2.10. Training loss of the learning linear elastic constants example

An initial guess is needed to start optimization for the L-BFGS optimizer. In principle, an ideal algorithm is desired to handle any initial guess. However, a reasonable initial guess is usually needed for an optimization problem, as a bad initial guess might cause the algorithm to converge slowly or lose convergence. For example, a negative initial guess of Young's modulus will cause stiffness matrix not positive-definite. Abaqus will stop running as the stiffness matrix does not physically make sense. This will cause the algorithm to stop running and lose convergence. For this problem, the initial guess of the material properties are listed in Table 2.2. The initial weights and biases of the neural network model were initialized by pre-training the model based on the initial guess of the material properties. The Diff1 column shows the difference between the initial guess and the target of the material properties. As one can tell, the difference between the initial guess and the target material properties ranges from -25% to -104%. Noting that the initial guess does not have to be limited to this range, a larger range can also lead to an accurate learning result. Diff2 represents

the learned material properties deviate from the target material properties. Although only several points on the exterior surface were selected for the training, the material properties of the two constituents can still be learned very accurately.

Table 2.2. Comparison between target, initial, and learned material properties

	Target	**Initial**	**Learned**	**Diff1**	**Diff2**
E_1(GPa)	276.00	394.25	276.01	-42.84 %	0.00 %
E_2 (GPa)	19.50	0.94	19.50	95.18 %	0.00 %
G_{12} (GPa)	70.00	50.00	70.00	28.57 %	0.00 %
v_{12}	0.28	0.20	0.28	28.57 %	0.00 %
G_{23} (GPa)	7.62	0.24	7.62	96.85 %	0.00 %
E_m(GPa)	4.76	9.73	4.76	-104.41 %	0.00 %
v_m	0.36	0.45	0.36	-25.00 %	0.00 %

2.3 Couple FE with Positive-Definite Deep Neural Networks (FE-PDNN)

In this section, we coupled FE with Cholesky decomposed positive-definite deep neural networks (PDNN) to form FE-PDNN. This modification imposes the positive-definite constraint to the stiffness matrix predicted by DNN and thus improves the convergence stability of the training process.

2.3.1 Implementation of FE-PDNN

A material stiffness matrix is a symmetric positive-definite real matrix. Then, the Cholesky decomposition factors a positive-definite matrix $\mathcal{M}$ into:

$$\mathcal{M} = \boldsymbol{L}_c \boldsymbol{L}_c^T \tag{2.25}$$

where $\boldsymbol{L}_c$ is a Cholesky factored lower triangular matrix with positive diagonal entries. To apply the positive-definite constraint to the stiffness matrix, DNN is trained to build the mapping between strain and the lower triangular matrix $\boldsymbol{L}_c$

$$\boldsymbol{L}_c = \mathrm{NN}(\boldsymbol{\varepsilon}; \boldsymbol{\theta}) \tag{2.26}$$

where NN represents deep neural networks, $\boldsymbol{\varepsilon}$ is the strain, $\boldsymbol{\theta}$ represents weights and biases of neural networks. $\boldsymbol{L}_c$ will multiply its transpose matrix to form the stiffness matrix $\mathcal{M}$. This type of neural networks is defined as PDNN since the stiffness matrix predicted by $\mathcal{M}$ remains positive-definite. The constitutive relation can be expressed the same as Eq. (2.12). The data flow of FE-PDNN can be summarized using Figure 2.11.

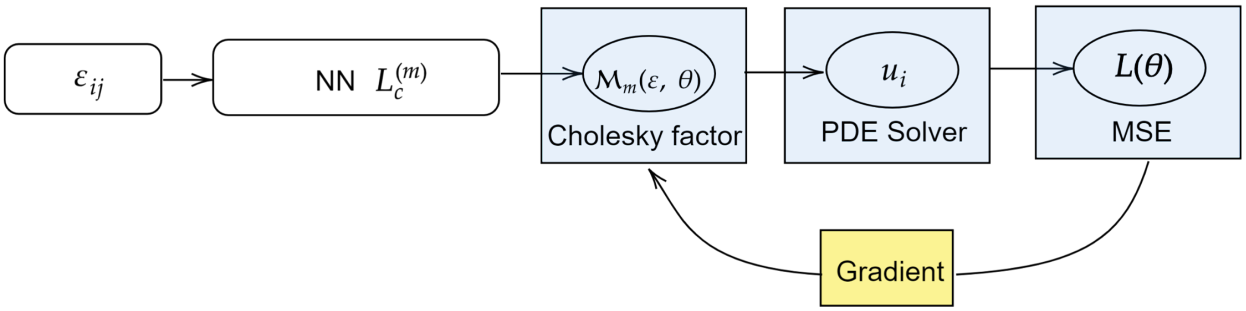


Figure 2.11. Data flow of the FE-PDNN mechanics system

For FE-PDNN, the training flow needs to be modified. The flowchart of the training of FE-PDNN is shown in Figure 2.12. The training scheme is very similar to the training process shown in Figure 2.6. The difference is that the DNN will be trained to predict the lower triangular matrix of the stiffness matrix. The predicted lower triangular matrix will multiply its transpose matrix to form a positive-definite stiffness matrix. The PDNN predicted constitutive law and load will be transferred into the Abaqus, which solves the FE model and computes the gradient of the displacement with respect to the constitutive law parameters via DSA [92]. This change applied a hard constraint to the DNN predicted result and guaranteed the stiffness matrix predicted by PDNN is positive-definite.

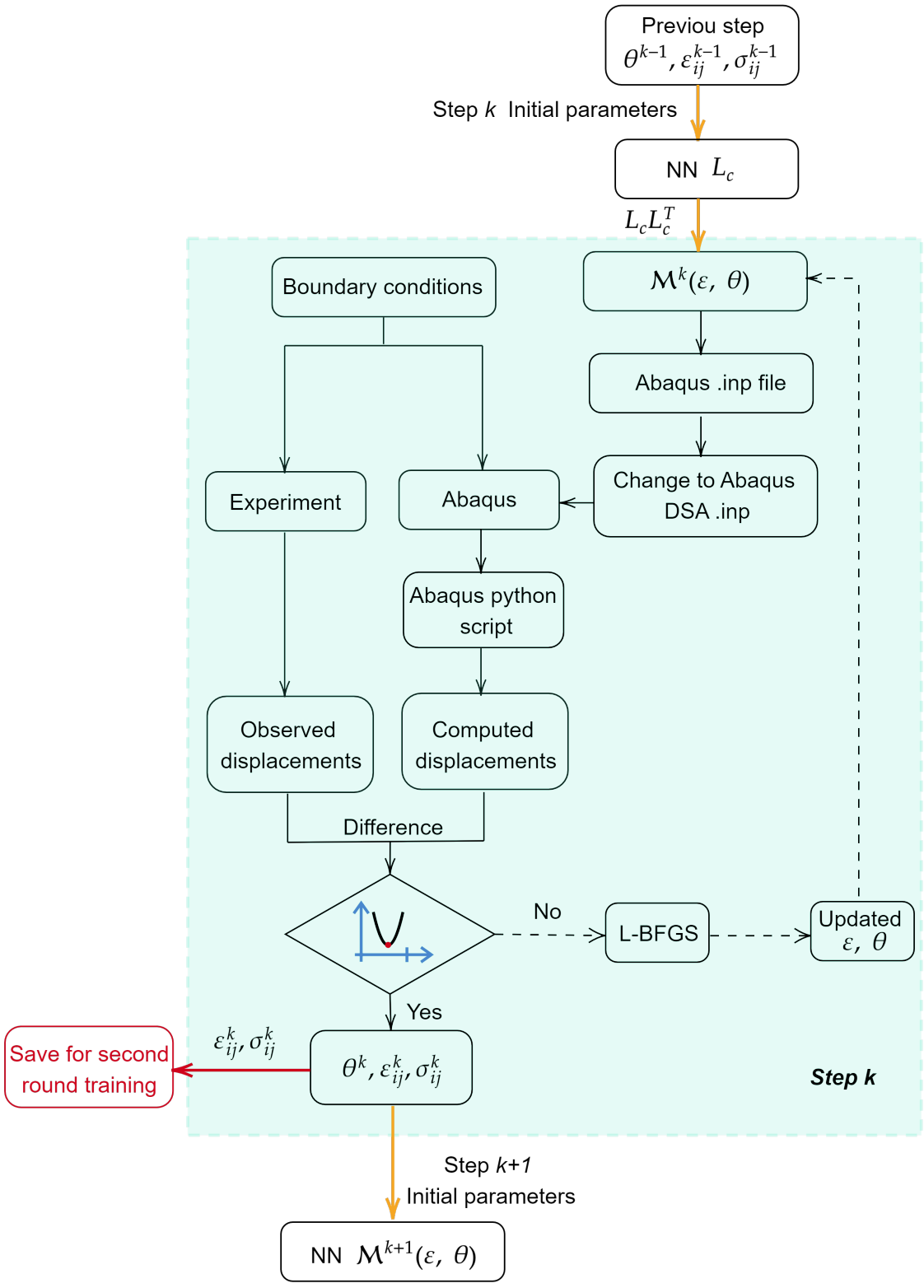


Figure 2.12. Flowchart of the training of the FE-PDNN mechanics system

2.3.2 Example 2: Learn the damage constitutive law of a unidirectional fiber-reinforced composite laminate with FE-PDNN

Example 2 problem description

In this section, a fiber-reinforced laminate example is presented to test the capability of FE-PDNN on learning damage constitutive law of composite materials. The Abaqus build-in damage model of fiber-reinforced composites was used to generate observation data. The layup angles of the laminate are $[10]_6$. It is not necessary to combine the lamination theory with FE-PDNN for this problem, as this plate is homogeneous through the thickness and can be regarded as a plane-stress problem. The length and width of the laminate are $L \times H = 138 \times 25$ mm, the ply thickness is 0.28 mm. As shown in Figure 2.13, the center of the left surface $x = 0$ is pinned with $u_x = u_y = 0$. The other area along $x = 0$ is constrained with $u_x = 0$. A uniform displacement is applied at the right end surface with $u_x = 0.90$ mm.

Figure 2.13. Boundary conditions of the fiber-reinforced composite laminate

The material of the model is assumed to be a fiber-reinforced composite with damage behavior. The plasticity can be ignored in this analysis, as the damage in a fiber-reinforced composite usually initiates without large plastic deformation. The material properties are shown in Table 2.3. For the damage model implemented by Abaqus, the damage initiation is determined by the Hashin failure criterion [25], [58], and the damage evolution is based on the fracture energy dissipation during the damage process [95]. The constitutive law of the material can be expressed

$$\boldsymbol{\sigma} = \boldsymbol{C}_d \boldsymbol{\varepsilon} \tag{2.27}$$

where $\boldsymbol{C_d}$ is the degraded elasticity matrix

$$\boldsymbol{C_d} = \frac{1}{D}\begin{bmatrix} (1-d_f)\,E_1 & (1-d_f)\,(1-d_m)\,\nu_{21}E_1 & 0 \\ (1-d_f)\,(1-d_m)\,\nu_{12}E_2 & (1-d_m)\,E_2 & 0 \\ 0 & 0 & (1-d_s)\,GD \end{bmatrix} \tag{2.28}$$

and the corresponding damaged compliance matrix is

$$\mathbf{H} = \begin{bmatrix} \frac{1}{(1-d_f)E_1} & -\frac{\nu_{21}}{E_2} & 0 \\ -\frac{\nu_{12}}{E_1} & \frac{1}{(1-d_m)E_2} & 0 \\ 0 & 0 & \frac{1}{(1-d_s)G_{12}} \end{bmatrix} \tag{2.29}$$

where $D = 1-(1-d_f)\,(1-d_m)\,\nu_{12}\nu_{21}$, d_f, d_m, d_s reflect the current state of fiber damage, matrix damage, and shear damage respectively. The DNN model is used to predict the lower triangular matrix of the Jacobian of $\boldsymbol{\sigma(\varepsilon)}$. Thus, we have

$$\mathcal{M}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) = \boldsymbol{L}_c\boldsymbol{L}_c^T = \begin{bmatrix} \frac{\partial\sigma_{11}}{\partial\epsilon_{11}} & \frac{\partial\sigma_{11}}{\partial\epsilon_{22}} & \frac{\partial\sigma_{11}}{\partial\gamma_{12}} \\ \frac{\partial\sigma_{11}}{\partial\epsilon_{22}} & \frac{\partial\sigma_{22}}{\partial\epsilon_{22}} & \frac{\partial\sigma_{22}}{\partial\gamma_{12}} \\ \frac{\partial\sigma_{11}}{\partial\gamma 12} & \frac{\partial\sigma_{22}}{\partial\gamma 12} & \frac{\partial\sigma_{12}}{\partial\gamma_{12}} \end{bmatrix} \tag{2.30}$$

For this problem, $\mathcal{M}(\boldsymbol{\varepsilon},\boldsymbol{\theta})$ can be regarded as $\boldsymbol{C_d}(\boldsymbol{\varepsilon},\boldsymbol{\theta})$. Since the fiber-reinforced composite is assumed to be transversely isotropic, $\mathcal{M}(\boldsymbol{\varepsilon},\boldsymbol{\theta})$ reduces to

$$\mathcal{M}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) = C_d(\boldsymbol{\varepsilon},\boldsymbol{\theta}) = \begin{bmatrix} C_{11}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) & C_{12}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) & 0 \\ C_{12}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) & C_{22}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) & 0 \\ 0 & 0 & C_{66}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) \end{bmatrix} \tag{2.31}$$

It is noted that the simplification in Eq. (2.31) is not a constraint for the FE-PDNN. It is possible to use a fully populated matrix if the extension and shear coupling does exist in the material.

In Eq. (2.31), PDNN is fully coupled with Abaqus as $\mathcal{M}$ is a function of $\boldsymbol{\varepsilon}$. The PDNN inputs need to be derived from the PDNN outputs of the previous iteration. Besides, all

Table 2.3. Laminate material properties and strength parameters [96]

	Material properties
Elastic properties:	
E_1(MPa)	$123,520$
E_2(MPa)	$6,516$
G_{12}(MPa)	$2,494$
ν_{12}	0.32
Tensile strengths:	
X(MPa)	$1,429$
Y(MPa)	41
Compressive strengths:	
X'(MPa)	530
Y'(MPa)	145
Shear strengths:	
S(MPa)	83.4
T, MPa	83.4

the PDNN outputs will contribute to the displacement. Thus, it is necessary to compute the displacement gradient of each node with respect to all the outputs during backward propagation.

The dimensions of the inputs and outputs of PDNN are three and four respectively, since the in-plane strains and independent elements in the Jacobian are used as input and output. The FE-PDNN structure for this problem is presented in Figure 2.14.

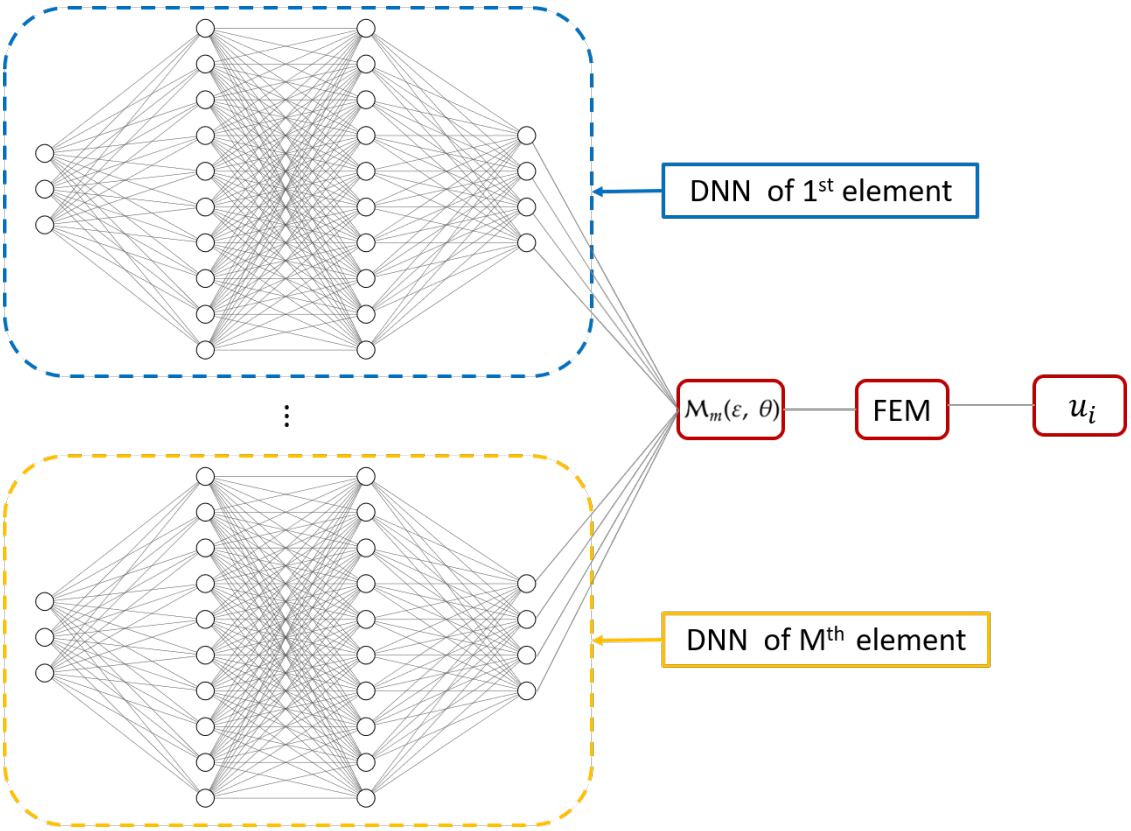


Figure 2.14. The structure of FE-PDNN for a damage problem

The explicit form of the modified backward propagation equation is

$$\begin{aligned}
\frac{\partial \boldsymbol{L}}{\partial \boldsymbol{\theta}} =& \frac{1}{N}\sum_{i=1}^{N} 2\left(\mathbf{u}_i - \hat{\mathbf{u}}_i\right)\left(\sum_{m=1}^{M}\frac{\partial \mathbf{u}_i}{\partial \mathcal{M}_m}\frac{\partial \mathcal{M}_m}{\partial \boldsymbol{\theta}}\right) \\
=& \frac{2}{N}\left[\left(u_1-\hat{u}_1\right)\left(\frac{\partial u_1}{\partial y_{11}}\frac{\partial y_{11}}{\partial \boldsymbol{\theta}}\right)+\left(u_1-\hat{u}_1\right)\left(\frac{\partial u_1}{\partial y_{12}}\frac{\partial y_{12}}{\partial \boldsymbol{\theta}}\right)+\cdots+\left(u_1-\hat{u}_1\right)\left(\frac{\partial u_1}{\partial y_{M4}}\frac{\partial y_{M4}}{\partial \boldsymbol{\theta}}\right)\right]+ \\
& \frac{2}{N}\left[\left(u_2-\hat{u}_2\right)\left(\frac{\partial u_2}{\partial y_{11}}\frac{\partial y_{11}}{\partial \boldsymbol{\theta}}\right)+\left(u_2-\hat{u}_2\right)\left(\frac{\partial u_2}{\partial y_{12}}\frac{\partial y_{12}}{\partial \boldsymbol{\theta}}\right)+\cdots+\left(u_2-\hat{u}_2\right)\left(\frac{\partial u_2}{\partial y_{M4}}\frac{\partial y_{M4}}{\partial \boldsymbol{\theta}}\right)\right]+ \\
& \vdots \\
& \frac{2}{N}\left[\left(u_N-\hat{u}_N\right)\left(\frac{\partial u_N}{\partial y_{11}}\frac{\partial y_{11}}{\partial \boldsymbol{\theta}}\right)+\left(u_N-\hat{u}_N\right)\left(\frac{\partial u_N}{\partial y_{12}}\frac{\partial y_{12}}{\partial \boldsymbol{\theta}}\right)+\cdots+\left(u_N-\hat{u}_N\right)\left(\frac{\partial u_N}{\partial y_{M4}}\frac{\partial y_{M4}}{\partial \boldsymbol{\theta}}\right)\right]
\end{aligned} \tag{2.32}$$

where y_{pq} is the q_{th} neural networks output of the p_{th} element. $\frac{\partial \mathcal{M}_m}{\partial \boldsymbol{L}_c^{(m)}}\frac{\partial \boldsymbol{L}_c^{(m)}}{\partial \boldsymbol{\theta}}$ is simplified to $\frac{\partial \mathcal{M}_m}{\partial \boldsymbol{\theta}}$ since it can be handled by the automatic differentiation package.

For this problem, the FE model converges with 368 CPS4R elements. Thus, the rest of the analysis was based on this element size. Besides, this model can also be regarded as x and y axis-symmetric if the observation reference point is moved to the center of the model ($x = 69, y = 12.5$). After modifying the controlled displacement to 0.45 mm (due to symmetry), a quarter of the model was used for the analysis. After trial-and-error testings, the DNN structure was identified as $[3, 40, 40, 40, 4]$, which indicates that the neural networks has three inputs, four outputs, and three hidden layers with each hidden layer having 40 neurons. The training data was split into 31 datasets, which means that the training has 31 steps. The training process followed the flowchart described in Figure 2.12. For this study, we selected the ReLU function as the activation function. ReLU activation function has gained massive popularity since it is computationally less expensive, avoids vanishing gradient problem, and has simpler mathematical operations. However, ReLU could have the "Dying ReLU" problem, which could cause gradients to fail to flow during backpropagation, and the weights are not updated. The "Dying ReLU" problem can be tackled by using a small learning rate and modifying the initialization of NN. In our implementation, we pretrained the NN before embedding it into FEM. Besides, we also set the learning rate to be around 10^{-4}, which is a small learning rate compared to the common learning rate of 10^{-3}.

Example 2 results and discussion

Figure 2.15 shows the plot of the averaged stress versus strain at the right end surface. The averaged stress is computed by using the force divided by the laminate cross-sectional area. The averaged strain is obtained using the displacement divided by the length of the model. As one can tell from the plot, the damage initiated at $\varepsilon_{11} = 0.0047$. Once the damage was initiated, the strain-stress relation becomes clearly not linear. Then, the laminate reached peak stress at $\varepsilon_{11} = 0.0058$. After the peak stress, the laminate degraded quickly and failed abruptly. From this plot, one can also observe that FE-PDNN can learn the damage behavior very well, as the differences between observation and FE-PDNN are within 1%.

The learning capability of FE-PDNN can be further confirmed in Figures 2.16 and 2.17. Figure 2.16 compares the contour plots of the observed and learned displacements. Step 15 ($\sigma_{\text{ave}} = 160.4$ MPa) and 31 ($\sigma_{\text{ave}} = 162.6$ MPa) are the damage initiated step and last steps, respectively. The material is severely damaged at step 31. These plots show that the observed and learned displacements agree with each other very well. Additionally, The contour plots at steps 15 and 31 show a great difference near the left end surface. The blue area at the left bottom end surface of step 31 is smaller than that of step 15, which indicates there is more deformation in this area at step 31. This change was caused by the damage of the material. Figure 2.17 shows the comparison of the σ_{11} at step 15 and 31. Again, one can conclude that the maximum and minimum stresses agree with each other very well, the contour shape matches excellently. Noteworthy is that the stress did not participate in the training of the FE-PDNN. The stresses are inversely determined from the training process. This demonstrated that FE-PDNN can learn the damage constitutive law of a severely damaged material.

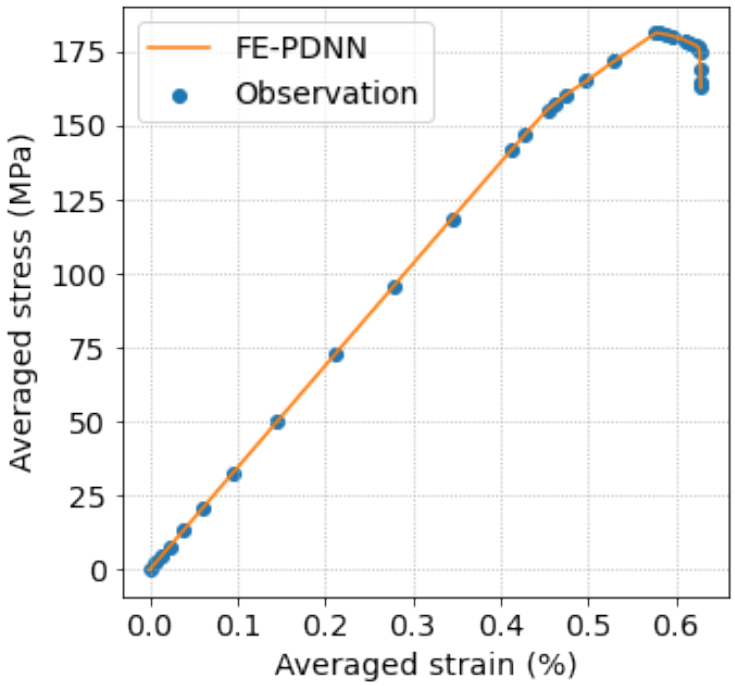
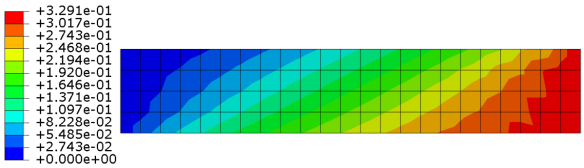
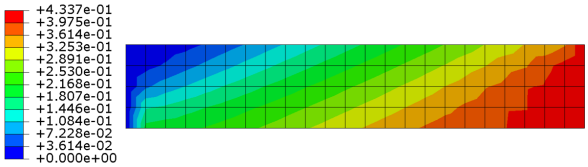


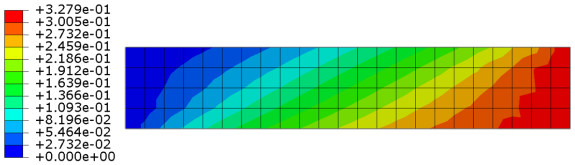
Figure 2.15. Averaged stress versus strain of the $[10]_6$ laminate



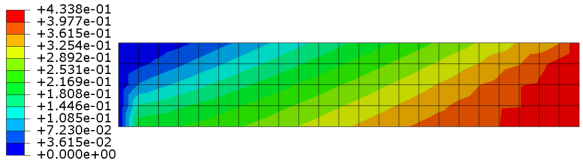
(a) Observed u_1 at step 15



(b) Observed u_1 at step 31

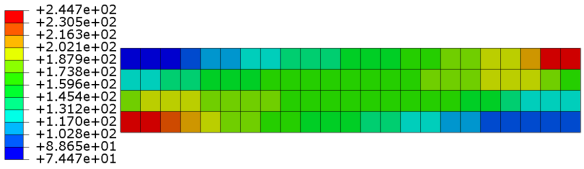


(c) Learned u_1 at step 15

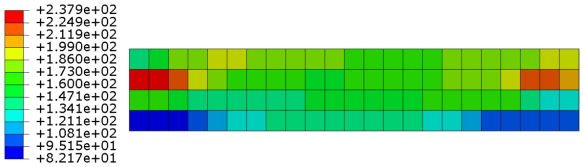


(d) Learned u_1 at step 31

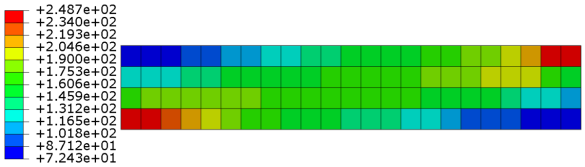
Figure 2.16. Comparison of the contour plots of u_1 of the $[10]_6$ laminate



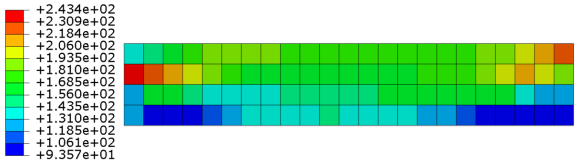
(a) Observed σ_{11} at step 15



(b) Observed σ_{11} at step 31



(c) Learned σ_{11} at step 15

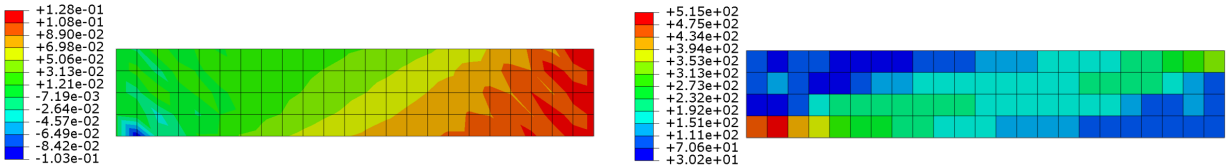


(d) Learned σ_{11} at step 31

Figure 2.17. Comparison of the contour plots of σ_{11} of the $[10]_6$ laminate

Additionally, a comparison was done between the learning capability of FE-DNN and FE-PDNN. Figure 2.18 presents the learning result of FE-DNN, which did not apply positive-definite constraint to the DNN. Figure 2.18a presents the displacement learned by FE-DNN. One can tell that FE-DNN failed to learn the displacement at step 31, as the contour shape of Figure 2.18a is non-smooth and cannot match the observed displacement at Figure 2.16b. This is because some elements already have severe damage, thus the diagonal terms of the stiffness matrix degraded significantly. This caused the training convergence issue as some diagonal terms were shifted to the negative value. As mentioned earlier, a negative diagonal term of the stiffness matrix causes the strain energy density no longer positive-definite. A

FE analysis stops running if the negative strain energy density occurs. Figure 2.18b shows the σ_{11} learned by FE-DNN. Again, FE-DNN failed to learn the stress at step 31.



