Michigan Technological University
Digital Commons @ Michigan Tech

Dissertations, Master's Theses and Master's Reports

2021

DEFORMATION MANIFOLD LEARNING MODEL FOR MULTI WALLED CARBON NANOTUBES

Shashank S. Pathrudkar
Michigan Technological University, sspathru@mtu.edu

Copyright 2021 Shashank S. Pathrudkar

Recommended Citation

Pathrudkar, Shashank S., "DEFORMATION MANIFOLD LEARNING MODEL FOR MULTI WALLED CARBON NANOTUBES", Open Access Master's Thesis, Michigan Technological University, 2021.
https://doi.org/10.37099/mtu.dc.etdr/1173

Follow this and additional works at: https://digitalcommons.mtu.edu/etdr
Part of the Materials Science and Engineering Commons, and the Mechanical Engineering Commons

المنارة للاستشارات

DEFORMATION MANIFOLD LEARNING MODEL FOR MULTI WALLED CARBON NANOTUBES

By

Shashank Pathrudkar

A THESIS

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

In Mechanical Engineering

MICHIGAN TECHNOLOGICAL UNIVERSITY

2021

© 2021 Shashank Pathrudkar

للاستشارات المنارة

This thesis has been approved in partial fulfillment of the requirements for the Degree of MASTER OF SCIENCE in Mechanical Engineering.

Department of Mechanical Engineering - Engineering Mechanics

Thesis Advisor: *Dr. Susanta Ghosh*

Committee Member: *Dr. Amartya Banerjee*

Committee Member: *Dr. Ranjit Pati*

Committee Member: *Dr. Soumik Sarkar*

Department Chair: *Dr. William Predebon*

Dedication

To my parents and grandparents

without whom I would neither be who I am nor I would have been able to contribute to this work.

Contents

List of Figures . . . . **ix**

Preface . . . . **xiii**

Acknowledgments . . . . **xv**

List of Abbreviations . . . . **xvii**

Abstract . . . . **xix**

1 Introduction and Motivation **1**

1.1 Two Dimensional Materials and Multi Walled Carbon Nanotubes . 1

1.2 Atomistic-Continuum Models 3

1.3 Data driven approaches 4

1.4 Dimensionality Reduction 6

2 Foliation Model and Data Generation **9**

2.1 Kinematics of the Foliation Model 10

2.2 Data generation of MWCNT using Foliation Model 14

3 Deformation Manifold Learning Model 17
3.1 Proposed Dimensionality Reduction Technique 18
3.1.1 FPCA 18
3.1.2 Proposed Constrained–FPCA 20
3.2 Learning in the Reduced Dimension through Deep Neural Networks 22
3.3 Limitations of FPCA 28
4 Results, Discussions and Future Scope 33
4.1 Dimensionality Reduction through Proposed Constrained FPCA . . 33
4.2 Accuracy 35
4.3 Interpretability of the DML model 40
4.4 Conclusions 41
References 43

List of Figures

2.1 Kinematics of Foliation model showing undeformed configuration space, parametric space, deformed configuration and the maps between them. Reprinted from [75]. 10

3.1 A detailed schematics of the present framework involving the data generation via simulation and the proposed Deformation Manifold Learning(DML) model. The DML model includes: the dimensionality reduction and the DNN–based learning. Inputs and outputs of DNN are shown with green arrows. Yellow arrows represent the prediction via the DML model to obtain the deformation for a given input. Reprinted from [74]. 18

3.2 Learning Curves for the DNNs. Training loss (- - -) and Validation loss (- - -) are plotted against the number of epochs for DNN used for CoF-PCs corresponding to (a) Torsional deformation, (b) in–plane bending deformation, and (c) out–of–plane bending deformation. Reprinted from [75]. 25

3.3 Cross–sections of MWCNT obtained using two approaches of DML model (- - -) that uses FPCA: (a) FPCA coupled with DNN and (c) FPCA coupled with the constrained–DNN. These two approaches are compared against the AC model (—). (b) and (d) are the close–up views of the blue boxes corresponding to (a) and (c). (b) Shows a discontinuity in the DML model and (d) shows increases error in prediction everywhere. Reprinted from [75]. 29

3.4 Error balancing through L-curve. Each point on the plot corresponds to a λ value. Reprinted from [75]. 30

3.5 Comparison of mean squared error (in the actual deformation) and error in constraint vs twisting angle at different penalty parameters ($\lambda = 0.00$, $\lambda = 0.09$ and $\lambda = 1.00$). Reprinted from [75]. 31

4.1 Cumulative % variance captured by principal components for MWCNTs under torsion (left) and bending (right). Reprinted from [75]. . 34

4.2 CoFPCs for the outermost wall of 40 walled tube under 25° twist at a particular cross–section. Reprinted from [75]. 34

4.3 Normalized CoFPCs shown as colorbar for $N_w = 10, 15, 20, 25, 30, 35, 40$. CoFPCs are shown for first 16 FPCs. CoFPCs are normalized by the maximum value for each tube. Reprinted from [75]. 35

4.4 Correlation plots for test set CoFPCS of (a) in–plane deformation in torsion, (b) in–plane and (c) out–of–plane deformation in bending. $R = 0.9943$(a), 0.9931(b), 0.9991(c). Reprinted from [75]. 35

4.5 (a) Twisted 40 walled CNT obtained via AC (top) and DML (bottom) model. (b) Radial deformation colormap (Red: high, Blue: low). Alternate walls of cross–sections obtained via AC (—) and DML (- - -) models, for 10, 20, 30, and 40 walled CNTs. Reprinted from [75]. . . 37

4.6 Bent 35–walled CNT obtained via AC (a,c) and DML (b,d) model.(c) and (d) show colormap of radial deformations corresponding to (a) and (b). (e–g) Alternate walls of cross–sections obtained via AC (—) and DML (- - -) model.Reprinted from [75]. 38

4.7 Comparison of AC (top) and DML (bottom) models for a 32 walled CNT (system which is not a part of training data) under torsion (a) and bending (b). Reprinted from [75]. 39

4.8 Energy comparison for (15, 25, 32, 40 –walled) MWCNTs under torsion (a) and bending (b) via AC (○) and DML model (+). 32 walled CNT (+) is an unknown system. The lines (—) and (—) are drawn to highlight pre– and post–buckling regimes. Reprinted from [75]. . . . 39

4.9 Functional Principal components of MWCNTs under torsion (top) and bending (bottom).Reprinted from [75]. 41

Preface

Most of the content of this thesis is published in the following journal article:

"Interpretable machine learning model for the deformation of multiwalled carbon nanotubes", Upendra Yadav, Shashank Pathrudkar, and Susanta Ghosh, *Physical Review B*, American Physical Society, 2021.

American Physical Society has granted permission to the authors to use the published article in this thesis. Co-authors of that paper, Upendra Yadav and Dr. Susanta Ghosh have permitted me to use the published article in this thesis. I am grateful to Upendra Yadav and Dr. Susanta Ghosh to permit me for using the text and figures from the original article and the supplementary material of the article. In that paper, I contributed to the formulation and execution of the proposed dimensionality reduction technique and the development of Deep Neural Networks.

Acknowledgments

I would like to express my sincere gratitude towards my advisor Dr. Susanta Ghosh for letting me pursue my MS studies under his able guidance and inspiring me throughout my journey. His support and constructive criticism has helped me paved my way and overcome the obstacles in this journey. Without his direction it would have been impossible for me to attempt this challenging work.

I am grateful to Dr. Amartya Banerjee, University of California, Los Angeles, Dr. Ranjit Pati, Michigan Technological University, and Dr. Soumik Sarkar, Iowa State University for taking out their time and being a part of my committee. I thank them for their feedback on my work.

I would like to thank my parents and my brother without whom I would not have been able to reach this point in my life. I am grateful for their support, love, prayers, and sacrifices, and I am certain that I will receive their support in future endeavors.

The role of Michigan Technological University, Mechanical Engineering-Engineering Mechanics department, and Dr. William Predebon has been vital in this journey and I thank them for providing all the necessary lab equipment, employment opportunities, and other facilities. I acknowledge the high-performance computing resources from the SUPERIOR computing facility at Michigan Tech and the Extreme Science and

Engineering Discovery Environment(XSEDE), which is supported by the NSF grant number ACI-1548562. (Request number: MSS190003, MSS200004). I would like to thank Dr. Susanta Ghosh for granting me access to these supercomputing facilities.

My journey till this point was fueled by excellent teachers who took the effort to bring out the best in me. I would thank all of them, especially Dr. Susanta Ghosh, Dr. Ibrahim Miskioglu, Dr. Trisha Sain, Dr. Kui Zhang, Dr. Timothy Havens, and Dr. Anthony Pinar.

I am blessed with friends and lab mates who have always encouraged me and stood with me in all of my ups and downs in this journey. I would like to thank members of Dr. Susanta Ghosh's lab, Upendra Yadav, and Revanth Mattey for helping me understand numerical methods and continuum mechanics. They always shared their deep understanding of the topic, which helped me construct this work. I would also thank other members Ponkrshnan Thiagarajan, Parag Nikam, and Pushkar Khairnar. I would thank my friends Rushikesh Kulkarni, Chinmay Sathe, Swapnil Bamane, Prashik Gaikwad, Swarnima Pardeshi, and Revanth Mattey to make this ride at Michigan Technological University smooth for me. I would like to express my gratitude towards Aishwarya Unta and Shreyas Kulkarni for their long-time friendship and empathy. I would never be able to thank them enough.

I humbly extend my thanks to all the people who are a part of this journey and pray to receive their continued support.

List of Abbreviations

2D	Two Dimensional
3D	Three Dimensional
c-FPCA	Constrained Functional Principal Component Analysis
CNT	Carbon Nanotube
CoFPCs	Coefficients of Functional Principal Components
DML	Deformation Manifold Learning
DNN	Deep Neural Network
MWCNT	Multiwalled Carbon Nanotube
FDA	Functional Data Analysis
FPCA	Functional Principal Component Analysis
HPC	High Performance Computing
ML	Machine Learning
PCA	Principal Component Analysis
TEM	Transmission electron microscopy

Abstract

Two-Dimensional (2D) materials are being studied widely by researchers due to their superior material properties over the bulk materials. Since the isolation of graphene in 2004, graphene has gained popularity amongst the 2D materials community. Graphene when rolled into sheets form Carbon Nanotubes (CNTs) which possess excellent mechanical and electrical properties. Concentric stacks of CNTs yield Multi-walled Carbon Nanotubes (MWCNTs) which are superior to CNTs in certain aspects. It has been well established that the deformation of CNTs and MWCNTs change their mechanical and electrical properties significantly. This has opened doors for CNTs into numerous applications and also piqued the need of studying the deformation characteristics of CNTs. Efforts have been made by researchers to develop models that approximate the geometry of CNTs and simulate them under given loading conditions. Atomistic models, Continuum models, and atomistic-continuum models have been used to simulate the deformation of CNTs. These models have been accurate in generating the deformed CNTs and are in good agreement with the experimental results. The models have also been proven to work well for MWCNTs having millions of atoms. Despite being accurate these models require high computation power which is a bottleneck in the wide use of these models. In this work, we present a data-driven model to predict the deformation of MWCNTs under torsional and bending loads.

Million atom MWCNTs are discretized and represented through a proposed dimensionality reduction technique described as constrained-Functional Principal Component Analysis. Further, learning is performed using Deep Neural Networks (DNNs) in the dimensionally reduced space. The proposed framework accurately predicts the deformation of MWCNTs and is in good agreement with the atomistic-physics simulations. The proposed model has an edge over traditional models in regards to the computational time and computational power required. The model yields dominant patterns of deformation which explain the prediction capability of the model. This makes our model comprehensible. The model is currently developed for MWCNTs and is presented here, but the model can be extended to other 2D materials and can form a basis towards the use of data-driven approaches for exploring the mechanics and physics of 2D materials.

