

Hybrid Computational-Machine Learning Models with Uncertainty Quantification for Aortic Dissection

by

Minglang Yin

B.S., Northwestern Polytechnical University; Xi'an, China P.R., 2016

M.S., Brown University; Providence, RI, 2018

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in Center for Biomedical Engineering, School of Engineering at Brown University



PROVIDENCE, RHODE ISLAND

October 2022

© Copyright 2022 by Minglang Yin

This dissertation by Minglang Yin is accepted in its present form
by Center for Biomedical Engineering, School of Engineering as satisfying the
dissertation requirement for the degree of Doctor of Philosophy.

Date_____

George Em Karniadakis, Ph.D., Advisor

Recommended to the Graduate Council

Date_____

Kareen L. K. Coulombe, Ph.D., Reader

Date_____

Joseph J. Crisco, Ph.D., Reader

Date_____

Yuri Bazilevs, Ph.D., Reader

Approved by the Graduate Council

Date_____

Thomas A. Lewis, Dean of the Graduate School

Vitae

Education

- **Ph.D.**, Biomedical Engineering, Brown University 2018 - 2022
Advisor: George Em Karniadakis
- **M.Sc.**, Fluids and Thermal Sciences, Brown University 2016 - 2018
Advisor: George Em Karniadakis
Thesis: *3D/1D Computed Fractional Flow Reserve Comparison in Coronary Artery Disease*
- **B.S.**, Aeronautical Engineering, Northwestern Polytechnical University, China 2012 - 2016
Advisor: Weiwei Zhang
Thesis: *Reduced-Order Aerodynamic Modeling and Study on Generalization Capability*

Positions

- Jan. 2022 - Aug. 2022, Consultant, Warren Alpert Medical School, Brown University, PI: Roy. K. Aaron
- Jan. 2018 - Aug. 2022, Research Assistant, Division of Applied Mathematics, Brown University, Advisor: George Em. Karniadakis

- Aug. 2017, Lecturer, Pre-college program, Introduction to Mechanical Engineering, Brown University
- Jan. - Jul. 2016, Research Assistant, School of Aeronautics, Northwestern Polytechnical University

Publications

1. **M. Yin**, E. Zhang, Y. Yu, G.E. Karniadakis. Interfacing finite elements with deep neural operators for fast multiscale modeling of mechanics problems. *Computer Methods in Applied Mechanics and Engineering*, (2022): 115027
2. **M. Yin**, E. Ban, E. Zhang, B. Rego, C. Cavinato, J.D. Humphrey, G.E. Karniadakis. Simulating progressive intramural damage leading to aortic dissection using an operator-regression neural network, *Journal of The Royal Society Interface*, (2022): 20210670.
3. S. Goswami, **M. Yin**, Y. Yu, G.E. Karniadakis. A physics-informed variational DeepONet for predicting the crack path in brittle materials, *Computer Methods in Applied Mechanics and Engineering*, 391 (2022): 114587.
4. S. Cai, Z. Mao, Z. Wang, **M. Yin**, G.E. Karniadakis, Physics-informed neural networks in fluid mechanics: A review, *Acta Mechanica Sinica* (2022): 1-12.
5. A. Blumens*, **M. Yin***, Y. Hasegawa, Z. Li, and G.E. Karniadakis. Multiscale parareal algorithm for long-time mesoscopic simulations of microvascular blood flow in zebrafish, *Computational Mechanics*, 68, 1131-1152 (2021).
6. **M. Yin**, X. Zheng, J.D. Humphrey, G.E. Karniadakis. Non-invasive inference of thrombus material properties with physics-informed neural networks,

- Computer Methods in Applied Mechanics and Engineering, 375 (2021): 113603.
7. E. Zhang, **M. Yin**, G.E. Karniadakis. Physics-informed neural networks for nonhomogeneous material identification in elasticity imaging, AAAI Conference (2020).
 8. **M. Yin**, A. Yazdani, and G.E. Karniadakis. One-dimensional modeling of fractional flow reserve in coronary artery disease: uncertainty quantification and bayesian optimization, Computer Methods in Applied Mechanics and Engineering, 353 (2019): 66-85.
 9. D. Hopper, D. Jaganathan, J. Orr, J. Shi, F. Simeski, **M. Yin**, J.T.C. Liu. Heat transfer in nanofluid boundary layer near adiabatic wall, Journal of Nanofluids, 7.6 (2018): 1297-1302.
 10. **M. Yin**, J. Kou, W. Zhang. A reduced-order aerodynamic model with high generalization capability based on neural network, Acta Aerodynamica Sinica, 35.02 (2017): 205-213.
 11. J. Kou, W. Zhang, and **M. Yin**. Novel Wiener models with a time-delayed nonlinear block and their identification, Nonlinear Dynamics, 85.4 (2016): 2389-2404.

Conferences

1. Jun. 2022, USNC/TAM (Austin, TX), Multiscale Modeling with Operator-Learning Neural Networks
2. Jun. 2022, USNC/TAM (Austin, TX), Imaging-Driven Modeling of Dissection Progression in the Aorta

3. Apr. 2022, SIAM UQ (Online), Multiscale modeling with operator-regression neural networks
4. Nov. 2021, IMECE (Online), Predicting injection-caused delamination in aortic walls using DeepONet
5. Oct. 2021, IACM Computational Fluids Conference (Online), Imaging-driven inference of biomaterial properties with physics-informed neural networks
6. Jul. 2021, USNCCM16 (Online), Data-driven modeling of injection-caused delamination on aortic walls using DeepONet
7. Nov. 2020, APS DFD (Online), Non-invasive inference of thrombus material properties with physics-informed neural networks
8. Mar. 2020, Mach Conference (Accepted), Physics-informed neural networks for solving forward and inverse problem with phase field models
9. Nov. 2019, APS DFD (Seattle, WA), Comparison of multi-scale models for blood flow in zebrafish brain, APS Division of Fluid Dynamics
10. Oct. 2019, BMES Annual Meeting (Philadelphia, PA) (Poster), Numerical study on hemodynamics of brain vasculature in Early zebrafish Life
11. Jan. 2019, SIAM CSE (Spokane, WA), Parameter inference and uncertainty quantification in simulating blood flow in coronary arteries

Invited Talks

1. Jan. 2022, Johns Hopkins University, Department of Biomedical Engineering, (Online) Multiscale Modeling and Machine Learning for Biomedicine

2. Aug. 2021, Northwestern Polytechnical University, School of Aeronautics, (Online), Physics-informed machine learning and its application in multi-scale modeling
3. Aug. 2021, Parallel-in-Time (PinT) Workshop (Online), Time parallel in PDEs using machine learning tools
4. Apr. 2021, NVIDIA GTC (Online), Non-invasive inference of thrombus material properties with physics-informed neural networks

Trainings

- Jul. 2019, San Diego Supercomputing Center summer institute on High Performance Computing and Data Science, San Diego, CA
- Oct. 2019, Integrating Machine Learning with Multiscale Modeling for Biomedical, Biological, and Behavioral Systems, Bethesda, MD
- Aug. - Oct. 2021, Integrated Medical Sciences III: Cardiovascular, Warren Alpert Medical School, Brown University, RI

Honors and Awards

- Distinguished Fellowship Finalist, Department of Biomedical Engineering, Johns Hopkins University 2022
- Sigma Xi honor society member 2021
- Conference award, 16th U.S. National Congress on Computational Mechanics 2021

- Travel award, San Diego Supercomputing Center summer institute on High Performance Computing and Data Science 2019
- Undergraduate Scholarship, Northwestern Polytechnical University 2014

Acknowledgments

Many wonderful individuals have helped me to accomplish my Ph.D. journey.

First and foremost, I would like to express my gratitude to my advisor, Dr. George Karniadakis, who has supported me tremendously in my academic career. George acts as a role model in my academic life, whose sharp mind, vision, diligence, and endless energy have and will motivate and inspire me for a lifetime. Joining the Crunch group is a life-changing experience for me, during which I participated in state-of-the-art research in multi-disciplines and have met wonderful collaborators and friends across the fields.

To my committee members, Dr. Kareen Coulombe, Dr. Joseph Crisco, and Dr. Yuri Bazilevs, thank you so much for being a wonderful board for my thesis and for advising and guiding me with your diverse expertise. A special thank you and gratitude to my collaborator, Dr. Jay Humphrey, at Yale for his unbounded knowledge in various fields, insightful guidance, and leadership of our projects. I am also grateful to Dr. Roy Aaron from Brown Medical School for supporting my postdoc application and offering me such a great opportunity to work with him in clinical research.