(a) FE-DNN learned u_1 at step 31 (b) FE-DNN learned σ_{11} stress at step 31

Figure 2.18. The learning result of FE-DNN for the $[10]_6$ laminate

2.4 Couple the Classical Lamination Theory with FE-PDNN Mechanics System

2.4.1 Lamination theory enhanced FE-PDNN

A composite laminate is formed by stacking multiple layers of fiber-reinforced composites. The composite layers are usually parallelly stacked together with different in-plane fiber orientations. For the classical lamination theory, the Kirchhoff assumptions [97] are made to derive the relationship between displacements and strains. The 3D solid displacements and 2D plate displacements can be related as

$$\begin{aligned} u_1(x_1, x_2, x_3) &= \bar{u}_1(x_1, x_2) - x_3 \bar{u}_{3,1} \\ u_2(x_1, x_2, x_3) &= \bar{u}_2(x_1, x_2) - x_3 \bar{u}_{3,2} \\ u_3(x_1, x_2, x_3) &= \bar{u}_3(x_1, x_2) \end{aligned} \tag{2.33}$$

where u_i are the 3D solid displacements and $\bar{u}_\alpha$ are the 2D plate displacements. Under the infinitesimal deformation assumption, the plate strains are defined as

$$\begin{aligned} \epsilon_{\alpha\beta}(x_1, x_2) &= \frac{1}{2}\left(\bar{u}_{\alpha,\beta} + \bar{u}_{\beta,\alpha}\right) \\ \kappa_{\alpha\beta}(x_1, x_2) &= -\bar{u}_{3,\alpha\beta} \end{aligned} \tag{2.34}$$

where $\epsilon_{\alpha\beta}$ are the plate strains, $\kappa_{\alpha\beta}$ are the curvatures of the reference surface. The lamination theory uses plate stress resultants defined as

$$\begin{aligned} N_{\alpha\beta} &= \langle\langle \sigma_{\alpha\beta} \rangle\rangle \\ M_{\alpha\beta} &= \langle\langle x_3 \sigma_{\alpha\beta} \rangle\rangle \end{aligned} \tag{2.35}$$

where the double angle brackets represents integration through the thickness $\langle\langle \cdot \rangle\rangle = \int_{\frac{-h}{2}}^{\frac{h}{2}} \cdot dx_3$. The indices α and β take value 1 or 2.

The constitutive law of a plate can be expressed as

$$\begin{Bmatrix} \boldsymbol{N} \\ \boldsymbol{M} \end{Bmatrix} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B} & \boldsymbol{D} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\epsilon} \\ \boldsymbol{\kappa} \end{Bmatrix} \tag{2.36}$$

with

$$\boldsymbol{A} = \langle\langle \boldsymbol{Q} \rangle\rangle, \quad \boldsymbol{B} = \langle\langle x_3 \boldsymbol{Q} \rangle\rangle, \quad \boldsymbol{D} = \left\langle\left\langle x_3^2 \boldsymbol{Q} \right\rangle\right\rangle \tag{2.37}$$

and

$$\boldsymbol{Q} = \boldsymbol{S}_e^{-1} = \begin{bmatrix} \frac{1}{E_1} & -\frac{\nu_{21}}{E_2} & \frac{\eta_{12,1}}{G_{12}} \\ -\frac{\nu_{12}}{E_1} & \frac{1}{E_2} & \frac{\eta_{12,2}}{G_{12}} \\ \frac{\eta_{1,12}}{E_1} & \frac{\eta_{2,12}}{E_2} & \frac{1}{G_{12}} \end{bmatrix}^{-1} \tag{2.38}$$

where $\boldsymbol{A}$ is the extension stiffness matrix, $\boldsymbol{B}$ is the extension-bending coupling stiffness matrix, and $\boldsymbol{D}$ is the bending stiffness matrix. $\boldsymbol{S}_e$ is the plane-stress reduced compliance matrix, $\boldsymbol{Q}$ is the plane-stress reduced stiffness matrix.

To enable DNN to learn the constitutive law of lamina based on the structural response of laminate, the FE-PDNN is coupled with the classical lamination theory. To achieve that, the DNN is trained to map between 3D strains and the Cholesky decomposed lower triangular matrix ($\boldsymbol{L}_c(\boldsymbol{\varepsilon}; \boldsymbol{\theta})$) of the lamina $\boldsymbol{Q}$ matrix. This mapping is shown below

$$\boldsymbol{L}_c(\boldsymbol{\varepsilon}; \boldsymbol{\theta}) = \mathrm{NN}(\boldsymbol{\varepsilon}; \boldsymbol{\theta}) \tag{2.39}$$

where NN represents DNN, $\boldsymbol{\varepsilon}$ and $\boldsymbol{\theta}$ are the strains and DNN parameters respectively. The lower triangular matrix will multiply its transpose matrix to form a positive-definite stiffness matrix

$$\mathcal{M}(\boldsymbol{\varepsilon};\boldsymbol{\theta}) = \boldsymbol{Q}(\boldsymbol{\varepsilon};\boldsymbol{\theta}) = \boldsymbol{L}_c\boldsymbol{L}_c^T \tag{2.40}$$

Eq. (2.40) indicates the plane-stress reduced stiffness matrix is a function of strains and DNN parameters. The plane-stress reduced stiffness matrix will follow Eq. (2.37) to form the plate stiffness matrix. Clearly, the plate stiffness matrix is a function of strains and DNN parameters. It can be expressed as

$$\boldsymbol{A}(\boldsymbol{\varepsilon};\boldsymbol{\theta}) = \langle\langle \boldsymbol{Q}(\boldsymbol{\varepsilon};\boldsymbol{\theta})\rangle\rangle \quad \boldsymbol{B}(\boldsymbol{\varepsilon};\boldsymbol{\theta}) = \langle\langle x_3\boldsymbol{Q}(\boldsymbol{\varepsilon};\boldsymbol{\theta})\rangle\rangle \quad \boldsymbol{D}(\boldsymbol{\varepsilon};\boldsymbol{\theta}) = \left\langle\left\langle x_3^2\boldsymbol{Q}(\boldsymbol{\varepsilon};\boldsymbol{\theta})\right\rangle\right\rangle \tag{2.41}$$

The data flow of coupling lamination theory with FE-PDNN can be summarized in Figure 2.19. The proposed method enables DNN to learn lamina constitutive law based on structured level response. Besides, the DNN is trained to predict the lower triangular matrix of the lamina plane-stress reduced stiffness matrix, which guarantees that the plate stiffness matrix is positive-definite. Finally, the proposed approach ensures that the learned relationship follows the governing systems implemented in the FE code.

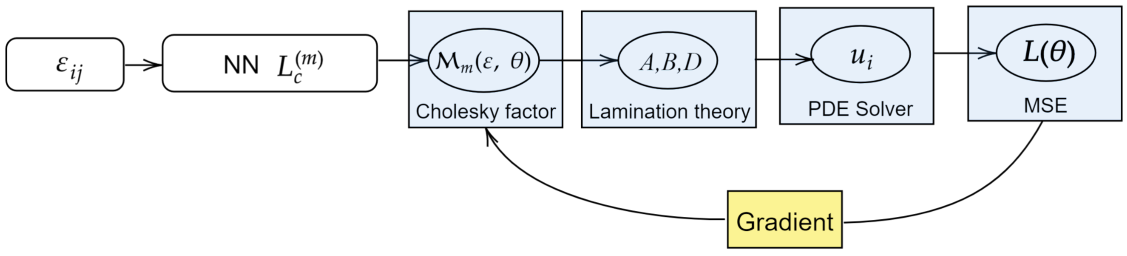


Figure 2.19. The data flow of coupling of lamination theory with the FE-PDNN mechanics system

2.4.2 Example 3: Learn damage constitutive law of lamina with lamination theory enhanced FE-PDNN

Example 3 problem description

For this example, the laminate is assumed to have the same geometry and material properties as the example in Section 2.3.2. The displacement is set to be 1.3 mm. The layup angles are $[0/45]_s$. Since the layup angles are not uniform, it would be expensive to directly apply FE-PDNN to learn the lamina constitutive law of the laminate. Therefore, the lamination theory enhanced FE-PDNN was applied to learn the constitutive law. This change needs to modify the FE elements to S4R elements. Again, the DNN model is used to predict the lower triangular matrix of plane-stress reduced stiffness matrix. Since the lamina is assumed to be transversely isotropic. We have

$$\mathcal{M}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) = \boldsymbol{L}_c\boldsymbol{L}_c^T = \boldsymbol{Q}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) = \begin{bmatrix} Q_{11}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) & Q_{12}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) & 0 \\ Q_{12}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) & Q_{22}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) & 0 \\ 0 & 0 & Q_{66}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) \end{bmatrix} \tag{2.42}$$

The stiffness matrix in Eq. (2.42) is expressed in the material coordinate system. To derive the plate constitutive stiffness matrix, one needs to transform the stiffness matrix from the material coordinate system to the laminate coordinate system

$$\boldsymbol{Q'}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) = \boldsymbol{R}_{\sigma_e}\boldsymbol{Q}(\boldsymbol{\varepsilon},\boldsymbol{\theta})\boldsymbol{R}_{\sigma_e}^T \tag{2.43}$$

with

$$\boldsymbol{R}_{\sigma_e} = \begin{bmatrix} \mathrm{Cos}(\gamma)^2 & \mathrm{Sin}(\gamma)^2 & -2\,\mathrm{Cos}(\gamma)\,\mathrm{Sin}(\gamma) \\ \mathrm{Sin}(\gamma)^2 & \mathrm{Cos}(\gamma)^2 & 2\,\mathrm{Cos}(\gamma)\,\mathrm{Sin}(\gamma) \\ \mathrm{Cos}(\gamma)\,\mathrm{Sin}(\gamma) & -\mathrm{Cos}(\gamma)\,\mathrm{Sin}(\gamma) & \mathrm{Cos}(\gamma)^2 - \mathrm{Sin}(\gamma)^2 \end{bmatrix} \tag{2.44}$$

where γ is the layup angle, $\boldsymbol{Q}'$ and $\boldsymbol{Q}$ are the in-plane reduced stiffness matrices in the material and laminate coordinate system, respectively. From Eq. (2.37), we can derive

$$\begin{aligned}
\boldsymbol{A}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) &= \begin{bmatrix} A_{11}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) & A_{12}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) & A_{16}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) \\ A_{12}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) & A_{22}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) & A_{26}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) \\ A_{16}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) & A_{26}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) & A_{66}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) \end{bmatrix} \\
\boldsymbol{D}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) &= \begin{bmatrix} D_{11}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) & D_{12}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) & D_{16}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) \\ D_{12}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) & D_{22}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) & D_{26}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) \\ D_{16}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) & D_{26}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) & D_{66}(\boldsymbol{\varepsilon},\boldsymbol{\theta}) \end{bmatrix}
\end{aligned} \tag{2.45}$$

The $\boldsymbol{B}$ matrix vanishes due to the symmetry of the laminate. Although the $\boldsymbol{A}$ and $\boldsymbol{D}$ matrices are fully populated matrix, they are derived from $\boldsymbol{Q}$ matrix. Thus, the $\boldsymbol{A}$ and $\boldsymbol{D}$ matrices can be uniquely determined by the four independent variables in the $\boldsymbol{Q}$ matrix. The plate equilibrium equations are shown in Appendix A.2. Thus, the governing equations can be formulated to solve for the displacement of the laminate [97]. The solved displacement can be used to form the MSE loss function. The neural network of this problem has the same structure as the previous example. Thus, the explicit form of the modified backward propagation equation can also be represented by Eq. (2.32). The change is that $\frac{\partial \mathbf{u}_i}{\partial \mathcal{M}_m}$ needs to be computed via the lamination theory. It is noted that although the symbol to express the gradient is the same, the physical meaning is different, because the governing equations for the plate model and the solid model are different. The top layer displacement was used as the observation data to train FE-PDNN.

Example 3 results and discussion

Figure 2.20 plots the averaged stress versus strain at the right end surface. Again, the averaged stress and strain are computed by the same method as the previous example. The damage was initiated at step 14 ($\sigma_{\text{ave}} = 344.7$ MPa). However, the influence to the curve was not obvious. Then, at step 18 ($\sigma_{\text{ave}} = 564.1$ MPa) the curve starts to change obviously and quickly loses the stiffness. From Figure 2.20, one can observe that the FE-PDNN agrees

well with the observation data. This indicates that it is possible to couple the lamination theory with FE-PDNN to learn the lamina constitutive law.

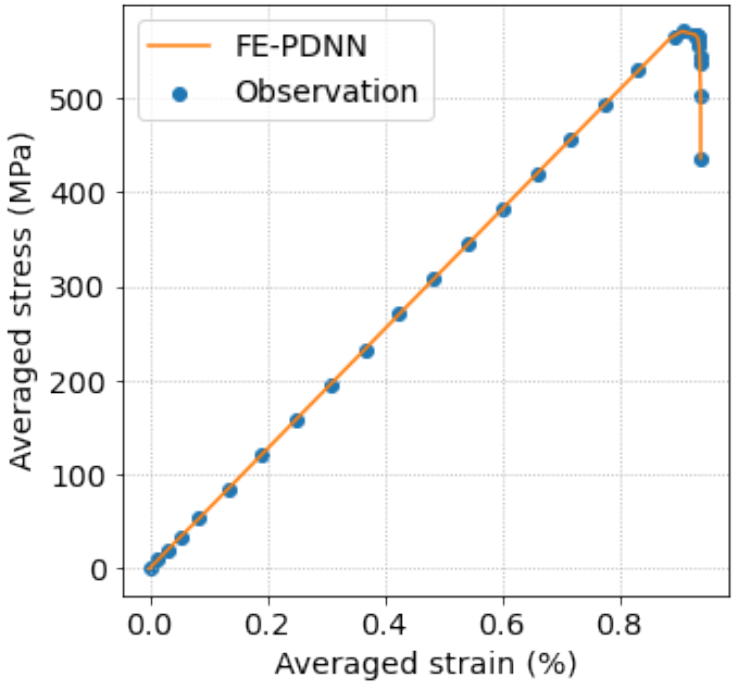
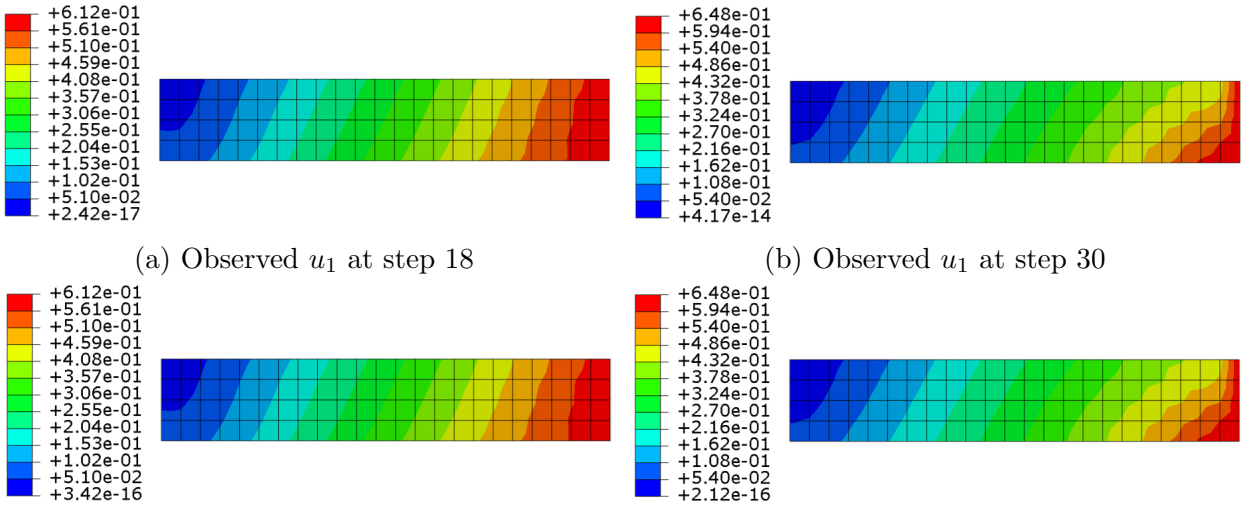


Figure 2.20. The plot of averaged stress versus strain of the $[0/45]_s$ laminate

This observation can be confirmed by Figures 2.21 and 2.22, which plot the displacement and stress at the top layer of the laminate. Figures 2.21a and 2.21b show the observed displacement at step 18 ($\sigma_{\text{ave}} = 564.1$ MPa) and 30 ($\sigma_{\text{ave}} = 430.0$ MPa). As one can tell, the displacement is not uniform in the laminate. This is because the ply orientations of the laminate are not uniform. The 45 degree ply generates shear stress in the laminate. Besides, with the increasing load, the displacement contour shape has been changed. This is due to the damage in the laminate. Figures 2.21c and 2.21d present the displacement learned by FE-PDNN, which applied the positive-definite constraint to the stiffness predicted by DNN. As shown in Figures 2.21c and 2.21d, all the contour plots match the observed plots very well. The Mises stress result is presented in Figure 2.22. Figures 2.22a and 2.22b show the Mises stress of the observation data. In Figure 2.22a, the maximum Mises stress is at the upper right and bottom left corner. However, the maximum stress in Figure 2.22b has changed to the bottom right corner. This change was due to the severe damage at the upper

right corner, as the severely damaged elements take less load under the same displacement. These results demonstrate that coupling lamination theory with FE-PDNN enables DNN to learn the lamina constitutive law based on the laminate structural response. Figures 2.22c and 2.22d present the stresses learned by FE-PDNN. One can tell that FE-PDNN accurately predicts the stress. There is a slight difference between observed and learned stress at the last step. However, the overall contour shape and the maximum stress match each other very well. Again, a training was done using FE-DNN. The result is presented in Figure 2.23. The result shows that FE-DNN failed to learn the constitutive law at step 30, as the contour plots of displacement and stress cannot match the observed result. The results of Mises stress have a worse agreement than those of the displacements.



(a) Observed u_1 at step 18

(b) Observed u_1 at step 30

(c) FE-PDNN learned u_1 at step 18

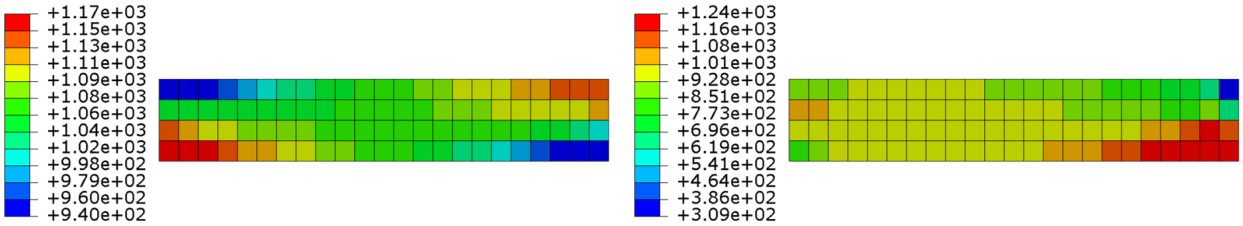
(d) FE-PDNN learned u_1 at step 30

Figure 2.21. Comparison of the contour plots of u_1 of the $[0/45]_s$ laminate

2.4.3 Example 4: Learn lamina damage constitutive law based on a laminate made of E-glass/MY750 material using experimental data

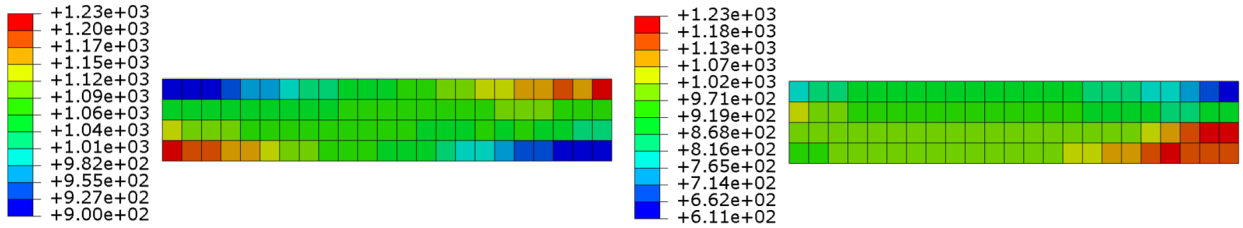
Example 4 problem description

In this example, we fed the experimental data into the lamination theory enhanced FE-PDNN to learn lamina constitutive law. The experimental tests were carried out by the Defence Evaluation and Research Agency (DERA) and collected in the World-Wide Failure



(a) Observed Mises stress at step 18

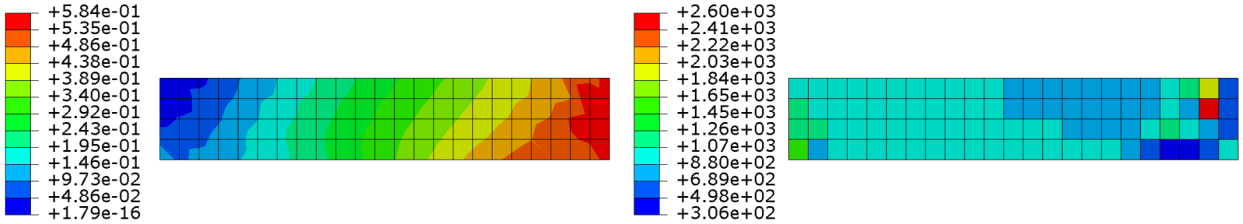
(b) Observed Mises stress at step 30



(c) Learned Mises stress at step 18

(d) Learned Mises stress at step 30

Figure 2.22. Comparison of the contour plots of Mises stress of the $[0/45]_s$ laminate



(a) FE-DNN learned u_1 at step 30

(b) FE-DNN learned Mises stress at step 30

Figure 2.23. The learning result of FE-DNN for the $[0/45]_s$ laminate

Exercise I [98]. The laminate coupons were made of E-glass/MY750 lamina with layup angles $[0/90]_s$. All the plies have an identical thickness of 0.475 mm and the overall thickness of the laminate is 1.9 mm. The coupons are 25 mm wide, 200 mm long and a gauge length of 100 mm, which leaves a 50 mm distance at each end for bonding and taps. The width of the FE model was the same as the specimen, the length was set to be 100 mm. The boundary conditions are the same as the example in Section 2.3.2, which is shown in Figure 2.13. The lamina properties and strength parameters are listed in Table 2.4. A total of five uniaxial tension tests were carried out. The initial failure was recorded at a strain of $\varepsilon_x = 0.375\%$, which corresponds to 117.5 MPa stress on the coupon. The mean failure strain in the loading direction was $\varepsilon_x = 2.69\%$ with 590 MPa stress. The coupons finally failed by

fiber fracture. A typical averaged stress-strain curves is shown in Figure 2.24. The E_1 and ν_{12} are re-calibrated based on this curve.

Table 2.4. The lamina properties and strength parameters of E-glass/MY750 [98], [99]

	Material properties
Elastic properties:	
E_1(MPa)	$42,000$
E_2(MPa)	$16,200$
G_{12}(MPa)	$5,830$
ν_{12}	0.12
Tensile strengths:	
X(MPa)	$1,280$
Y(MPa)	40
Compressive strengths:	
X'(MPa)	800
Y'(MPa)	145
Shear strengths:	
S(MPa)	73
T(MPa)	73

From Eq. (2.37), we can calculate the plate stiffness matrix as

$$\begin{aligned} \boldsymbol{A} &= \begin{bmatrix} 55598.80 & 3714.23 & 0 \\ 3714.23 & 55598.80 & 0 \\ 0 & 0 & 11077.00 \end{bmatrix} \text{(N/mm)} \\ \boldsymbol{D} &= \begin{bmatrix} 22286.90 & 1117.36 & 0 \\ 1117.36 & 11165.00 & 0 \\ 0 & 0 & 3332.33 \end{bmatrix} \text{(N} \cdot \text{mm)} \end{aligned} \tag{2.46}$$

The $\boldsymbol{B}$ matrix vanishes due to the symmetry of the laminate, which imply that the extension-bending and shear-twist couplings should vanish. Besides, $A_{16} = A_{26} = D_{16} = D_{26} = 0$. This indicates that there are no extension-shear, extension-twist, and bending-twisting couplings for this laminate. Thus, the strain is uniform in each layer of the laminate

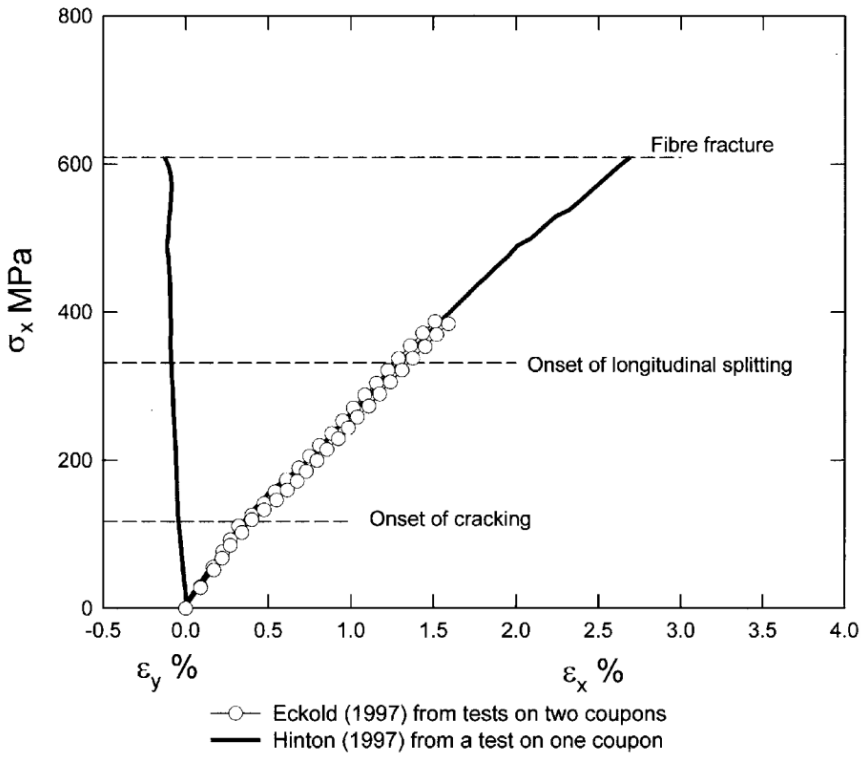


Figure 2.24. The averaged strain-stress curves for $[0/90]_s$ E-glass/MY750 laminate under uniaxial tensile loading with $\sigma_x : \sigma_y = 1 : 0$ [98]

under uniaxial tensile loading. We can take advantage of this fact to just use the averaged strain-stress data to train FE-PDNN to discover the constitutive law of lamina. The force and displacement can be reproduced using the stress and strain multiplying the corresponding area ($A = 47.5\,\text{mm}^2$) and length ($l = 100$ mm) of the laminate, respectively.