Chapter 1 presented here introduces the topic, provides motivation to the work, and discusses previous works found in the literature related to the topic. Chapter 2 describes the *Foliation model*. Kinematics of the *Foliation model* are discussed because the Foliation model was used to generate data train the machine learning model. Chapter 3 presents the framework of the Deformation Manifold Learning model as presented in the article. It also elaborates on the proposed dimensionality reduction technique. Results, conclusions, and future scope of the work are presented in Chapter 4.

Chapter 1

Introduction and Motivation

1.1 Two Dimensional Materials and Multi Walled Carbon Nanotubes

Andre Geim and Konstantin Novoselov did the first successful isolation of graphene in 2004 [53] and subsequently, superiority of graphene and other two dimensional materials over bulk materials has been proven by eminent researchers. Since then rapid progress has been made to utilize the superior mechanical, thermal, chemical, and electrical properties of Graphene. Along with these properties, the strength and unique geometry of graphene structures allow them to undergo large deformations in a reversible manner [13]. Graphene sheet when rolled into a tube form Carbon Nanotube (CNT) or more specifically Single Wall Carbon Nanotube (SWCNT). Multiple concentric tubes form Multi Walled Carbon Nanotube (MWCNT), which is a two-dimensional arrangement of carbon atoms forming a strong and stable honeycomb lattice, which interacts with the neighboring graphene sheets and substrates through weak vanderWaals interactions. Like graphene CNTs possess very high strength and high stiffness. Young's Modulus of CNTs is found out to be in the order of TPa. In 1996 M.M.J.Treacy et al reported an average value of 1.8 TPa for Young modulus of SWCNTs [70]. Similarly, Youngs Modulus for MWCNTs is also in the order of TPa [16, 44]. Min-Feng Yu et al showed in 2000 that the strength of MWCNTs lies in the range of 11 to 63 GPa [78]. Exceptionally high axial strength and stiffness make CNTs an ideal candidate as nanoscale reinforcement fibers in composites. Apart from mechanical properties CNTs also possess good electrical properties which can be exploited in NanoElectroMechanical Systems, Electrodes, Super capacitors, etc [66]. Deformation of CNTs has a significant effect on the mechanical and electrical properties of CNTs [15, 56, 72]. This makes it important for us to understand the deformation characteristics of CNTs to make better use of CNTs and understand the physics of CNTs. CNTs and MWCNTs have a complex deformation morphology with a nonlinear elastic response to loading in the post-buckling regime [2, 40]. Various models have been developed to simulate CNTs and understand their deformation characteristics which are discussed in section 1.2.

1.2 Atomistic-Continuum Models

Philippe Poncharal et al gave high-resolution Transmission electron microscopy (TEM) image of a bent nanotube, showing the characteristic wavelike distortion of CNTs [59]. Researchers have been working on developing computational models to simulate deformed Carbon Nanotubes and compare them against the experimental images. Two major types of models developed for CNTs are based on i) Atomistic-physics of carbon-carbon bonds and ii) Continuum theory. Atomistic models are based on the atomistic physics of carbon-carbon bonds and they are simulated using molecular dynamics simulations [18]. Molecular dynamics simulations are accurate and reliable, but they are prohibited for large systems and are computationally expensive. Continuum theory based models are built upon thin shell theory and assume CNTs to be thin shells [55]. Continuum shell models have proven to be nearly accurate but they have prompted long debates about the value of the thickness of the shell. The thin shell model also neglects the atomistic physics associated with multiple layers of graphene. To address these issues several atomistic-continuum models are proposed in the literature [3, 57, 69]. Atomistic-continuum models take into consideration the interatomic potentials and are based on extensions of Cauchy-born rule [3, 5, 26, 57]. Atomistic-continuum models of multi-layered graphene and MWCNTs based on empirical potentials take a lot of computation power to evaluate van der Waals energy and forces. In all these models interwall interactions are modeled using atom-atom interaction potentials like Lennard-jones potential. In all these approaches every layer of graphene or every wall of MWCNT are to be modeled separately.

To avoid the need for individual treatment of each wall and save on the computation time required bulk elastic models for MWCNTs or multilayer graphene sheets are developed and have qualitatively reproduced deformation patterns of MWCNTs observed in experiments, but at the expense of crude approximations [20]. Foliation model developed by Susanta Ghosh et al is a bulk model for multilayer graphene materials that consider the standard interatomic potentials between carbon atoms (intralayer) and the weak van der Waals interaction between carbon atoms (interlayer) is also an atomistic-continuum model which offers unique benefit over other models [21]. Foliation model is discussed in greater detail in Chapter 2. In this work we use the Foliation model to generate data to train our Machine Learning model.

1.3 Data driven approaches

Atomistic-continuum models like surface model [5] and Foliation model [21] provide a computational advantage over their fully atomistic counterparts but they are still computationally expensive and inaccessible for smaller computing systems. For example, a system handled by surface model which contains 31 million atoms required 10 million gauss points and was executed on 512 processors in parallel. Foliation model provides about 10 fold computational advantage over the Surface model but still needs enormous computation power. To simulate large systems like multilayered graphene or MWCNTs these models are executed using High-Performance Computing (HPC) systems. To alleviate the need of computation power we seek the possibility of data-driven models which learn the patterns in available data and predict for unknown inputs. In general terms data-driven models provide a map between inputs and outputs of the available data. Data-driven models are developed using a variety of Machine Learning models according to the type of data available at hand.

Machine Learning methods have gained popularity amongst researchers in the past few years. Deep Neural Networks (DNNs) a class of Machine Learning methods have been widely used across a variety of fields because of their property to act as a universal approximator. Deep Learning [41] has shown ground breaking success in the field of medical sciences [6, 12, 37, 48, 67, 81], material sciences [10, 19, 79], object and image recognition [11, 31, 39, 58, 80], natural language processing [14, 17] and many other fields. Deep Learning is also being used by researchers working on two-dimensional materials [47, 49, 61, 75].

In this work, we present Deformation Manifold Learning (DML) model which can predict the deformation of MWCNTs under torsional and bending loads [75]. The model uses Deep Neural Networks for learning purposes. Additional details on the DNN used in this work, input and output parameters of the DNN, and components of DML model are given in further chapters.

1.4 Dimensionality Reduction

Deep Neural Networks (DNNs) [32, 41] are intensely investigated for accelerating mechanics, physics, and materials research [33, 46, 73], however, so far most of the applications are limited to the prediction of low–dimensional properties, such as material moduli. On the contrary, discretized material deformation requires prediction in a high–dimensional space. Though the continuum deformation is in the ambient space, the discretized data lies in a very high–dimensional space. For instance, large thick MWCNTs require several millions of degrees of freedom to describe their deformation [1, 21].

Deep Learning models can predict low dimensional (e.g. CNN, Autoencoder[27, 28]) or high dimensional outputs (e.g. Encoder-Decoder [6, 24]). However, these Deep Learning models require high dimensional inputs. State–of–the–art DNNs cannot accurately predict high–dimensional targets from a few input features. The objective of the present study is to create an ML model to accurately and efficiently predict high–dimensional discretized deformations of MWCNTs as output from low–dimensional inputs. This necessitates the reduction of the dimension of the output.

An additional challenge for the MWCNTs is that the deformed configuration is a non–linear manifold. Thus the reduced–dimension (called *latent space*) of the deformation is non–linear. Commonly used dimensionality reduction techniques [71] such as Principal Component Analysis (PCA) and classical Metric Multidimensional Scaling are inapplicable for the present problem since they are based on linear models. Nonlinear Dimensionality Reduction techniques (also called Manifold Learning) such as Isomap, Locally–Linear Embedding, and Umap are designed to identify the low–dimensional non–linear manifold structure of the data [43, 50]. In these techniques, an approximate low–dimensional neighborhood graph embedded in the high–dimension is obtained following the manifold structure of the data. However, to accurately predict the deformation of MWCNTs we need an accurate, smooth, and functional representation of the mapping from the high–dimensional to a low–dimensional manifold such that it respects the constraints of the deformation. Just visualization or approximate discrete representations of the low–dimensional manifold are not sufficient for the present purpose. Functional data analysis (FDA) is a branch of statistics that analyzes functional data providing information about curves or surfaces varying over a continuum. Functional Principal Component Analysis (FPCA) [62, 77] provides a smooth functional representation of the data, which is analogous to Kosambi–Karhunen–Loève Expansion [36, 68]. FPCA represents a stochastic process through a linear combination of an infinite number of orthogonal functions. These orthogonal functions are the functional counterparts of principal components in the standard

Principal Component Analysis. However, we found that the FPCA cannot respect any geometric constraint of the system since these orthogonal functions need not satisfy any constraint. MWCNTs has periodicity constraint along the circumference due to its cylindrical structure. As a consequence FPCA yields discontinuous and erroneous predictions for MWCNTs, which is demonstrated in the present work (see Sec. 3.3).

In the present work, we propose to extend FPCA by designing a basis set of functions to satisfy this constraint exactly. We call the proposed technique constrained–FPCA (c—FPCA). The proposed c–FPCA technique alleviates the *curse of dimensionality* by providing low–dimensional functional representations for the deformations of MWCNTs.

Chapter 2

Foliation Model and Data Generation

In this chapter, we discuss Foliation model and the way we obtain training data using the Foliation model. Foliation model is a three-dimensional continuum model for layered crystalline materials. In [21] Susanta Ghosh et al validate the deformation of MWCNTs obtained using Foliation model against the deformation obtained by Surface model [3, 5] and experiments [59]. The central idea of the Foliation model is that the layered crystalline material is viewed as Foliation. Thus the location of the individual graphene layer is no longer tracked, instead it is considered as a continuous stack of leaves. The constitutive model for this bulk is derived from the atomic interactions. The interaction energy between the walls results from van der Waals forces and the elastic energy along the leaves of Foliation is a result of bonded interactions between the atoms. The unique advantage Foliation model offers over other conventional anisotropic models is that it can be readily discretized using finite elements and we can coarse grain the system in all directions. This gives Foliation model a computational edge over other models. Data driven model requires a tremendous amount of data for training purposes. More the merrier. Therefore, Foliation model proved beneficial for us to generate the training data given its computational superiority.

2.1 Kinematics of the Foliation Model

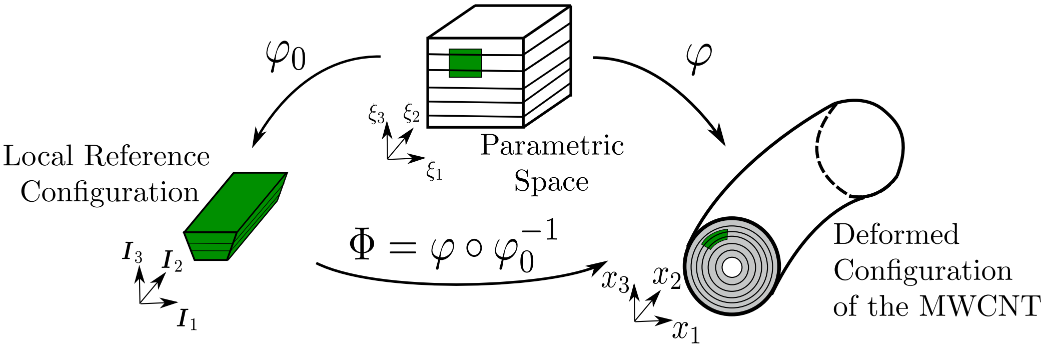
φ_0
φ
ξ_3 ξ_2
ξ_1
Parametric Space
Local Reference Configuration
I_3 I_2
I_1
$\Phi = \varphi \circ \varphi_0^{-1}$
x_3 x_2
x_1
Deformed Configuration of the MWCNT

Figure 2.1: Kinematics of Foliation model showing undeformed configuration space, parametric space, deformed configuration and the maps between them. Reprinted from [75].