Many thanks to my collaborators, who are now or used to be Crunch members: Dr. Yue Yu, Dr. Zhen Li, Dr. Somdatta Goswami, and Mr. Enrui Zhang. Your knowledge of mathematics, computational sciences, and solid mechanics helped

me greatly in accomplishing my research. I would also like to acknowledge the Yale team, Dr. Ehsan Ban, Dr. Bruno Rego, Dr. Cristina Cavinato, Dr. David Li, and Dr. Bart Spronck, such a pleasure to work with you. I learned vascular biomechanics from theoretical, computational, and experimental perspectives through working with you. I want to express my gratitude to Dr. Zhiping Mao, Dr. Shengze Cai, Dr. Zhicheng Wang, Dr. Ansel Blumers, Dr. Yosuke Hasegawa, Dr. Xiaoning Zheng, Dr. Alireza Yazdani, Dr. Jiaqing Kou, Dr. Weiwei Zhang, Dr. Hopper, Dr. Jaganathan, Dr. Orr, Dr. Shi, Mr. Simeski, and Dr. Liu for helping me to accomplish our research.

I also want to acknowledge: Dr. Aniruddha Bora, Dr. Adar Kahana, Dr. Guansheng Li, Dr. Zixiang Liu, Dr. He Li, Dr. Kamaliyoti Nath, Dr. Mengjia Xu, Dr. Ameya Jagtap, Dr. Khemraj Shukla, Varun Kumar, Zhen Zhang, Qian Zhang, Sicheng Liu, and Zongren Zou for your supports and help.

At last, I would like to thank my family and friends. I could not finish this journey without your support from you. No words could express even a fraction of my gratitude for your love, support, and company.

Contents

Vitae	iv
Acknowledgments	x
1 Introduction	1
1.1 Specific Aims	2
1.2 Background	4
1.3 Prior Research	5
1.4 Significance and Innovations	7
1.5 Organization of Dissertation	8
2 Simulating Progressive Intramural Damage Leading to Aortic Dissection Using DeepONet – an Operator-Regression Neural Network	10
2.1 Abstract	11
2.2 Introduction	12
2.3 Methods	17
2.3.1 Phase Field Model	17
2.3.2 Histologically Motivated Microstructure	20
2.3.3 Linear Model	22
2.3.4 DeepONet	23
2.4 Results	26
2.4.1 Data Preparation	26
2.4.2 Damage Field Prediction (<i>Net D</i>)	28
2.4.3 Pressure–Volume Curve (<i>Net P-V</i>)	33
2.5 Discussion	38
2.A Additional Details	42
2.B Prediction of Displacement Fields	42
2.C Field Predictions Based on Strut Maps	45
2.D Exploratory Analysis of Pressure–Volume Data	47
2.E Linear Model Prediction of the Pressure–Volume Curve	49
2.F Iterative Prediction of the Damage Field Using Logistic Regression	55

3	Computational Modeling of Aortic Mechanics Based on Microimaging	61
3.1	abstract	62
3.2	Introduction	63
3.3	Methodology	66
3.3.1	Smoothed Particle Hydrodynamics	68
3.3.2	Four-fiber Model	73
3.3.3	Two-photon microscopy	74
3.4	Results	75
3.4.1	Benchmark testing	75
3.4.2	Imaging reconstruction	76
3.4.3	Imaging-driven simulations	80
3.5	Discussion	82
3.6	Acknowledgments	84
3.A	Parallel efficiency	84
3.B	Bilayer model parameter fitting	86
4	Non-invasive Inference of Thrombus Material Properties with Physics-Informed Neural Networks	88
4.1	Abstract	89
4.2	Introduction	90
4.3	Methods	93
4.3.1	Cahn-Hilliard and Navier-Stokes Equations	93
4.3.2	Physics-Informed Neural Networks (PINNs)	94
4.4	Results	97
4.4.1	Inference of Permeability	98
4.4.2	Inference of viscoelastic Modulus	107
4.5	Discussion	117
4.A	<i>A posteriori</i> practical identifiability analysis	120
4.B	Supplementary results for the inference of viscoelasticity	124
4.C	Inferring permeability from a pre-trained model	124
4.D	Relative error over time	125
5	Uncertainty Quantification of Nonlinear Ex-vivo Aortic Biomechanics with Functional Priors Learning and Posterior Estimation	127
5.1	Abstract	128
5.2	Introduction	128
5.3	Methodology	132
5.3.1	Four-Fiber family model	133
5.3.2	Learning functional priors: PI-GAN and DeepONet	134
5.3.3	Posterior inference	137
5.4	Results	138
5.4.1	Case I: 1D Constitutive Law	139
5.4.2	Case II: 2D Constitutive Law	142
5.4.3	Case III: 2D Constitutive Law with Ex-vivo Data	146
5.5	Discussion	149

6	Interfacing Finite Elements with Deep Neural Operators for Fast Multiscale Modeling of Mechanics Problems	152
6.1	Abstract	153
6.2	Introduction	154
6.3	Methodology	158
6.3.1	Deep Operator Network (DeepONet)	158
6.3.2	Coupling Methods	160
6.4	Results	163
6.4.1	Poisson equation	164
6.4.2	Heat equation	170
6.4.3	Elastoplasticity	174
6.4.4	Hyperelasticity	179
6.5	Discussion	187
6.A	Smoothed Particle Hydrodynamics	190
6.B	Accuracy of FEM and SPH	194
6.C	Network Training Details	195
6.D	Data Generation	196
6.D.1	Random field generation	196
6.D.2	Sampling in elastoplasticity	198
7	Conclusion	199
7.1	Summary	200
7.2	Open Questions and future work	201

List of Tables

2.1	Model architectures for predicting damage progression. The trunk net is chosen as a fully-connected neural network (FNN) with three hidden layers while the branch net is a convolutional neural network (CNN) with three blocks. Each block contains a convolutional layer, a LeakyReLU layer, and a batch normalization layer.	30
2.2	Model architecture for the P-V curve. The trunk net is a fully-connected neural network (FNN) with three hidden layers. The branch net can be a FNN or a convolutional neural network (CNN) with three blocks, where each block contains a convolutional layer, a LeakyReLU layer, and a batch normalization layer. [4, 8, 12] indicates the number of channels in each block.	37
2.3	Error Table for predicting P-V curves using different models, both high (DeepONet) and low (linear regression) fidelity. See also Fig. S8 - S11.	38
3.1	Parameters used in the benchmark mean-behavior model. The parameters are fitted based on biaxial data of a WT model.	76
3.2	Mass fractions of each constituent for the WT/mgR model. The parameters are collected from histological analysis.	78
3.3	Parameters used in the WT/mgR multilayer model. The parameters are fitted and collected based on biaxial fitting and histological imaging processing [62]. <i>eln</i> : elastin, <i>int</i> : intralamellar substances, <i>adv</i> : adventitia.	79
4.1	Summary of the mean relative L_2 error for the thrombus in a cavity over half and full time window. Training points in the cavity problem are sampled from $t \in [0.03, 0.63]$. The half window relative errors are defined as the errors from $t \in [0.03, 0.33]$, whereas the full window relative errors are defined as the errors from $t \in [0.03, 0.63]$	110
4.2	Summary of the mean relative L_2 error for the biofilm problem over half and full time window. Training points in the biofilm problem are sampled from $t \in [0.02, 1.16]$. The half window relative errors are defined as the errors from $t \in [0.02, 0.56]$, whereas the full window relative errors are defined as the errors from $t \in [0.02, 1.16]$	114
5.1	Setup for the three examples.	138
5.2	Structure and size of the PI-GAN in Case I. FNN: fully-connected neural networks.	139

5.3	Four-fiber family parameters and their range. All of the training data are generated with parameters uniformly sampled within the range. The data is adopted from [62].	140
5.4	Structure and size of the PI-GAN in Case II.	142
6.1	Setup for the four examples. D-D: Dirichlet-Dirichlet, R-D: Robin-Dirichlet, N-D: Neumann-Dirichlet, R-N: Robin-Neumann	163
6.2	Parameters value of the HGO model. We set the value of k_1, k_2 , and α the same for $i = 1$ and 2	180
6.3	Wall time comparison for SPH and DeepONet per iteration in the hyperelastic problem. The excessive cost of SPH is exacerbated because we use a time-dependent SPH solver to simulate a static problem. . . .	185