The FE-PDNN can determine the strain-stress relationship of lamina based on the laminate structural response. But the experimental data of strain and stress in each layer are not available, as the experiment only provides the averaged strain-stress relation.

To provide a comparison to FE-PDNN learned strain-stress relation, an analytical progressive failure analysis was performed. The lamina was assumed to fail according to the plane-stress simplified Tsai-Wu failure criterion

$$f = F_1\sigma_{11} + F_2\sigma_{22} + F_{11}\sigma_{11}^2 + F_{22}\sigma_{22}^2 + 2F_{12}\sigma_{11}\sigma_{22} + F_{66}\sigma_{12}^2 \tag{2.47}$$

where

$$F_1 = \frac{1}{X} - \frac{1}{X'}, \qquad F_2 = \frac{1}{Y} - \frac{1}{Y'}$$
$$F_{11} = \frac{1}{XX'}, \qquad F_{22} = \frac{1}{YY'}$$
$$F_{66} = \frac{1}{S^2}, \qquad F_{12} = -\frac{1}{2}\frac{1}{XX'}$$

The accuracy of the analytical failure analysis highly depends on the strength parameters. Besides, the degradation factor, which is used to calculate the residual stiffness after the initial failure of a material, is commonly used in the damage analysis [27], [99]. For this example, the strength parameters are selected following the WWFE I data [98]. The degradation factor is set to be 0.1, which is calibrated based on the averaged stress-strain data.

A constitutive model should not only be able to match the existing observations but also able to make blind predictions. The previous examples have shown the capability of the FE-PDNN mechanics system to learn unknown constitutive laws based on limited structured level response. But the blind prediction capability of the mechanics system has not been demonstrated yet. To test the blind prediction capability, the virtual tests were performed via analytical failure analysis, which is to imitate the first round training of FE-PDNN to generate the correct strain-stress data.

The tests assumed that the laminate has the same geometry and boundary conditions as the example in Section 2.4.2. Besides, the laminate is made of the E-glass/MY750 material and fails according to the Tsai-Wu failure criterion. According to Eqs. (2.43) and (2.44), the change of layup angle can result in extension-shear couplings Q_{16} and Q_{26}, which can generate inhomogeneous stress fields in the laminate. The Latin Hypercube technique is used to sample the layup angles. For demonstration purposes, 40 total layup angles are selected in the range from 0 to 90 degrees. We avoided the angles of 10, 25, and 45 degrees intentionally, as the blind prediction was performed based on a $[10/25/45]_s$ laminate. The analytical failure analysis was performed to compute the strain-stress in the material coordinate system at

various layup angles. For simplicity, we assumed that once a layer failed then the stiffness matrix of that layer is degraded to be zero. A DNN model would be trained based on these data and perform the progressive damage blind prediction of the $[10/25/45]_s$ laminate.

The averaged strain-stress response and the strength ratio α of each layer were used to evaluate the performance of the blind prediction. The strength ratio is defined in the form [32]

$$f\left(\boldsymbol{\sigma}_{\mathrm{cr}}\right) = f(\alpha\boldsymbol{\sigma}) = 1 \tag{2.48}$$

where $f(\cdot)$ denotes the failure criterion, $\boldsymbol{\sigma}_{cr}$ is the critical stresses, $\boldsymbol{\sigma}$ is the current stress state. For the Tsai-Wu failure criterion, since it contains both linear and quadratic terms of the stress components, we will have

$$f\left(\alpha\boldsymbol{\sigma}\right) = \alpha^2 a + \alpha b = 1 \tag{2.49}$$

Thus, the strength ratio can be calculated as

$$\alpha = \frac{-b \pm \sqrt{b^2 + 4a}}{2a} \tag{2.50}$$

where

$$\begin{aligned} a =& F_{11}\sigma_{11}^2 + F_{22}\sigma_{22}^2 + F_{66}\sigma_{12}^2 + 2F_{12}\sigma_{11}\sigma_{22} \\ b =& F_1\sigma_{11} + F_2\sigma_{22} \end{aligned} \tag{2.51}$$

Eq. (2.50) has two solutions. The strength ratio will take the smaller positive one of the two solutions. A material will be safe if $\alpha > 1$ and will fail if $\alpha \leq 1$.

Example 4 results and discussion

Figure 2.25 plots the averaged strain-stress relationship. The plots show FE-PDNN matches the experimental data very well. Besides, the analytical progressive failure analysis also shows good agreement with the experimental data. This result is expected, as the accuracy of the analytical approach mainly depends on the selection of damage parameters. For this example, the damage parameters are calibrated from experiments data. Figure 2.25

also shows that FE-PDNN can predict the global response more accurately comparing with the analytical failure analysis.

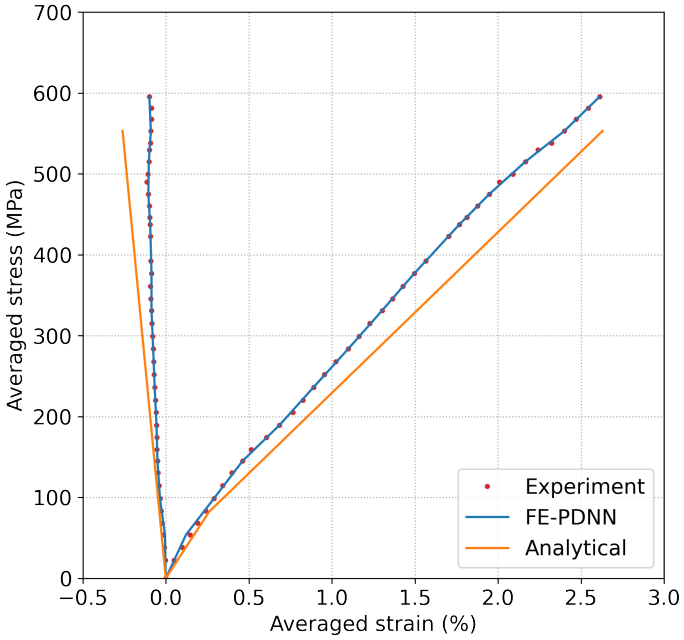


Figure 2.25. Averaged stress versus strain of the $[0/90]_s$ laminate

Figure 2.26 presents σ_{11} versus ε_{11} at the 0-degree layer. One can tell that the analytical analysis shows a linear relationship. This is because the 0-degree layer is the last layer that failed during the loading. The initial failure stress of this layer would be the ultimate stress of the laminate. Thus, the stress-strain curve of the analytical result at the 0-degree layer shows a linear relationship. Figure 2.26 also shows that the stress-strain relationships between the two methods are close, with the maximum difference between the two methods is about 10%. This result confirmed the feasibility of learning constitutive law with FE-PDNN.

The averaged strain-stress curve of $[10/25/45]_s$ laminate is shown in Figure 2.27. The result shows that although the strain-stress data of 10, 25, and 45 degrees layers are not used for training the DNN, DNN still predicted the averaged strain-stress very well, as the errors between DNN blind prediction and the target solution are within 2%. This indicates DNN can provide an excellent prediction for the global response of the laminate.

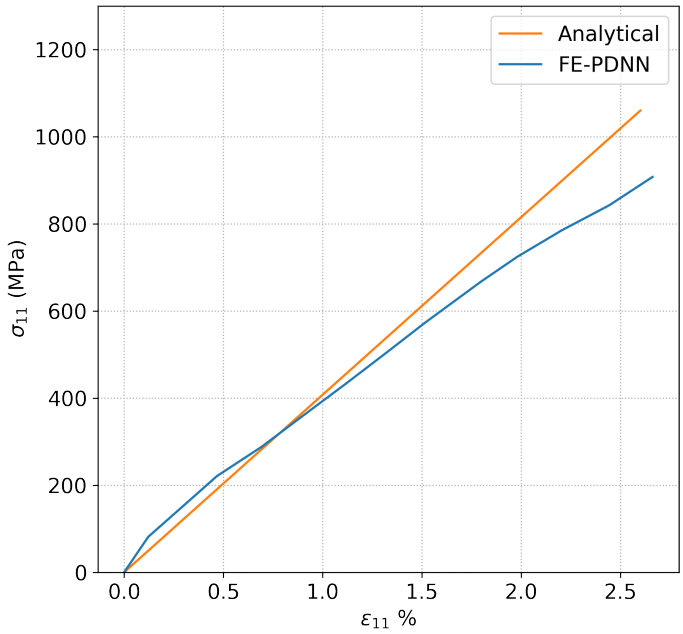


Figure 2.26. Stress versus strain of the 0-degree layer

Figure 2.28 shows the strength ratio of different layers. From this plot, one can tell that the 45 degrees layer failed first since the strength ratio of this layer reached to one with $N_{11} = 252$ N/mm, which is the initial failure extension force. Next, the 25-degrees layer failed with $N_{11} = 313$ N/mm. Finally, the 10-degrees layer reached ultimate stress with $N_{11} = 365$ N/mm. This failure sequence is expected, as the change of layup angles from 10 to 45 degrees will increase the extension-shear coupling effect. Considering that the shear strength parameter is significantly smaller compared to other strength parameters, the ply that has larger shear stress will fail earlier. From this plot, one can also observe that the strength ratio predicted by DNN has an excellent match with the analytical result. Since the strength ratio is calculated based on the stress in each layer, it indicates that DNN predicted the stress in each layer accurately.

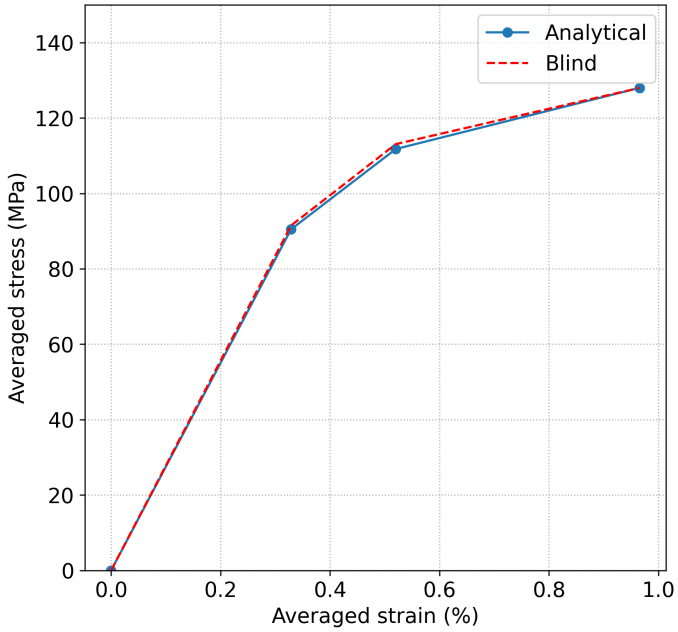


Figure 2.27. The plot of averaged stress versus strain of the $[10/25/45]_s$ laminate

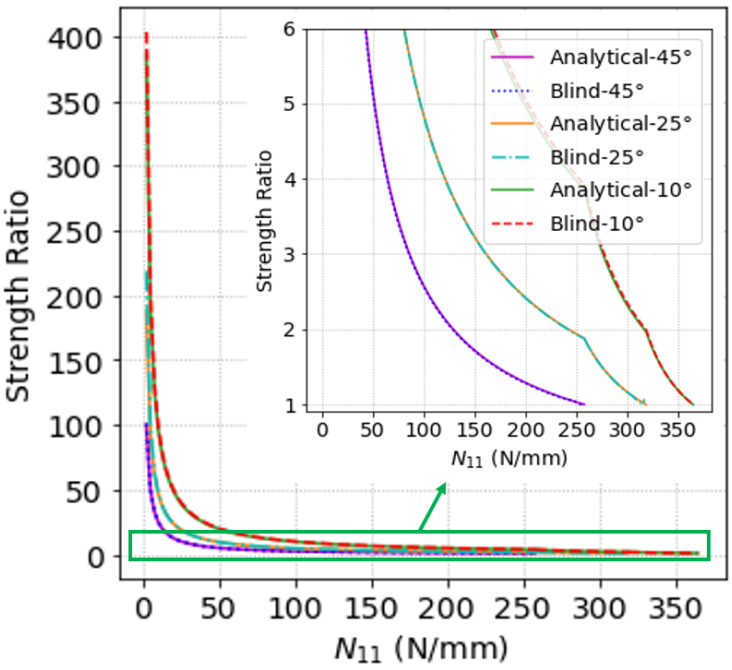


Figure 2.28. The plot of strength ratio of the $[10/25/45]_s$ laminate

3. DISCOVER FAILURE CRITERIA OF COMPOSITES WITH MACHINE LEARNING

Over the past few years, a few experimental failure data of composites have been collected. It would be of interest to leverage the existing data to improve the prediction of failure criteria. In this chapter, we developed to implement a framework that combines sparse regression with compressed sensing to discover failure criteria from data. Following the phenomenological failure models, we divided the failure of composites into tensile and compressive fiber modes, tensile and compressive matrix modes. Besides, we used an optimization approach to generate a conservative failure criterion and satisfy the engineering design needs. Three examples were studied with this framework. The first example was presented to demonstrate the capability of the framework. The data was generated by the Hashin failure criterion and added various degrees of noise. The proposed framework was implemented to discover the failure criterion from the noised data. For the second example, the proposed method was used to discover failure criteria from the experimental data which are collected from the WWFE I. In the third example, we used an optimization approach to enforce a constraint so that the predicted results are smaller than the majority of the experimental data. The results show that the proposed method can discover the failure criterion accurately. In addition, with the enforced constraint, the proposed method can yield a conservative failure criterion.

3.1 Theory of Sparse Regression and Compressed Sensing

The regression analysis is a process to estimate the relationship between an outcome variable and independent variables (also called features) [100]. The regression of a system can be expressed as

$$\mathbf{y} = \boldsymbol{\Theta}(\mathbf{X})\,\xi \tag{3.1}$$

where $\mathbf{y} \in \mathbb{R}^{m\times n}$ is the measurements of the outcome variables, $\boldsymbol{\Theta}(\mathbf{X}) \in \mathbb{R}^{m\times p}$ is a matrix that includes all the candidate functions, which are formed from the linear combination of the features, $\xi \in \mathbb{R}^{p\times n}$ is the matrix of coefficients.

For a system with n features and m states, the features in each state can be expressed as $\mathbf{x} = [x_1\ x_2\ \cdots\ x_n]^T \in \mathbb{R}^n$. Then, $\mathbf{X}$ can be explicitly expressed as

$$\mathbf{X} = \begin{bmatrix} (\mathbf{x}^1)^T \\ (\mathbf{x}^2)^T \\ \vdots \\ (\mathbf{x}^m)^T \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_n^1 \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^m & x_2^m & \cdots & x_n^m \end{bmatrix} \tag{3.2}$$

where x_i^j is the ith feature in jth state. $\mathbf{y}$ can be expressed as

$$\mathbf{y} = \begin{bmatrix} (\mathbf{y}^1)^T \\ (\mathbf{y}^2)^T \\ \vdots \\ (\mathbf{y}^m)^T \end{bmatrix} = \begin{bmatrix} y_1^1 & y_2^1 & \cdots & y_n^1 \\ y_1^2 & y_2^2 & \cdots & y_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ y_1^m & y_2^m & \cdots & y_n^m \end{bmatrix} \tag{3.3}$$

where y_i^j is the ith outcome in jth state. Next, we can form $\boldsymbol{\Theta}(\mathbf{X})$ which includes the candidate functions

$$\boldsymbol{\Theta}(\mathbf{X}) = [\mathbf{X}^0 \quad \mathbf{X} \quad \mathbf{X}^{P_2} \quad \mathbf{X}^{P_3} \quad \cdots \quad \sin(\mathbf{X}) \quad \cos(\mathbf{X}) \quad \sin(2\mathbf{X}) \quad \cos(2\mathbf{X}) \quad \cdots\ ] \tag{3.4}$$

where $\mathbf{X}^0$ and $\mathbf{X}$ are the zero-order and first-order candidate function matrix respectively. $\mathbf{X}^{P_i}$ represents higher-order candidate function matrix. For example, the quadratic nonlinearities $\mathbf{X}^{P_2}$ is

$$\mathbf{X}^{P_2} = \begin{bmatrix} (x_1^1)^2 & x_1^1 x_2^1 & \cdots & (x_2^1)^2 & x_2^1 x_3^1 & \cdots & (x_n^1)^2 \\ (x_1^2)^2 & x_1^2 x_2^2 & \cdots & (x_2^2)^2 & x_2^2 x_3^2 & \cdots & (x_n^2)^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ (x_1^m)^2 & x_1^m x_2^m & \cdots & (x_2^m)^2 & x_2^m x_3^m & \cdots & (x_n^m)^2 \end{bmatrix} \tag{3.5}$$

For many physics systems, only a few terms are important, which makes ξ sparse in the space of the possible candidate functions. However, a standard regression to solve for ξ will

yield a nonzero solution for each element. Thus, it is hard to use the standard regression approach to identify the governing equations.

If ξ is sparse, then one method to identify the important entries in ξ is to add the L_1 regularization to the regression to form a Least Absolute Shrinkage and Selection Operator (LASSO) [101], [102]

$$\min_{\xi}\{(\mathbf{\Theta}(\mathbf{X})\xi - \mathbf{y})^2 + \lambda \|\xi\|\} \tag{3.6}$$

where λ is a coefficient that tunes the weights of the candidate functions. LASSO shrinks the coefficients of the unimportant functions to zero and thus can remove the influence of these functions completely. The process to find ξ is closely related to compressed sensing, which is a popular technique in reconstructing high-fidelity signals using limited measurements [103], [104]. Compressed sensing will normalize $\mathbf{\Theta}(\mathbf{X})$ to ensure the restricted isometry property which provides a guide for finding the measurement matrix for compressive sensing [103], [105]. For example, if $\mathbf{\Theta}(\mathbf{X})$ is normalized by dividing elements in each column by the L_2 norm of the column

$$(\mathbf{\Theta}')_{ij} = (\mathbf{\Theta})_{ij}/L_2(j) \tag{3.7}$$

with

$$L_2(j) = \sqrt{\sum_{i=1}^{M}[(\mathbf{\Theta})_{ij}]^2} \tag{3.8}$$

Then, $\mathbf{y}$ can be expressed as

$$\mathbf{y} = \mathbf{\Theta}' \cdot \xi' \tag{3.9}$$

with

$$\xi' = \xi \cdot L_2 \tag{3.10}$$

As a result, ξ is given as

$$\xi = \xi'/L_2 \tag{3.11}$$

3.2 Discover Failure Criterion of Composites Using Spare Regression with Compressed Sensing

Generally speaking, if a failure criterion is stress-based, it can be expressed as

$$f(\sigma) = 1 \tag{3.12}$$

where $f(\cdot)$ represents the failure function. The purpose of the failure criterion is to predict the failure of the material under a certain loading state.

Hashin failure criterion distinguishes the failure modes of the unidirectional lamina into four failure modes, namely, tensile and compressive fiber failure modes, and tensile and compressive matrix failure modes. The equations of the four modes under a plane-stress state are given as follows:

Tensile fiber mode ($\sigma_{11} \geq 0$)

$$f = \left(\frac{\sigma_{11}}{X}\right)^2 + \left(\frac{\sigma_{12}}{S}\right)^2 = 1 \tag{3.13}$$

Compressive fiber mode ($\sigma_{11} < 0$)

$$f = \frac{\mid \sigma_{11} \mid}{X'} = 1 \tag{3.14}$$

Tensile matrix mode ($\sigma_{22} \geq 0$)

$$f = \left(\frac{\sigma_{22}}{Y}\right)^2 + \left(\frac{\sigma_{12}}{S}\right)^2 = 1 \tag{3.15}$$

Compressive matrix mode ($\sigma_{22} < 0$)

$$f = \left[\left(\frac{Y'}{2R}\right)^2 - 1\right]\frac{\sigma_{22}}{Y'} + \left(\frac{\sigma_{22}}{2R}\right)^2 + \left(\frac{\sigma_{12}}{S}\right)^2 = 1 \tag{3.16}$$

where X and Y are tensile strengths in the principal directions of the material (x_1, x_2 respectively), X' and Y' are the compressive strengths, R and S are shear strengths in the planes of symmetry ($x_2 - x_3$, $x_1 - x_2$ respectively).

To implement the sparse regression, we followed the Hashin failure criterion to divide the failure of composites into tensile and compressive fiber modes, tensile and compressive matrix modes. For the fiber failure modes, we assumed that the independent features are σ_{11} and σ_{12}. While for matrix failure modes, the independent features are σ_{22} and σ_{12}. It is needed to point out that the assumption of the independent features is not a constraint for sparse regression. If the data of other features are available, it is possible to include them in the analysis. For example, for the compressive fiber failure mode, we use σ_{11} and σ_{12} as the independent features. If the data of the influence from σ_{22} is available, the sparse regression can also incorporate σ_{22} as a feature.

Based on these settings, we can set the features $\mathbf{X} = [\sigma_{11}\ \sigma_{22}\ \sigma_{12}]$, then the collection of nonlinear functions with the features $\mathbf{\Theta}(\mathbf{X})$ can be constructed. Since the stresses are measured at the failure state, we have all the elements in f are equal to one. Thus, we can set up a sparse regression problem to determine the sparse vectors of coefficients as

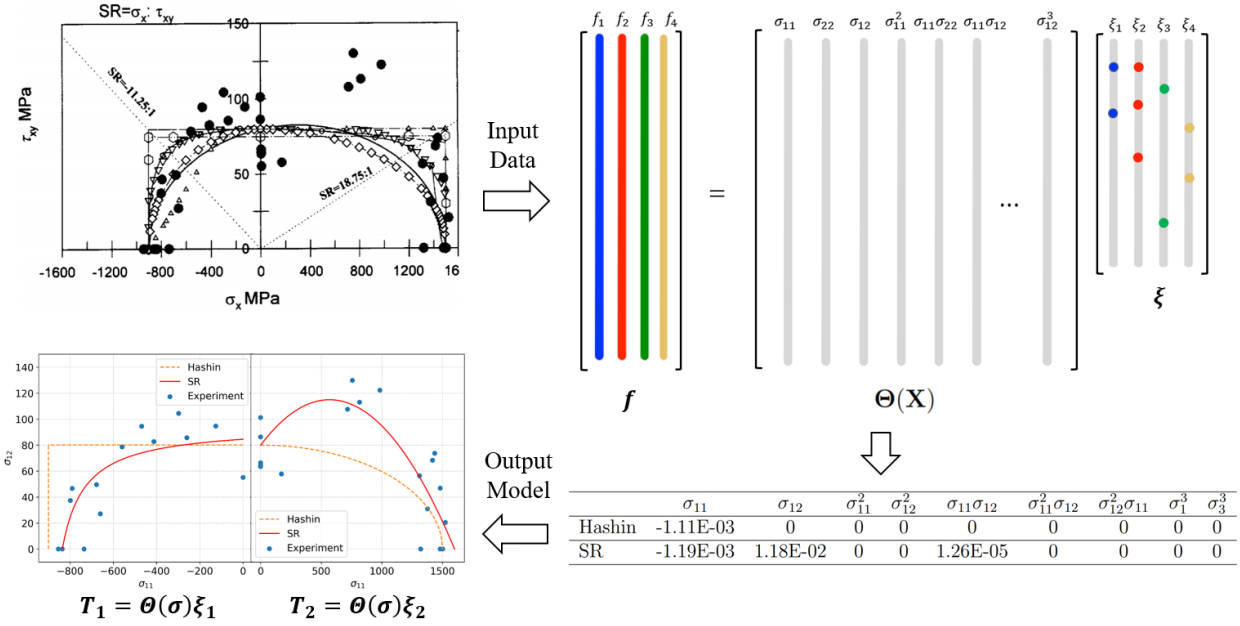
$$\mathbf{f} = \mathbf{\Theta}(\mathbf{X})\xi \tag{3.17}$$

where ξ is the collection of the sparse vectors of coefficients of different failure modes. Eq. (3.17) can be illustrated in Figure 3.1.

The value of an element in ξ_i indicates whether a corresponding candidate function in $\mathbf{\Theta}(\mathbf{X})$ is active or not. Besides, since we assumed that fiber modes and matrix modes are governing by different features, the values of the element in ξ_i that are corresponding to the non-influential candidate functions shall be zero. Thus, Eq. (3.17) can be further simplified to the form shown in Figure 3.2. In Figure 3.2, the regression of different failure modes has been separated. f_i is the failure index, which equals one at failure state. $\mathbf{\Theta}(\mathbf{X})$ matrix has been reduced to only include influential candidate functions. Once ξ_i is determined, the governing equation for each failure mode can be constructed as follows

$$\mathbf{f}_i = \mathbf{\Theta}_i\left(\mathbf{X}\right)\xi_i \tag{3.18}$$

It is noted that summation convention is not applicable to Eq. (3.18).



	σ_{11}	σ_{12}	σ_{11}^2	σ_{12}^2	$\sigma_{11}\sigma_{12}$	$\sigma_{11}^2\sigma_{12}$	$\sigma_{12}^2\sigma_{11}$	σ_1^3	σ_3^3
Hashin	-1.11E-03	0	0	0	0	0	0	0	0
SR	-1.19E-03	1.18E-02	0	0	1.26E-05	0	0	0	0

Figure 3.1. The schematic of sparse regression to discover failure criterion

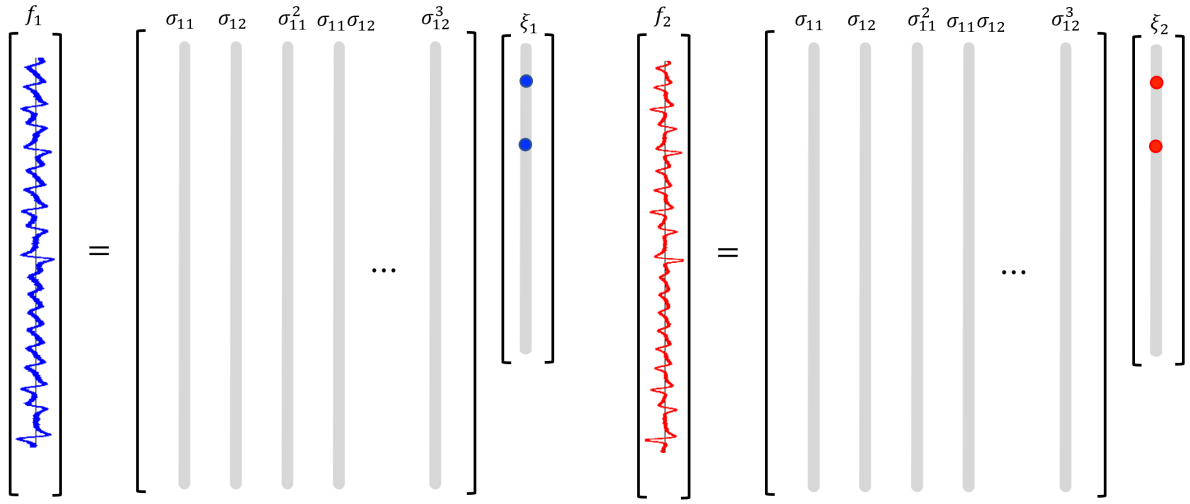
3.3 Applications

This section presented three examples based on sparse regression with compressed sensing. The first example used sparse regression to demonstrate its capability to discover the failure criterion. The Hashin failure criterion was used to generate the test data. Then various degrees of noise was added to the test data. For the second example, the proposed method was used to discover the failure criteria from the experimental data from WWFE I. For the third example, we used an optimization approach to obtain a conservative criterion which enforced a constraint to the discovered criterion to shift the predicted data to be smaller than the experimental data.