Let a body in parametric space be denoted by $\bar{V}$. The parametric space is described by coordinates $\{\xi_1, \xi_2, \xi_3\}$. The deformed configuration be in the Euclidean space $\mathbb{R}^3$ described by x_1, x_2, x_3. And let the local reference configuration be in the space described by I_1, I_2, I_3. A map from parametric space to euclidean space is given by

φ, another map φ_0, maps parametric space to the local reference configuration. We define deformation map using φ and φ_0 as $\Phi = \varphi \circ \varphi_0^{-1}$. Figure 2.1 taken from [75] shows φ_0, φ and Φ.

The deformation gradient can be obtained using deformation map ϕ as

$$\boldsymbol{F} = D_\xi \varphi D_\xi \varphi_0^{-1} \tag{2.1}$$

We are simulating $(5,5),(10,10),\cdots,(5\,N_w,5\,N_w)$ MWCNTs walls, with $N_w = 10,15,20,25,30,35,40$ where N_w is the number of walls. These MWCNTs are obtained by bending the arm-chair configuration of graphene. The MWCNTs are defined by $(5m,5m)$. Where m represents the m-th wall in a MWCNT. For example in a 20-walled MWCNT, we will have (5,5), (10,10) , (15,15)....(100,100) CNTs. In the ideal configuration of MWCNT the gap between the tubes is constant and is kept as equilibrium distance for van der Waals interaction. The graphene sheet which is rolled to form these MWCNTs is neither sheared not stretched. Considering this the reference configuration can be expressed as

$$\boldsymbol{T}_0(\xi) = \begin{bmatrix} 2\pi[R_{in}\xi^3 + R_{out}(1-\xi^3)] & 0 & 0 \\ 0 & L & 0 \\ 0 & 0 & R_{out} - R_{in} \end{bmatrix} \tag{2.2}$$

where,

R_{in}: Inner radius

R_{out}: Outer radius

L: Length of the MWCNT

Considering homogenized stretch and shear-free deformation for reference configuration, the deformation gradient can now be expressed as

$$\boldsymbol{F} = D_{\xi}\varphi \boldsymbol{T}_0^{-1} \tag{2.3}$$

In the Foliation model the parallel surfaces (parallel distribution of walls of MWCNTs) are refered to as *leaf* of foliation. The Cauchy-Green strain tensor of the leaves is expressed as

$$\boldsymbol{C}^l = (\boldsymbol{T}_0^l)^{-T} \boldsymbol{g} (\boldsymbol{T}_0^l)^{-1} \tag{2.4}$$

where, $\boldsymbol{g}$ is the metric tensor of each leaf at each point and is given by

$$\boldsymbol{g}_{\alpha} = \frac{\partial \varphi}{\partial \xi^{\alpha}} \tag{2.5}$$

The curvature tensor is obtained by pullback of the Weingertan map and is computed as,

$$\mathscr{K} = (\boldsymbol{T}_0^l)^{-T} \boldsymbol{k} (\boldsymbol{T}_0^l)^{-1} \tag{2.6}$$

where, $\boldsymbol{k}$ is the curvature of each leaf, and is given by

$$\boldsymbol{k}_{\alpha\beta} =< \boldsymbol{n}, \boldsymbol{g}_{\beta,\alpha} > \tag{2.7}$$

where, $\boldsymbol{n}$ is the unit normal vector.

Considering the exponential Cauchy-Born rule given in [5], the energy density corresponding to the bonded interactions is given by $W(\boldsymbol{C}^l, \mathscr{K})$. The interatomic energy (bonded) is computed using the Brenner potential, and the interlayer van der Waals interaction (non-bonded) is a function of the distance (t) between the walls. Non bonded potential is obtained by taking the sum of pair–wise 6–12 Lennard–Jones (LJ) potential over all the atoms pairs in the system, which is given by,

$$V_{vdW} = -\frac{|V_g(t_0)|}{0.6}\left[\left(\frac{t_0}{t}\right)^4 - 0.4\left(\frac{t_0}{t}\right)^{10}\right] \tag{2.8}$$

Where, t_0 is the equilibrium spacing and $V_g(t_0)$ is the well–depth. The total volumetric effective local van der Waals interaction for a MWCNT according to the FOliation model is given by

$$\mathscr{V}(\lambda^t, \boldsymbol{C}^l) = \frac{N_w - 1}{N_w}\frac{1}{t_0 J^l} V_{vdw}(\lambda^t t_0) \tag{2.9}$$

where,

λ^t: Transversal stretch representing the compaction of the walls.

J^l: Leaf Jacobian determinant.

N_w: Number of walls in the MWCNT.

Details of the above equations are provided by Susanta Ghosh et al in [21]. The total internal energy of the MWCNT is expressed as

$$\Pi_{int}(\varphi) = \int_{\bar{V}} (W(\boldsymbol{C}^l, \mathscr{K}) + \mathscr{V}(\lambda^t, \boldsymbol{C}^l))dV_0 \tag{2.10}$$

The total energy for the system considering the external applied force can be written as

$$\Pi_{total}(\varphi) = \Pi_{int}(\varphi) - \Pi_{ext}(\varphi) \tag{2.11}$$

The above total energy is minimized with respect to the deformation map to obtained deformed shape of MWCNT under given boundary conditions.

2.2 Data generation of MWCNT using Foliation Model

To learn the deformation pattern of the entire MWCNT, multiple sets of simulations would be required. Also, it is very difficult to predict the entire MWCNT system which has millions of degrees of freedom. To reduce the number of simulations we decomposed the domain into several cross–sections (N_{cs}) at regular intervals along its length. Due to the periodicity of the rippling deformation, this decomposition strategy increases the size of the data set. Let's assume that the ξ_3 co–ordinate of the m–th tube in the parametric domain is given by ξ_3^m and each tube is discretized along the ξ_2 direction as $\{\xi_2^n\}_{n=1}^{N_{cs}}$. Where N_{cs} denotes the number of cross–sections taken along the tube length. Thus, discretizing the deformed configuration (Φ) along ξ_1 (circumference) and ξ_2 (length) yields a collection of deformed cross–sections, which are parametric curves of ξ_1, as $\Phi \rightarrow \{\Phi_1(\xi_1, \xi_2^n, \xi_3^m), \Phi_2(\xi_1, \xi_2^n, \xi_3^m), \Phi_3(\xi_1, \xi_2^n, \xi_3^m)\}$, $m = 1, \cdots, N_w; n = 1, \cdots, N_{cs}$. These curves can be further reparametrized in the cylindrical coordinate as $\{\theta(\xi_1), \Phi_2(\theta(\xi_1), \xi_2^n, \xi_3^m), r(\theta(\xi_1), \xi_2^n, \xi_3^m)\}$. Following the decomposition technique, the total deformation of each point (ξ_1, ξ_2, ξ_3) for m–th wall at n–th cross–section of a MWCNT can be represented through two parts, (i) an in–plane radial deformation $r(\theta(\xi_1), \xi_2^n, \xi_3^m)$ in the undeformed cross–sectional plane, and (ii) axial deformation $\Phi_2((\theta(\xi_1), \xi_2^n, \xi_3^m))$.

Chapter 3

Deformation Manifold Learning Model

The proposed semi–supervised ML model includes two steps (i) unsupervised dimensionality reduction (via proposed c–FPCA) of the deformed manifold and (ii) supervised learning (via DNN) of deformation in the reduced dimension. Henceforth, the proposed ML model is referred to as the *Deformation Manifold Learning* (DML) model, shown in Fig. 3.1. It takes the details of the MWCNT system and its boundary conditions as inputs and predicts its high–dimensional discretized deformation. This chapter describes the components of DML model.

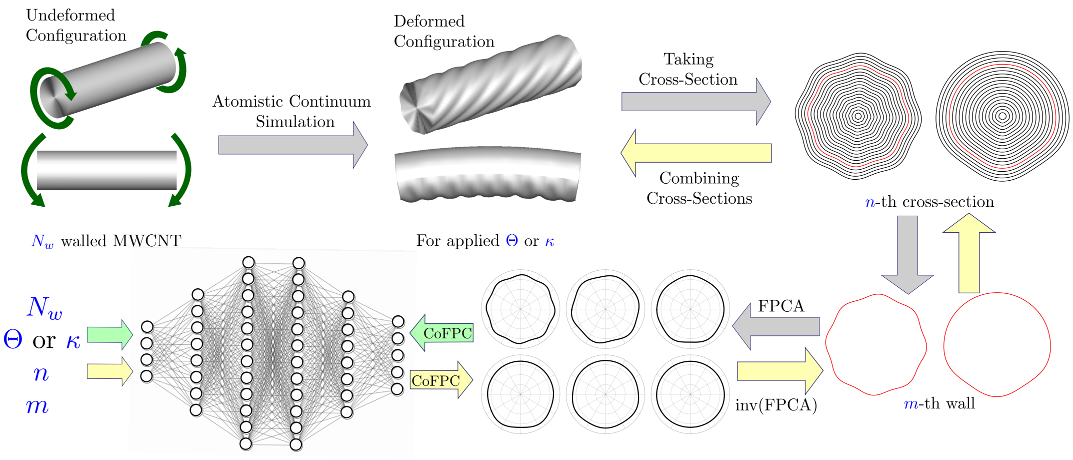
Undeformed Configuration
Atomistic Continuum Simulation
Deformed Configuration
Taking Cross-Section
Combining Cross-Sections
N_w walled MWCNT
For applied Θ or κ
n-th cross-section
N_w
Θ or κ
n
m
CoFPC
CoFPC
FPCA
inv(FPCA)
m-th wall

Figure 3.1: A detailed schematics of the present framework involving the data generation via simulation and the proposed Deformation Manifold Learning(DML) model. The DML model includes: the dimensionality reduction and the DNN–based learning. Inputs and outputs of DNN are shown with green arrows. Yellow arrows represent the prediction via the DML model to obtain the deformation for a given input. Reprinted from [74].

3.1 Proposed Dimensionality Reduction Technique

3.1.1 FPCA

The cross–sections of the MWCNTs are given by the mapped (ξ_1, ξ_3) planes for different ξ_2 in the deformed configuration, which constitutes the data set $\{r_i(\theta)\}_{i=1}^{N}$ of length N. Let us assume the radial deformations of each tube are sampled from a stochastic process $R(\theta)$, $\theta \in \mathcal{T} = (0, 2\pi)$, such that its second derivative is square–integrable. This smoothness of the deformation map is a necessary condition since the energy of a MWCNT is a function of curvature of its walls. We suppose that $R(\theta)$ can take any of the values $r_i(\theta) \in \mathscr{H}^2(\mathcal{T})$, $i = 1, \cdots, N$. Where $\mathscr{H}^2(\mathcal{T})$ is Hilbert space. We denote the $L^2(\mathcal{T})$ inner product of functions ϕ_i , $\phi_j \in \mathscr{H}^2(\mathcal{T})$ with $\langle \phi_i, \phi_j \rangle : \int_{\mathcal{T}} \phi_i(\theta)\, \phi_j(\theta)\, d\theta$.

Let the mean and the covariance functions of $R(\theta)$ are denoted by $\mu(\theta)$ and $v(\theta, \vartheta) = \mathrm{Cov}(R(\theta), R(\vartheta))$. Invoking the Kosambi–Karhunen–Loève Expansion theorem [36, 68], the centered process can be expressed as

$$R(\theta) - \mu(\theta) = \sum_{k=1}^{\infty} \bar{c}_k\, \psi_k(\theta) \tag{3.1}$$

Here, $\bar{c}_k = \langle (R(\theta) - \mu(\theta)), \psi(\theta) \rangle$. Where $\psi_k(\theta)$, $k = 1, 2, \cdots$, are the orthonormal eigenfunctions of the following eigenvalue problem $\int_{\mathcal{T}} v(\theta, \vartheta)\, \psi(\theta) d\vartheta = \lambda\, \psi(\theta)$.