List of Figures

1.1	Thesis architecture and the challenges to address.	3
2.1	Sketch of the aortic wall and its microstructure. (a) The aortic wall consists of three layers: intima, media, and adventitia. (b) Artistic rendering of the lamellar microstructure of the aortic media by O’Connell et al. [191], reproduced with permission, plus (c) its representative volume element for the finite element simulation. 1: medial layer, 2: elastic lamellae, 3: lamellar media, 4: elastic (interlamellar) strut, 5: top view of the interlamellar struts (in white) shown as a strut map. z : axial, θ : circumferential, r : radial.	14
2.2	Histologically motivated microstructure of the murine aortic wall under biaxial loading. (a) 3D imaging of the murine arterial wall by multiphoton microscopy. Aortic wall (b) initiating and (c) propagating intralamellar delaminations, including histologically motivated microstructure for the phase-field model. Green: elastic lamellae. Blue: Smooth muscle cell nuclei. Red: Collagen in the adventitia. Scale bar: $100\ \mu m$.	21
2.3	Summary of the linear P-V model. At each volume V_k , the corresponding pressure P_k is predicted from the elastin strut map s_{ij} and a set of volume-specific linear regression coefficients a_{ijk} . Elastin struts ($s_{ij} = 1$) are displayed in black, while non-elastin matrix ($s_{ij} = 0$) pixels are displayed in white.	23
2.4	A schematic of DeepONet for fluid injection-induced dissection problems and its training data. (a) An unstacked DeepONet for learning an operator $G : u \rightarrow G(u)$ that takes two inputs, $[u(x_1), u(x_2), \dots, u(x_m)]$ and y , and returns the scalar $G(u)(y)$. The trunk net is a fully-connected neural network (FNN) whereas the architecture of the branch net varies depending on the problem. (b) Training data for the problem. First, we train <i>Net D</i> whose input u contains the elastic struts map and the current surface displacement map to predict the damage field at the current state. Next, the predicted damage field is taken as an input for <i>Net P-V</i> for predicting the corresponding pressure-volume curve for the injection fluid. Each network is trained separately and coupled for prediction.	25

2.5	3D snapshots for injection-induced delamination in the phase-field model at (a) an early stage and (b) a progressed stage. Note that (c, d) show enlarged views with details: (1) elastic lamellae, (2) damage region, (3) elastic struts, (4) intralamellar media, (5) a strut being torn, and (6) a broken strut. (e) Illustration of a pressure-volume response and its corresponding dissection state. A pressure drop results from irreversible tearing of the elastic struts and matrix indicated in the circle, with their corresponding displacement fields on the right. The computational geometry is symmetric with respect to the transparent part in (c) and (d), hence only half of the damage and displacement fields (for the transparent right side) are shown for illustration, with grey pixels showing struts and red indicating damage or high displacement. . . .	27
2.6	Image downsampling. Damage, displacement, and strut images can be downsampled by taking the average within a 4×4 kernel (yielding 2×2 image domains for the linear regression model, 6×6 image domains for the DeepONet model). Note that the physical meaning for each field after downsampling is: (a) the percentage of damage within a pixel, (b) mean displacement, and (c) percentage of strut within a pixel. . . .	29
2.7	Triples of a single training case for predicting damage progression. A dataset triplet is composed of images $\mathbf{u}$ (strut and displacement fields), a vector $\mathbf{y}$ containing pixel coordinates x_i, y_j and volume of injection V_k , and the damage percentage D_i	30
2.8	Training/Testing results of DeepONet for predicting damage progression. Model predictions and true damage fields are shown in the first and second columns, while the third column presents their absolute difference. (a, b) show the results for a training case at steps 0 and 60, while (c, d) represent a testing case at steps 0 and 48. The damage field indicates the percentage of torn material in the region at a particular pixel. . . .	31
2.9	Errors for predicting damage fields against the number of training cases and epochs. The network architecture is shown in Table 2.1. The shaded area indicates the one standard deviation averaged from 5 runs with different initiation. . . .	32
2.10	Triples of a single training case for the P-V response. A dataset triplet is composed of images $\mathbf{u}$ (strut and damage fields), volume of injection V_k , and its corresponding pressure P_k	33
2.11	Learning the pressure-volume curve using different input data and trunk network architectures. (a) Errors and deviations of different network architectures trained to learn the pressure-volume curve. The trunk net is chosen as a fully-connected network (FNN) for each run, while the branch net is chosen as a convolutional neural network (CNN) or a FNN, respectively. The networks trained with damage fields have a lower testing error than those without, given the same training cases. The error bars show one standard deviation from 5 runs with different initiation. (b, c) The training and testing errors are plotted against the number of training cases. FNN and CNN (trunk net) trained with damage fields are plotted in blue, while those without damage fields are plotted in red. (d, e) Using the same color style, the losses vs training epochs are plotted for FNN and CNN, respectively. . . .	35

2.12	DeepONet for predicting P-V responses. Model predictions (dashed lines) for training cases 1 and 2 are compared with the true data (black line, first column). In particular, model predictions with and without damage fields are plotted in red and blue, respectively. The middle and right columns compare the testing data with models trained with(red)/without(blue) the current damage field. Notice that the orange line represents the predicted P-V curves based on the prediction of <i>Net D</i> . The branch net is a FNN trained with 1900 training cases. The true data were generated from the phase-field finite element solver. Note that the pressure values are normalized to $O(1)$ by dividing by 100.	36
2.13	L_2^2 errors with different regularization parameters for Net P-V and Net D. We train a Net P-V and a Net D with 1,000 training cases using different regularization parameter α .	43
2.14	Training/Testing results of DeepONet for predicting displacement fields. Model predictions and true displacement fields are shown in the first and second columns, while the third column presents their absolute difference. (a, b) show the results for a training case at steps 0 and 57, while (c, d) represent for a testing case at steps 0 and 47. The length unit is μm .	44
2.15	Mean square errors (MSE) for predicting displacement fields against the number of training cases and epochs. The network architectures are the same in Table 1. The shaded area represents one standard deviation from 5 runs with different initiation.	45
2.16	Testing results of DeepONet for predicting damage fields based on partial information. (a) Model predictions and (b) true displacement fields at steps 0, 20, and 40 are shown, whereas the predictions are only based on strut maps. (c) Testing loss of models with different inputs versus injection volume. The red bars indicate testing loss (L_2 error) of model predictions with strut and damage field as input whereas the blue bars are predictions based only on strut maps. The error bars show the one standard deviation from 5 runs with different initialization.	46
2.17	Testing results of DeepONet for predicting displacement fields based on partial information. (a) Model predictions and (b) true damage fields at steps 0, 20, and 40 are shown, whereas the predictions are based only on strut maps. (c) Testing loss of models with different inputs versus injection volume. The red bars indicate testing loss (L_2 error) of model predictions with strut and damage fields as input whereas the blue bars are predictions only based on strut maps. The error bars show the one standard deviation from 5 runs with different initializations.	47
2.18	Maximum volume and pressure distributions. Because the data-generating finite element simulations of dissection were executed until failure, the training and test data used to develop and validate the P-V predictive models presented herein exhibited a large range of maximum injection volume and pressure values. (a) Kernel density estimation of the maximum volume data yielded two primary sub-populations of samples: one tightly concentrated around $\sim 3500 \mu\text{m}^3$ and another more variable group of samples that failed earlier. The overall distribution is well described by a two-component Gaussian mixture. (b) The distribution of maximum pressure values was unimodal and nearly Gaussian. GMM, Gaussian mixture model; KDE, kernel density estimation.	49