3.3.1 Example 5: Discover the failure criterion from noisy analytical data

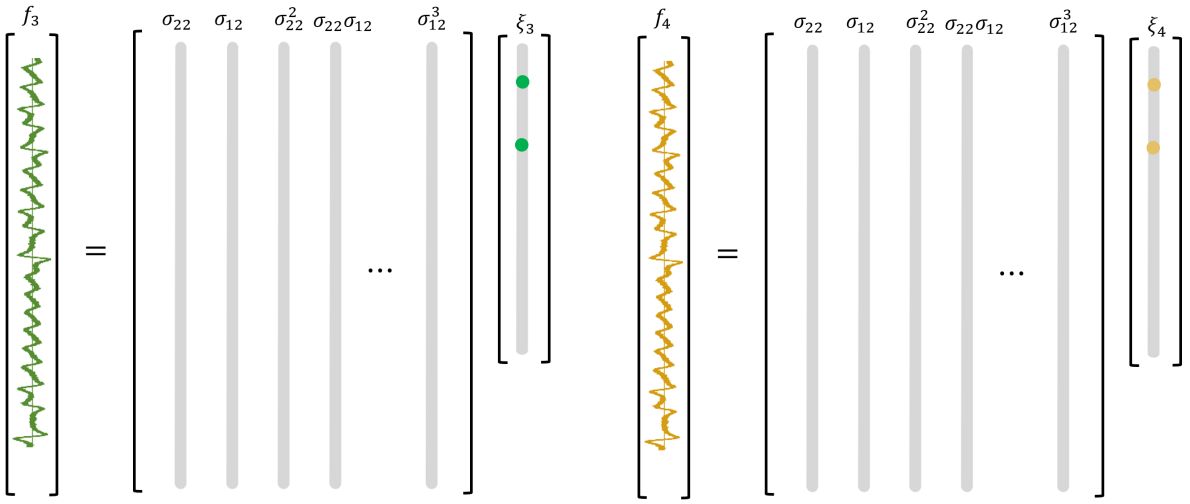
Example 5 problem description

In this example, we used the Hashin failure criterion to generate the data of the failure envelope. Then, these data are divided into four groups according to their failure modes. Realistically, the measurements of stresses from experiments contain noise, which could be



(a) Tensile fiber mode

(b) Compressive fiber mode



(c) Tensile matrix mode

(d) Compressive matrix mode

Figure 3.2. Sparse regression to solve for active terms of different failure modes

caused by many factors, such as loading imperfection, geometry differences of specimens, sensor measurement errors, etc. Thus, we added a certain degree of noise to the data to mimic the real situation. Then, Eq. (3.18) does not hold exactly, which can be modified to

$$\mathbf{f}_i = \mathbf{\Theta}_i(\mathbf{X})\,\xi_i + \eta\mathbf{Z} \tag{3.19}$$

where $\mathbf{Z}$ is modeled as a matrix of independent and identically distributed Gaussian entries with zero mean and noise magnitude η. Next, the noised data will be transferred to sparse regression to determine the failure criterion governing equations. To solve for the coefficients vector ξ_i, one method is to use LASSO Eq. (3.6), which applies L_1 regularization that promotes sparsity and works well with noised data. However, LASSO is computationally expensive for large data sets. Thus, following Brunton et. al. [17], we applied the sequential threshold least-squares method to solve the coefficient vectors. The algorithm of the code is presented in Appendix A.3. It is needed to note that the previous studies to identify governing equations from data have shown to be quite sensitive to noise [103], [106]. In contrast, the algorithm presented in Appendix A.3 is remarkably robust to noise [17].

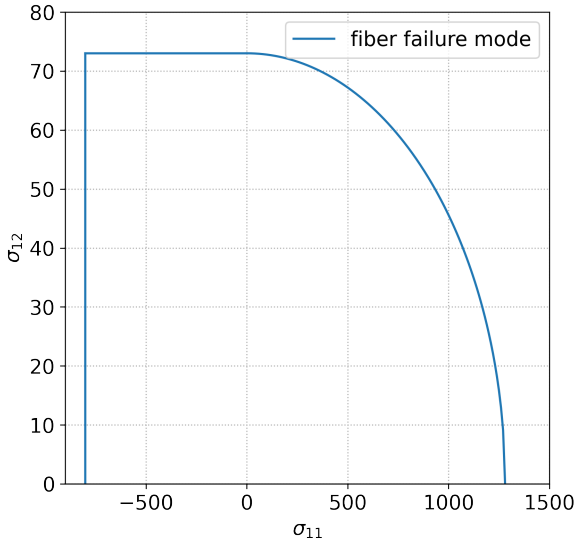
The E-Glass/MY750 composite is selected for demonstration. The strength properties of this material are listed in Table 3.1 [107]. Based on these strength parameters and the Hashin failure criterion, the failure envelope is shown in Figure 3.3.

Table 3.1. Strength parameters of E-Glass/MY750

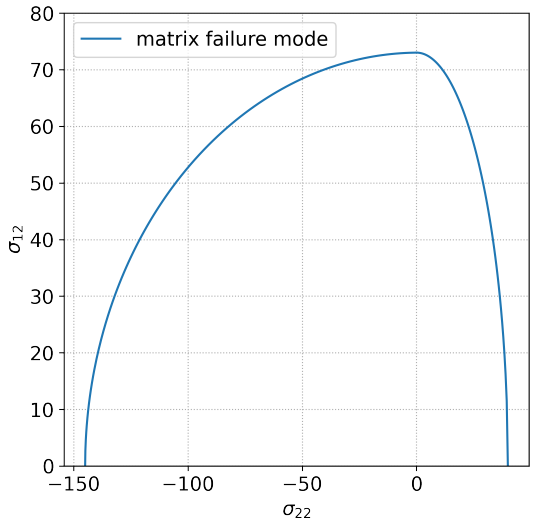
X(MPa)	X'(MPa)	Y(MPa)	Y'(MPa)	S(MPa)	R(MPa)
1280	800	40	145	73	50

Example 5 results and discussion

Figure 3.3 shows that the failure envelope of the compressive fiber mode is formed by two straight lines. This is because the Hashin failure criterion assumed the compressive fiber mode can be determined by the single value of X', which is shown in Eq. (3.14). Since the envelope of the compressive fiber mode is simple and can be easily identified from sparse



(a) Fiber failure mode

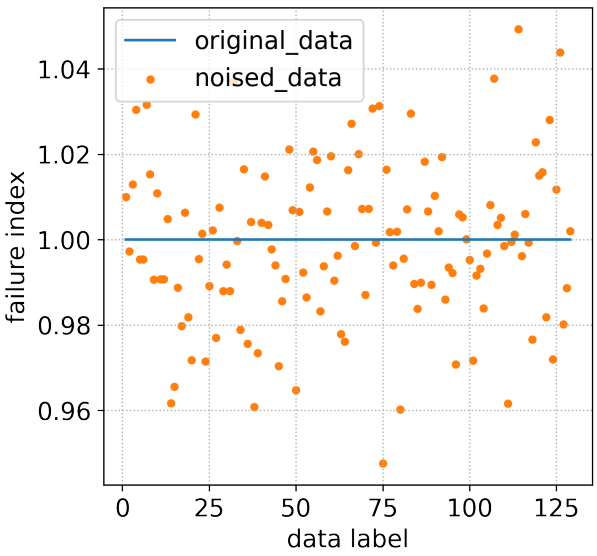


(b) Matrix failure mode

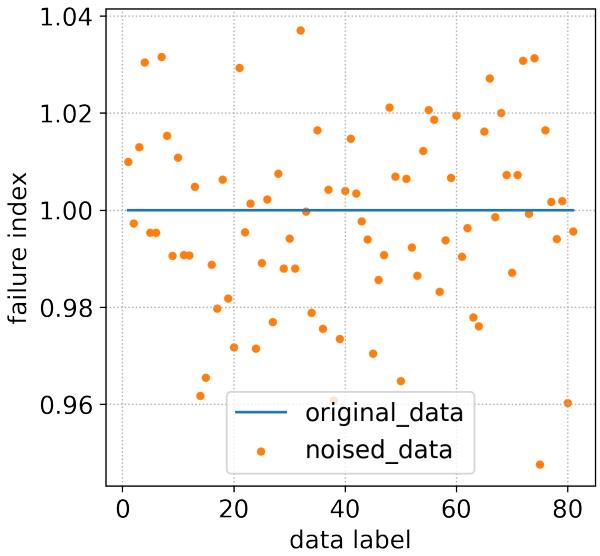
Figure 3.3. Failure envelops of E-glass/MY750 according to Hashin failure criterion

regression, for this example, we will focus on the analysis of other failure modes. Besides, since the measured stresses from experiments have noise, it indicates that the failure index may not be one with the measured stresses. Thus, for this demonstration example, we added zero mean Gaussian noise with variance η to the failure indices to test the robustness of the method according to Eq. (3.19). Figure 3.4 shows the distribution of the failure indices with $\eta = 0.02$.

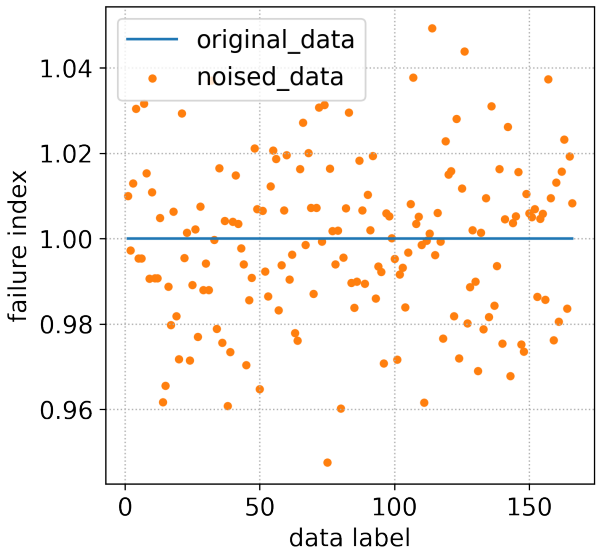
As one can tell from tables 3.2 to 3.4, although the higher-order candidate functions and stresses coupling candidate functions are selected, the sparse regression and compressed sensing method identified these candidate functions are inactive, and their coefficients are zero. This result shows the advantage of the proposed method over the traditional regression, as the traditional method has difficulty identifying inactive candidate functions, as it typically results in coefficients for all the candidate functions. Besides, one can also tell that coefficients of the active candidate functions for tensile fiber and tensile matrix modes are identified accurately since the differences between the exact values and spare regression identified values are within 1.05 %. The accuracy can be further confirmed from the plots of



(a) Tensile fiber mode



(b) Tensile matrix mode



(c) Compressive matrix mode

Figure 3.4. Comparison between original failure indices and noised failure indices with $\eta = 0.02$

the failure envelope in Figures 3.5a and 3.5b. Figures 3.5a and 3.5b show that the exact and identified failure envelopes are very close, with the maximum differences within 1%. For the compressive matrix failure mode, while the coefficients of the quadratic terms are identified well, the coefficient of the linear term is identified with 6.67% difference. The reason for the linear term identification being less accurate is that the linear term of σ_{22} has less influence

on the failure index for this material. This can be confirmed from the plot of the failure envelope in Figure 3.5c, which shows that the exact and identified failure envelopes are very close with the difference smaller than 1%.

The identified coefficients of the failure criterion is presented in Tables 3.2, 3.3, and 3.4. In these tables, the error of nonzero coefficients are computed as

$$\text{error} = \frac{|\ \text{Exact value} - \text{Identified value}\ |}{\text{Exact value}} \times 100 \tag{3.20}$$

Table 3.2. Comparison between exact and identified coefficients of the tensile fiber mode

	σ_{11}	σ_{12}	σ_{11}^2	σ_{12}^2	$\sigma_{11}\sigma_{12}$	$\sigma_{11}^2\sigma_{12}$	$\sigma_{12}^2\sigma_{11}$	σ_{11}^3	σ_{12}^3
Exact value	0	0	6.104E-07	1.877E-04	0	0	0	0	0
Identified value	0	0	6.040E-07	1.878E-04	0	0	0	0	0
Error (%)	0	0	1.046	0.522	0	0	0	0	0

Table 3.3. Comparison between exact and identified coefficients of the tensile matrix mode

	σ_{22}	σ_{12}	σ_{22}^2	σ_{12}^2	$\sigma_{22}\sigma_{12}$	$\sigma_{22}^2\sigma_{12}$	$\sigma_{12}^2\sigma_{22}$	σ_{22}^3	σ_{12}^3
Exact value	0	0	6.250 E-04	1.877 E-04	0	0	0	0	0
Identified value	0	0	6.249 E-04	1.870 E-04	0	0	0	0	0
Error (%)	0	0	0.0144	0.367	0	0	0	0	0

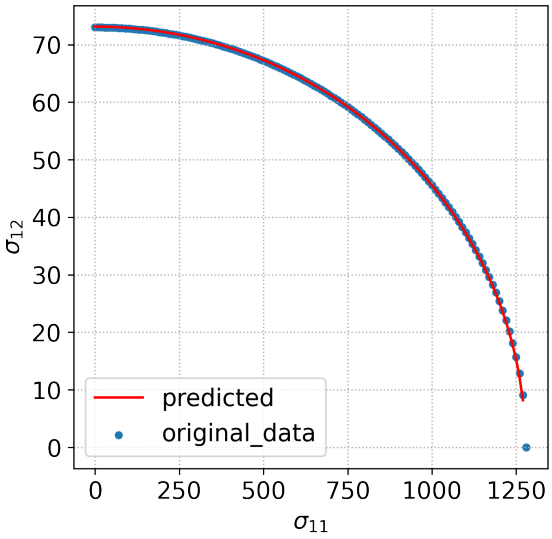
Table 3.4. Comparison between exact and identified coefficients of the compressive matrix mode

	σ_{22}	σ_{12}	σ_{22}^2	σ_{12}^2	$\sigma_{22}\sigma_{12}$
Exact value	-9.415 E-05	0	4.691E-05	1.877 E-04	0
Identified value	-8.787 E-05	0	4.702 E-05	1.872 E-04	0
Error (%)	6.673	0	0.234	0.2675	0
	$\sigma_{22}^2\sigma_{12}$	$\sigma_{12}^2\sigma_{22}$	σ_{22}^3	σ_{12}^3	
Exact value	0	0	0	0	
Identified value	0	0	0	0	
Error (%)	0	0	0	0	

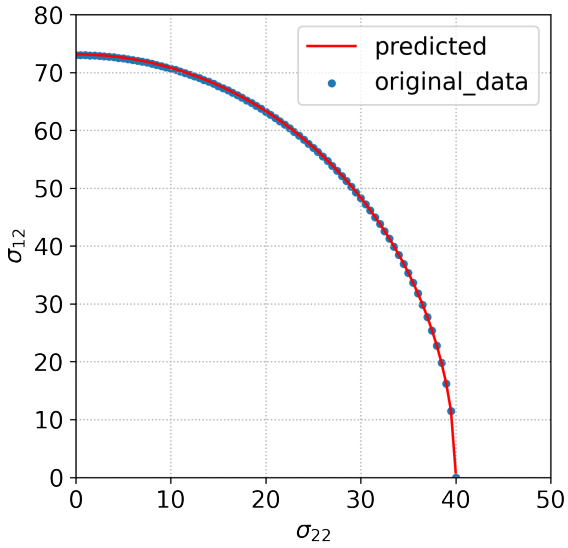
Figure 3.6 shows the plots of the exact and identified failure envelopes under different noise degrees of η. As one can tell that sparse regression can accurately identify the failure envelope with $\eta = 0 - 0.1$. Note that $\eta = 0.1$ could lead to about 20% maximum differences comparing to the exact values. The proposed method failed to identify the failure criterion at $\eta = 0.2$, which leads to about 40% differences. It is needed to point out that a filter may be added to the noised data to facilitate the process of identifying the failure criterion. But, for this example, we are interested in testing the capability of the proposed method, so the noised data is fed into sparse regression directly.

Figure 3.7 presents the errors between exact and identified failure criterion. The comparison were conducted by comparing the values of the exact and predicted σ_{12} under a certain σ_{11} or σ_{22} at failure. In this figure, the plot of the $\eta = 0.2$ scenario is omitted, because the identification of failure criterion failed for this scenario. The errors are computed via Eq. (3.20) .

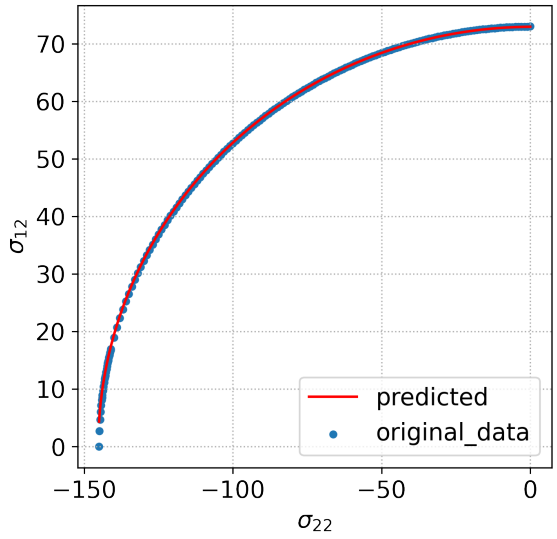
The y-axis of Figure 3.7 is on a logarithmic scale. As one can tell from the figure, when $\eta = 0$, the sparse regression can identify the failure envelope almost identical to the exact envelope since the errors are smaller than 10^{-11} %. Besides, the figure also shows that with the increase of noise magnitude η, the error also becomes larger. But sparse regression can



(a) Tensile fiber mode



(b) Tensile matrix mode

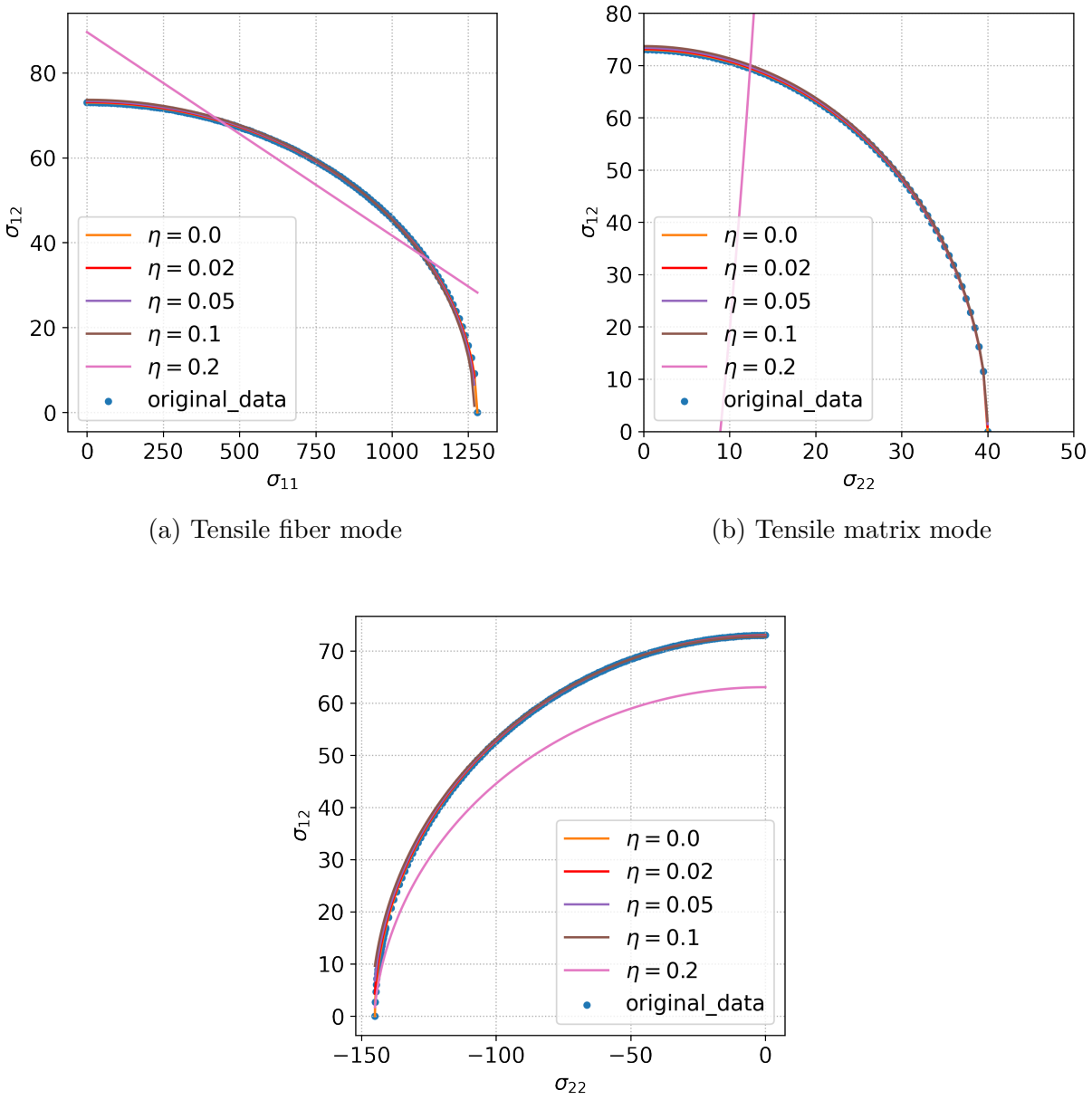


(c) Compressive matrix mode

Figure 3.5. Plots of exact and identified failure envelopes

still identify the failure envelope at $\eta = 0.1$. Finally, Figure 3.7 also shows that when the predicted value is close to zero, the error becomes large. This result is expected, as we expressed the error in percentage. Then if there is a small disturbance at these points, the error percentage could be large as the exact value is small. It is needed to point out that

this large error will not influence the accuracy of the identified failure criterion. Because when one variable is small, the other variable(s) could dominate the failure, which can be accurately measured.

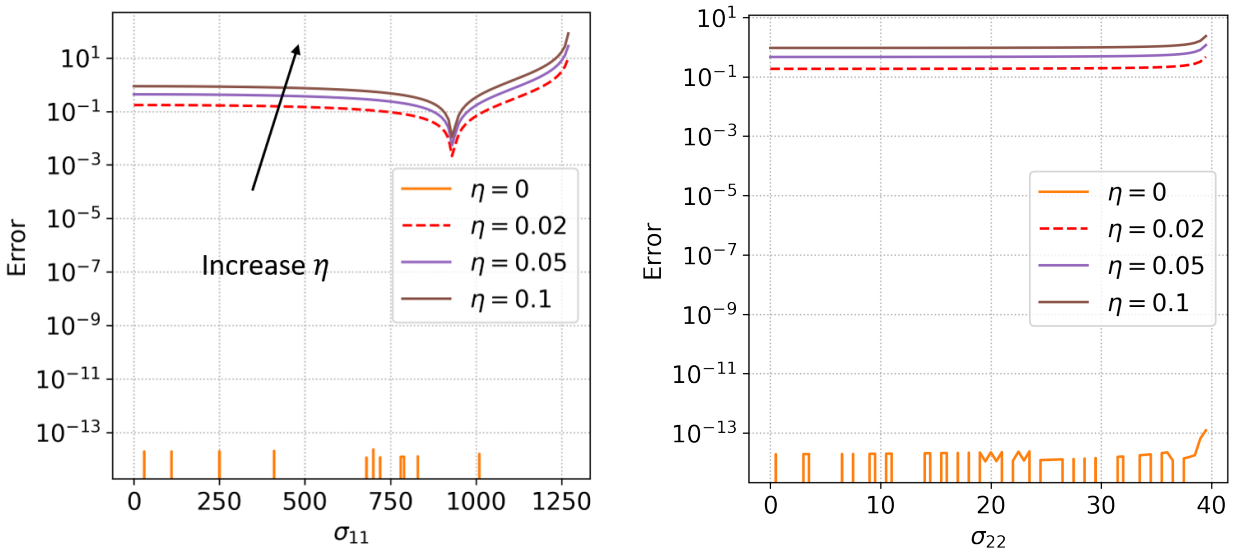


(a) Tensile fiber mode

(b) Tensile matrix mode

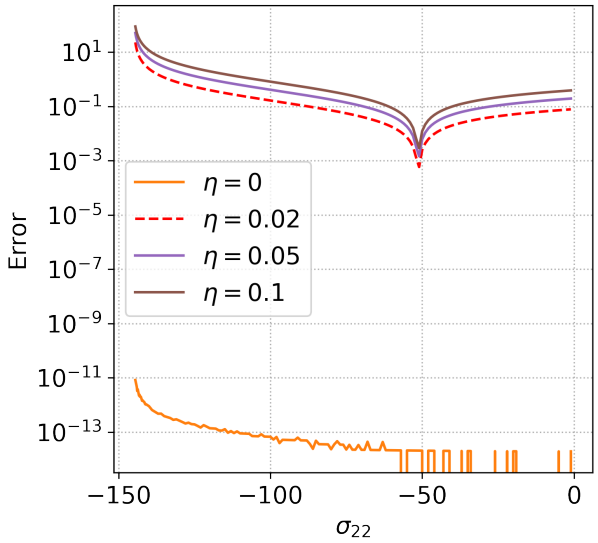
(c) Compressive matrix mode

Figure 3.6. Plots of exact and identified failure envelopes with different noise magnitudes



(a) Tensile fiber mode

(b) Tensile matrix mode



(c) Compressive matrix mode

Figure 3.7. Errors between exact and identified failure envelopes with different η

3.3.2 Example 6: Discover failure criterion from experimental data

Example 6 problem description

For the example in section 3.3.1, we demonstrated discovering failure criteria from experimental data. As mentioned earlier, we followed the phenomenological failure models and divided the failure of composites into tensile and compressive fiber modes, tensile and

compressive matrix modes. The unidirectional composites biaxial failure data were collected from WWFE I report [108]. Unfortunately, the report did not provide the data of the fiber failure mode and matrix failure mode of the same material. Thus, we used the data of T300/BSL914C to identify fiber failure criterion and E-glass/LY556 to identify matrix failure criteria respectively. Besides, the noise in the experimental data was strong, since the experiments were conducted by various researchers with various specimens for the same material. Thus, depending on the noise, a filter [109] was applied to de-noise the data.

The biaxial failure stress envelope for unidirectional T300/BSL914C carbon/epoxy lamina is reported in longitudinal and shear loading (σ_1 versus τ_{12}, which are expressed as σ_{11} and σ_{12} in this paper). These tests were done by Schelling and Aoki [107]. The specimens were manufactured as axially wound tubes with prepreg T300/BSL914C carbon/epoxy. The tubes were 32 mm in diameter and $1.9 - 2.3$ mm thick with fiber volume fraction $V_f = 0.56$. The compression and torsion tests were carried out at different laboratories. Figure 3.8a plots of the experimental data of stress failure envelope. One interesting observation was that the shear failure stresses tended to increase when the axial tension stress was applied and decreased at higher tensile stresses. In addition, there was a wide range of scattering of the shear strength when $\sigma_{11} = 0$. The report did not explain these different values. Table 3.5 provided the strength parameters of T300/BSL914C.

Table 3.5. Strength parameters of T300/BSL914C

X(MPa)	X'(MPa)	Y(MPa)	Y'(MPa)	S(MPa)	R(MPa)
1500	900	27	200	80	41

For the E-glass/LY556 material, the biaxial failure stress envelope was provided in transverse and shear loading (σ_2 versus τ_{12}, which are expressed as σ_{22} and σ_{12} in this paper). The tests were conducted by Hütter et. al. with circumstantially filament-wound tubes [110]. The internal diameter and thickness of the tubes were 60 mm and 2 mm respectively. The fiber volume fraction of the material was $V_f = 0.62$. The strength parameters of E-glass/LY556 are listed in Table 3.6. The results of this series of tests are provided in

Figure 3.8b. For the failure of this material, the result also showed that the shear stresses increased with the applying of compressive transverse stresses and then decreased at higher compressive stresses.