These eigenfunctions, $\psi_k(\theta)$, are henceforth referred to as functional principal components (functional–PCs). Assuming a finite set of eigenfunctions is sufficient to approximate the centered stochastic process, $R(\theta) - \mu(\theta)$, its i–th sample can be written as

$$r_i(\theta) - \mu(\theta) \approx \sum_{k=1}^{K} \bar{c}_{ik}\, \psi_k(\theta),\ \ i = 1, \cdots, N \tag{3.2}$$

Interpretation of eigenfunctions: The first eigenfunction ψ_1 represents the principal mode of variation of the data set. The k–th eigenfunction ψ_k is the k–th most dominant mode of variation orthogonal to $\{\psi_i\}_{i=1}^{k-1}$. To solve the aforementioned eigenvalue problem in $\mathscr{H}^2(\mathcal{T})$ we choose a convenient finite–dimensional basis and look for solutions in terms of that predefined basis. However, choosing any arbitrary basis for FPCA will not work, since the deformed configurations of MWCNTs have geometric constraints that need to be satisfied by the eigenfunctions and hence also needs to be satisfied by the basis. Erroneous predictions via FPCA is demonstrated in Sec. 3.3. To solve the eigenvalue problem while satisfying a constraint can be a difficult task, in the following we reformulate the FPCA in a function space whose every element satisfies the constraint exactly.

3.1.2 Proposed Constrained–FPCA

In order to alleviate the above mentioned problem, a basis $\mathcal{B} = \{\phi_k \in \mathscr{H}^2(\mathcal{T}),\, g(\phi_k) = 0,\, k = 1, \cdots, K\}$ is chosen. This basis $\mathcal{B}$, encodes the geometric constraint (periodic constraint) of the deformation of MWCNTs via the function $g(\phi_k) = 0$, which is crucially important and specializes the FPCA for the systems with any geometric constraint. We call this novel technique constrained–FPCA (c–FPCA). We rewrite the data set, $\{r_i(\theta)\}_{i=1}^{N}$, the eigenfunction $\psi(\vartheta)$, and the covariance function $v(\vartheta, \theta)$ in terms of the basis $\mathcal{B}$ and solve the aforementioned eigenvalue problem to obtain the functional–PCs, $\psi_k(\vartheta)$. Subsequently, the function $r_i(\theta)$ is represented in terms of functional–PCs using the Eq. 3.2 and their corresponding coefficients $(\bar{c}_{ik})$ are referred here as coefficients of functional–PCs (CoFPCs). The i–th sample can be written in terms of the basis, $\mathcal{B}$, as

$$r_i(\theta) - \mu(\theta) = \sum_{k=1}^{K} c_{ik}\, \phi_k(\theta),\ \ i = 1, \cdots, N, c_{ik} \in \mathbb{R} \tag{3.3}$$

In this work, we have chosen the Fourier Basis for ϕ_k. The eigenfunction's representation in the basis $\mathcal{B}$ as

$$\psi(\theta) = \sum_{k=1}^{K} b_k \phi_k(\theta),\ \ b_k \in \mathbb{R} \tag{3.4a}$$

$$\boldsymbol{\psi}(\theta) = \boldsymbol{\phi}(\theta)^t\, \mathbf{b},\ \ \mathbf{b} \in \mathbb{R}^K \tag{3.4b}$$

The covariance function can be written in the basis $\mathcal{B}$ as

$$v(\vartheta, \theta) = \frac{1}{N} \boldsymbol{\phi}(\vartheta)^t\, \mathbf{C}^t\, \mathbf{C}\, \boldsymbol{\phi}(\theta) \tag{3.5}$$

the principal component weight functions $\psi_k(s)$ given as

$$\psi_k = \underset{\substack{\|\psi\|=1, \langle \psi, \psi_j \rangle = 0 \\ \text{for } j=1,\cdots,k-1}}{\arg\max}\ \mathrm{Var}\left(\int_{\mathcal{T}} (\mathrm{R}(\theta) - \mu(\theta)) \psi(\theta) \mathrm{d}\theta \right) \tag{3.6}$$

should satisfy the eigenvalue problem. Hence, the eigenvalue problem can be rewritten as

$$\begin{aligned}\int_{\mathcal{T}} v(\vartheta,\theta)\psi(\theta)dt &= \frac{1}{N}\int_{\mathcal{T}} \boldsymbol{\phi}(\vartheta)^t\,\boldsymbol{C}^t\,\boldsymbol{C}\boldsymbol{\phi}(\theta)\boldsymbol{\phi}(\theta)^t\boldsymbol{b}\,dt \\ &= \boldsymbol{\phi}(\vartheta)^t N^{-1}\boldsymbol{C}^t\boldsymbol{C}\boldsymbol{W}\boldsymbol{b} = \lambda\boldsymbol{\phi}(\vartheta)^t\,\mathbf{b}\end{aligned} \tag{3.7}$$

Where the $K \times K$ symmetric matrix $\mathbf{W}$ such that $W_{i,j} = \langle \phi_i, \phi_j \rangle$. Defining $\mathbf{u} = \mathbf{W}^{1/2}\mathbf{b}$, the above equation can be expressed as a symmetric eigenvalue problem

$$N^{-1}\mathbf{W}^{1/2}\boldsymbol{C}^t\boldsymbol{C}\boldsymbol{W}^{1/2}\mathbf{u} = \lambda\,\mathbf{u} \tag{3.8}$$

Which can be solved for the eigenvector $\mathbf{u}$. The components of each eigenfunction can be found as $\mathbf{b} = \mathbf{W}^{-1/2}\mathbf{u}$. The principal basis can be computed as $\psi(\vartheta) = \boldsymbol{\phi}(\vartheta)^t\,\mathbf{b}$.

The dimension of the problem is significantly reduced by obtaining a K (number of functional–PCs) much smaller than the size of the discretized $r_i(\theta)$.

3.2 Learning in the Reduced Dimension through Deep Neural Networks

Various machine learning algorithms are prescribed in the literature to perform regression and classification [9, 30], but DNNs have the unique capability to act as universal approximators [32] and hence are used to generate highly complex nonlinear maps between inputs and outputs.

We have used Deep Neural Networks (DNNs) to map the MWCNT system parameters to its deformation in the reduced dimension. The DNN architecture takes the Geometry parameters and Boundary conditions as input and outputs CoFPCs. The 4 Inputs for the proposed DNN are: Geometry parameters (i) total number of walls in the MWCNT (N_w), (ii) the wall number ($m, m = 1, \cdots, N_w$), and (iii) the length coordinate ($\Phi_2(\xi_2^n), n = 1, \cdots, N_{\text{cs}}$); (iv) Boundary Conditions: Angle of twist (Θ) or Curvature (κ), per unit length. The dimension of the output layer is the number of CoFPCs, which is decided based on the accuracy required (in c–FPCA), details of which are provided in Chapter 4.

In supervised learning [41] DNNs are trained using some set of known inputs and outputs before we use them to predict for unknown inputs. DNN ($\mathcal{N}$) is a composite function of weights $\bar{\mathbf{w}}$ and biases $\boldsymbol{\beta}$ that maps inputs x_i to y_i. The objective is to update these weights $\mathbf{w}$ and biases $\boldsymbol{\beta}$ to minimize the difference between true output y and predicted output $\hat{y}$, defined by a cost function $\mathcal{J}(y, \hat{y})$:

$$\min_{\bar{\mathbf{w}}} \quad \mathcal{J}(y, \mathcal{N}(x_i, \bar{\mathbf{w}})) \tag{3.9}$$

This is done iteratively by using Stochastic Gradient Descent (SGD) and Backpropagation algorithm [42]. Further, this set of trained weights and biases $\bar{\mathbf{w}} = (\mathbf{w}, \boldsymbol{\beta})$ is used to predict unknown output $y_i = \mathcal{N}(x_i, \bar{\mathbf{w}})$ for given input x_i.

Three DNNs are trained for predicting the following deformations of MWCNTs: (i) In–plane deformation under torsion, (ii) in–plane, and (ii) out–of–plane (axial) deformation under bending. Unlike torsion, in bending the axial deformation is not negligible, hence we have used two DNNs for in–plane and axial deformations.

Now we discuss the details of the DNN used in the proposed model. As mentioned earlier we train 3 different networks for torsion, bending in–plane, and bending out–of–plane. Data for these DNNs was obtained by simulating 7 MWCNTs ($N_w = 10, 15, 20, 25, 30, 35, 40$). Deformed MWCNTs were systematically discretized to obtain the data points used for training.

DNNs are prone to overfitting while training, which limits their capability to predict outputs for data set outside the training set. To overcome this issue multiple regularization strategies and normalization [34] strategies are prescribed. In the present work, we adopt regularization strategies viz. elastic net regularization and early stoppage [60, 82]. Details of these strategies are as follows,

Early Stopping: The number of epochs used for training purposes was set at a maximum limit of 2000, however, early stopping was used with a “patience” of 100

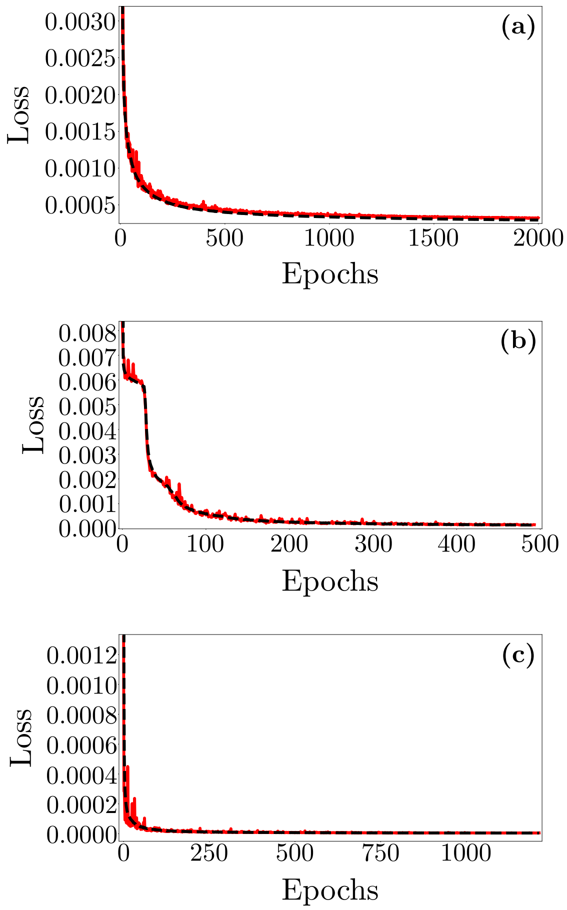

Figure 3.2: Learning Curves for the DNNs. Training loss (- - -) and Validation loss (- - -) are plotted against the number of epochs for DNN used for CoFPCs corresponding to (a) Torsional deformation, (b) in–plane bending deformation, and (c) out–of–plane bending deformation. Reprinted from [75].

epochs to avoid overfitting of the network [60, 65]. Learning curves for the DNNs used are presented in Fig. 3.2. The use of early stopping ensured generalization performance of our model but to further improve performance for test data we used naive elastic net regularization [82].