2.19	Statistical analysis of the pressure-volume data. (a) Percentiles and (b) correlations of observed pressure as a function of the injection volume. (c) Standard deviation of observed pressure as a function of the injection volume. (d) Standard deviation of volume-specific pressure plotted against the mean pressure observed at the same volume, showing a nearly linear relation. (e) While at low volumes, the pressure data is positively skewed, the skewness is closer to zero at higher volumes. (f) Similarly, pressure distributions are leptokurtic at low volumes, but approach a kurtosis of 3 at higher volumes. Together, these metrics suggest that the pressure distributions at high volumes are nearly Gaussian (cf Fig. 2.18(b)).	50
2.20	Preprocessing of the strut map for linear regression. (a) In the first version of the model, the input elastin strut map data is downsampled to half of its size along each dimension. In downsampling, each adjacent 2×2 block of pixels in the strut map is reduced to one pixel, whose value is set to the mean in the original pixel block. (b) In the second version of the model, the original strut map is first cropped to include only the top-left region surrounding the injection, and then the cropped strut map is downsampled, leading to a substantially reduced input space.	51
2.21	Absolute values of the least-squares linear P-V model coefficients. (a) Due to the large input space in the first version of the model, the regression results in spurious coefficient values in regions far from the injection site, suggesting possible overfitting. (b) The reduced input space yielded similar results in the remaining coefficients. In both models, pressure predictions at low injection volumes (e.g. $\leq 1000 \mu\text{m}^3$) are dominated by the elastin microstructure along the immediate boundary of the initial injection region, but broader regions of the tissue contribute to the model at higher volumes ($\sim 2000 \mu\text{m}^3$).	52
2.22	Performance of the linear P-V model for three representative samples. (a) In the first version of the model, the uncertainties in predicted pressure values estimated from the training data are substantially smaller than the errors observed in validation against the test data, indicating that the model overfits the training data. (b) In the second version of the model, which utilizes a cropped and downsampled strut map as its input, the uncertainties estimated from the training and test data are approximately equal. CI, credible interval.	53
2.23	Results of the linear model for predicting P-V responses. Model predictions (dash lines) for training cases 1 and 2 are compared with the true data (black line, first column). In particular, model predictions with and without damage fields are plotted in red and blue, respectively. The other columns compare the testing data with models trained with (red)/without (blue) the current damage field. The branch net is FNN trained with 1900 training cases. The true data were generated from the phase-field finite element solver.	54
2.24	Neighborhood normalized stiffness maps. Maps of the tissue stiffness averaged over 1×1 , 3×3 , and 5×5 pixel neighborhoods, for a representative sample. The stiffness values were computed directly from the elastin strut map, and normalized to the prescribed elastin stiffness, which is $20\times$ greater than that of the non-elastin matrix.	57

2.25	Logistic regression results. (a) Best-fit coefficients for the logistic regression model, showing contributions of damaged neighboring pixels and effective local tissue stiffness. (b) Corresponding map of the contribution of local tissue stiffness to the probability of tearing.	57
2.26	Logistic regression-based damage prediction. True (i.e. finite element-derived) and predicted (by the logistic regression model) maps of damage, shown at frames one-third ($t_{1/3}$) and two-thirds ($t_{2/3}$) of the way through the simulation as well as at the final simulation time (t_1). The damaged region is displayed in white, while the locations of elastin struts are shown as red dots.	59
3.1	Schematic of imaging-driven modeling of aortic mechanics. (a) Aorta isolation from mouse models. A segment of the descending thoracic aorta (DTA) was isolated from euthanized mice and was (1) cannulated and mounted on a biaxial loading device. (b) The segmented aorta was pressurized and loaded to in vivo states where two-photon imaging and mechanical data were collected. The collected data were analyzed separately: (c) the mechanical data were used to quantify mechanical parameters in the four-fiber model and (d) the imaging data were post-process in a specific pipeline for geometry reconstruction. (e) Detailed computational modeling was developed using parameters fitted from (c) and particle representation of the microstructure from (d).	67
3.2	“spiky” kernel in SPH with the kernel radius h.	68
3.3	Algorithms that introduce deposition prestretch in SPH. Left: The inner wall is fixed with incremental deposition prestretch (ΔG). The wall is free to move in the radial direction while being bounded in the circumferential direction. Right: Once the deposition prestretch is imposed, the inner wall is moved in the radial direction until the end of the moving step.	72
3.4	Inflation test and wall stresses predicted by the mean-behavior SPH model. Left: SPH particles at the initial (blue) and loaded (orange) configuration. Right: Circumferential stresses predicted from the SPH (red dots) and an analytical thick-wall model during inflation.	77
3.5	Procedures of imaging processing and microstructure reconstruction. (a) A stack of raw images that represent elastic lamellae was processed, (b) filtered, and (c) thresholded through a customized pipeline. (d) The thresholded images are converted into particles with collagen (yellow) coating on the outer and smooth muscle filled in the intralamellar space (blue). (e) The reconstructed geometry is transformed in a cylindrical coordinate with curvature.	77
3.6	Reconstructed microstructure for the (a) mgR and (b) WT model. Mass fraction of each constituent: (Elastin, WT) 44.58%, (SMC, WT) 32.12%, (Adventitia, WT) 23.30%; (Elastin, mgR) 45.99%, (SMC, mgR) 21.68%, (Adventitia, mgR) 32.33%.	79
3.7	Microstructure and stresses distribution in the WT (top) and mgR (bottom) predicted from SPH simulations. The stresses are simulated at diastole (80 mmHg).	80
3.8	Distribution of the radial (σ_r) and circumferential (σ_θ) stresses in the WT (left) and mgR (right) model. The predicted stresses from SPH are plotted with the blue solid lines with the dashed lines representing analytical solutions.	81

3.9	Benchmark testing for parallel efficiency of LAMMPS SMD. We test the code efficiency for a fixed size problem run on different numbers of processors. Speedup factors of LAMMPS SMD (out implementation) are bounded by the theoretical speedup.	85
3.10	Fitting from the four-fiber model (red lines) with best-fit parameters and biaxial testing data (dots) for the WT mouse.	86
3.11	Fitting from the four-fiber model (red lines) with best-fit parameters and biaxial testing data (dots) for the mgR (Marfan) mouse.	87
4.1	Schematic of a PINN for solving inverse problem for Cahn-Hilliard and Navier-Stokes PDEs. Circled by blue boxes, <i>Net U</i> and <i>Net W</i> represent surrogate models for the PDEs solution whose derivatives can be computed with automatic differentiation (AD). The computed derivatives are used in the loss function to restrict model outputs such that they satisfy the system of PDEs in Ω . For inverse problems, the residual between sensor measurements $u _{data}$ and model outputs $\hat{u}_\Omega$ are included in the loss function. We use ADAM to optimize the model parameters θ (weights and biases) and search the unknown values of the material parameters from the PDEs λ to minimize the loss function.	95
4.2	2D flow past a thrombus. (a) The computational domain is a channel with walls on the top and bottom boundaries with the inlet flow $u(t, y)$ entering from the left side.; $\phi=1$ corresponds to fluid. A thrombus with a permeable core $\phi = -1$ and shell $\phi = 0$ is present at the bottom boundary. (b) Sampling points for inferring permeability include initial points ($\star$) at the time t_0 , inner points ($\star$) from t_1 to t_n , boundary points ($\star$) on boundaries, and point measurements ($\star$) with PDE solutions.	98
4.3	Prediction and error of 2D flow past a thrombus for various permeability values at $t = 0.63$. Representative snapshots of the reference data for permeability value κ (a) 50 (b) 0.5 (c) 0.005 (d) 0.0005 are shown against the predicted phase and velocity field. The first column shows the reference simulation results while the second shows the results from the PINNs. The third column shows the absolute value of the difference between references and model predictions. The network is informed by 10,000 point measurements drawn from $t \in [0.33, 0.63]$	100
4.4	Performance and history of PINN on predicting permeability of 2D flow past a thrombus with κ ranging from 0.0005 to 50. (a) Comparison of the inferred values with the reference values. (b) Mean relative L_2 errors for phase and velocity field when κ varies from 0.0005 to 50. The plots in (c, d, e, f) illustrate the inference of the permeability with respect to the number of iterations of PINN when $\kappa = 50$, $\kappa = 0.5$, $\kappa = 0.005$, and $\kappa = 0.0005$. The number of training points is 10,000.	101
4.5	Effect of the number of training points on the accuracy of PINNs. $N_{total} = 100,000$. Plots (a, d, g, j) show comparison of reference and inferred permeability for a different number of training points from 0.1% of the total number of points to 40%. The second and the third column show the mean of relative L_2 error for (b, e, h, k) velocity field and (c, f, i, l) phase field over time for different numbers of training points. We shaded the area where the error is lower than 1%.	103