Table 3.6. Strength parameters of E-glass/LY556

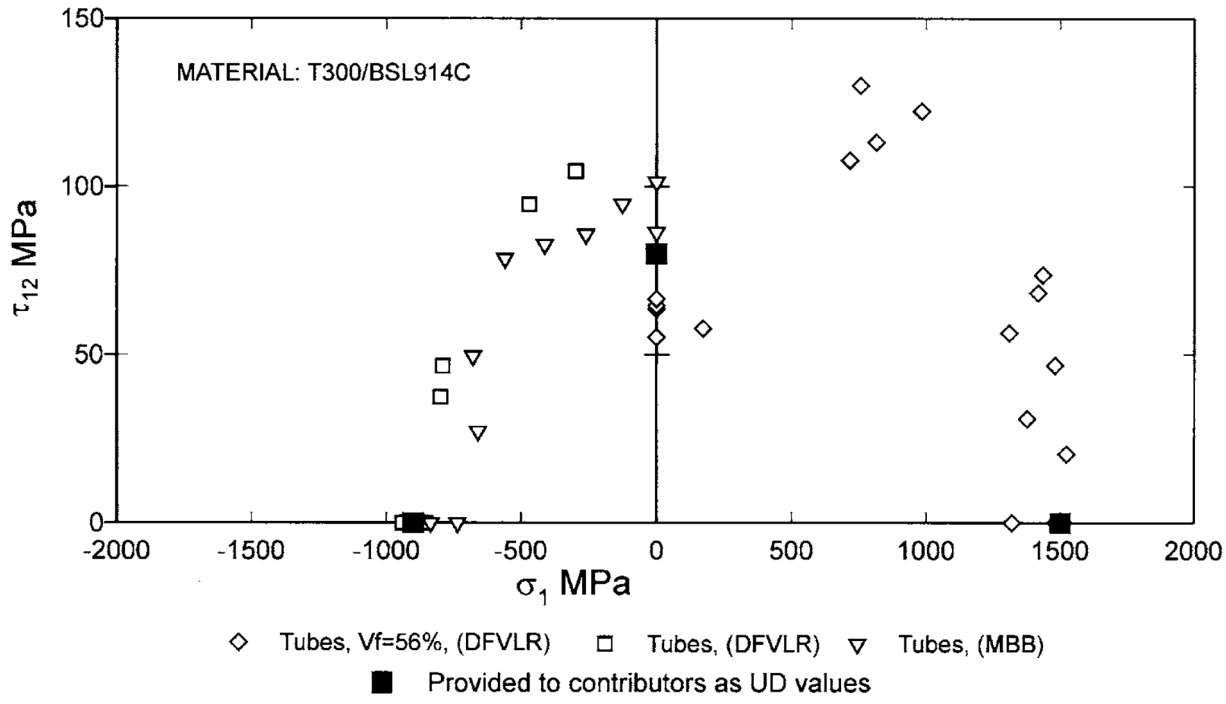
Y(MPa)	Y'(MPa)	S(MPa)	R(MPa)
40	137.4	61.2	39.9

Example 6 results and discussion

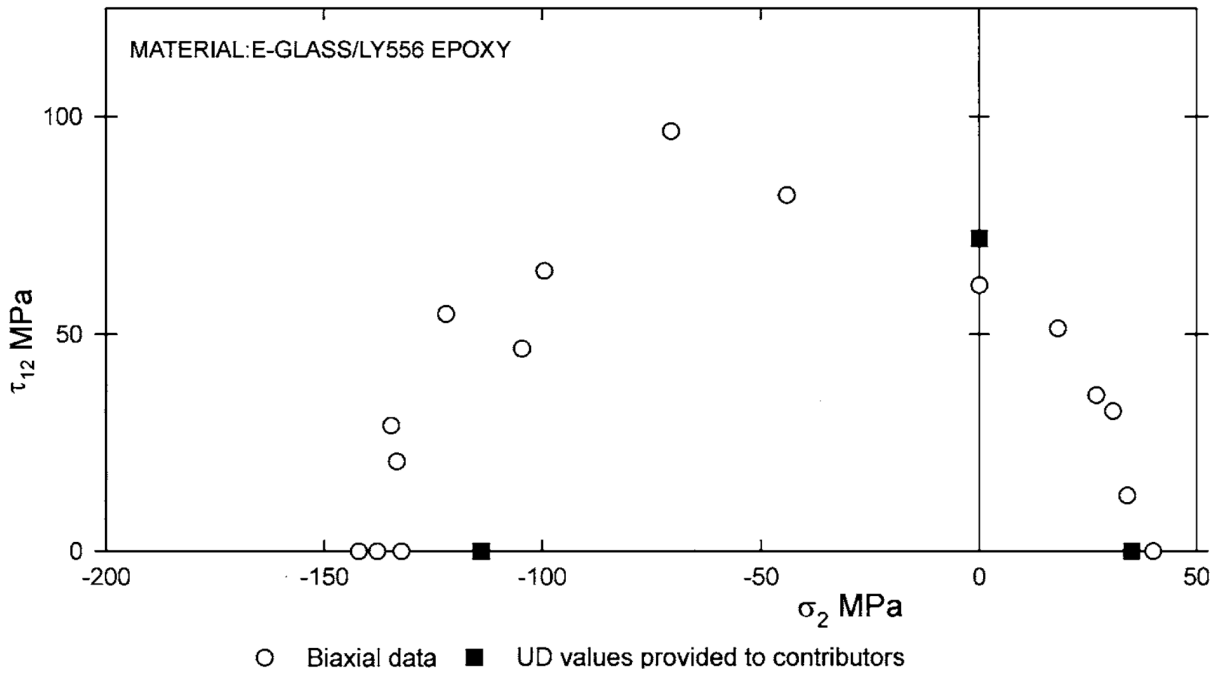
Figure 3.9 presents the failure envelope identified from experimental data. In this figure, the blue dots represent the experiment result. The orange dashed line and red line are the results of the Hashin failure criterion and sparse regression respectively. Figure 3.9a shows the identified failure criterion of compressive fiber mode. One can tell that the identified failure criterion of compressive failure mode improved the prediction comparing to the Hashin failure criterion. While the Hashin failure criterion predicted the failure envelope as a rectangle, the sparse regression method identified the failure criterion as a hyperbola, which captured the trend of experimental data. The active coefficients of the identified failure criterion are presented in Table 3.7. These coefficients indicate that σ_{12} and the coupling between σ_{22} and σ_{12} play an active role in the failure of compressive fiber mode. There is no contribution from σ_{12}^2 in the identified failure criterion.

Figure 3.9b shows the comparison of tensile fiber mode failure envelope between the Hashin failure criterion and sparse regression result. As one can tell, the sparse regression result successfully captured the phenomenon that the shear failure stresses tended to increase when the axial tension stress was applied and decrease at higher tensile stress. While for the Hashin failure criterion, it failed to account for this phenomenon. The result of the identified failure criterion is presented in Table 3.8. It is noted that linear terms related with σ_{11} and σ_{12} play a significant role in tensile fiber mode

For E-glass/LY556 material, the identified failure envelope result is presented in Figures 3.9c and 3.9d. These figures shows that the results based on the identified failure criterion



(a) T300/BSL914C



(b) E-Glass/LY556

Figure 3.8. Biaxial failure stress envelope for T300/BSL914C and E-Glass/LY556 from WWFE I

match the experiment result well, as the identified failure envelope is located in the middle of the data and captured the trend of the scattering of the experiment data. In addition, the results also indicate that while the sparse regression identified failure criterion is more conservative compared with the Hashin failure criterion for tensile matrix failure mode, it becomes less conservative at compressive matrix failure mode. But the sparse regression result fits the experimental data better. The coefficients of the active term for these two failure modes are summarized in Tables 3.9 and 3.10.

Table 3.7. Failure criteria coefficients of the compressive fiber mode

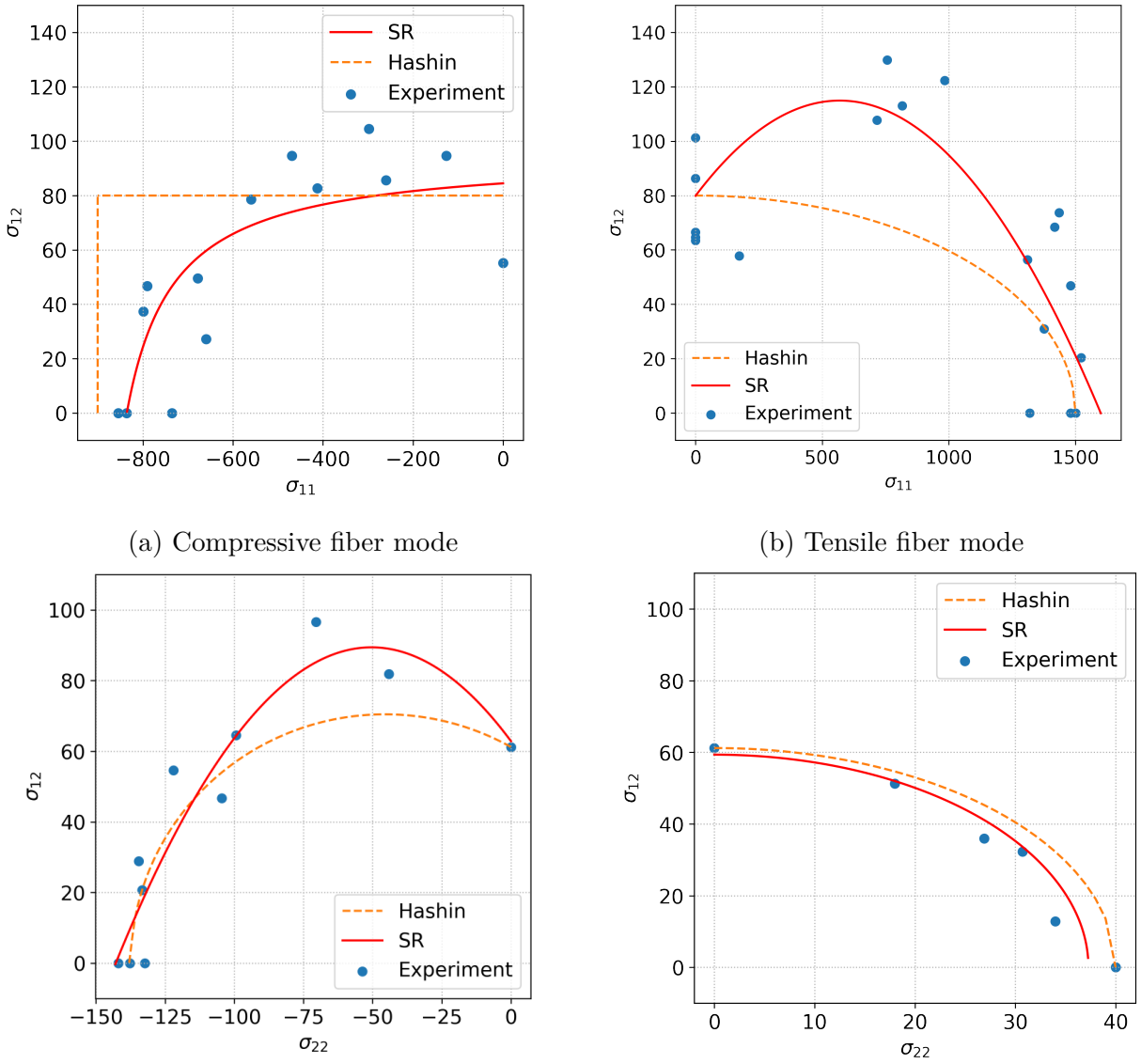
	σ_{11}	σ_{12}	σ_{11}^2	σ_{12}^2	$\sigma_{11}\sigma_{12}$	$\sigma_{11}^2\sigma_{12}$	$\sigma_{12}^2\sigma_{11}$	σ_1^3	σ_3^3
Hashin	-1.11E-03	0	0	0	0	0	0	0	0
SR	-1.19E-03	1.18E-02	0	0	1.26E-05	0	0	0	0

Table 3.8. Failure criteria coefficients of the tensile fiber mode

	σ_{11}	σ_{12}	σ_{11}^2	σ_{12}^2	$\sigma_{11}\sigma_{12}$	$\sigma_{11}^2\sigma_{12}$	$\sigma_{12}^2\sigma_{11}$	σ_1^3	σ_3^3
Hashin	0	0	4.44E-07	1.56E-04	0	0	0	0	0
SR	-1.54E-03	1.25E-02	1.35E-06	0	0	0	0	0	0

Table 3.9. Failure criteria coefficients of the compressive matrix mode

	σ_{22}	σ_{12}	σ_{22}^2	σ_{12}^2	$\sigma_{22}\sigma_{12}$	$\sigma_{22}^2\sigma_{12}$	$\sigma_{12}^2\sigma_{22}$	σ_{22}^3	σ_{12}^3
Hashin	1.43E-02	0	1.57E − 04	2.67E − 04	0	0	0	0	0
SR	1.68E-02	1.59E-02	2.01E-04	0	0	0	0	0	0



(a) Compressive fiber mode

(b) Tensile fiber mode

(c) Compressive matrix mode

(d) Tensile matrix mode

Figure 3.9. Result of identified failure criterion from experimental data

Table 3.10. Failure criteria coefficients of the tensile matrix mode

	σ_{22}	σ_{12}	σ_{22}^2	σ_{12}^2	$\sigma_{22}\sigma_{12}$	$\sigma_{22}^2\sigma_{12}$	$\sigma_{12}^2\sigma_{22}$	σ_{22}^3	σ_{12}^3
Hashin	0	0	2.67E-04	6.24E-04	0	0	0	0	0
SR	0	0	7.21E-04	2.83E-04	0	0	0	0	0

Additionally, we conducted a study to compare the performance of the sparse regression and DNN under the limited experimental data. For the current dataset, we only have the stresses at failure. This means the failure indices are one for all the cases. If we feed this dataset into DNN directly, DNN would try to predict the failure index to be one for every stress combination. To enable DNN to evaluate the failure state of a stress state, we took advantage of the concept of strength ratio (α), which is defined as

$$f\left(\boldsymbol{\sigma}_{cr}\right) = f(\alpha\boldsymbol{\sigma}) = 1 \tag{3.21}$$

The physical meaning of strength ratio is defined as a factor that, when multiplying with the current stress state $\boldsymbol{\sigma}$ gives the failure index of one. Following Tian and Tao, et. al. [82], we normalized the stress to unit stress vectors. The strength ratio was calculated by dividing the original failure stress by the corresponding component in the unit stress vector, i.e. $\alpha = \sigma_{11}/\sigma_{11}^{n}$, where σ_{11} and σ_{11}^{n} are the original and normalized stresses, respectively. A DNN model with three hidden layers was identified. The structure of the DNN model is $[3, 20, 20, 20, 1]$. This structure means the neural network has three inputs ($\sigma_{11}, \sigma_{22}, \sigma_{12}$), one output ($\alpha$), and three hidden layers with each of the hidden layers having 20 neurons. For the input of DNN, the non-available components are set to zero. To prevent overfitting, the L2 regularization [111], which adds the weighted sum of all the neuron's squared weights to the total loss, was utilized in the training of the neural network. The activation function is the rectified linear unit (ReLU) function [112] and is shown in Figure 3.10.

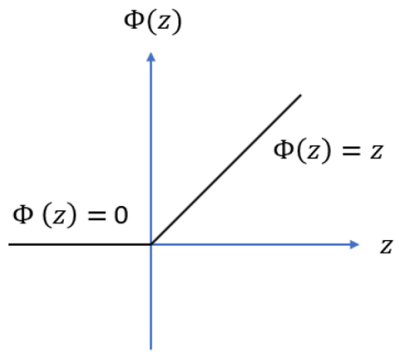


Figure 3.10. ReLU activation function

Figure 3.11 presents the result of the comparison between sparse regression and DNN. The result shows that DNN could generate a prediction that matches closer to the experimental data. However, since the dataset is too small, DNN cannot predict the result accurately in certain regions. For example, for the tensile fiber mode, at $\sigma_{11} = [0, 200]$ MPa region, there is a drastic change in the prediction. This drastic change does not physically make sense and is caused by insufficient data points in this region. In contrast, sparse regression does not have this issue and can capture the trend of the experimental data well. This comparison indicates that sparse regression is superior to DNN under a limited data size.

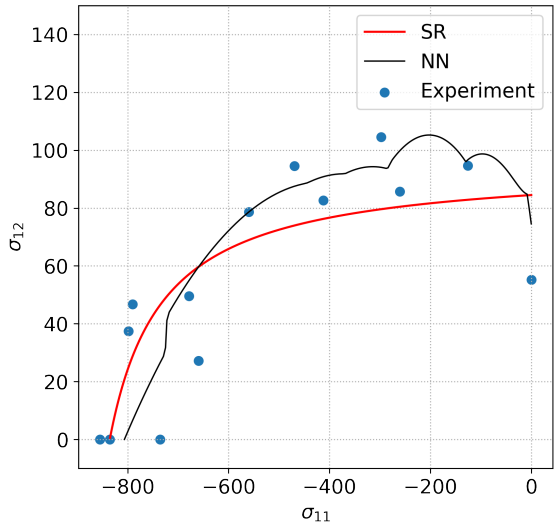
3.3.3 Example 7: Enforce constraint to yield conservative failure criterion

Example 7 problem description

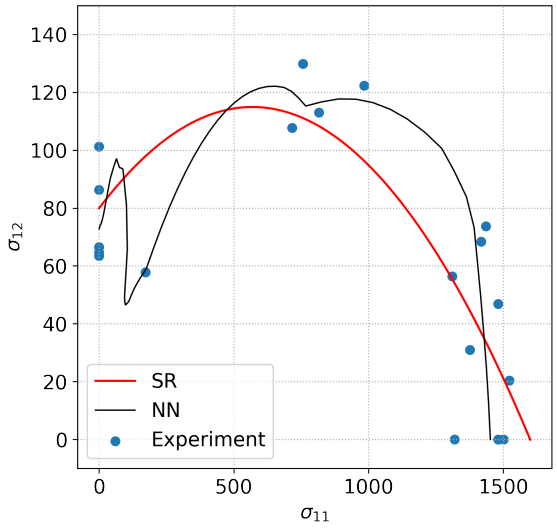
The previous two examples in this chapter have shown the feasibility to apply sparse regression with compressed sensing to discover failure criterion. With the sequential threshold least-squares algorithm, the losses between failure criterion predicted data and training data are minimized. The failure envelope is located in the middle of the data. Nevertheless, a conservative design is usually required in engineering design for the safety issue. The conservative design requires the model predicted data to be smaller than the experimental data. Inspired by Mattias et. al. [112] and Tao et. al. [111], we can use an optimization approach to enforce a constraint to the discovered criterion to shift the predicted data to be smaller than the experimental data. The constraint is expressed as an inequality equation and shown below

$$\sigma_p^{(i)} - \sigma_{\text{exp}}^{(i)} \leq 0 \tag{3.22}$$

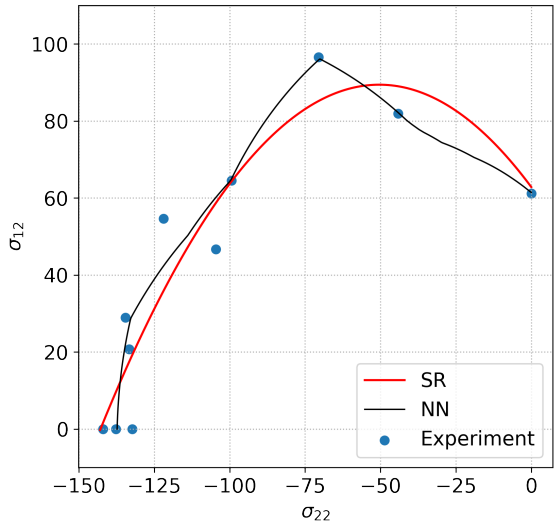
where $\sigma_p^{(i)}$ and $\sigma_{\text{exp}}^{(i)}$ are the ith failure criterion predicted data and experimental data, respectively. This constraint indicates that the predicted result should be equal to or smaller than the corresponding experimental data. To ensure the fulfilling of the inequality constraint, we can apply a penalty function $\Phi(\cdot)$. The penalty function was chosen as the ReLU function [112], which is shown in Figure 3.10. This figure indicates that negative errors will be shifted to be zero.



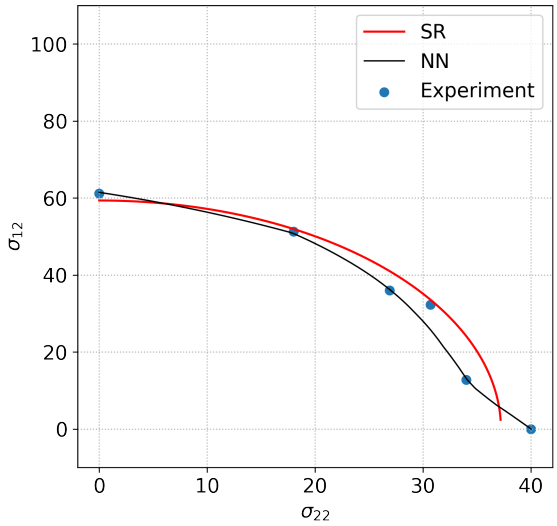
(a) compressive fiber mode



(b) tensile fiber mode



(c) compressive matrix mode



(d) tensile matrix mode

Figure 3.11. Result comparison between sparse regression and DNN

Next, we can can add the penalty function caused constraint loss to the original mean square error (MSE) loss. Then the total loss becomes:

$$L_t = \frac{\beta}{2n}\sum_{i=1}^{n}\left(\sigma_p^{(i)} - \sigma_{\exp}^{(i)}\right)^2 + \frac{\gamma}{2n}\sum_{i=1}^{n}\Phi\left(\sigma_p^{(i)} - \sigma_{\exp}^{(i)}\right)^2 \tag{3.23}$$

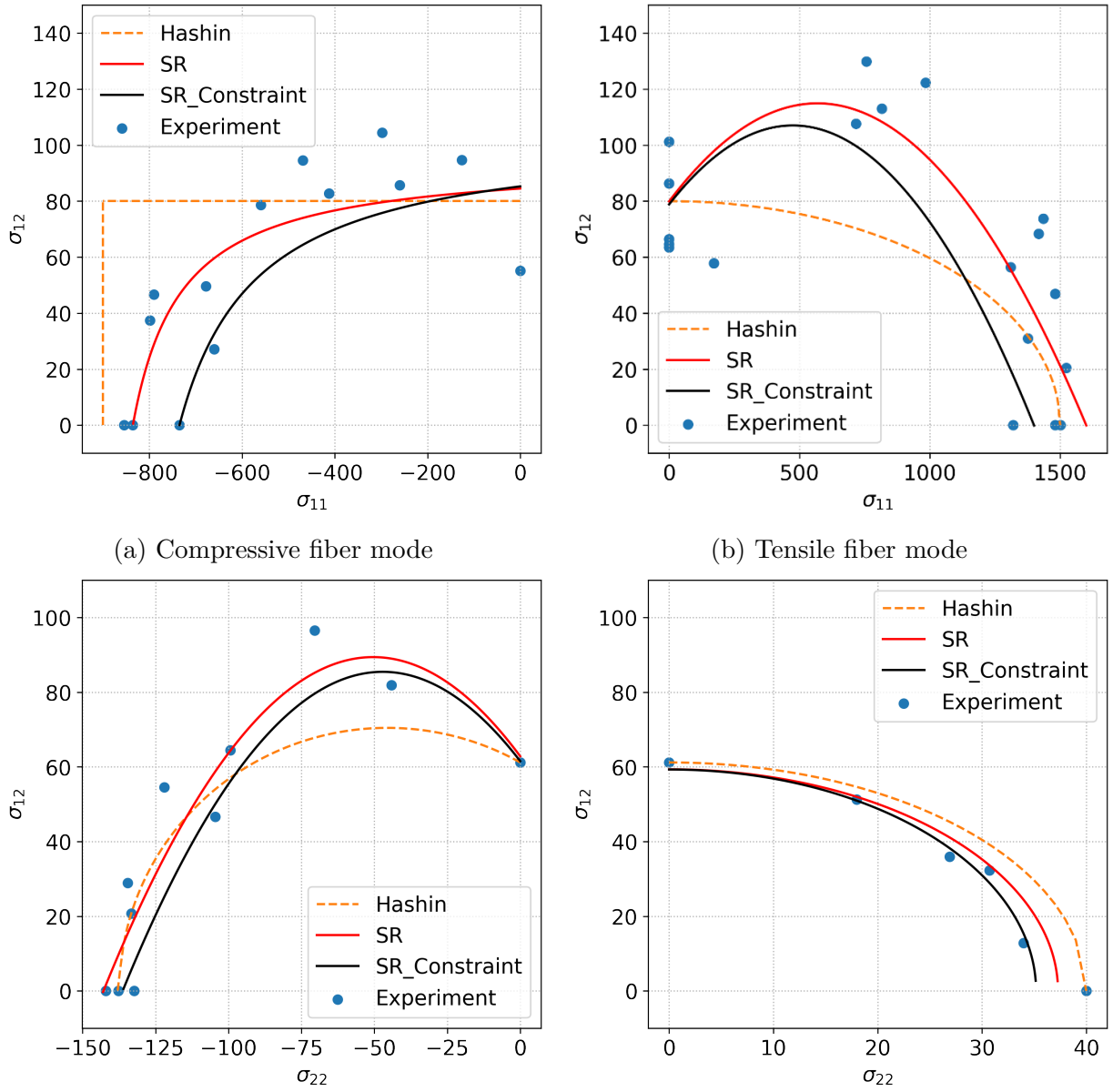
where L_t is the total loss function, β and γ are the weights of the original loss and constraint loss respectively. These two weights are added to tune the influence of original loss and constraint loss. An optimization algorithm will be employed to minimize the total loss. In this study, we selected ADAM optimizer [113] to tune the model. It is important to note that the inequality constraint in Eq. (3.23) is applied as a soft constraint, which can lead the prediction toward yielding $\sigma_p^{(i)}$ less than $\sigma_{\text{exp}}^{(i)}$, but cannot guarantee the prediction will always satisfy the constraint. However, by tuning the weights of the losses, it is possible to get a model with all or majority of the the $\sigma_p^{(i)}$ equal or smaller than the corresponding σ_{exp}.

For demonstration, we applied the proposed method to the example in Section 3.3.2. The weights of the MSE loss and constraint loss were set to be 1 and 10^{-4} correspondingly. The learning rate of the ADAM optimizer was 10^{-5}.

Example 7 results and discussion

The result is shown in Figure 3.12. In these plots, the blue dots, orange dashed line, and red line represent the experimental, Hashin failure criterion, and sparse regression result, respectively. The black line represents the result of sparse regression with constraint. One can tell that the spares regression with constraint can capture the trend of the experimental data well. In addition, the prediction made by the sparse regression with constraint is more conservative, as the majority of the experimental data lie outside of the failure envelope. This has achieved the goal of yielding a conservative failure criterion that can be directly used in a design. For this problem, we stopped keeping tuning the weight of the constraint loss to shift all the experimental data lie in the outside of the failure envelope. The experimental data has strong noise, because the experiments were conducted by various specimens at different labs. Besides, keeping tuning the envelope to be smaller than the experimental data will decrease the accuracy of the prediction significantly. The coefficients of the conservative failure criterion is shown in Tables 3.11-3.14. The SR_c represents the result from the sparse regression with constraint. One can tell that the optimization did not change the type of the failure envelope since the features of the model and the order of the coefficients are the same as the result of the sparse regression. The coefficients are slightly changed to make the

failure envelope being more conservative. These tables along with Figure 3.12 demonstrate the feasibility of yielding a conservative failure criterion with sparse regression.