Elastic Net Regularization: A combination of $\mathcal{L}_1$ and $\mathcal{L}_2$ regularization called elastic net regularization [82] is used, which overcomes the individual drawbacks of $\mathcal{L}_1$ and $\mathcal{L}_2$ regularization. The cost function including $\mathcal{L}_1$ and $\mathcal{L}_2$ regularization can be written as

$$\tilde{\mathcal{J}}(y, \mathcal{N}(x_i, \bar{\mathbf{w}})) = \mathcal{J}(y, \mathcal{N}(x_i, \bar{\mathbf{w}})) + \alpha\Omega(\bar{\mathbf{w}}) \ \alpha \in \mathbb{R} \tag{3.10}$$

Where $\Omega(\bar{\mathbf{w}}) = ||\bar{\mathbf{w}}||_{\mathcal{L}_1}$ and $\Omega(\bar{\mathbf{w}}) = ||\bar{\mathbf{w}}||_{\mathcal{L}_2}$ for $\mathcal{L}_1$ and $\mathcal{L}_2$ regularization respectively. The cost function for elastic net regularization can be written as

$$\tilde{\mathcal{J}}(y, \mathcal{N}(x_i, \bar{\mathbf{w}})) = \mathcal{J}(y, \mathcal{N}(x_i, \bar{\mathbf{w}})) + \alpha_1\Omega(||\bar{\mathbf{w}}||_1) + \alpha_2\Omega(||\bar{\mathbf{w}}||_2), \alpha_1, \alpha_2 \in \mathbb{R} \tag{3.11}$$

We performed a resampling procedure of k–fold cross–validation on the data set to validate the performance of the model on independent fractions of the data set [8].

Feature–normalization techniques improve the performance of the network and reduce the training and inference time. Various methods like Statistical or Z–score Normalization, Min–Max normalization, Median Normalization, Sigmoid normalization, Statistical Column Normalization are prescribed in literature for performing Normalization [34]. In the present work, we use Z–score normalization on the number of walls (N_w), the number of cross–sections (N_{cs}), and twisting angle (Θ). Min–max normalization is used for the wall number. Normalization methods are judiciously chosen and they have improved the correlation.

Hyperparameter optimization is necessary along with the above mentioned techniques. Hyperparameter tuning helps to achieve high accuracy in DNNs. Grégoire Montavon, Geneviève B. Orr and Klaus-Robert Müller have gathered neural network tricks from the world's most prominent neural networks researchers [51] which have helped researchers to improve the performance of their neural networks, we have implemented a few of the relevant tricks for regularization and hyperparameter tuning in our model. DNNs used in the DML model consisted of approximately 40k learning parameters, 6 hidden layers of varying sizes. Weights were initialized using glorot (xavier) initialization. The input layer has 4 nodes, Hidden layers have 32, 64, 128, 128, 64, 64 nodes, Output layer has 16 or 6 nodes depending on the deformation (torsion/bending). From the discretized deformed MWCNTs, we had 0.44 million datapoints for torsion and 0.32 million datapoints for bending. Out of the available data, 80% is used for training, 20% is used for testing. Out of 80% of the training data, we use 20% data as a validation set during training. Adam optimizer was used to minimize the cost function on the *latent space* targets [38]. Learning rate was set to 0.001 as recommended in [38]. The Rectified Linear Unit (ReLU) is used as an activation function in our network which is a better choice of activation function to avoid problems of vanishing gradient or exploding gradient as compared to other activation functions like Sigmoid [22, 23].

DNNs used in DML model are built and trained in Tensorflow 1.1.14 on Python 3.6. We used the Bridges facility available at Pittsburgh Supercomputing Center for training purposes. Bridges' GPU-AI partition consists of NVIDIA Tesla V100 GPU, which was used for training purposes. The time required per iteration was $\approx 12s$ and $\approx 8s$ for DNNs used for MWCNTs under torsion and bending respectively.

MWCNT simulations through Foliation model were done at XSEDE-Bridges supercomputer which is equipped with Intel Haswell (E5-2695 v3) CPUs with 14 cores per CPU running at 2.3-3.3 GHz with 128GB RAM and 35MB cache. Postprocessing of the data was performed on the same system. We used an in house desktop system: 2 processors, 18-core Intel Xeon Gold 5220 CPU running at 2.20 GHz with 64 GB RAM, 24.65MB cache and 512 GB SATA SSD for i) Discretization of the data and other data preparation tasks ii) Dimensionality reduction iii) Inference from the Deep Neural Network iv) Plotting the high dimensional deformation data. Discretization of the data and dimensionality reduction were performed in MATLAB. Parallelization was achieved by using MATLAB's Parallel Computing Toolbox.

3.3 Limitations of FPCA

In this section, we show that the failure of FPCA to satisfy the constraint exactly leads to inaccurate dimension reduction and hence inaccurate prediction of the deformation via the DML model. To show this limitation of FPCA two approaches are taken here:

(i) FPCA coupled with DNNs

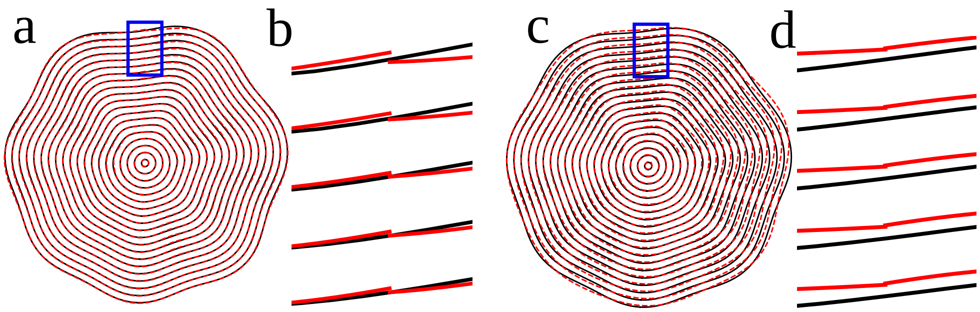

Figure 3.3: Cross–sections of MWCNT obtained using two approaches of DML model (- - -) that uses FPCA: (a) FPCA coupled with DNN and (c) FPCA coupled with the constrained–DNN. These two approaches are compared against the AC model (—). (b) and (d) are the close–up views of the blue boxes corresponding to (a) and (c). (b) Shows a discontinuity in the DML model and (d) shows increases error in prediction everywhere. Reprinted from [75].

(ii) FPCA coupled with constrained–DNNs.

These approaches were used to predict the deformation of twisted MWCNTs. In the first approach, we use DNN on the reduced dimension, which is obtained via FPCA. We found that it yields discontinuity in the deformation due to the violation of constraints by the FPCA as shown in Fig. 3.3a,b. To overcome the discontinuity, in the second approach we have enforced the constraint through the DNN by modifying its objective function:

$$\tilde{\mathcal{J}}(y, \mathcal{N}(x_i, \bar{\mathbf{w}})) = \mathcal{J}(y, \mathcal{N}(x_i, \bar{\mathbf{w}})) + \lambda \mathcal{E}_P \tag{3.12}$$

where $\mathcal{E}_P$ is the error due to the violation of the constraint. The unknown parameter λ decides the degree of enforcement of the constraint. The penalty parameter, λ, provides a balance between the two errors. While the constraint-DNN could reduce the discontinuity, it significantly compromises accuracy everywhere else as shown in

Fig. 3.3c, d.

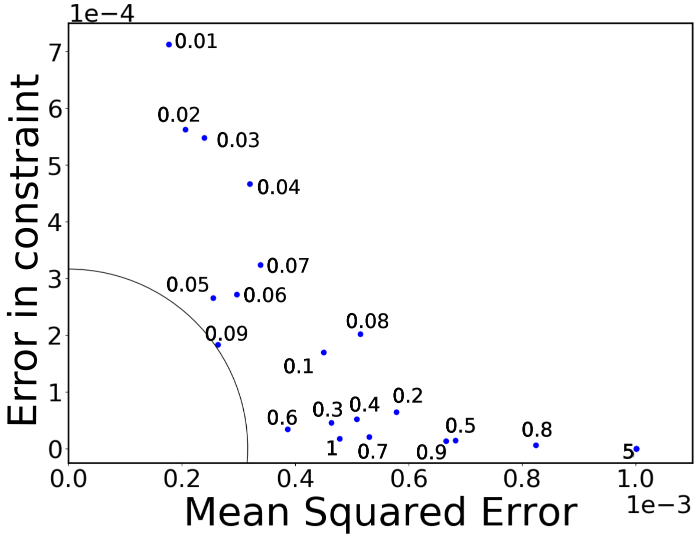

Figure 3.4: Error balancing through L-curve. Each point on the plot corresponds to a λ value. Reprinted from [75].

The unknown parameter λ decides the degree of enforcement of the constraint. For higher values of penalty parameter, the optimization will be dominated by the error in constraint ($\mathcal{E}_P$). For Lower values of penalty parameter, the optimization will be dominated by $\mathcal{J}(y, \mathcal{N}(x_i, \bar{\mathbf{w}}))$. Clearly, there is a need to find out the optimum penalty parameter λ to balance between two errors, which is obtained here by using the L-curve method [29]. A plot of these two errors is used to obtain the optimum λ as shown in Fig. 3.4. Point closest to the origin, gives the optimum penalty parameter. Optimum penalty parameter is used in the modified objective function for training.

The balance between these two errors for various values of λ is shown in Fig. 3.5, which demonstrates the effect of the penalty parameter on error in constraint and mean squared error for different twisting angles. The error in constraint decreases

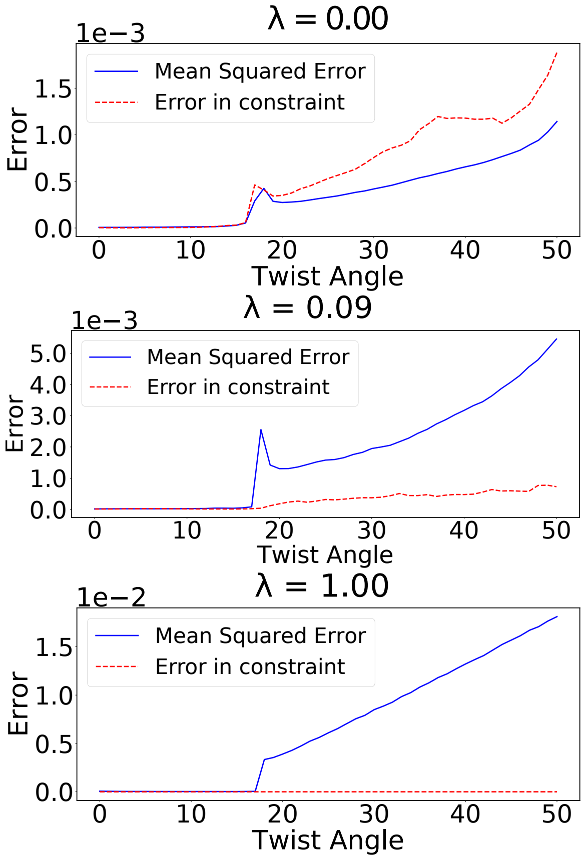

Figure 3.5: Comparison of mean squared error (in the actual deformation) and error in constraint vs twisting angle at different penalty parameters ($\lambda = 0.00$, $\lambda = 0.09$ and $\lambda = 1.00$). Reprinted from [75].

as λ increases. Even for optimal λ, mean square error is quite high (order of 10^{-3}). Note that the constraint can be satisfied very accurately for very large λ. However, as expected, we found that for large λ the Mean Squared Error increases significantly (Fig. 3.5 (bottom)).

To satisfy the constraint exactly we propose an extension to FPCA, termed as constrained-FPCA (c–FPCA). The proposed c–FPCA strategy reduces the error by at least two orders of magnitude compared to the constrained DNN strategy).The c–FPCA is described in Section Chapter 3 Section 3.1.2.