4.6	Predictions of 2D flow past a thrombus trained with noisy phase field measurements. (a) Inferred permeability and (c) mean relative L_2 errors between the model predictions and the reference for phase ϵ_ϕ and velocity field ϵ_v . Noise is added to the phase field with the noise level ranged from 0 to 20%. Here, 10,000 data points are scattered in the spatio-temporal domain as the training data to infer the permeability.	104
4.7	2D flow past a thrombus with phase dependent permeability. (a) History of network losses (Loss PDE, Loss IC, Loss BC, and Loss Data) and (b) inferred the permeability κ as a function of ϕ . (c) The values of a and b at each epoch. (d) Comparison of phase field and velocity field for $\kappa(\phi)$ at $t = 0.78$ and their absolute error. The core permeability is 0.001 and the shell permeability is set 1 as the actual values. 10,000 data points are scattered in the spatio-temporal domain from 30 snapshots ($t \in [0.03, 0.93]$) as the training data to infer the permeability.	105
4.8	(a) Schematic for viscoelastic thrombus in a cavity, (b) history losses for each term, and (c) inference of λ_e. (a) We impose a time-dependent sinusoidal vertical velocity $v(t, x) = -(1 - \cos(2\pi t)) \sin(2\pi x)$ at the top boundary, and set the left and right sides as periodic. We changed the weight for each loss term at epoch 600,000. Training data points are scattered among the spatio-temporal domain to train the network, including 20,000 phase field points from the inner spatio-temporal domain, 2,000 pressure measurements at boundaries, and 1,000 initial points. Loss PDE : loss for the PDEs residuals, Loss BC : loss for boundary conditions, Loss IC : loss for initial conditions, Loss Data : loss for measurements data.	107
4.9	Viscoelastic thrombus in a cavity. The first column presents the phase field distribution ϕ , velocity field v , and pressure distribution p from reference data, respectively. The second column shows the same field predictions from the model at $t = 0.48$. The absolute difference between the data and the model predictions are plotted in the third column. . .	109
4.10	2D flow past a viscoelastic biofilm setup. (a) The computational domain is a channel with wall boundaries on the top and bottom sides, and with flow $u(t)$ entering from the left side. A viscoelastic biofilm is present at the bottom boundary. (b) Training points include initial points (★) at time t_0 , inner points (★) from t_1 to t_n , boundary points (★) on the boundaries, and point measurements (★) with PDE solutions. The sampling points are refined within the dashed line area to improve the results of the PINN model.	110
4.11	History of (a) network losses and inference for (b, c) λ_e and κ against the number of training epochs for the biofilm problem. The full time window is from 0.02 to 1.16.	112
4.12	2D flow past a viscoelastic biofilm. With λ_e and κ unknown at the same time, we sample 15,000 phase field data, 2,000 pressure points at boundaries, and 1,000 initial points as the training data for parameters inference and field regression. The time is from 0.02 to 1.16. Representative snapshots (at $t_0 = 0.02$, $t_1 = 0.44$, and $t_2 = 0.86$) of the reference (a) phase field and (b) velocity fields are shown against the predicted phase and velocity from the model. The first column shows the reference fields from simulation, the second shows the predicted results from PINNs, and the third column shows the absolute value of the difference between the references and the model predictions.	113

4.13	Effect of the number of training points on the prediction value of λ_e with κ value as known. (a, g) Predicted λ_e versus the number of training points for case with zero κ and non-zero κ . (b-f, h-l) The mean relative error L_2 versus the number of training points for each field when $\kappa = 10$ and $\kappa = 0.01$. We also plotted the inference results and L_2 error given various training data sources such as $\phi + u + p$, $\phi + p$, and only with ϕ . For the one with pressure information and phase field, inner (Ω) and boundary (∂) pressure measurements are used respectively to train the neural network. We shaded areas with lower than 5% of error. $t \in [0.02, 1.16]$, $N_{total} = 200,000$	115
4.14	2D flow past a viscoelastic biofilm trained with noisy measurements with λ_e unknown. (a) The inferred λ_e . (b) presents the mean relative L_2 errors between the model predictions and the references for each field. Noise is added to the phase field with the noise level ranged from 0 to 20%. Here, 16,000 (8%) data points are scattered in the spatio-temporal domain.	116
4.15	A posteriori practical identifiability analysis of κ to the phase field. (a-d) Phase field difference between κ is set as true values and inferred values.	121
4.16	2D flow past a viscoelastic biofilm for $\lambda_e = 15$ and 0.1. The first two lines show the phase field different at $t_0 = 0.44$ and $t_1 = 0.86$. The middle two lines and the last two lines show the comparison of velocity and pressure at t_0 and t_1 . The last column presents the absolute difference between fields data when $\lambda_e = 15$ and 0.1.	122
4.17	Results for the biofilm case with additional results. We train the PINN model with $p + u$ and $\phi + u$ as comparisons, and we plot the inference results and errors on the top of Fig. 4.13 with opaque lines indicating the new results. The shaded areas indicate that the error is lower than 5%.	123
4.18	Effect of using a pre-trained model on the inference of $\kappa = 0.005$ and 0.0005. (a, c) compare the history of inferred κ from the model with and without pre-training versus epochs. (b, d) shows the training loss versus epochs for each model. The pre-trained model is adopted as the weights and biases for $\kappa = 0.5$ at last epochs.	125
4.19	Relative L2 error over the full time window for the (a) cavity and (b) biofilm example.	126
5.1	Schematic of learning functional priors (yellow box) and posterior estimation (green box). (1) Prior knowledge of the function family will inform a physics-informed generative adversarial network (PI-GAN) in the training stage. (2) Upon the completion of training PI-GAN, the model will be incorporated in a sampler to (3) estimate the posterior conditioned on new data D	131
5.2	Physics-informed generative adversarial networks (PI-GANs). The generator of PI-GAN is a DeepONet with a vector of noise ξ and stretch λ as the input of the branch and trunk net. The network predicts the corresponding stress $\sigma_{\theta,r}$ which can be calculated by taking the partial derivatives (physics-informed) of the intermediate output (strain energy density W). The generated data and real data will be transmitted to the discriminator for estimating their difference.	134

5.3	Uncertainty quantification of 1D four-fiber model. a: 1280 random samples of σ_θ . b: Mean and standard deviation band computed from Bayesian inference with functional priors. c: Mean and standard deviations predicted by Gaussian process.	140
5.4	Prediction errors at different noise scales and number of points. The color represents the magnitude of error defined in Equ. 5.9.	142
5.5	Mean and standard deviations (STD) for inferring σ_θ given five sparse measurements. (a) Mean (red) and two STD (yellow) of σ_θ versus true stress (blue). Line comparison at $\lambda_z = 1.51$ and $\lambda_\theta = 1.51$ are plotted in (b) and (c). The network is only trained to with σ_θ	143
5.6	Informative inference of stresses and energy based on 8 measurements of σ_θ and σ_z. The mean (red) and 2 STD (orange) are plotted with the true data (blue) for (a) σ_θ , (b) σ_z , and (c) W. The network is trained with both σ_θ and σ_z	144
5.7	Uninformative inference of stresses and energy based on 8 measurements of σ_θ. The mean (red) and 2 STD (orange) are plotted with the true data (blue) for (a) σ_θ , (b) σ_z , and (c) W. The network is trained with both σ_θ and σ_z	145
5.8	Stresses inference for three different genotypes: (a) wild-type, (b) Fbn1mgR/mgR, and (c) hBAC+mWT. The contours of nonlinear regression (Nonlin. Reg.) are calculated with all data collected from ex-vivo experiments whereas the functional prior (func. prior) estimates the mean of subject-specific constitutive relations only based on 6-7 points cluster at high stretches. Taking the nonlinear regression results as reference, the prediction errors range from 0.05 to 0.11. . . .	147
6.1	Schematic of multiscale modeling with DeepONet. (a) In a multiscale mechanics system, traction $T(x, t)$ acts on the boundary of domain Ω , which is decomposed into sub-domains described by a variety of microscopic models. A macroscopic model describes the response in the bulk region whereas nonlinear, microscopic, or data-driven models capture the detailed response in the small regions. (b) DeepONet, composed of a branch and a trunk net, is able to learn the response of the microscopic systems (from simulated or multi-modal data) and serve as a surrogate in the multiscale system. MD: molecular dynamics, DPD: dissipative particle dynamics, SPH: smoothed particle hydrodynamics, PD: peridynamics, NL FEM: nonlinear finite element, FEM: finite element, MF: multifidelity.	157
6.2	Schematic architecture of DeepONet. DeepONet learns the mapping operator G from an input function g to its corresponding output function $G(g)$. The input of branch and trunk net are g and $\mathbf{y} \in \mathbb{R}^p$, which represents a discretized function from $g(x_1)$ to $g(x_m)$ and information such as coordinates and time, respectively. Note that the output dimension of the trunk net, n , is consistent with that of the branch net. The final output $G(g)(\mathbf{y})$ is computed as the dot product of $\mathbf{b}$ and $\mathbf{t}$. . .	158