(a) Compressive fiber mode

(b) Tensile fiber mode

(c) Compressive matrix mode

(d) Tensile matrix mode

Figure 3.12. Result of identified failure criterion with enforced constraint

Table 3.11. Conservative failure criteria coefficients of compressive fiber mode

	σ_{11}	σ_{12}	σ_{11}^2	σ_{12}^2	$\sigma_{11}\sigma_{12}$	$\sigma_{11}^2\sigma_{12}$	$\sigma_{12}^2\sigma_{11}$	σ_1^3	σ_3^3
Hashin	-1.11E-03	0	0	0	0	0	0	0	0
SR	-1.19E-03	1.18E-02	0	0	1.26E-05	0	0	0	0
SR_c	-1.36E-03	1.17E-02	0	0	1.30E-05	0	0	0	0

Table 3.12. Conservative failure criteria coefficients of tensile fiber mode

	σ_{11}	σ_{12}	σ_{11}^2	σ_{12}^2	$\sigma_{11}\sigma_{12}$	$\sigma_{11}^2\sigma_{12}$	$\sigma_{12}^2\sigma_{11}$	σ_1^3	σ_3^3
Hashin	0	0	4.44E-07	1.56E-04	0	0	0	0	0
SR	-1.54E-03	1.25E-02	1.35E-06	0	0	0	0	0	0
SR_c	-1.51E-03	1.27E-02	1.58E-06	0	0	0	0	0	0

Table 3.13. Conservative failure criteria coefficients of compressive matrix mode

	σ_{22}	σ_{12}	σ_{22}^2	σ_{12}^2	$\sigma_{22}\sigma_{12}$	$\sigma_{22}^2\sigma_{12}$	$\sigma_{12}^2\sigma_{22}$	σ_{22}^3	σ_{12}^3
Hashin	1.43E-02	0	$1.57\mathrm{E}-04$	$2.67\mathrm{E}-04$	0	0	0	0	0
SR	1.68E-02	1.59E-02	2.01E-04	0	0	0	0	0	0
SR_c	1.65E-02	1.62E-02	2.00E-04	0	0	0	0	0	0

Table 3.14. Conservative failure criteria coefficients of tensile matrix mode

	σ_{22}	σ_{12}	σ_{22}^2	σ_{12}^2	$\sigma_{22}\sigma_{12}$	$\sigma_{22}^2\sigma_{12}$	$\sigma_{12}^2\sigma_{22}$	σ_{22}^3	σ_{12}^3
Hashin	0	0	2.67E-04	6.24E-04	0	0	0	0	0
SR	0	0	7.21E-04	2.83E-04	0	0	0	0	0
SR_c	0	0	8.07E-04	2.84E-04	0	0	0	0	0

4. MACHINE LEARNING ASSISTED COMPOSITE ROTOR BLADE PLANFORM DESIGN

Composite materials are increasingly used in the design of rotor blades due to their high stiffness-to-weight ratio and excellent fatigue life. Currently, the composite rotor blade planform design intensively focuses on optimizing planform parameters to achieve higher performance. However, the strength of the material is rarely considered in the planform design. One issue is that the failure analysis of the rotor blade is usually performed by an expensive multiscale analysis which requires calculating the sectional properties, computing the rotor blade global response, and reproducing local stress for each layer of the composite blade. This process is inefficient when applying to an optimization framework, as whenever there is a new structural load, this analysis procedure needs to be performed again. Unfortunately, millions of load cases can be accumulated during the blade design optimization. The ignoring of strength analysis may result in the blade working in an unsafe or low safety factor region, as composite materials are anisotropic and susceptible to various failure mechanisms. In this chapter, we propose to optimize the composite rotor blade planform design with strength consideration. The optimized design can improve the performance of the aircraft and ensure the structure is within the safety margin. To reduce the computational cost of the cross-sectional failure analysis, we will use machine learning model to construct a beam-level failure criterion surrogate model to replace the physics-based cross-sectional failure analysis. The surrogate model is constructed based on the Timoshenko beam model via Artificial neural networks (ANN), where the mapping will be between blade loads and the strength ratios of the cross-section.

4.1 Structure Level Failure Criterion

4.1.1 Tsai-Wu failure criterion and strength ratio

Many failure criteria have been proposed to analyze the failure of composites. Some commonly used failure criteria are max stress (or strain) failure criterion, max shear stress (or strain) failure criterion, Tsai-Hill failure criterion, Hashin failure criterion, and Tsai-Wu

failure criterion [10]. In this chapter, the composite materials are assumed to fail according to the Tsai-Wu failure criterion since it is a simple yet general enough formula that can account for multiaxial stress interactions and the difference between tensile and compressive strengths. The Tsai-Wu failure function can be expressed as

$$\begin{aligned} f(\boldsymbol{\sigma}) = {} & F_{11}\sigma_{11}^2 + F_{22}\sigma_{22}^2 + F_{33}\sigma_{33}^2 + F_{44}\sigma_{23}^2 + F_{55}\sigma_{13}^2 + F_{66}\sigma_{12}^2 \\ & + 2F_{12}\sigma_{11}\sigma_{22} + 2F_{13}\sigma_{11}\sigma_{33} + 2F_{23}\sigma_{22}\sigma_{33} \\ & + F_1\sigma_{11} + F_2\sigma_{22} + F_3\sigma_{33} \end{aligned} \tag{4.1}$$

with

$$\begin{aligned} & F_{11} = \frac{1}{X_t X_c}, \quad F_{22} = \frac{1}{Y_t Y_c}, \quad F_{33} = \frac{1}{Z_t Z_c}, \\ & F_{44} = \frac{1}{R^2}, \quad F_{55} = \frac{1}{T^2}, \quad F_{66} = \frac{1}{S^2}, \\ & F_{12} = \frac{1}{2}\left(\frac{1}{Z_t Z_c} - \frac{1}{X_t X_c} - \frac{1}{Y_t Y_c}\right) \\ & F_{13} = \frac{1}{2}\left(\frac{1}{Y_t Y_c} - \frac{1}{X_t X_c} - \frac{1}{Z_t Z_c}\right) \\ & F_{23} = \frac{1}{2}\left(\frac{1}{X_t X_c} - \frac{1}{Y_t Y_c} - \frac{1}{Z_t Z_c}\right) \\ & F_1 = \frac{1}{X_t} - \frac{1}{X_c}, \quad F_2 = \frac{1}{Y_t} - \frac{1}{Y_c}, \quad F_3 = \frac{1}{Z_t} - \frac{1}{Z_c} \end{aligned} \tag{4.2}$$

where $X_t, Y_t, Z_t, X_c, Y_c, Z_c, R, T, S$ are the strength constants. They can be determined from experiments. Under a specific failure criterion, a stress state is called a critical stress state if

$$f\left(\sigma_{cr}\right) = 1 \tag{4.3}$$

where $f\left(\cdot\right)$ represents the failure criterion. The strength ratio under a stress condition is defined as

$$f\left(\sigma_{cr}\right) = f\left(\alpha\sigma\right) = 1 \tag{4.4}$$

For the Tsai-Wu failure criterion, since it has both linear and quadratic terms of the stress components, the strength ratio can be solved as

$$f(\alpha\sigma) = \alpha^2 a + \alpha b = 1 \tag{4.5}$$

where

$$\begin{aligned} \mathrm{a} =& F_{11}\sigma_{11}^2 + F_{22}\sigma_{22}^2 + F_{33}\sigma_{33}^2 + F_{44}\sigma_{23}^2 + F_{55}\sigma_{13}^2 + F_{66}\sigma_{12}^2 \\ &+ 2F_{12}\sigma_{11}\sigma_{22} + 2F_{13}\sigma_{11}\sigma_{33} + 2F_{23}\sigma_{22}\sigma_{33} \\ b =& F_1\sigma_{11} + F_2\sigma_{22} + F_3\sigma_{33} \end{aligned} \tag{4.6}$$

The solution of Eq. (4.5) will be

$$\alpha = \frac{-b \pm \sqrt{b^2 + 4a}}{2a} \tag{4.7}$$

Eq. (4.7) has two solutions. The strength ratio should take the smaller positive one. Besides, from the definition of the strength ratio, a material will be safe if $\alpha > 1$ and fail if $0 < \alpha < 1$, which is different from the commonly defined failure index. The reason we use the strength ratio is that strength ratio is proportional to the load, while failure index is not necessary proportional to the load.

4.1.2 Failure criterion of structures

In this study, the Variational Asymptotic Beam Section (VABS) [83] code is used to perform the physics based cross-sectional failure analysis. The process of calculating strength ratio at the beam level is shown in Figure 4.1. In this analysis, we need to use VABS to

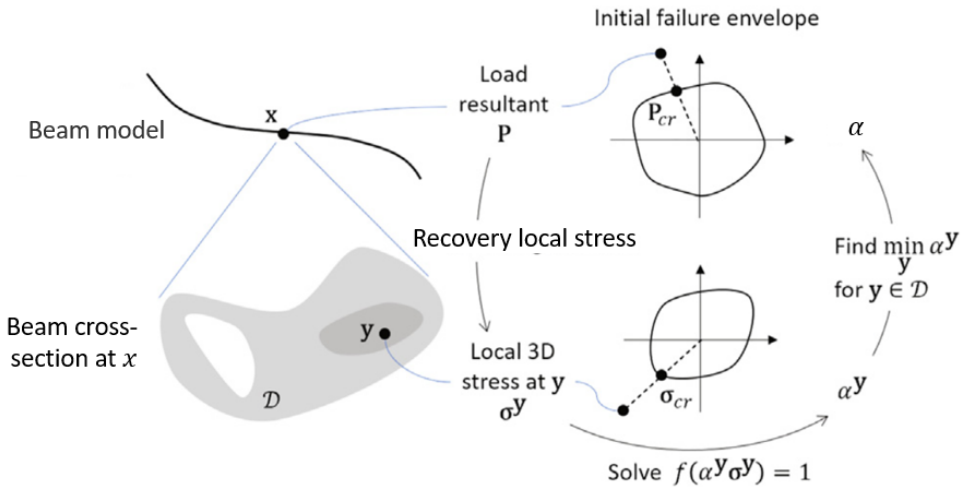


Figure 4.1. The process of calculating structural level strength ratio [32]

calculate the effective properties of the cross-section. Then, the beam analysis will take the effective properties and calculate the rotorcraft response at the global level. From the global response, we can obtain blade loads P, which is an array containing forces and moments corresponding to a specific beam model. For example, for a Timoshenko Beam model, $P = (F_1, F_2, F_3, M_1, M_2, M_3)$ where F_1 is the tensile force, F_2 and F_3 are shear forces, M_1 is the twisting moment, M_2 and M_3 are bending moments. Next, VABS will be used to compute the local 3D stress fields, which will be used to solve for the strength ratio according to the failure criterion at each material point of the cross-section. The minimum strength ratio among all the material points will be the strength ratio of the cross-section $\left(\alpha^{SG}\right)$.

To construct a beam level failure criterion is to find a relation

$$\alpha = \alpha(P) \tag{4.8}$$

If the stress analysis is linear elastic, the critical loads of the cross-section can be expressed as

$$P_{cr} = \alpha P \tag{4.9}$$

where P_{cr} is the critical load of the cross-section. P_{cr} is the properties of a blade like strength parameters to a material. Eq. (4.8) and (4.9) indicate that one can directly get the strength ratio of the cross-section with the structural loads as inputs. It would be beneficial to find such a relation, as this relation can avoid the computational expensive multiscale failure analysis. However, the analytical form of the failure criterion at the structural level is typically not available, since a structure can have various components and complicated geometries, it would be difficult to find a analytical model to describe the strength ratio. Fortunately, surrogate modeling can be a useful tool to construct the beam level failure criterion to reduce computational cost.

For surrogate modeling, conducting simulation to generate the training data is one of the most important steps in the modeling process. In this study, we will use VABS to perform the cross-sectional failure analysis to generate the training database. The process is shown as Workflow 1 in Figure 4.2. The process is very similar to conduct a regular

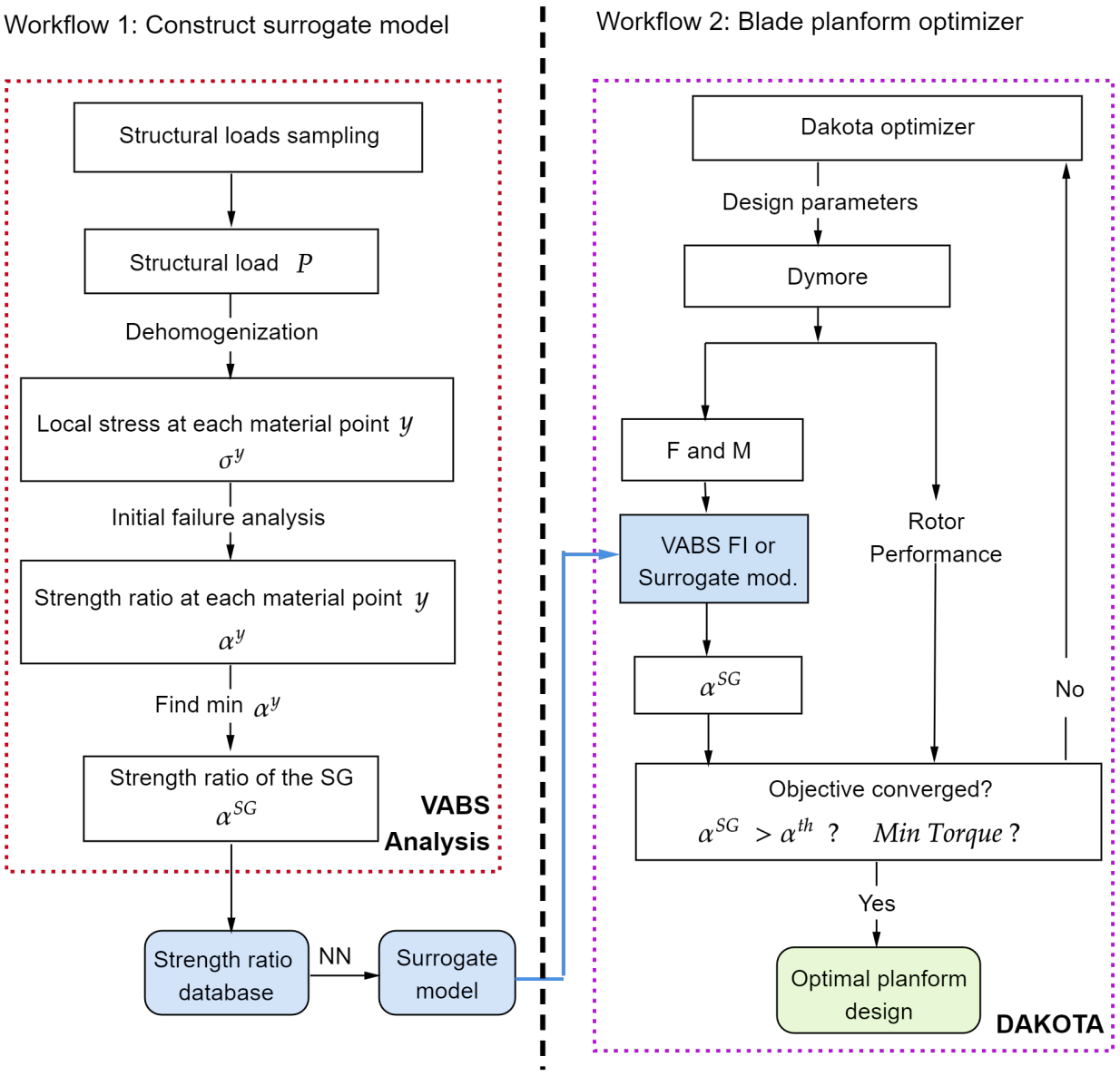


Figure 4.2. Flowchart of rotor blade planform optimization

cross-sectional failure analysis, except that we will use a high-dimensional surface sampling technique to generate various combinations of the structural loads. The sampled load cases will be transferred to VABS. Then, VABS will carry out the failure analysis and compute the strength ratio at each material point. The minimum values among all the strength ratios will be the strength ratio of the cross-section (α^{SG}). By repeating doing the analyses for all

the load cases, we will form a database. The database will be used to form a surrogate model for the beam level failure criterion. The trained surrogate model will be embedded into the Workflow 2, which will conduct the comprehensive analysis to compute the performance and extract the corresponding forces and moments in the rotor blade. The surrogate model will take the forces and moments to compute the strength ratio of the airfoil.

4.2 Rotor Blade Planform Design Process

4.2.1 Machine learning assisted failure criterion of an UH-60A cross-section

In this study, the Artificial neural network (ANN) is used to construct a surrogate model for the beam level failure criterion. The reason we selected ANN instead of the Gaussian process is that ANN is more attractive to deal with big datasets and complicated functions. In this work, we will construct a surrogate model with the Timoshenko beam model, which has six load components and could result in a big training data size. The mapping of ANN will be constructed between sectional loads P and strength ratio of the blade cross-section. The architecture of a ANN model is shown in Figure 4.3.

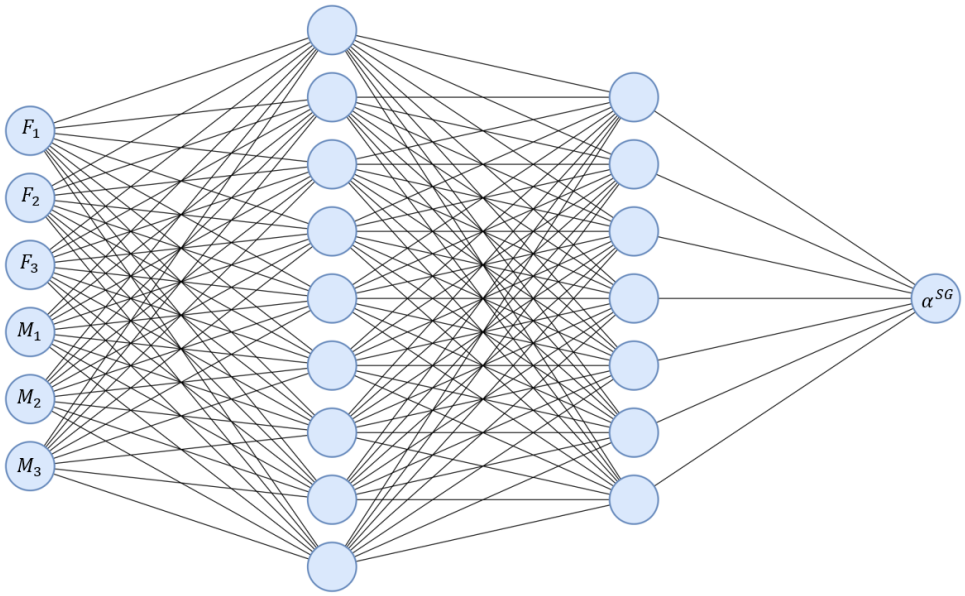


Figure 4.3. The architecture of a structural level failure criterion neural networks model

A UH-60A composite rotor blade will be used in this study [82]. The cross-section of the blade is shown in Figure 4.4. For the cap, the base layer is aluminum with a thickness of 0.01 inches. The main material for the cap is AS4 12k/E7K8 of ply thickness 0.0054 inches. For the box spar and overwrap, the base layer is T300 15k/976 with a ply thickness of 0.0053 inches. The box spar and overwrap main material are T650-35 3k 976 fabric with a ply thickness of 0.0062 inches. Table 4.1 presents the layup angles and the number of plies of the composites. The region in the cap is filled with Rohacell 70. The region in the overwrap is filled with Plascore PN2-3/16OX3.0. The non-structural mass is made of lead. The properties of these materials are listed in Table 4.2.

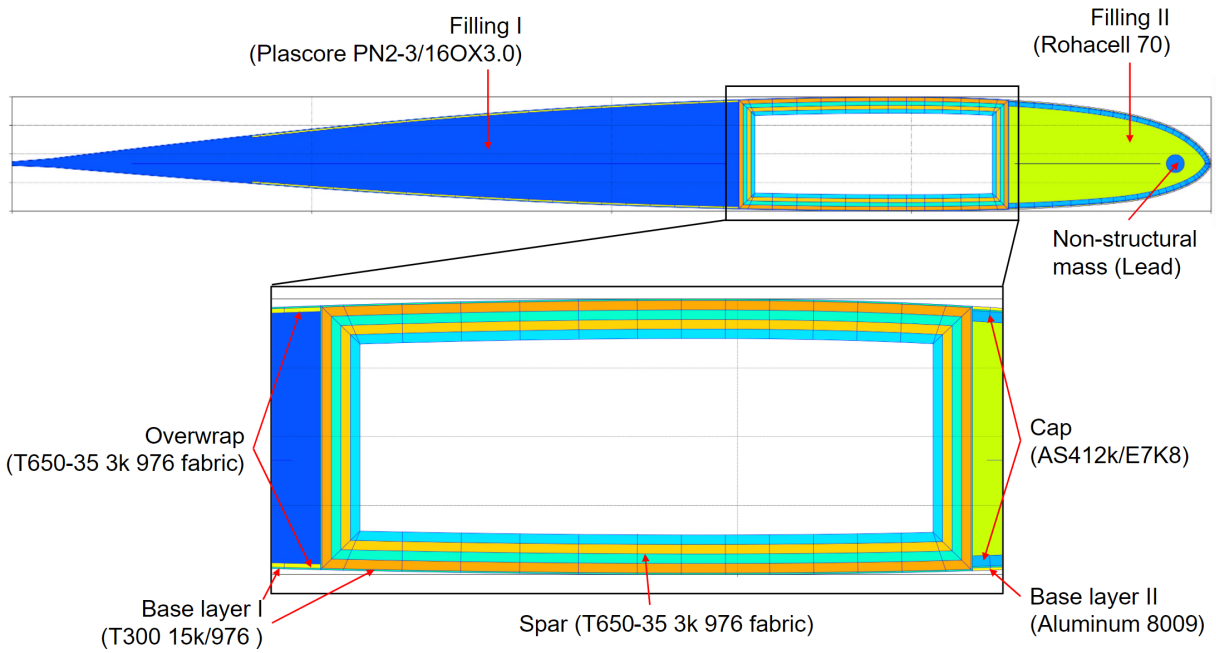


Figure 4.4. The cross-section of an UH-60A composites rotor blade

4.2.2 Planform design with strength considerations

The rotor blade planform optimization will be performed with the DAKOTA optimization package [114], which is a multilevel parallel object-oriented framework for design optimization. The framework is shown in Figure 4.2. As mentioned earlier, the framework consists of two interdependent workflows: the surrogate model construction workflow and the ro-

Table 4.1. The composites design of spar, cap and overwrap

Component	Layer	Plies n	Orientation θ	Lamina type
Spar	1	11	−21	T650-35 3k 976 fabric
	2	11	38	
	3	11	−84	
	4	11	−29	
Cap	1	14	−46	AS4 12k/E7K8
Overwrap	1	5	47	T650-35 3k 976 fabric

tor blade optimization workflow. The surrogate model construction workflow is descried in Section 4.1.2. The rotor blade optimization workflow begins according to the user-defined optimization algorithm selected through Dakota. Then, with the substitution of the planform design parameters, a rotor model will be generated. The Dymore comprehensive code [115] will run to compute the performance and extract the corresponding forces and moments in the rotor blade. VABS or the surrogate model will take the forces and moments to compute the strength ratio of the cross-section. The performance metrics and strength ratio of the cross-section will be returned to the Dakota optimizer. The simulation proceeds according to the chosen optimization method.

4.3 Application

4.3.1 Example 8: Optimize planform of UH60A helicopter with strength consideration for forward flight

Example 8 problem description

For this example, we created a simplified UH60A helicopter model in Dymore, which serves the purpose to demonstrate the optimization workflow. The model was created to have four blades with uniform composites blade cross-section. The cross-section is shown on Figure 4.4 with properties presented in Table 4.3. The blade was set to have a length of 26.833 ft, rotating at 27.02 rad/s. The airfoil of the blade was set to be SC1095. An airfoil table, which defines the airfoil lift, drag, and moment coefficients as a function of angle of

Table 4.2. Material properties of the components of an UH60 blade

Name	Quantity	Value	Thickness
Aluminum 8009	ρ	$0.271959 \times 10^{-3}(\text{lbf} \cdot \text{sec}^2/\text{in}^4)$	-
	E	$13.1 \times 10^6(\text{psi})$	
	ν	0.330	
Lead	ρ	$1.060957 \times 10^{-3}(\text{lbf} \cdot \text{sec}^2/\text{in}^4)$	-
T300 15k/976	ρ	$0.149716 \times 10^{-3}(\text{lbf} \cdot \text{sec}^2/\text{in}^4)$	0.0053 (in)
	E_1	$19.6 \times 10^6(\text{psi})$	
	E_2	$1.34 \times 10^6(\text{psi})$	
	ν_{12}	0.348	
	G_{12}	$0.910 \times 10^6(\text{psi})$	
Rohacell 70	ρ	$7.040895 \times 10^{-6}(\text{lbf} \cdot \text{sec}^2/\text{in}^4)$	-
	E_1, E_2, E_3	$13.1 \times 10^3(\text{psi})$	
	ν_{12}, ν_{13}	0.315	
	ν_{23}	0.300	
	G_{12}, G_{13}, G_{23}	$4.12 \times 10^3(\text{psi})$	
Plascore PN2-3/16OX3.0	ρ	$4.509066 \times 10^{-6}(\text{lbf} \cdot \text{sec}^2/\text{in}^4)$	-
	E_1, E_3	$1.00 \times 10^3(\text{psi})$	
	E_2	$20.0 \times 10^3(\text{psi})$	
	ν_{12}, ν_{23}	0.010	
	ν_{13}	0.300	
	G_{12}	$3.50 \times 10^3(\text{psi})$	
	G_{13}	$1.00 \times 10^3(\text{psi})$	
	G_{23}	$5.80 \times 10^3(\text{psi})$	
AS4 12k/E7K8	ρ	$0.145973 \times 10^{-3}(\text{lbf} \cdot \text{sec}^2/\text{in}^4)$	0.0054 (in)
	E_1	$19.3 \times 10^6(\text{psi})$	
	E_2	$1.23 \times 10^6(\text{psi})$	
	ν_{12}	0.32	
	G_{12}	$7.31 \times 10^6(\text{psi})$	
T650-35 3k 976 fabric	ρ	$0.145973 \times 10^{-3}(\text{lbf} \cdot \text{sec}^2/\text{in}^4)$	0.0062 (in)
	E_1	$10.4 \times 10^6(\text{psi})$	
	E_2	$10.0 \times 10^6(\text{psi})$	
	ν_{12}	0.06	
	G_{12}	$0.8 \times 10^6(\text{psi})$	

attack and Mach numbers, was provided to compute the airloads. The air properties are presented in Table 4.4. In Table 4.4, the v_i represents the flight speed of the helicopter. For example, $v_1 = 215.6$ ft/s means the helicopter flight forward with a speed 215.6 ft/s. This

set of flight speed represents the P5231 forward flight condition. The trim of the thrust are $F_3(\text{lift}) = 20,467\text{lbs}$, $M_1(\text{rolling moment}) = 1669.8$ lb·ft, $M_2(\text{pitching moment}) = -3058.3$ lb·ft.