Chapter 4

Results, Discussions and Future Scope

4.1 Dimensionality Reduction through Proposed Constrained FPCA

The proposed c-FPCA dimensionality reduction could capture 99% variability of the deformation data set through only 14 and 4 functional–PCs for torsion and bending respectively, as shown in Fig. 4.1. To capture 99.9% variability, the corresponding numbers are 16 and 6 respectively. The associated (16 and 6) CoFPCs are used as the outputs of DNNs. To obtain the functional–PCs we started with 64 basis functions to represent data vectors of size up to several hundred. This demonstrates up to two orders of magnitude dimensionality reduction via the present approach. Owing to the high accuracy of c–FPCA, DNNs need to learn in significantly reduced dimensions, yielding higher accuracy. Further, c–FPCA returns only a few dominant modes as shown in figure 4.2 and figure 4.3 having a perspicuous pattern, which makes it easier for DNN to learn.

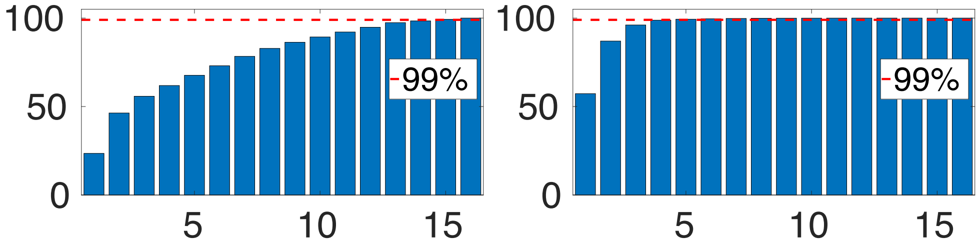

Figure 4.1: Cumulative % variance captured by principal components for MWCNTs under torsion (left) and bending (right). Reprinted from [75].

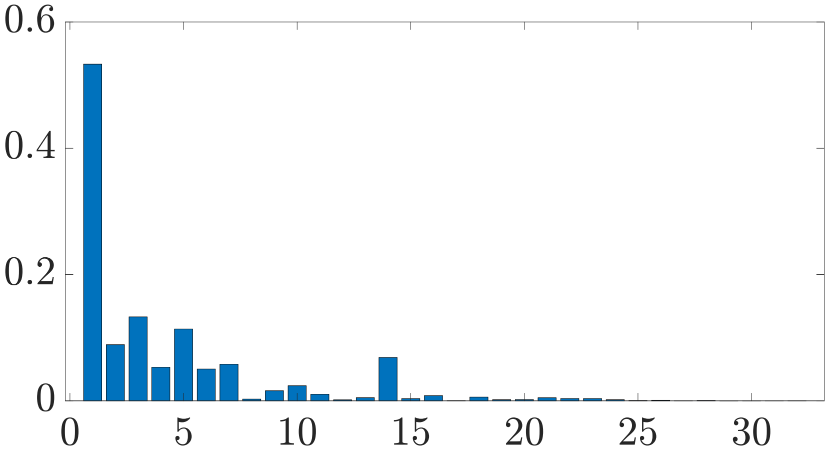

Figure 4.2: CoFPCs for the outermost wall of 40 walled tube under 25° twist at a particular cross–section. Reprinted from [75].

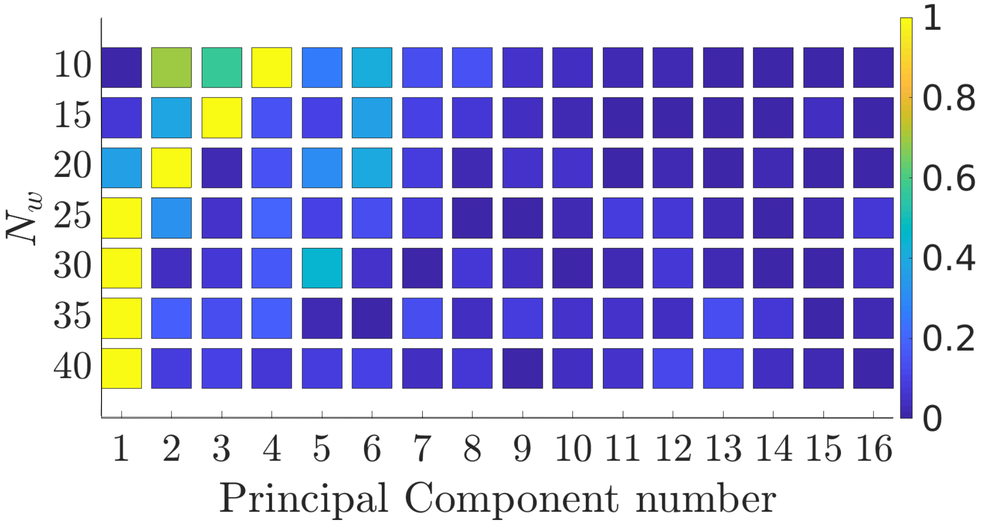

Figure 4.3: Normalized CoFPCs shown as colorbar for $N_w = 10, 15, 20, 25, 30, 35, 40$. CoFPCs are shown for first 16 FPCs. CoFPCs are normalized by the maximum value for each tube. Reprinted from [75].

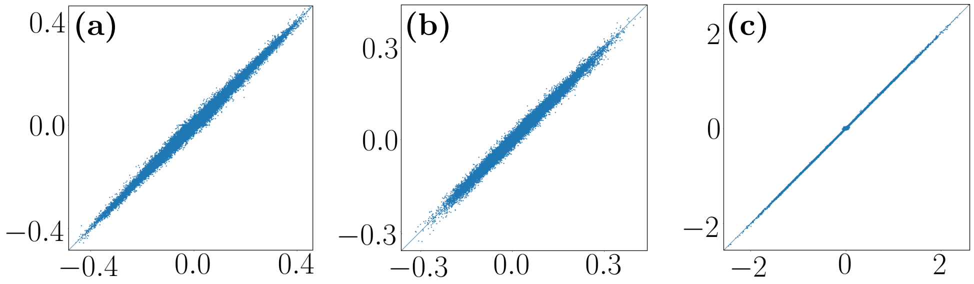

Figure 4.4: Correlation plots for test set CoFPCS of (a) in–plane deformation in torsion, (b) in–plane and (c) out–of–plane deformation in bending. $R = 0.9943$(a), 0.9931(b), 0.9991(c). Reprinted from [75].

4.2 Accuracy

While predicting through the DML model, for a given MWCNT system and loading, at first, the DNN predicts the CoFPCs, which lie in the low–dimensional *latent space*. Subsequently, the high–dimensional deformed cross-sections containing all the walls

(Fig. 4.5 (bottom) and Figs. 4.6 e, f, g) is obtained through inverse c–FPCA.Further, these deformed cross–sections are concatenated through the length coordinate to generate the 3D deformed shape. Since the functional–PCs are non–zero almost everywhere, it is imperative that we predict CoFPCs very accurately. To achieve very high accuracy for DNNs we have adopted the following strategies: (i) regularization techniques, (ii) hyper–parameter tuning, and (iii) features–normalization, (see Sec. 3.2).

The high accuracy of the DNNs is demonstrated through very low relative–mean squared error (order of 10^{-4}) for the validation data and excellent correlations ($R > 0.993$) for the test data as shown in Fig. 4.4.

Predictions by the proposed DML model is compared against the AC model for two types of systems: (i) known systems but unknown loading, (ii) unknown systems and unknown loading, Deformation morphologies under torsion and bending obtained through AC and DML models are provided for the known and unknown systems in Fig. (4.5,4.6) and in Fig. 4.7 respectively. The proposed DML model matches remarkably well with the AC model for unknown loading as evident from the deformed surfaces and cross–sections. Their match is quite accurate even when both the system and the loading are unknown (as long as the unknown system is within the range of the training data). This obviates the need for AC simulations for such systems, yielding huge computational savings. However, if an MWCNT lies way outside the range of the training data its accuracy might go down since it might exhibit a deformation

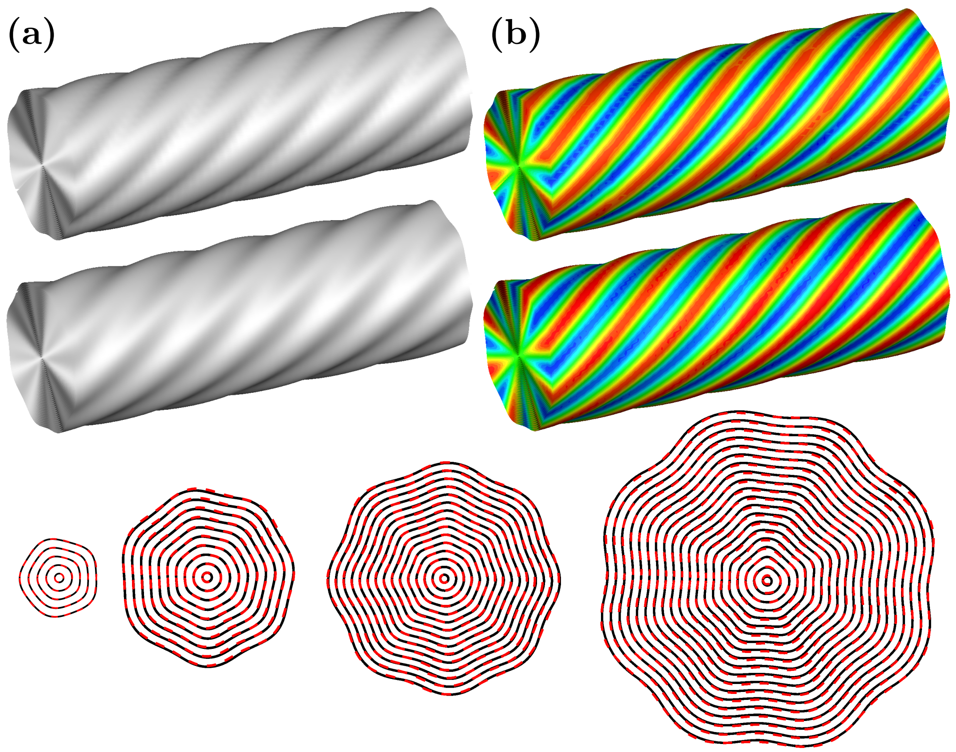

Figure 4.5: (a) Twisted 40 walled CNT obtained via AC (top) and DML (bottom) model. (b) Radial deformation colormap (Red: high, Blue: low). Alternate walls of cross–sections obtained via AC (—) and DML (- - -) models, for 10, 20, 30, and 40 walled CNTs. Reprinted from [75].

pattern that doesn't occur in the training.