6.3	A flexible framework in domain decomposition. The proposed coupling framework is able to adopt either (a) overlapping or non-overlapping domain decomposition coupled with Dirichlet, Neumann, or Robin boundary conditions at the interface. As an illustration, (b) presents a Robin-type boundary condition imposed on model I with interfacial solution $\hat{u}(x_1)$ updated by a relaxation scheme. Dirichlet and Neumann boundary can be imposed by adjusting the value of R_1 and R_2 . The relaxation parameter, θ , is either fixed at a value or updated dynamically. $T_{II}^n(x_1)$ represents the Neumann-related information at x_1 from Model II, e.g., traction or derivatives.	162
6.4	Setup of the Poisson equation. (a) A 1×1 unit square is decomposed into two overlapping regions, Ω_{NN} (light blue) and Ω_{FEM} (meshed). Ω_{FEM} is bounded by Γ_1 and Γ_0 , whereas Ω_{NN} is a 0.3×0.3 square in the center with boundary Γ_2 . (b) DeepONet takes $u_{FEM} _{\Gamma_2}$ as input and yields the solution in Ω_{NN} , denoted as $u_{NN} _{\Omega_{NN}}$. (c) Formulation of an overlapping D-D method. Given the boundary condition on Γ_0 , the coupling framework iteratively updates the corresponding boundary condition on Γ_1 . The relaxation parameter θ is either fixed or updated based on the Aitken's rule [311].	164
6.5	Results of coupling FEM and DeepONet for the Poisson equation. (a-b) Model predictions from the FEM and DeepONet with the corresponding absolute errors in Ω_{FEM} and Ω_{NN} . (c) Model predictions at the interfaces ($u _{\Gamma_1}$ and $u _{\Gamma_2}$) and the true solution (red line). The relative errors of model predictions at the interfaces are far less than 1%. Please see Fig. 6.4 for Ω_{NN} and Ω_{FEM}	167
6.6	Parametric studies of the Poisson equation. (a) Given a boundary condition on Γ_0 , the convergence history of the coupling model varies with the coupling method (Robin or Dirichlet) and parameter θ . (b) The model is trained with data generated from a random field with $\alpha = 5$. We test the generalization ability represented by the relative errors of the coupling model for cases corresponding to $\alpha = 1$ to 10. (c) Box plots of relative errors for DeepONets trained with 100 to 900 cases. Each column shows the coupling results by testing with 50 different cases. The shaded area indicates that the relative error on Γ_1 is less than 1%. Notice that we present the error with respect to the ground truth to show the convergence and accuracy of our framework.	168
6.7	Setup of the time-dependent problem (heat equation). (a) The spatio-temporal domain is decomposed into two sub-domains with Γ as the interface between FEM and NN domain. (b-c) Formulation of the coupling framework for the heat equation. The framework first solves for $u_{FEM}^{n+1,k+1}$ from FEM with updated information at the interface $\tilde{u}$ at k -th step. Then, the computed derivative at Γ is transmitted into DeepONet, which predicts the spatial derivative in Ω_{NN} (c). The solution in Ω_{NN} is computed based on a time-stepping scheme, followed by a relaxation update (b). The initialization at $k = 0$ and $n = 0$ is described in the main text.	170

6.8	Results of coupling FEM and DeepONet for the heat equation. (a) Model predictions from DeepONet and FEM. DeepONet predicts the solution in $x \in \Omega_{NN}$ while FEM predicts the solution for $x \in \Omega_{FEM}$. The black line at $x = 0.5$ denotes the interface of the two domains. The solution at $x = 0.75$ (indicated by the white dashed line) is presented in (b). (c) The mean square errors of FEM (red) and DeepONet (blue) vs time (t).	173
6.9	Setup of the elastoplasticity problem. (a) A $2l \times 2l$ solid plate is clamped on the bottom edge with a traction t_0 distributed on the top edge. (b) The yellow region (Ω_{FEM}) is modeled with linear elasticity, whereas the blue region (Ω_{NN}) is modeled with elastoplasticity. Interfacial displacement (Dirichlet boundary condition) $u _{\Gamma}$ computed from FEM is transmitted to DeepONet as input. (c) The DeepONet estimates the stress components in Ω_{NN} , $\sigma_{NN} _{\Omega_{NN}}$, based on which the interfacial traction $t_{NN} _{\Gamma}$ can be calculated accordingly. The predicted traction is updated and provided to FEM as a Neumann-type boundary condition. In this example, we set the interior circle radius $r_i = 0.1$, the interface circle radius $r_o = 0.3$, and the plate size $l = 1$	174
6.10	Results of coupling FEM and DeepONet in elastoplasticity. (a-b) Results of the normal stress in y direction (σ_{22}) and equivalent plastic strain ($\bar{\epsilon}^P$). From left to right: True value, FEM/DeepONet predicted value from coupling, and their absolute errors. (c) Displacement and traction at the interface Γ . True values and predicted values from coupling are presented. Solid lines: displacement and traction of true data in x (blue) and y (red) directions. Dashed lines: displacement and traction of the model predictions in x (yellow) and y (black) directions. (d) The history of the relative errors of the coupling model at the interface. ϵ_{FEM} and ϵ_{NN} refer to the L_2 error of displacement and traction from FEM and DeepONet, respectively. Notice that we present the error with respect to the ground truth to show the convergence and accuracy of our framework.	178
6.11	Setup of the hyperelasticity problem. (a) A unit square is decomposed into Ω_{NN} and Ω_{FEM} . We impose a non-uniform traction boundary condition on the top edge and fix the displacement at the bottom and train multiple DeepONets to represent the mechanics of an SPH model. The material is reinforced by fibers in the vertical direction. Information at the interface (displacement u and first Piola-Kirchhoff stress P) is transmitted between DeepONet and FEM. (b) Traction and displacement in different directions are exchanged at the interface Γ . FEM predicts the external domain while the DeepONet is trained based on SPH data. (c) Two types of DeepONet are proposed to predict the mechanics of the system: predicting stresses based on displacement information (green boxes) and vice versa (yellow boxes).	182
6.12	Results of coupling FEM and DeepONet in the hyperelasticity problem. Results of (a) displacement in Ω_{FEM} and (b) P_{22} in Ω_{NN} . From left to right: prediction from SPH, prediction from FEM, and their absolute differences. (c) Predicted displacement and traction at the interface. Predictions from both FEM and DeepONets are compared with the true data. Solid lines: displacement and traction of true data in x (blue) and y (red) directions. Dashed lines: displacement and traction of the model predictions in x (yellow) and y (black) directions.	184

6.13	L_2 errors of the interfacial displacement in the hyperelastic multiscale model with a Robin boundary. The red line with Robin coefficient $R=0.25$ converges fastest among the testing cases. The black solid line represents the relative error history of a Neumann boundary condition ($R=0$). The shaded area indicates that the relative error is less than 2%. Notice that we present the error with respect to the ground truth to show the convergence and accuracy of our framework.	186
6.14	Comparison of SPH and FEM. (a) We impose the same Dirichlet boundary condition on the top and internal circle for a FEM and SPH model. (b-c) Interfacial displacement and traction of both models. . . .	194
6.15	Network size for various problems. We tabulate the branch/trunk network size for the four examples in Sec. 6.4 with the number of training epochs.	195
6.16	History of Training/testing errors of DeepONet for different problems.	197
6.17	Details of network setup (inputs/outputs and MSE errors of training/testing) for each example in Sec. 6.4.	197

CHAPTER ONE

Introduction

1.1 Specific Aims

Aortic dissection is responsible for significant morbidity and mortality in children, young and older adults [79, 189, 88]. One possible outcome for an artery undergoing dissection is that the crack turns inward and forms a false lumen, while the other one can be the crack turning outward and causing lethal internal bleeding. Although most of the studies focused on the stage with a developed false lumen, there has been modest attention on the initiation and propagation of aortic dissection. One hypothesis states that aggregates of glycosaminoglycans (GAGs) play an important mechanical role in facilitating aortic dissection, followed by many *in silico* investigations [103, 272, 104]. However, the underlying biomechanical factors (such as elastin alignment and intralamellar collagen) are neglected, which are imperative in predicting the dissection outcomes. This significant knowledge gap causes difficulty in understanding the mechanism of the initiation and propagation of aortic dissection.