Table 4.3. Beam properties of UH60A blade [82]

m $(\text{lbf}\cdot\text{sec}^2/\text{in}^2)$	C_{GJ} $(10^6\ \text{lbf}\cdot\text{in}^2)$	C_{EIf} $(10^6\ \text{lbf}\cdot\text{in}^2)$	C_{EIc} $(10^6\ \text{lbf}\cdot\text{in}^2)$	d_2^{SC} (in)	d_2^{CG} (in)
0.00092	24.25	22.31	830.9	-5.18	-6.03

Table 4.4. Air properties

ρ (slug /ft^3)	v_{sound} (ft/s)	v_1 (ft/s)	v_2 (ft/s)	v_3 (ft/s)	Lift slop	Drag coeff.
0.0023	1121.06	215.6	0.0	-30	5.73	0.010

This example tries to optimize the planform of the blade to minimize the torque in the rotor and enforce the strength constraint under the same flight condition. For the same flight condition, a smaller torque in the rotor indicates the helicopter requires less power from the engine to keep the flight condition. Table 4.5 shows the design variables. As shown in the table, we selected the dihedral (θ_d), sweep (θ_s), and twist (θ_t) angles of the tip of the blade. These design variables are visualized in Figure 4.5. The Genetic algorithm (GA) [116] is used to carry out the optimization. Three cases were studied in this example. Case 1 optimizes the planform of the blade to minimize the torque without considering the strength of the blade. Case 2 and 3 takes α^{SG} as a constraint and use VABS and ANN-based surrogate model to calculate the α^{SG} respectively. The α^{SG} of the blade should be larger than the predefined threshold. For this case, we set $\alpha^{\text{threshold}} = 1.10$.

Example 8 results and discussion

Figure 4.6 shows the validation of beam level failure criterion surrogate model constructed with ANN. Figure 4.6a shows the comparison between ANN predicted and test strength ratios. Note that the test data does not participate in the training of the ANN model. For

Table 4.5. Design variables of the helicopter planform design

Variable	Type	Description	Range
θ_d	continuous	Dihedral angle of the tip of the blade	$[-6°, 6°]$
θ_s	continuous	Sweep angle of the tip of the blade	$[-6°, 6°]$
θ_t	continuous	Twist angle of the tip of the blade	$[-10°, 10°]$

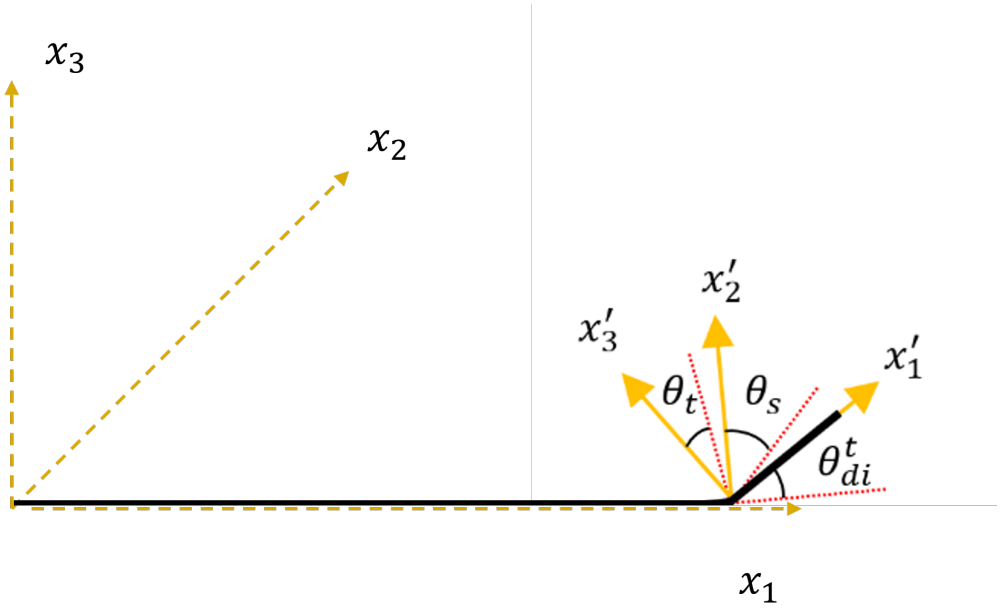
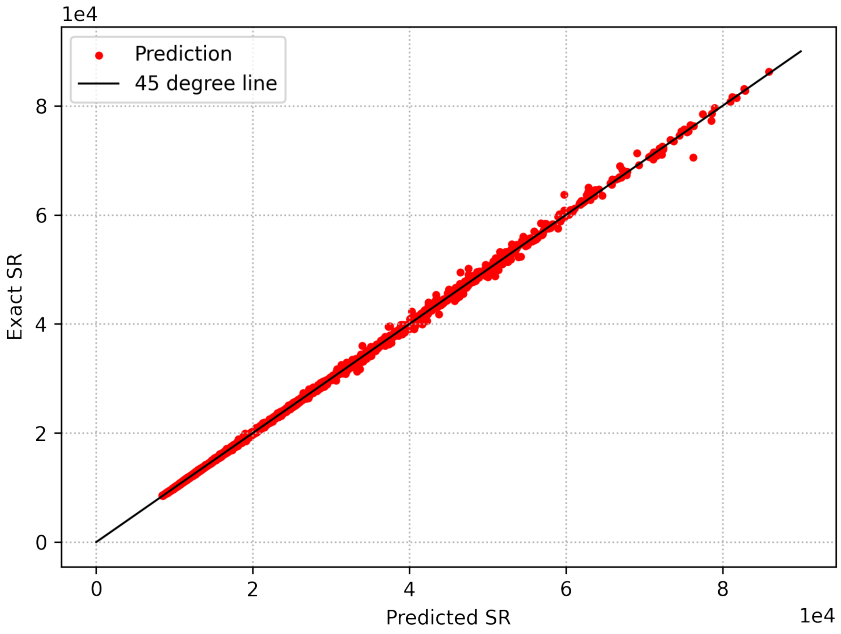


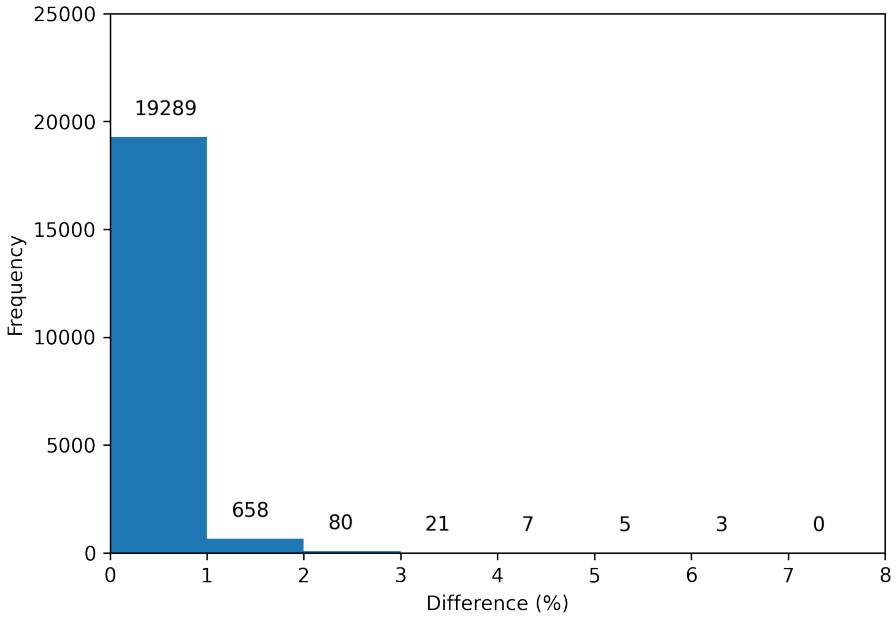
Figure 4.5. Sketch of blade planform design variables

this plot, the horizontal axis and vertical axis represent the predicted data and test data, respectively. The black line is the reference line, which has a slope of 45 degrees. This indicates that the closer the data point locates to the reference line, the better the result will be. As one can tell, ANN can predict the strength ratios very well, as all the data points are very close to the 45-degree line. This can be confirmed from the histogram shown in Figure 4.6b. Figure 4.6b shows that over 98% predictions have errors smaller than 2 %. No prediction has an error larger than 7%. In addition, the coefficient of determination $R^2 = 0.998$. These statistical metrics indicate that ANN can construct an accurate beam level failure criterion.

Figure 4.7 shows the comparison of 2D failure envelopes between ANN and VABS results. These envelopes are the projections of the failure function on the 2D planes. These plots



(a) Comparison between ANN predicted and VABS calculated strength ratios



(b) Histogram of the ANN prediction

Figure 4.6. Validation of ANN constructed beam level failure criterion surrogate model

provide another intuitive way of comparing the quality of the surrogate model. For these envelopes, a load case is safe if it lies on the inside of the failure envelope. Conversely, if a load case lies on the outside of the failure envelope, this load case is unsafe. Overall, one can tell that the results from ANN and VABS match each other very well. Figures 4.6 and 4.7 show confidence on employing the ANN constructed beam level failure criterion surrogate model.

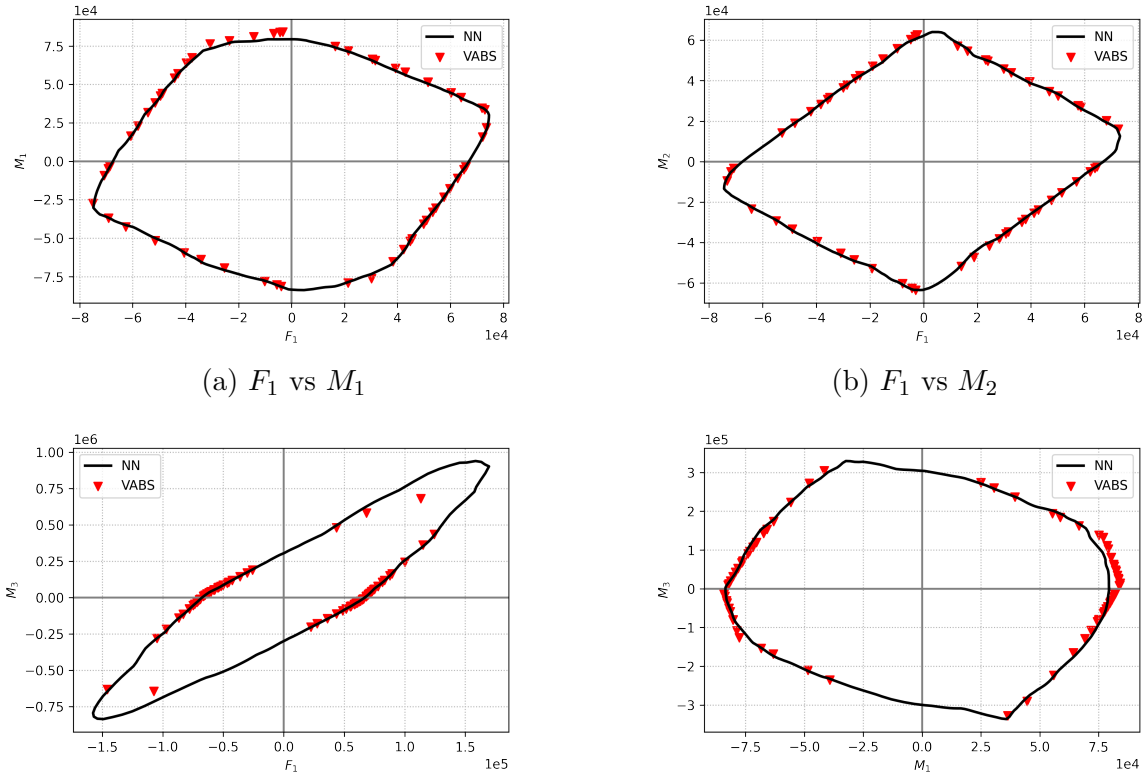


Figure 4.7. Comparison of 2D failure envelopes between ANN and VABS

Figure 4.8 shows the convergence of the optimization. Figures 4.8a and 4.8b present the torque and strength ratio constraint convergence respectively. As one can tell, all three scenarios can reach the optimum design, as all of the torques can converge and the strength ratio constraints could reach the preset threshold. In addition, Figure 4.8 also indicates that the cases with constraint tend to converge to torques that are larger than the case without constraint. This result is reasonable, as the strength constraint could limit the decrease of

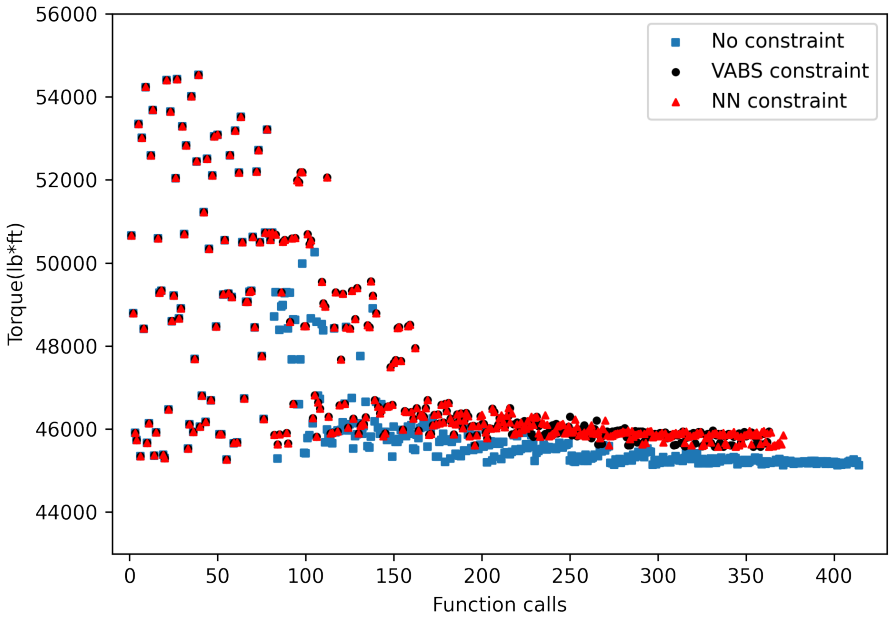
the torque for the sake of safety. Table 4.6 presents the comparison of the strength ratios of all the design scenarios. This table shows that although the optimization without constraint can lead to smaller torque in the rotor, the strength ratio is 0.92, which is smaller than 1. It means this design could have structure failure during the forward flight of the helicopter. For the VABS and ANN surrogate model constraint designs, both strength ratios reached values that are larger than 1.1. This indicates that the optimization with strength constraint can yield a safer design.

Figure 4.9 shows the comparison of torques of the optimum designs. Again, we observe that the optimized designs tend to converge to smaller torques than the base design. This indicates that the optimized case requires less power to remain in the same flight condition. Besides, from this plot, we can observe that the VABS and ANN surrogate model-based optimizations converge to larger torques compared to the design without constraint. This confirmed our observation from Figure 4.8.

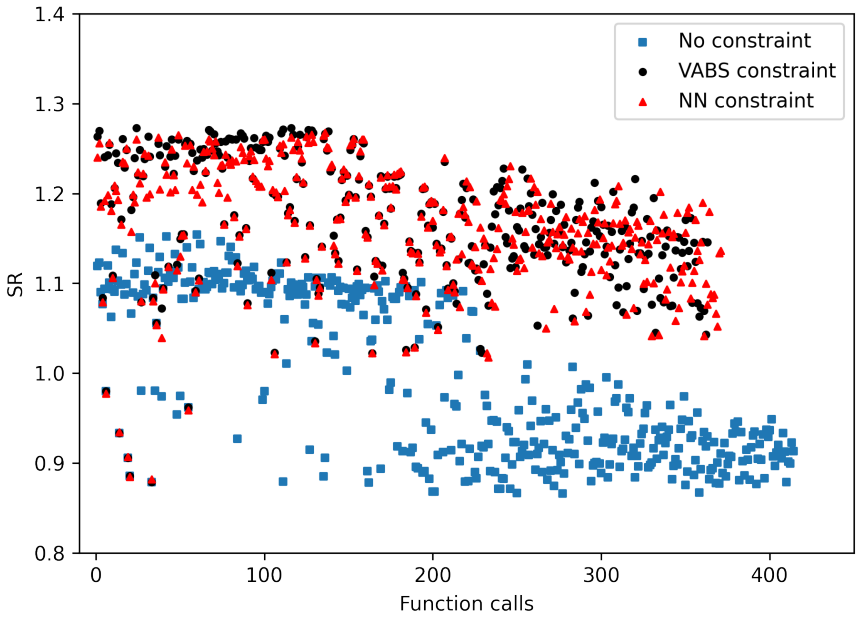
Figure 4.10 presents the comparison of the trims for different cases. This figure shows that the optimized and base designs have some differences, especially for the rolling moment. However, the averages of the trim all reached the target values, with the differences are smaller than 0.1%. This result indicates that the differences in the trim will not impact the flight significantly. All these results show that the optimization achieved our objective.

Table 4.6 lists the design variables for the base design and the three optimized designs. This table shows that θ_d for the three optimized cases are close to each other. Besides, when comparing the design with and without constraint, we can tell that the θ_t will affect the result greatly, as the case with and without constraints are -6.49 and -9.43 degree, respectively.

Finally, we conducted a comparison of the computational time cost for all these scenarios. From table 4.6, we can tell that the cost of the VABS (version 4.0) constraint optimization is 32.64 hours. The ANN constraint optimization cost is 17.66 hours. But, we also need to take the cost of generating the database and training into consideration, for this case, the time costs to generate the database and train the surrogate model are 6.22 hours and 1.51 hours, respectively. So the total time cost of surrogate modeling is 24.39 hours. Thus, we can tell the cost of VABS 4.0 based optimization is more expensive compared to the ANN surrogate model constraint optimizations. Additionally, the new version of VABS (version



(a) Plot of torque



(b) Plot of strength ratio

Figure 4.8. Convergence plots of the blade planform optimization

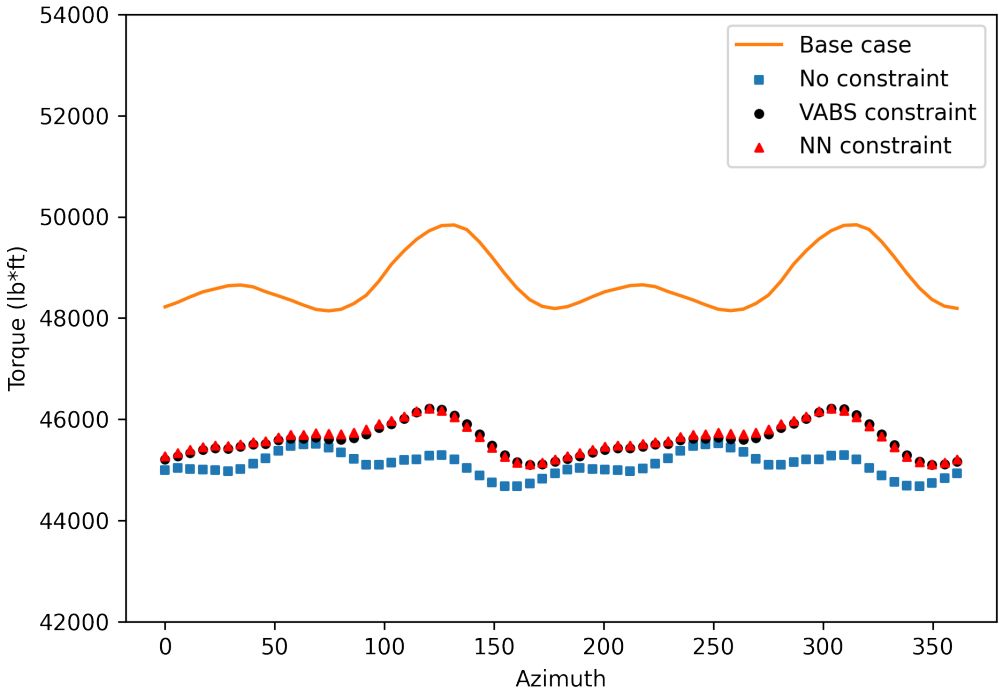
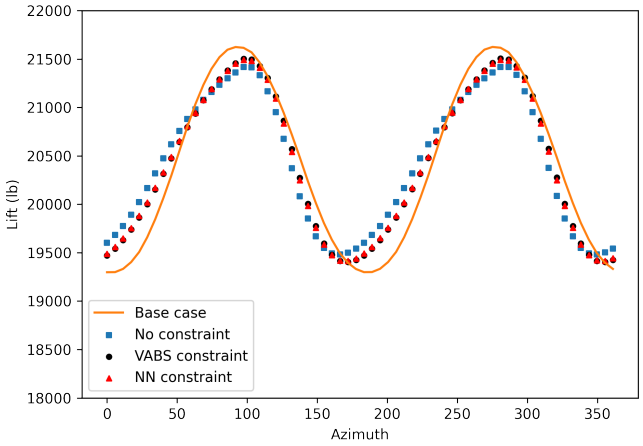


Figure 4.9. Comparison of torque for different optimum cases

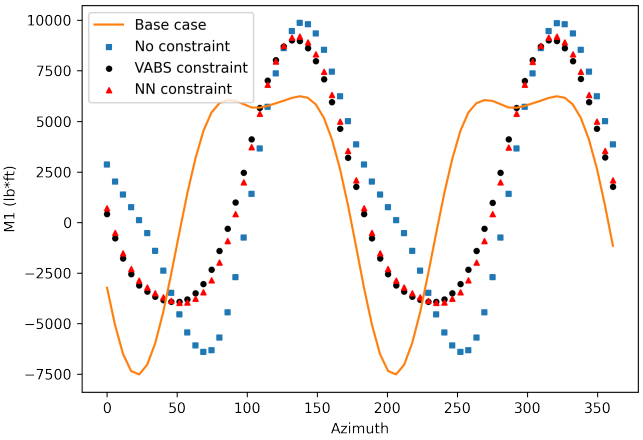
4.1) added a new capability to analyze multiple load cases parallel, which can reduce the computational cost of the blade failure analysis significantly. For this example, the cost to complete the optimization with the VABS 4.1 is 23.16 hours. This cost is about break even with ANN constraint optimization. However, since the helicopter needs to analyze various flight conditions, the ANN surrogate model is still useful to save time on the helicopter design. From these observations, we can conclude that the design with the ANN constraint failure criterion can reduce the optimization cost and yield a safer blade design.

Table 4.6. Results for different optimization cases

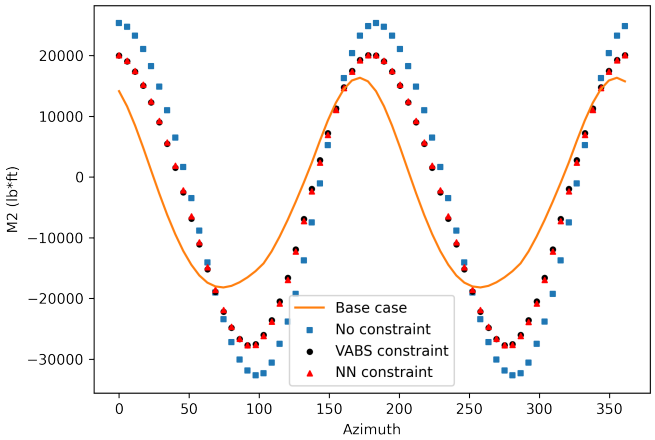
	θ_d	θ_t	θ_s	Lift (lbs)	Torque (lb·ft)	SR	VABS 4.0 time (hrs)	VABS 4.1 time (hrs)
Base case	2.00	0	0	20426.17	48736.14	1.04	-	-
No cstr.	4.72	-9.43	-5.26	20431.48	45128.47	0.92	16.21	-
VABS cstr.	4.72	-6.49	-5.82	20426.59	45587.29	1.13	32.64	23.16
NN cstr.	4.75	-6.49	-5.66	20424.83	45627.87	1.12	17.66	-



(a) Lifting force



(b) Rolling moment



(c) Pitching Moment

Figure 4.10. Comparison of trim for different optimum cases

5. CONCLUSION AND FUTURE WORKS

5.1 Conclusion

In this work, we have shown the implementation of machine learning techniques to discover unknown governing laws of composites. Besides, we also presented an application of machine learning to reduce the computational time of the failure analysis of composites structures .

To learn the constitutive law of composites, we presented a FE coupled deep neural networks mechanics system. This framework makes it possible to use partially experimental data to learn the constitutive law in a form-free manner, which avoids the accuracy loss caused by the presumed expressions in the constitutive law. Besides, we applied a positive definite constraint to form FE-PDNN. This modification solves the convergence robustness issue of learning the constitutive law of a severely damaged material. In addition, the lamination theory is introduced to the FE-PDNN mechanics system that enables FE-PDNN to learn the lamina constitutive law based on the structural response of laminate.

To discover the failure criteria with machine learning and fit the small data size constraint, we have implemented the sparse regression with compressed sensing to discover failure criteria of composites. The proposed method does not need Bigdata to train the model and can identify the most important candidate functions that govern the data. Following the Hashin failure criterion, we divided the failure of composites into tensile and compressive fiber modes, tensile and compressive matrix modes. Additionally, we used an optimization approach to enforce a constraint to the discovered criterion to shift the predicted data to be smaller than the experimental data. This approach can yield a conservative failure criterion and satisfy the engineering design needs.

Finally, we demonstrated employing machine learning to reduce the computational time of the failure analysis of composites structures. We constructed a beam level failure criterion surrogate model with ANN to replace the physics-based strength analysis. The surrogate model was constructed based on the Timoshenko beam model. VABS was used to generate the training data. The trained surrogate model was deployed into the Dakota framework to accomplish the optimization study. The result showed that while the optimization with the

surrogate model reduced the computational time significantly, it can also achieve the same accuracy as the physics-based optimization.

5.2 Future Work

The proposed FE-PDNN framework is remarkably general and can be extended to various applications. Firstly, it would be of interest to apply experimental data from digital image correlation (DIC) tests to the FE-PDNN framework. This would enable the simulation with the real-world data directly. It has the potential to improves both efficiency and accuracy of a simulation, which promotes simulation one step closer to the digital twin. Moreover, it is not limited to discover the mechanical behavior of composites, it can also apply to study the multi-physics effects of materials, such as thermal effect, piezoelectric effect, and electromagnetic effect, etc.

Secondly, I would like to extend the FE-DNN framework to couple with other models, i.e. multiscale model, to promote the understanding of complex material behavior. For some advanced materials, it is difficult to carry out tests to measure the response of the material directly, i.e. the nonlinear constitutive behavior of the yarns of woven composites. This poses challenges to determining the properties of a material. FE-DNN is naturally fit to solve these types of problem since it does not need direct responses. By coupling with other models, FE-DNN can help to calibrate the classical models or construct the constitutive model in a form-free manner with neural networks. This work will facilitate the innovation of the fundamental understanding of advanced materials.

Thirdly, we have employed an optimization approach to design a composites rotor blade. Further, it would be interesting to implement ML techniques to promote the design of advanced materials and structures. Specifically, given the specific performance needs of a material or structure (such as strength and stiffness, fatigue life, etc), the ML algorithm can immediately come up with the design parameters, i.e. for composites, it output the types of matrix and fiber, fiber volume fraction, and the microstructure of the material, etc. The procedure is like the inverse procedure of the blade cross-section design optimization. This study has the potential to reduce design costs and improve design efficiency significantly. In

addition, the ML-designed model can be connected to additive manufacturing equipment, which can print the material or structure directly. This system will avoid computationally expensive design optimization and labor-intensive prototyping. It will significantly improve the efficiency of the design and characterization of advanced materials and structures.

REFERENCES

[1] N. A. Maddock, N. L. James, D. R. McKenzie, and J. F. Patrick, "Technological advances for polymers in active implantable medical devices," in *The Design and Manufacture of Medical Devices*, Elsevier, 2012, pp. 239–272.

[2] S. Chand, "Review carbon fibers for composites," *Journal of Materials Science*, vol. 35, no. 6, pp. 1303–1313, 2000.

[3] B. Bitesize, *Composite materials-using materials gcse chemistry (single science) revision.* [Online]. Available: https://www.bbc.co.uk/bitesize/guides/ztrwng8/revision/6.

[4] Sundar, *What is a reinforcement and matrix in composites?* [Online]. Available: https://extrudesign.com/what-is-a-reinforcement-and-matrix-in-composites/.