To quantify the accuracy, we compute the relative error in the predicted deformed configurations. The maximum relative error is found to be ≈ 1% for the 32–walled CNT, which is an unknown system under unknown loading. To quantify the accuracy, we compute the relative error in the predicted deformed configurations as

$$\frac{1}{N_w}\sum_{m=1}^{N_w}\frac{\langle(\mathbf{\Phi}^{\mathrm{AC}}(\xi_1,\xi_2,\xi_3^m)-\mathbf{\Phi}^{\mathrm{DML}}(\xi_1,\xi_2,\xi_3^m))^2\,\rangle}{\langle\mathbf{\Phi}^{\mathrm{AC}}(\xi_1,\xi_2,\xi_3^m)^{\,2}\,\rangle} \tag{4.1}$$

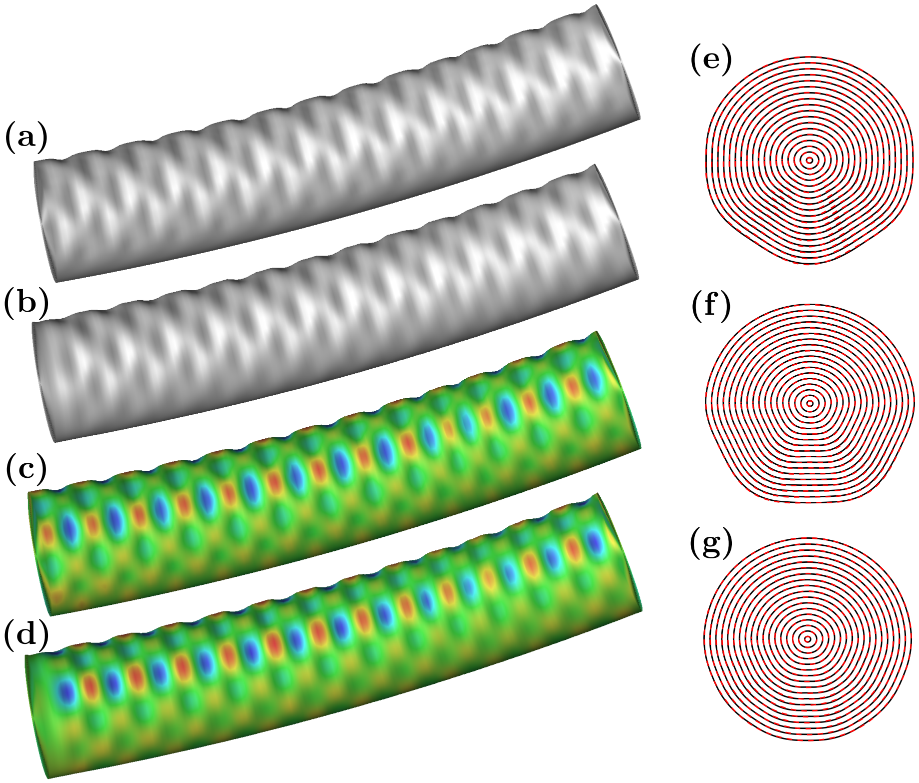

Figure 4.6: Bent 35–walled CNT obtained via AC (a,c) and DML (b,d) model.(c) and (d) show colormap of radial deformations corresponding to (a) and (b). (e–g) Alternate walls of cross–sections obtained via AC (—) and DML (- - -) model.Reprinted from [75].

Where the symbol $\langle \bullet \rangle$ denotes integral over ξ_1, ξ_2 planes: $\langle \bullet \rangle := \int \bullet \, d\xi_1 \, d\xi_2$.

We attribute the high accuracy of the proposed model to the accuracy in both the dimensionality reduction and learning through DNNs. The deformation obtained from CoFPCs (output of the DML model) is used to compute the energy via a discretization. The total energy computed through the DML and the AC model matches very well for both known and unknown systems, as shown in Fig. 4.8.

The proposed model is significantly more efficient than the AC model. The AC model

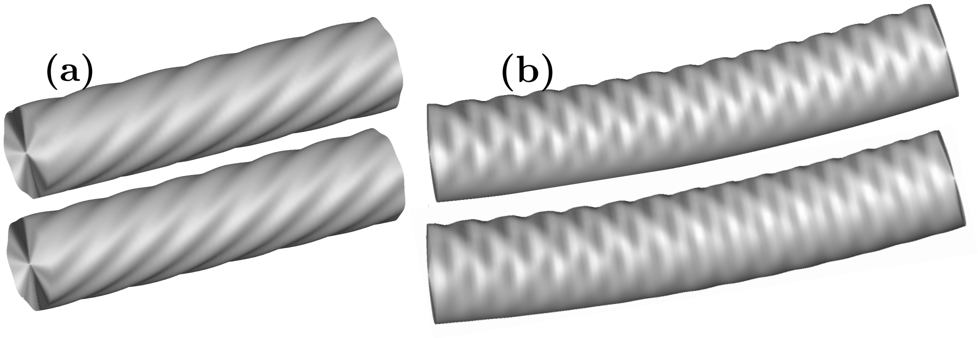

Figure 4.7: Comparison of AC (top) and DML (bottom) models for a 32 walled CNT (system which is not a part of training data) under torsion (a) and bending (b). Reprinted from [75].

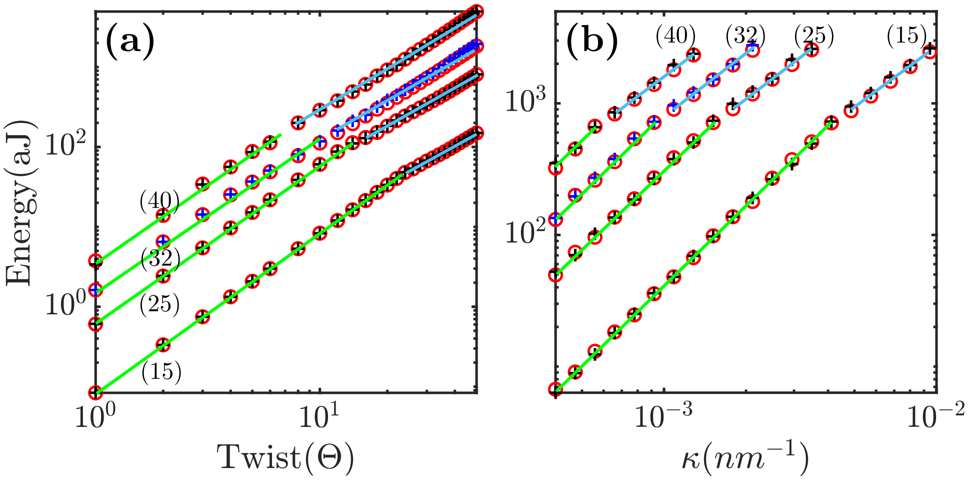

Figure 4.8: Energy comparison for (15, 25, 32, 40 –walled) MWCNTs under torsion (a) and bending (b) via AC (○) and DML model (+). 32 walled CNT (+) is an unknown system. The lines (—) and (—) are drawn to highlight pre– and post–buckling regimes. Reprinted from [75].

requires tens or hundreds of *total CPU* hours in parallel processing to simulate each of the MWCNTs. Whereas, inference via the proposed model (upon training), requires only about ten seconds for an unknown MWCNT.

4.3 Interpretability of the DML model

Despite their applicability and accuracy, ML models are often criticized as "black–box" or non–comprehensible. Recently there is a surge in efforts to produce intelligible knowledge about the problem through ML models, this ability is referred to as interpretability [52]. The proposed model is interpretable since the *latent space* of deformation is comprehensible through the functions spanning it.

Recently several efforts have been made to overcome the "black box" nature of ML models and to make them more comprehensible to humans, through formulating interpretation techniques [52]. Herein, we explore the model–based–interpretability (as defined in [52]) of the proposed model. The proposed model can extract dominant (principal) modes of deformed configurations and their relative contribution in an unsupervised manner. A few principal components of the deformation of MWCNTs under torsion and bending are shown in Fig. 4.3. The rippling deformation of MWCNTs under torsion follows a sequence of ridge and furrows, whereas, in case of bending it resembles the diamond buckling pattern [1]. These key patterns of deformation are captured through the functional principal components (Fig. 4.9). So far these key deformed patterns were approximately–identified manually for individual MWCNTs. The principal components of deformation automatically identified in the present model show qualitative similarity with those identified manually in [4, 5, 83].

These functional–PCs are universal since they are obtained from the entire data set. This fact enhances the model's predictive capability on unseen systems and hence explains the generalizability (performance for unseen systems) of the model. The DNNs learn the reduced dimension spanned by the functional–PCs. The principal modes of deformations are easy to comprehend thus enhance the understanding of how the proposed model works, which makes it an interpretable model.

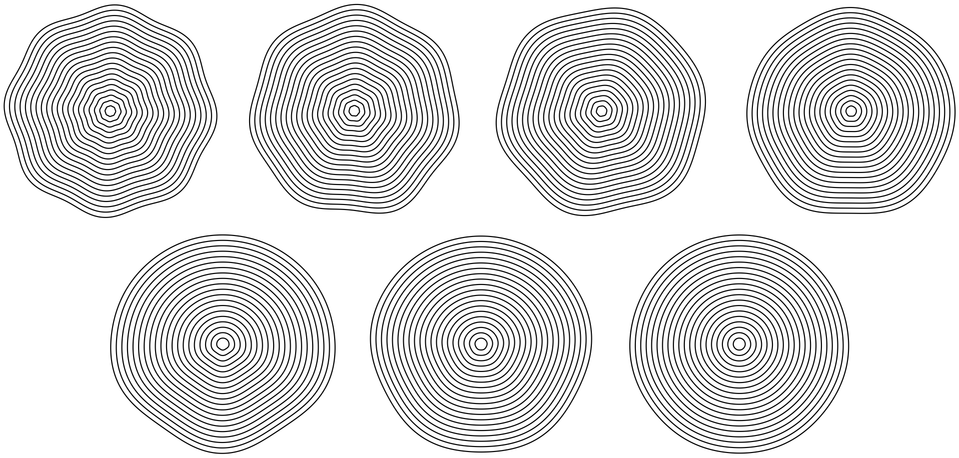

Figure 4.9: Functional Principal components of MWCNTs under torsion (top) and bending (bottom).Reprinted from [75].

4.4 Conclusions

In this study, a novel interpretable machine learning model is proposed, which predicts high–dimensional deformed configurations of MWCNTs accurately and efficiently using only 4 inputs. It combines an unsupervised dimensionality reduction of the deformed configuration space and supervised learning in the reduced space.

To conclude this study, we summarize its main features. *Firstly*, a novel dimensionality reduction technique is proposed that extends FPCA to respect the constraints of deformation exactly. This improves accuracy in low–dimensional representation of deformation and enables accurate prediction of high–dimensional deformation of MWCNTs. *Secondly*, the proposed model is remarkably accurate for unknown systems and unknown loading. This capability eliminates expensive AC simulations for systems beyond what is used in the training, yielding a massive gain in computational efficiency. *Thirdly*, the principal components are comprehensible and thus help to elucidate how the model predicts high–dimensional deformation through learning the space of functional–PCs, leading to model–interpretability.

In future, the proposed model will serve as a basis for the exploration of machine learning for nanotubes and 2D materials. A similar model can be developed for Boron nitride nanotubes and other nanotubes. Proposed constrained-FPCA can be used for dimensionality reduction of manifolds of deformed 2D materials like graphene. This will enable us to develop machine learning models for the deformation of 2D materials.

References

[1] Arias, I. and Arroyo, M. (2008). *Phys. Rev. Lett.*, 100(085503).

[2] Arroyo, M. and Arias, I. (2008). Rippling and a phase-transforming mesoscopic model for multiwalled carbon nanotubes. *Journal of the Mechanics and Physics of Solids*, 56(4):1224–1244.

[3] Arroyo, M. and Belytschko, T. (2002). An atomistic-based finite deformation membrane for single layer crystalline films. *Journal of the Mechanics and Physics of Solids*, 50(9):1941–1977.

[4] Arroyo, M. and Belytschko, T. (2003). *Phys. Rev. Lett.*, 91(21):215505.

[5] Arroyo, M. and Belytschko, T. (2004). Finite element methods for the non-linear mechanics of crystalline sheets and nanotubes. *International Journal for Numerical Methods in Engineering*, 59(3):419–456.

[6] Balu, A., Nallagonda, S., Xu, F., Krishnamurthy, A., Hsu, M.-C., and Sarkar, S.

(2019). A deep learning framework for design and analysis of surgical bioprosthetic heart valves. *Scientific reports*, 9(1):1–12.

[7] Bažant, Z. and Cedolin, L. (1991). *Stability of Structures: Elastic, Inelastic, Fracture and Damage Theories*. Oxford University Press.

[8] Bengio, Y. and Grandvalet, Y. (2004). *J. machine learning research*, 5(Sep):1089–1105.

[9] Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

[10] Cai, J., Chu, X., Xu, K., Li, H., and Wei, J. (2020). Machine learning-driven new material discovery. *Nanoscale Advances*, 2(8):3115–3130.