In this thesis, we propose to investigate GAGs-initiated aortic dissection and its relevant biomechanical systems by developing imaging-driven multimodality and multifidelity computational and artificial intelligence models. Specifically, synthetic and *ex vivo* data, such as subject-specific geometries, regional wall displacement, and collagen/elastin alignment, will be incorporated in machine learning and particle-based models for investigating the initiation and propagation of aortic dissection under the influence of subject-specific biomechanical factors. In addition, we will inform relevant *ex vivo*/synthetic data to novel physics-informed models for inferring underlying biomaterial and biomechanical properties. Thus, we propose the following specific aims:

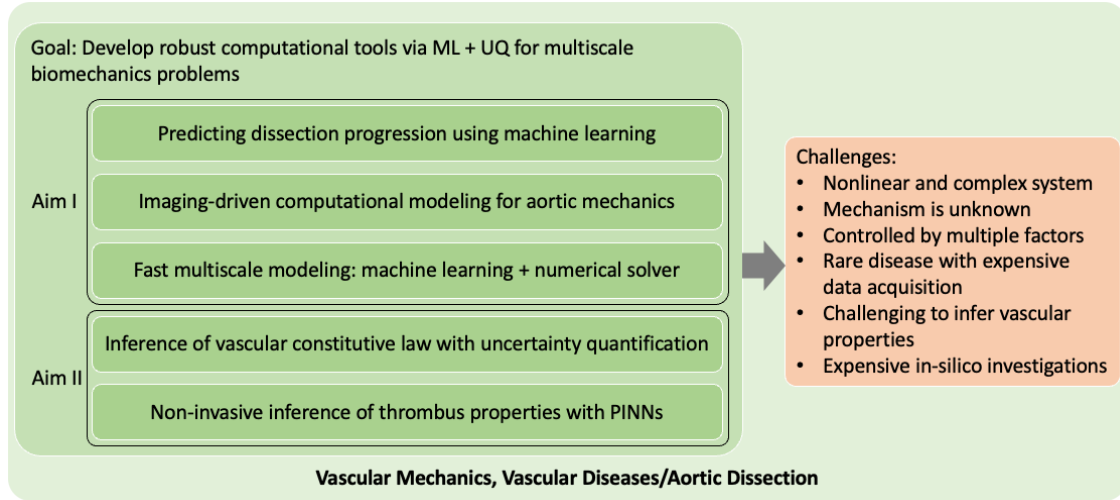


Figure 1.1: Thesis architecture and the challenges to address.

Aim 1: Use novel ex-vivo data collected from mouse models in conjunction with simulated data to develop computational models for investigating intramural delamination processes and its relevant biomechanical systems. Toward this end, we will develop neural operators (DeepONet) informed by continuum-level finite element models and imaging-driven particle models to investigate vascular biomechanics and the early stage of aortic delamination.

Aim 2: Develop machine learning models to quantify uncertainty of subject-specific biomechanical properties of the aorta based on measurable data. Toward this end, I will develop physics-informed machine learning to non-invasively infer the regional biomaterial properties of a thrombus in an artery based on synthetic imaging data. Then, we will develop a probabilistic framework to learn the functional knowledge of the constitutive model of the aorta (described by the four-fiber family model) using generative adversarial network (GAN) and DeepONet. The trained models will be combined with a Bayesian inference framework to enable efficient quantification of the posterior of biomechanical stresses given sparse measurements.

Fig. 1.1 presents a schematic of this thesis. Under the umbrella of vascular mechanics and diseases, this thesis developed a series of robust computational models and machine learning models with uncertainty quantification to address the pressing challenges in understanding aortic dissection, e.g., nonlinearity and complexity of the system, unknown mechanism, expensive data acquisition and *in silico* investigations, etc. We will address Aim 1 in Chap 2, 3, and Chap 6. Chap 4 and 5 will address Aim II. Completing the proposed aims will contribute towards building an improved patient-specific probabilistic modeling of chronic dissection and interventional planning in the future.

1.2 Background

Thoracic aortic aneurysm and dissection cause significant morbidity and mortality [207, 192, 103], with estimates up to 10,000 to 15,000 deaths per year in the USA alone. Particularly troublesome is that aortic dissections are the cause of death in over 5% of all young people suffering sudden cardiac death [47, 264]; that is, this lethal condition affects young and old alike. Left ventricle contraction distends the aorta wall at systole, whereas the aortic wall recoils to propel forward blood circulation at diastole. The aorta must maintain structural integrity since it is subjected to long-term axial and radial stresses. However, any process that results in weakening of the aortic wall may lead to a higher local stress field which may initiate intramural dissections.

One hypothesis states that dissection may arise from aggregates of glycosaminoglycans (GAGs) in the media layer. Although GAGs compose only 2-5% weight of the aorta, they are important contributors in maintaining normal arterial structure

and function. Aggregations of GAGs, e.g. hyaluronan and versican, are primarily found within the extracellular space of the media in arteries. In particular, aggregates of versican are negatively charged, tending to imbibe water and generating a local Donnan swelling pressure. Increasing evidence [53, 127, 273] suggests that abnormal increases in GAGs that induce Donnan swelling pressure would be detrimental to structure integrity, for these pathological poolings of GAGs could lead to a devastating condition of local stress concentrations that may initiate intramural dissections. Initial delamination of the aortic wall propagates within the medial layer, but can turn inward and connect with the true lumen to form a so-called false lumen within the wall or turn outward and lead to a rupture in the presence of a compromised adventitia.

1.3 Prior Research

Most prior studies of the biomechanics of aortic dissections have followed along four lines. (i) In a series of papers from 1987-98, Roach and colleagues focused on the intramural (blood) pressure needed to propagate an initial medial defect within the thoracic aorta [91, 163, 36, 250, 266, 225]; they consistently found that static pressures needed to be 500 mmHg or more to dissect the normal wall. Prokop et al. [93, 267, 183] reported similar values under static loading, but emphasized that the pressure for dissection was lower (~ 200 mmHg) under pulsatile flow. (ii) More recently, multiple groups have used either *in vitro* flow loops or computational fluid dynamics models to estimate blood flow and pressure within the false lumen of a dissection [115, 206, 247, 116]. One of the main conclusions from these papers is “lack of a distal tear could result in increased diastolic pressure within the false lumen which may cause an increased risk of false lumen expansion and

rupture” [261]. Similar suggestions hold in the case of a partial thrombosis that blocks the outlet of a false lumen. (iii) In contrast, Holzapfel and colleagues reported experimental and computational results on the propagation of dissection in mode I tearing tests [256, 74, 243]. Among other findings, they showed that the energy of dissection was greater in axial than circumferential directions in the excised human abdominal aorta, which emphasizes the importance of anisotropy of the wall. Pasta et al. [197] performed similar tearing tests, albeit more appropriately on normal and diseased human ascending aortas. They reported that delamination strength was lower in the inner than the outer wall and that aortic disease lowers this strength further. Each of these three lines of investigation has provided new insight into the propagation of a pre-existing defect, but none have addressed the similarly important biomechanical mechanisms of initiation or subsequent early effects of intramural thrombus on the dissecting aortic wall. (iv) Finally, a few groups have performed finite element analyses of wall stress in the ascending aorta and arch to try to identify reasons for the preferential initiation of dissections in this region [185, 73, 168]. Most are inappropriately based on homogenized linear elasticity, hence such studies fail to account for the detailed histopathological or biomechanical and genetics-based data that are available. *In conclusion, failing to account for impacts from detailed mathematical models, bio-data, and in vitro geometry together contribute inaccuracy in predicting the propagation of delamination in aortic dissection.*

Remark: There exist three main hypotheses on explaining, from a mechanistic point of view, the initiation and propagation of aortic dissection. The first one assumes that the dissection may arise from an intimal defect on the intramural wall. Following this line, many insightful in-silico works assess the risk of a false lumen for further dissection [200] and investigate the stress distribution for an

artery with flaps [215]. Another theory hypothesizes that a dissection may occur at or near the region with a concentrated stress field: the subclavian arterial wall has higher stresses than its surrounding vessels due to a sharp bifurcation from the aortic arch [65]. The third is that aortic dissection may arise from accumulates of GAGs, where each contains a pooling of negatively charged solutions that draws water into it. The combination of osmotic pressure and internal stresses on the artery wall will predispose the artery to dissection. Based on this hypothesis, a series of in-silico studies use smoothed particle hydrodynamics [221, 4] and finite element model [226, 227] to illustrate the role of accumulates of GAGs on the progression of dissection. This research is based on the third hypothesis for studying the role of biomechanical factors in aortic delamination.