[5] A. Moisala, Q. Li, I. Kinloch, and A. Windle, "Thermal and electrical conductivity of single-and multi-walled carbon nanotube-epoxy composites," *Composites Science and Technology*, vol. 66, no. 10, pp. 1285–1288, 2006.

[6] H. Zhang, Z. Zhang, and K. Friedrich, "Effect of fiber length on the wear resistance of short carbon fiber reinforced epoxy composites," *Composites Science and Technology*, vol. 67, no. 2, pp. 222–230, 2007.

[7] A. Kelly, L. McCartney, W. Clegg, and R. Stearn, "Controlling thermal expansion to obtain negative expansivity using laminated composites," *Composites Science and Technology*, vol. 65, no. 1, pp. 47–59, 2005.

[8] A. De Luca and F. Caputo, "A review on analytical failure criteria for composite materials," *AIMS Materials Science*, vol. 4, no. 5, pp. 1165–1185, 2017.

[9] W. Cantwell and J. Morton, "The significance of damage and defects and their detection in composite materials: A review," *The Journal of Strain Analysis for Engineering Design*, vol. 27, no. 1, pp. 29–42, 1992.

[10] R. Talreja, "Manufacturing defects in composites and their effects on performance," *Polymer Composites in the Aerospace Industry*, pp. 83–97, 2020.

[11] D. Li, Q. Yao, C. Zhao, N. Jiang, and L. Jiang, "Charpy transverse impact failure mechanisms of 3d mwk composites at room and liquid nitrogen temperatures," *Journal of Aerospace Engineering*, vol. 28, no. 4, p. 04 014 106, 2015.

[12] F. Christin, "Design, fabrication, and application of thermostructural composites (tsc) like c/c, c/sic, and sic/sic composites," *Advanced Engineering Materials*, vol. 4, no. 12, pp. 903–912, 2002.

[13] A. C. Orifici, I. Herszberg, and R. S. Thomson, "Review of methodologies for composite material modelling incorporating failure," *Composite Structures*, vol. 86, no. 1-3, pp. 194–210, 2008.

[14] M. Egmont-Petersen, D. de Ridder, and H. Handels, "Image processing with neural networks—a review," *Pattern Recognition*, vol. 35, no. 10, pp. 2279–2301, 2002.

[15] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," *Emerging Artificial Intelligence Applications in Computer Engineering*, vol. 160, no. 1, pp. 3–24, 2007.

[16] X. Liu, F. Gasco, J. Goodsell, and W. Yu, "Initial failure strength prediction of woven composites using a new yarn failure criterion constructed by deep learning," *Composite Structures*, vol. 230, p. 111 505, 2019.

[17] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.

[18] F. Tao, X. Lyu, X. Liu, and W. Yu, "Learning composite constitutive laws via coupling abaqus and deep neural network," in *Proceedings of the American Society for Composites—Thirty-fifth Technical Conference*, 2020.

[19] F. Greco, L. Leonetti, R. Luciano, and P. Trovalusci, "Multiscale failure analysis of periodic masonry structures with traditional and fiber-reinforced mortar joints," *Composites Part B: Engineering*, vol. 118, pp. 75–95, 2017.

[20] W. Tan, F. Naya, L. Yang, T. Chang, B. Falzon, L. Zhan, J. Molina-Aldareguía, C. González, and J. Llorca, "The role of interfacial properties on the intralaminar and interlaminar damage behaviour of unidirectional composite laminates: Experimental characterization and multiscale modelling," *Composites Part B: Engineering*, vol. 138, pp. 206–221, 2018.

[21] J. Gu and P. Chen, "Some modifications of Hashin's failure criteria for unidirectional composite materials," *Composite Structures*, vol. 182, pp. 143–152, 2017.

[22] C. H. Wang and C. N. Duong, *Bonded Joints and Repairs to Composite Airframe Structures*. Academic Press, 2015.

[23] S. W. Tsai and E. M. Wu, "A general theory of strength for anisotropic materials," *Journal of Composite Materials*, vol. 5, no. 1, pp. 58–80, 1971.

[24] L. Kroll and W. Hufenbach, "A physically based failure criterion for laminated composites," *Mechanics of Composite Materials*, vol. 35, no. 4, pp. 277–284, 1999.

[25] Z. Hashin, "Failure Criteria for Unidirectional Fiber Composites," *Journal of Applied Mechanics*, vol. 47, no. 2, pp. 329–334, Jun. 1980.

[26] Z. Hashin and A. Rotem, "A fatigue failure criterion for fiber reinforced materials," *Journal of Composite Materials*, vol. 7, no. 4, pp. 448–464, 1973.

[27] A. Puck and H. Schürmann, "Failure analysis of FRP laminates by means of physically based phenomenological models," in *Failure Criteria in Fibre-Reinforced-Polymer Composites*, Elsevier, 2004, pp. 832–876.

[28] R. Cuntze, "The predictive capability of failure mode concept-based strength criteria for multi-directional laminates—part b," *Composites Science and Technology*, vol. 64, no. 3-4, pp. 487–516, 2004.

[29] S. T. Pinho, C. G. Dávila, P. P. Camanho, L. Iannucci, and P. Robinson, "Failure models and criteria for FRP under-in-plane or three-dimensional stress states including shear non-linearity," National Aeronautics and Space Administration, USA, Tech. Rep., 2005.

[30] M. Hinton and A. Kaddour, "The background to the second world-wide failure exercise," *Journal of Composite Materials*, vol. 46, no. 19-20, pp. 2283–2294, 2012.

[31] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Data-driven discovery of partial differential equations," *Science Advances*, vol. 3, no. 4, e1602614, 2017.

[32] S. Tian and W. Yu, "Adaptive sampling assisted surrogate modeling of initial failure envelopes of composite structures," *Composite Structures*, vol. 269, pp. 113–985, 2021.

[33] F. Tao, X. Liu, H. Du, and W. Yu, "Finite element coupled positive definite deep neural networks mechanics system for constitutive modeling of composites," *Computer Methods in Applied Mechanics and Engineering*, vol. 391, p. 114 548, 2022.

[34] G. W. Milton and A. Sawicki, "Theory of Composites. Cambridge Monographs on Applied and Computational Mathematics," *Applied Mechanics Reviews*, vol. 56, no. 2, B27–B28, 2003.

[35] J. Qu and M. Cherkaoui, *Fundamentals of Micromechanics of Solids*. Wiley Hoboken, 2006.

[36] R. Hill, "The elastic behaviour of a crystalline aggregate," *Proceedings of the Physical Society. Section A*, vol. 65, no. 5, p. 349, 1952.

[37] Z. Hashin and S. Shtrikman, "A variational approach to the theory of the elastic behaviour of multiphase materials," *Journal of the Mechanics and Physics of Solids*, vol. 11, no. 2, pp. 127–140, 1963.

[38] R. Hill, "A self-consistent mechanics of composite materials," *Journal of the Mechanics and Physics of Solids*, vol. 13, no. 4, pp. 213–222, 1965.

[39] T. Mori and K. Tanaka, "Average stress in matrix and average elastic energy of materials with misfitting inclusions," *Acta Metallurgica*, vol. 21, no. 5, pp. 571–574, 1973.

[40] C. Sun and R. Vaidya, "Prediction of composite properties from a representative volume element," *Composites Science and Technology*, vol. 56, no. 2, pp. 171–179, 1996.

[41] Z. Xia, Y. Zhang, and F. Ellyin, "A unified periodical boundary conditions for representative volume elements of composites and applications," *International Journal of Solids and Structures*, vol. 40, no. 8, pp. 1907–1921, 2003.

[42] J. Guedes and N. Kikuchi, "Preprocessing and postprocessing for materials based on the homogenization method with adaptive finite element methods," *Computer Methods in Applied Mechanics and Engineering*, vol. 83, no. 2, pp. 143–198, 1990.

[43] J. Fish, Q. Yu, and K. Shek, "Computational damage mechanics for composite materials based on mathematical homogenization," *International Journal for Numerical Methods in Engineering*, vol. 45, no. 11, pp. 1657–1679, 1999.

[44] W. Yu, "Simplified formulation of mechanics of structure genome," *AIAA Journal*, vol. 57, no. 10, pp. 4201–4209, 2019.

[45] X. Liu, K. Rouf, B. Peng, and W. Yu, "Two-step homogenization of textile composites using mechanics of structure genome," *Composite Structures*, vol. 171, pp. 252–262, 2017.

[46] F. Tao, X. Lyu, X. Liu, and W. Yu, "Multiscale analysis of multilayer printed circuit board using mechanics of structure genome," *Mechanics of Advanced Materials and Structures*, pp. 1–10, 2019.

[47] L. M. Kachanov, "Rupture time under creep conditions," *International Journal of Fracture*, vol. 97, no. 1-4, pp. 11–18, 1999.

[48] J.-L. Chaboche, "Continuous damage mechanics—a tool to describe phenomena before crack initiation," *Nuclear Engineering and Design*, vol. 64, no. 2, pp. 233–247, 1981.

[49] D. Krajcinovic and G. U. Fonseka, "The continuous damage theory of brittle materials, Part 1: General theory," *Journal of Applied Mechanics*, vol. 48, no. 4, pp. 809–815, 1981.

[50] A. Turon, P. P. Camanho, J. Costa, and C. Dávila, "A damage model for the simulation of delamination in advanced composites under variable-mode loading," *Mechanics of Materials*, vol. 38, no. 11, pp. 1072–1089, 2006.

[51] J. Lemaitre, *A Course on Damage Mechanics*. Springer Science & Business Media, 2012.

[52] Z. Gao, L. Zhang, and W. Yu, "A nonlocal continuum damage model for brittle fracture," *Engineering Fracture Mechanics*, vol. 189, pp. 481–500, 2018.

[53] P. Liu and J. Zheng, "Recent developments on damage modeling and finite element analysis for composite laminates: A review," *Materials & Design*, vol. 31, no. 8, pp. 3825–3834, 2010.

[54] R. Talreja, "A continuum mechanics characterization of damage in composite materials," *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, vol. 399, no. 1817, pp. 195–216, 1985.

[55] A. Matzenmiller, J. Lubliner, and R. Taylor, "A constitutive model for anisotropic damage in fiber-composites," *Mechanics of Materials*, vol. 20, no. 2, pp. 125–152, 1995.

[56] P. Maimí, P. P. Camanho, J. Mayugo, and C. Dávila, "A continuum damage model for composite laminates, Part I–constitutive model," *Mechanics of Materials*, vol. 39, no. 10, pp. 897–908, 2007.

[57] P. Maimí, P. P. Camanho, J. Mayugo, and C. Dávila, "A continuum damage model for composite laminates, Part II–computational implementation and validation," *Mechanics of Materials*, vol. 39, no. 10, pp. 909–919, 2007.

[58] I. Lapczyk and J. A. Hurtado, "Progressive damage modeling in fiber-reinforced materials," *Composites Part A: Applied Science and Manufacturing*, vol. 38, no. 11, pp. 2333–2341, 2007.

[59] F. Jiang and W. Yu, "Damage analysis by physically nonlinear composite beam theory," *Composite Structures*, vol. 182, pp. 652–665, 2017.

[60] Degrieck, and W. Van Paepegem, "Fatigue damage modeling of fibre-reinforced composite materials: Review," *Applied Mechanics Reviews*, vol. 54, no. 4, pp. 279–300, Jul. 2001.

[61] J. Fong, "What is fatigue damage?" In *Damage in Composite Materials: Basic Mechanisms, Accumulation, Tolerance, and Characterization*, ASTM International, 1982.

[62] T. Kirchdoerfer and M. Ortiz, "Data-driven computational mechanics," *Computer Methods in Applied Mechanics and Engineering*, vol. 304, pp. 81–101, 2016.

[63] K. Wang and W. Sun, "A multiscale multi-permeability poroplasticity model linked by recursive homogenizations and deep learning," *Computer Methods in Applied Mechanics and Engineering*, vol. 334, pp. 337–380, 2018.

[64] M. Mozaffar, R. Bostanabad, W. Chen, K. Ehmann, J. Cao, and M. Bessa, "Deep learning predicts path-dependent plasticity," *Proceedings of the National Academy of Sciences*, vol. 116, no. 52, pp. 26 414–26 420, 2019.

[65] F. E. Fossan, L. O. Müller, J. Sturdy, A. T. Bråten, A. Jørgensen, R. Wiseth, and L. R. Hellevik, "Machine learning augmented reduced-order models for ffr-prediction," *Computer Methods in Applied Mechanics and Engineering*, vol. 384, p. 113 892, 2021.

[66] B. Bahmani and W. Sun, "A kd-tree-accelerated hybrid data-driven/model-based approach for poroelasticity problems with multi-fidelity multi-physics data," *Computer Methods in Applied Mechanics and Engineering*, vol. 382, p. 113 868, 2021.

[67] H. J. Logarzo, G. Capuano, and J. J. Rimoli, "Smart constitutive laws: Inelastic homogenization through machine learning," *Computer Methods in Applied Mechanics and Engineering*, vol. 373, p. 113 482, 2021.

[68] H. You, Y. Yu, N. Trask, M. Gulian, and M. D'Elia, "Data-driven learning of nonlocal physics from high-fidelity synthetic data," *Computer Methods in Applied Mechanics and Engineering*, vol. 374, p. 113 553, 2021.

[69] R. Eggersmann, T. Kirchdoerfer, S. Reese, L. Stainier, and M. Ortiz, "Model-free data-driven inelasticity," *Computer Methods in Applied Mechanics and Engineering*, vol. 350, pp. 81–99, 2019.

[70] D. Z. Huang, K. Xu, C. Farhat, and E. Darve, "Predictive modeling with learned constitutive laws from indirect observations," *arXiv preprint arXiv:1905.12530*, 2019.

[71] K. Xu, D. Z. Huang, and E. Darve, "Learning constitutive relations using symmetric positive definite neural networks," *arXiv preprint arXiv:2004.00265*, 2020.

[72] H. R. Tamaddon-Jahromi, N. K. Chakshu, I. Sazonov, L. M. Evans, H. Thomas, and P. Nithiarasu, "Data-driven inverse modelling through neural network (deep learning) and computational heat transfer," *Computer Methods in Applied Mechanics and Engineering*, vol. 369, p. 113 217, 2020.

[73] M. Flaschel, S. Kumar, and L. De Lorenzis, "Unsupervised discovery of interpretable hyperelastic constitutive laws," *Computer Methods in Applied Mechanics and Engineering*, vol. 381, p. 113 852, 2021.

[74] J. Lim, L. Allen, R. B. Haehnel, and I. Dettwiller, "An examination of aerodynamic and structural loads for a rotor blade optimized with multi-objective genetic algorithm," in *Vertical Flight Society's 77th Annual Forum & Technology Display*, pp. 10–14.

[75] D. H. Hodges, "Review of composite rotor blade modeling," *AIAA Journal*, vol. 28, no. 3, pp. 561–565, 1990.

[76] J. L. Walsh, "Performance optimization of helicopter rotor blades," in *Air Force/NASA Symposium on Recent Advances in Multi-Disciplinary Analyses and Optimization*, 1991.

[77] E. R. Leon, A. L. Pape, M. Costes, J.-A. Désidéri, and D. Alfano, "Concurrent aerodynamic optimization of rotor blades using a Nash game method," *Journal of the American Helicopter Society*, vol. 61, no. 2, pp. 1–13, 2016.

[78] J. Nash, "Non-cooperative games," *Annals of Mathematics*, vol. 54, pp. 286–295, 1951.

[79] L. Wang, B. Diskin, R. T. Biedron, E. J. Nielsen, V. Sonneville, and O. A. Bauchau, "High-fidelity multidisciplinary design optimization methodology with application to rotor blades," *Journal of the American Helicopter Society*, vol. 64, no. 3, pp. 1–11, 2019.

[80] P. E. Gill, W. Murray, and M. A. Saunders, "Snopt: An sqp algorithm for large-scale constrained optimization," *SIAM Review*, vol. 47, no. 1, pp. 99–131, 2005.

[81] J. Lim, "Consideration of structural constraints in passive rotor blade design for improved performance," *The Aeronautical Journal*, vol. 119, no. 1222, pp. 1513–1539, 2015.

[82] S. Tian, F. Tao, H. Du, W. Yu, J. W. Lim, R. B. Haehnel, Y. Wenren, and L. D. Allen, "Structural design optimization of composite rotor blades with strength considerations," in *AIAA SciTech 2022 Forum*, 2022, p. 2454.

[83] W. Yu, V. V. Volovoi, D. H. Hodges, and X. Hong, "Validation of the variational asymptotic beam sectional analysis," *AIAA Journal*, vol. 40, no. 10, pp. 2105–2112, 2002.

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

[85] M. A. Nielsen, *Neural Networks and Deep Learning*. Determination press San Francisco, CA, USA, 2015, vol. 25.

[86] J. Brownlee, "A tour of machine learning algorithms," *Machine Learning Mastery*, vol. 25, 2013.

[87] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, *The Finite Element Method: its Basis and Fundamentals*. Elsevier, 2005.

[88] I. Farmaga, P. Shmigelskyi, P. Spiewak, and L. Ciupinski, "Evaluation of computational complexity of finite element analysis," in *2011 11th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*, IEEE, 2011, pp. 213–214.

[89] E. Tsironi, P. Barros, C. Weber, and S. Wermter, "An analysis of convolutional long short-term memory recurrent neural networks for gesture recognition," *Neurocomputing*, vol. 268, pp. 76–86, 2017.

[90] X. Liu, F. Tao, and W. Yu, "A neural network enhanced system for learning nonlinear constitutive relation of fiber reinforced composites," in *AIAA SciTech 2020 Forum*, 2020, p. 0396.

[91] J. Nocedal and S. Wright, *Numerical Optimization*. Springer Science & Business Media, 2006.

[92] A. Documentation and U. Manual, "Version 6.14," Tech. Rep., 2010.

[93] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, pp. 8026–8037.

[94] O. C. Zienkiewicz, R. L. Taylor, P. Nithiarasu, and J. Zhu, *The Finite Element Method*. McGraw-hill London, 1977, vol. 3.

[95] P. P. Camanho and C. G. Dávila, "Mixed-mode decohesion finite elements for the simulation of delamination in composite materials," National Aeronautics and Space Administration, USA, Tech. Rep., 2002.

[96] B. M. Willis, "Applying finite element analysis with a focus on tensile damage modeling of carbon fiber reinforced polymer laminates," Ph.D. dissertation, The Ohio State University, 2013.

[97] S. P. Timoshenko and S. Woinowsky-Krieger, *Theory of Plates and Shells*. McGraw-hill, 1959.

[98] M. Hinton, P. Soden, and A.-S. Kaddour, *Failure criteria in fibre reinforced polymer composites: the world-wide failure exercise*. Elsevier, 2004.

[99] K.-S. Liu and S. W. Tsai, "A progressive quadratic failure criterion for a laminate," *Composites Science and Technology*, vol. 58, no. 7, pp. 1023–1032, 1998.

[100] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis*. John Wiley & Sons, 2021.

[101] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[102] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*, 10. Springer Series in Statistics New York, 2001, vol. 1.

[103] W. Wang, R. Yang, Y. Lai, V. Kovanis, and C. Grebogi, "Predicting catastrophes in nonlinear dynamical systems by compressive sensing," *Physical Review Letters*, vol. 106, no. 15, p. 154 101, 2011.

[104] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[105] E. J. Candes, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 59, no. 8, pp. 1207–1223, 2006.

[106] M. Schmidt and H. Lipson, "Distilling free-form natural laws from experimental data," *Science*, vol. 324, no. 5923, pp. 81–85, 2009.

[107] P. Soden, M. Hinton, and A. Kaddour, "Lamina properties, lay-up configurations and loading conditions for a range of fibre reinforced composite laminates," in *Failure Criteria in Fibre-reinforced-polymer Composites*, Elsevier, 2004, pp. 30–51.

[108] M. Hinton, A. Kaddour, and P. Soden, "The world-wide failure exercise: Its origin, concept and content," in *Failure Criteria in Fibre-Reinforced-Polymer Composites*, Elsevier, 2004, pp. 2–28.

[109] R. W. Schafer, "What is a savitzky-golay filter?" *IEEE Signal Processing Magazine*, vol. 28, no. 4, pp. 111–117, 2011.

[110] U. Hutter, H. Schelling, and H. Krauss, "An experimental study to determine failure envelope of composite materials with tubular specimens under combined loads and comparison between several classical criteria," *AGARD Specialists Meeting on Failure Modes of Composite Mater*, 1975.

[111] F. Tao, X. Liu, H. Du, and W. Yu, "Physics-informed artificial neural network approach for axial compression buckling analysis of thin-walled cylinder," *AIAA Journal*, vol. 58, no. 6, pp. 2737–2747, 2020.

[112] M. Ohlsson, C. Peterson, and B. Söderberg, "Neural networks for optimization problems with inequality constraints: The knapsack problem," *Neural Computation*, vol. 5, no. 2, pp. 331–339, 1993.

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

[114] M. S. Eldred, K. R. Dalbey, W. J. Bohnhoff, B. M. Adams, L. P. Swiler, P. D. Hough, D. M. Gay, J. P. Eddy, and K. H. Haskell, *Dakota: A multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis. version 5.0, User's Manual*, 2010.

[115] O. A. Bauchau, *Dymore user's manual*, 2007.

[116] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, vol. 4, no. 2, pp. 65–85, 1994.

A. SUPPORTING MATERIAL

A.1 Introduction to L-BFGS Optimization Algorithm

Given a convex and twice continuously differentiable loss function $\boldsymbol{f}(\boldsymbol{x})$, where $\boldsymbol{x}$ is a d-dimension column vector. To minimize $\boldsymbol{f}(\boldsymbol{x})$, Newton's method tries to create a series $\boldsymbol{x}_k$ from the initial guess $\boldsymbol{x}_0$ that converges toward the minimum. Each iteration has the form [91]

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{t} = \boldsymbol{x}_k - \boldsymbol{H}_k^{-1} \nabla \boldsymbol{f}\left(\boldsymbol{x}_k\right) \tag{A.1}$$

where $\boldsymbol{H}_k$ is the Hessian matrix. By using second-order Taylor approximation, the second order of Taylor expansion of $\boldsymbol{f}$ at $\boldsymbol{x}_k$ is

$$\boldsymbol{f}\left(\boldsymbol{x}_k + \boldsymbol{t}\right) \approx \boldsymbol{f}\left(\boldsymbol{x}_k\right) + \nabla \boldsymbol{f}\left(\boldsymbol{x}_k\right) \boldsymbol{t} + \frac{1}{2} \nabla^2 \boldsymbol{f}\left(\boldsymbol{x}_k\right) \boldsymbol{t}^2 \tag{A.2}$$

The subsequent iterate $\boldsymbol{x_{k+1}}$ is defined to minimize Eq. (A.2) in $\boldsymbol{t}$ and $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{t}$. If the second derivative is positive and Eq. (A.2) is convex on $\boldsymbol{t}$, one get

$$\boldsymbol{0} = \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{t}} \left(\boldsymbol{f}\left(\boldsymbol{x}_k\right) + \nabla \boldsymbol{f}\left(\boldsymbol{x}_k\right) \boldsymbol{t} + \frac{1}{2} \nabla^2 \boldsymbol{f}\left(\boldsymbol{x}_k\right) \boldsymbol{t}^2 \right) = \nabla \boldsymbol{f}\left(\boldsymbol{x}_k\right) + \nabla^2 \boldsymbol{f}\left(\boldsymbol{x}_k\right) \boldsymbol{t} \tag{A.3}$$

Thus, the minimum can be found at

$$\boldsymbol{t} = -\frac{\nabla \boldsymbol{f}\left(\boldsymbol{x}_k\right)}{\nabla^2 \boldsymbol{f}\left(\boldsymbol{x}_k\right)} = -\boldsymbol{H}_k^{-1} \nabla \boldsymbol{f}\left(\boldsymbol{x}_k\right) \tag{A.4}$$

Newton's method converges fast and robust. However, it requires computing the inverse of the Hessian matrix, which can be too expensive for deep neural networks. In this work, the Limited-memory BFGS (L-BFGS) optimizer is implemented to train the model. L-BFGS is a Quasi-Newton method that approximates the inverse Hessian matrix using limited memory.

If one defines

$$\begin{aligned} \boldsymbol{s}_k &= \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \\ \boldsymbol{y}_k &= \boldsymbol{g}_{k+1} - \boldsymbol{g}_k \\ \boldsymbol{\rho}_k &= \frac{1}{\boldsymbol{y}_k^T \boldsymbol{s}_k} \end{aligned} \tag{A.5}$$

where $\boldsymbol{g}_k \equiv \nabla \boldsymbol{f}\left(\boldsymbol{x}_k\right)$. Then, with L-BFGS method, the inverse Hessian can be approximated [91]

$$\boldsymbol{H}_{k+1}^{-1} = \left(\boldsymbol{I} - \boldsymbol{\rho}_k \boldsymbol{s}_k \boldsymbol{y}_k^T\right) \boldsymbol{H}_k \left(\boldsymbol{I} - \boldsymbol{\rho}_k \boldsymbol{y}_k \boldsymbol{s}_k^T\right) + \boldsymbol{\rho}_k \boldsymbol{s}_k \boldsymbol{s}_k^T \tag{A.6}$$

This method is called as L-BFGS. It shares lots of features with Newton's method. But it computes the inverse Hessian with an approximation method, which reduced the computational cost significantly.

A.2 Plate Equilibrium Equations

The plate equilibrium equations are

$$\begin{aligned}
&\frac{\partial N_{11}}{\partial x_1} + \frac{\partial N_{12}}{\partial x_2} + p_1 = 0 \\
&\frac{\partial N_{21}}{\partial x_1} + \frac{\partial N_{22}}{\partial x_2} + p_2 = 0 \\
&\frac{\partial N_{13}}{\partial x_1} + \frac{\partial N_{23}}{\partial x_2} + p_3 = 0 \\
&\frac{\partial M_{12}}{\partial x_1} + \frac{\partial M_{22}}{\partial x_2} - q_1 - N_{23} = 0 \\
&\frac{\partial M_{11}}{\partial x_1} + \frac{\partial M_{21}}{\partial x_2} + q_2 - N_{13} = 0 \\
&\frac{\partial^2 M_{11}}{\partial x_1^2} + \frac{\partial^2 M_{22}}{\partial x_2^2} + 2\frac{\partial^2 M_{12}}{\partial x_1 \partial x_2} + \frac{\partial q_2}{\partial x_1} - \frac{\partial q_1}{\partial x_2} + p_3 = 0
\end{aligned}$$

where p_i and q_i are the plate distributed forces and moments over the reference surface.

A.3 Sparse Regression Algorithm

```
%% compute Sparse regression: sequential least squares
Xi = Theta\dXdt;  % initial guess: Least-squares

% lambda is our sparsification knob.
for k=1:10
    smallinds = (abs(Xi)<lambda);   % find small coefficients
    Xi(smallinds)=0;                % and threshold
    for ind = 1:n                   % n is state dimension
        biginds = ~smallinds(:,ind);
        % Regress dynamics onto remaining terms to find sparse Xi
        Xi(biginds,ind) = Theta(:,biginds)\dXdt(:,ind);
    end
end
```

Figure A.1. Sparse regression algorithm in MATLAB [17]

VITA

Fei Tao was born on Jan. 18, 1992, in Bozhou, Anhui, China. He graduated from Northwest A&F University, Shaanxi, China on July 2013 with a Bachelor's degree in Civil Engineering. In Fall 2013, he started his graduate program in Civil Engineering at Lyles School of Civil Engineering at Purdue University supervised under Prof. Bobet Antonio. In Fall 2016, he start the Ph.D. program at the School of Aeronautics and Astronautics at Purdue University working under guidance of Prof. Wenbin Yu. During the course of the Ph.D. program, he went for an internship at Dassault Systemes, West Lafayette from Jun. 2018 to Dec. 2018. At the internship, he worked as a solution consultant on various projects. His expertise is in the field of machine learning, finite element analysis, constitutive modeling, micromechanics, and mechanics of composite structures and materials.