[11] Cai, Z., Fan, Q., Feris, R. S., and Vasconcelos, N. (2016). A unified multi-scale deep convolutional neural network for fast object detection. In *European conference on computer vision*, pages 354–370. Springer.

[12] Ching, T., Himmelstein, D. S., Beaulieu-Jones, B. K., Kalinin, A. A., Do, B. T., Way, G. P., Ferrero, E., Agapow, P.-M., Zietz, M., Hoffman, M. M., et al. (2018). Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15(141):20170387.

[13] Chopra, N. G., Benedict, L. X., Crespi, V. H., Cohen, M. L., Louie, S. G., and Zettl, A. (1995). Fully collapsed carbon nanotubes. *Nature*, 377(6545):135–138.

[14] Collobert, R. and Weston, J. (2008). In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 160–167. Association for Computing Machinery.

[15] De Volder, M. F. L., Tawfick, S. H., Baughman, R. H., and Hart, A. J. (2013). *Science*, 339(6119):535–539.

[16] Demczyk, B. G., Wang, Y. M., Cumings, J., Hetman, M., Han, W., Zettl, A., and Ritchie, R. (2002). Direct mechanical measurement of the tensile strength and elastic modulus of multiwalled carbon nanotubes. *Materials Science and Engineering: A*, 334(1-2):173–178.

[17] Deng, L. and Liu, Y. (2018). *Deep learning in natural language processing*. Springer.

[18] Dumitrică, T. and James, R. D. (2007). Objective molecular dynamics. *Journal of the Mechanics and Physics of Solids*, 55(10):2206–2236.

[19] Fullwood, D. T., Niezgoda, S. R., and Kalidindi, S. R. (2008). Microstructure reconstructions from 2-point statistics using phase-recovery algorithms. *Acta Materialia*, 56(5):942–948.

[20] Garg, M., Pantano, A., and Boyce, M. (2007). An equivalent orthotropic representation of the nonlinear elastic behavior of multiwalled carbon nanotubes.

[21] Ghosh, S. and Arroyo, M. (2013). An atomistic-based foliation model for multilayer graphene materials and nanotubes. *Journal of the Mechanics and Physics of Solids*, 61(1):235–253.

[22] Glorot, X. and Bengio, Y. (2010). In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.

[23] Glorot, X., Bordes, A., and Bengio, Y. (2011). In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323.

[24] Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. (2018). *ACS central science*, 4(2):268–276.

[25] Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. (2018). *ACM computing surveys*, 51(5):1–42.

[26] Guo, X., Wang, J., and Zhang, H. (2006). Mechanical properties of single-walled carbon nanotubes based on higher order cauchy–born rule. *International Journal of Solids and Structures*, 43(5):1276–1290.

[27] Hanakata, P. Z., Cubuk, E. D., Campbell, D. K., and Park, H. S. (2018). *Phys. Rev. Lett.*, 121(25):255304.

[28] Hanakata, P. Z., Cubuk, E. D., Campbell, D. K., and Park, H. S. (2020). Forward and inverse design of kirigami via supervised autoencoder. *arXiv preprint arXiv:2008.05298*.

[29] Hansen, P. C. (1992). Analysis of discrete ill-posed problems by means of the l-curve. *SIAM Review*, 34(4):561–580.

[30] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer.

[31] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[32] Hornik, K., Stinchcombe, M., and White, H. (1989). *Neural Networks*, 2(5):359–366.

[33] Iten, R., Metger, T., Wilming, H., del Rio, L., and Renner, R. (2020). *Phys. Rev. Lett.*, 124:010508.

[34] Jayalakshmi, T. and Santhakumaran, A. (2011). *Int. J. of Computer Theory and Engineering*, 3(1):1793–8201.

[35] Jones, M. C. and Rice, J. A. (1992). *The American Statistician*, 46(2):140–145.

[36] Jorgensen, P. and Song, M. (2007). *J. of Mathematical Physics*, 48(10):103503.

[37] Khairnar, P., Thiagarajan, P., and Ghosh, S. (2020). A modified bayesian convolutional neural network for breast histopathology image classification and uncertainty quantification. *arXiv preprint arXiv:2010.12575*.

[38] Kingma, D. P. and Ba, J. (2014). *arXiv preprint arXiv:1412.6980*.

[39] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). In *Advances in Neural Information Processing Systems*, volume 2, pages 1097–1105.

[40] Kuzumaki, T., Hayashi, T., Ichinose, H., Miyazawa, K., Ito, K., and Ishida, Y. (1998). In-situ observed deformation of carbon nanotubes. *Philosophical magazine A*, 77(6):1461–1469.

[41] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

[42] LeCun, Y., Touresky, D., Hinton, G., and Sejnowski, T. (1988). A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school*, volume 1, pages 21–28. CMU, Pittsburgh, Pa: Morgan Kaufmann.

[43] Lee, J. A. and Verleysen, M. (2007). *Nonlinear Dimensionality Reduction*. Springer, 1st edition.

[44] Li, C. and Chou, T.-W. (2003). Elastic moduli of multi-walled carbon nanotubes and the effect of van der waals forces. *Composites Science and Technology*, 63(11):1517–1524.

[45] Lipton, Z. C. (2018). *ACM Queue*, 16(3):31–57.

[46] Lu, L., Dao, M., Kumar, P., Ramamurty, U., Karniadakis, G. E., and Suresh, S. (2020). *Proc. Natl. Acad. Sci. U.S.A.*, 117(13):7052–7062.

[47] M. Dieb, T., Hou, Z., and Tsuda, K. (2018). Structure prediction of boron-doped graphene by machine learning. *The Journal of chemical physics*, 148(24):241716.

[48] Mamoshina, P., Vieira, A., Putin, E., and Zhavoronkov, A. (2016). Applications of deep learning in biomedicine. *Molecular pharmaceutics*, 13(5):1445–1454.

[49] Masubuchi, S., Watanabe, E., Seo, Y., Okazaki, S., Sasagawa, T., Watanabe, K., Taniguchi, T., and Machida, T. (2020). Deep-learning-based image segmentation integrated with optical microscopy for automatically searching for two-dimensional materials. *npj 2D Materials and Applications*, 4(1):1–9.

[50] McInnes, L., Healy, J., and Melville, J. (2018). *arXiv preprint arXiv:1802.03426*.

[51] Montavon, G., Orr, G., and Müller, K.-R. (2012). Neural networks-tricks of the trade second edition. *Springer, DOI*, 10:978–3.

[52] Murdoch, J., Singh, C., Kumbier, K., Abbasi-Asl, R., and Yu, B. (2019). *Proc. Natl. Acad. Sci. U.S.A.*, 116(44):22071–22080.

[53] Novoselov, K. S., Jiang, D., Schedin, F., Booth, T., Khotkevich, V., Morozov, S., and Geim, A. K. (2005). Two-dimensional atomic crystals. *Proceedings of the National Academy of Sciences*, 102(30):10451–10453.

[54] Pak, M. and Kim, S. (2017). A review of deep learning in image recognition. In *2017 4th international conference on computer applications and information processing technology (CAIPT)*, pages 1–3. IEEE.

[55] Pantano, A., Parks, D. M., and Boyce, M. C. (2004). Mechanics of deformation of single-and multi-wall carbon nanotubes. *Journal of the Mechanics and Physics of Solids*, 52(4):789–821.

[56] Papadakis, S. J., Hall, A. R., Williams, P. A., Vicci, L., Falvo, M. R., Superfine, R., and Washburn, S. (2004). *Phys. Rev. Lett.*, 93(14):146101.

[57] Park, J. Y., Cho, Y., Kim, S. Y., Jun, S., and Im, S. (2006). A quasicontinuum method for deformations of carbon nanotubes. *Computer Modeling in Engineering and Sciences*, 11(2):61.

[58] Pathak, A. R., Pandey, M., and Rautaray, S. (2018). Application of deep learning for object detection. *Procedia computer science*, 132:1706–1717.

[59] Poncharal, P., Wang, Z., Ugarte, D., and De Heer, W. A. (1999). Electrostatic deflections and electromechanical resonances of carbon nanotubes. *science*, 283(5407):1513–1516.

[60] Prechelt, L. (2012). *Early Stopping — But When?*, pages 53–67. Springer.

[61] Rahman, A., Deshpande, P., Radue, M. S., Odegard, G. M., Gowtham, S.,

Ghosh, S., and Spear, A. D. (2020). A machine learning framework for predicting the shear strength of carbon nanotube-polymer interfaces based on molecular dynamics simulation data. *Composites Science and Technology*, page 108627.

[62] Ramsay, J. and Silverman, B. (2005). *Functional Data Analysis*. Springer.

[63] Ramsay, J. O. and Silverman, B. W. (2007). *Applied functional data analysis: methods and case studies*. Springer.

[64] Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326.

[65] Sarle, W. S. (1996). *Computing sci. and stats.*, pages 352–360.

[66] Schnorr, J. M. and Swager, T. M. (2011). Emerging applications of carbon nanotubes. *Chemistry of Materials*, 23(3):646–657.

[67] Shen, D., Wu, G., and Suk, H.-I. (2017). *Annual review of biomedical engineering*, 19:221–248.

[68] Stark, H. and Woods, J. W., editors (1986). *Probability, Random Processes, and Estimation Theory for Engineers*. Prentice-Hall, Inc., USA.

[69] Sun, Y. and Liew, K. M. (2008). Application of the higher-order cauchy–born rule in mesh-free continuum and multiscale simulation of carbon nanotubes. *International Journal for Numerical Methods in Engineering*, 75(10):1238–1258.

[70] Treacy, M. J., Ebbesen, T. W., and Gibson, J. M. (1996). Exceptionally high young's modulus observed for individual carbon nanotubes. *nature*, 381(6584):678–680.

[71] Van Der Maaten, L., Postma, E., and Van den Herik, J. (2009). Dimensionality reduction: a comparative review. *Journal Machine Learning Research*, 10:66–71.

[72] Williams, P., Papadakis, S., Patel, A., Falvo, M., Washburn, S., and Superfine, R. (2002). *Phys. Rev. Lett.*, 89:255502.

[73] Xie, T. and Grossman, J. C. (2018). *Phys. Rev. Lett.*, 120:145301.

[74] Yadav, U., Pathrudkar, S., and Ghosh, S. (2020). An interpretable machine learning model for deformation of multi-walled carbon nanotubes. *arXiv preprint arXiv:2011.08304*.

[75] Yadav, U., Pathrudkar, S., and Ghosh, S. (2021). Interpretable machine learning model for the deformation of multiwalled carbon nanotubes. *Physical Review B*, 103(3):035407.

[76] Yakobson, B. I., Brabec, C., and Bernholc, J. (1996). Nanomechanics of carbon tubes: instabilities beyond linear response. *Physical review letters*, 76(14):2511.

[77] Yao, F., Müller, H.-G., and Wang, J.-L. (2005). *Annals of Statistics*, 33(6):2873–2903.

[78] Yu, M.-F., Lourie, O., Dyer, M. J., Moloni, K., Kelly, T. F., and Ruoff, R. S. (2000). Strength and breaking mechanism of multiwalled carbon nanotubes under tensile load. *Science*, 287(5453):637–640.

[79] Zhang, X. and Garikipati, K. (2020). Machine learning materials physics: Multi-resolution neural networks learn the free energy and nonlinear elastic response of evolving microstructures. *Computer Methods in Applied Mechanics and Engineering*, 372:113362.

[80] Zhao, Z.-Q., Zheng, P., Xu, S.-t., and Wu, X. (2019). Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232.

[81] Zhou, S. K., Greenspan, H., and Shen, D. (2017). *Deep learning for medical image analysis*. Academic Press.

[82] Zou, H. and Hastie, T. (2005). *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320.

[83] Zou, J., Huang, X., Arroyo, M., and Zhang, S. (2009). *Journal of applied physics*, 105(3):033516.