1.4 Significance and Innovations

Due to the emergency nature, aortic dissection cannot be observed directly *in vivo*: cracks could propagate asymptotically and unconsciously. Hence, it is hard to monitor the early stage of dissection, not to mention studying the underlying biomechanics. **One crucial output of this computational study is to quantify, for the first time, the role of biomechanical conditions in either preventing or promoting the early-stage propagation of dissection on a vessel wall (Chap 2 and 3).** Using a computational model allows us to study the dissection progression of a subject-specific geometry under diverse biological and mechanical conditions, which will provide more quantitative insights to unveil their roles, respectively. An improved understanding of the mechanisms can enhance our knowledge in predicting, preventing, and managing aortic dissection in humans. **Another important output of this study comes from the development of probabilistic mod-**

els for quantifying biomechanical properties based on sparse measurements (Chap 4 and 5). Quantification of subject-specific biomechanical properties will, in turn, enable more accurate subject-specific modelings which can predict dissection on different mouse model. **Third**, although we focus on aortic dissection, the computational framework we developed in this research can be translated into other arterial dissections and damage as well, including intracranial, cervical, carotid, renal, and coronary [263].

This proposal is novel because we will employ novel multimodality imaging data to inform our models: using unique ex vivo data will allow us to delineate factors that initiate and propagate a dissection (Aim 1). We will understand the strong biological variability inherent to dissection by using different mouse models where each mouse model exhibits a biological variability to inform the proposed probabilistic predictive frameworks. Another innovation is using neural networks to combine ex vivo/synthetic data for characterizing material properties (aortic wall/thrombus) with uncertainty quantification (Aim 2).

1.5 Organization of Dissertation

This thesis is organized as follows: In Chap 2, we develop a data-driven surrogate model of the delamination process for differential strut distributions using DeepONet, a new operator–regression neural network. This surrogate model is trained to predict the pressure–volume curve of the injected fluid and the damage progression within the wall given a spatial distribution of struts, with *in silico* data generated using a phase-field finite element model. Then, Chap 3 presents a particle-based computational model for investigating dissection pro-

gression based on the ex-vivo mouse aorta. The computational geometries are subject-specific, reconstructed from two-photon imaging data. We perform a series of in-silico mechanical tests for the aorta of the wild-type and Marfan mice. In Chap 4 and 5, however, we develop machine learning models for quantification of biomechanical properties of thrombus and arterial wall, which are vital for accurate patient-specific modeling. We employ physics-informed neural networks (PINNs) to infer properties of biological materials using synthetic data. In particular, we successfully apply PINNs on inferring permeability and viscoelastic modulus from thrombus deformation data, which can be described by the fourth-order Cahn-Hilliard and Navier-Stokes Equations (Chap 4). In Chap 5, we propose to employ a probabilistic machine learning model in conjunction with a Bayesian sampling algorithm to efficiently estimate the posteriors of the constitutive relationships. The last chapter, however, serves as an exploratory attempt to accelerate multiscale simulations by combining machine learning and finite elements. Background that are more specific to each aim can be found at the beginning of each chapter.

CHAPTER TWO

Simulating Progressive Intramural Damage Leading to Aortic Dissection Using DeepONet – an Operator-Regression Neural Network

2.1 Abstract

Aortic dissection progresses mainly via delamination of the medial layer of the wall. Notwithstanding the complexity of this process, insight has been gleaned by studying *in vitro* and *in silico* the progression of dissection driven by quasi-static pressurization of the intramural space by fluid injection, which demonstrates that the differential propensity of dissection along the aorta can be affected by spatial distributions of structurally significant interlamellar struts that connect adjacent elastic lamellae. In particular, diverse histological microstructures may lead to differential mechanical behavior during dissection, including the pressure–volume relationship of the injected fluid and the displacement field between adjacent lamellae. In this chapter, we develop a data-driven surrogate model of the delamination process for differential strut distributions using DeepONet, a new operator–regression neural network. This surrogate model is trained to predict the pressure–volume curve of the injected fluid and the damage progression within the wall given a spatial distribution of struts, with *in silico* data generated using a phase-field finite element model. The results show that DeepONet can provide accurate predictions for diverse strut distributions, indicating that this composite branch-trunk neural network can effectively extract the underlying functional relationship between distinctive microstructures and their mechanical properties. More broadly, DeepONet can facilitate surrogate model-based analyses to quantify biological variability, improve inverse design, and predict mechanical properties based on multi-modality experimental data.

2.2 Introduction

Aortic dissection, a cardiovascular condition resulting in high mortality, manifests within the medial layer of the aortic wall due to physical separation of the lamellar units (Fig. 2.1(a)). The normal medial microstructure in Fig. 2.1(b) reveals a sandwich structure wherein elastic lamellae delimit intralamellar media (cells and extracellular matrix) and are connected by interlamellar struts (fibrillins, elastin, and collagens) [191]. One possible outcome for an artery undergoing dissection is that the tear turns inward and forms a false lumen, while another possibility involves the tear turning outward and causing vessel rupture; the latter scenario can be lethal. Although most studies have focused on the stage with a developed false lumen, there has been little investigation of the initiation and propagation of aortic dissection from a mechanistic perspective. This knowledge gap prevents further predictive modeling endeavors.

There exist several hypotheses aiming at bridging this knowledge gap. One hypothesis states that dissection may arise from an intimal defect; that is, a defect within the luminal surface. Following this line, many insightful *in silico* works assessed the risk of a false lumen for further dissection [200] and investigated the stress distribution for an artery with flaps [215, 115, 206, 116, 185, 73, 168]. These works provided new insights into the propagation of a pre-existing defect, but did not address the initiation of aortic dissection. Another theory hypothesizes that a dissection may occur at or near a region with a concentrated stress field; for example, the subclavian branch of the aorta has higher stresses due to a sharp geometric and material discontinuity [65].

Additionally, it has been hypothesized and demonstrated that aortic dissec-

tion may arise from focal accumulations of glycosaminoglycans (GAGs) [102, 226], which are highly negatively charged and thus imbibe and sequester water. The combination of Gibbs-Donnan swelling pressure and internal stresses on the arterial wall can predispose to dissection, leading to a higher probability of rupture. Based on this concept, a series of *in silico* studies have used smoothed particle hydrodynamics [221, 4, 3] and standard finite element modeling [226, 227] to illustrate the role of accumulations of GAGs on the progression of dissection. Seminal work by Roach and colleagues focused instead on directly measuring the intramural (blood) pressure needed to propagate an initial medial defect within the thoracic aorta [91, 163, 36, 250, 266, 225], which provided further insights into dissection from *in vitro* experiments. They consistently found that static pressures needed to be 500 mmHg or more to dissect the normal wall and examined the mechanical properties of the porcine aorta at different locations by injecting fluid quasi-statically. Their studies have shed light on the mechanism and characteristics of GAG-induced dissection initiation and propagation.

Most recently, Ban et al. [11] developed a phase-field finite element model to investigate aortic dissection with a histologically motivated microstructure; specifically, they simulated studies by Roach and colleagues [225] focusing on the pressure-volume (P-V) curve associated with the intramural fluid that initiates and drives intramural damage. The realistic microstructure (Fig. 2.1(b)) is simplified and represented in the phase-field model; Fig. 2.1(c) shows the top view of an elastic strut distribution, which varies substantially from sample to sample. The variability in the elastin architecture leads to a corresponding difference in mechanical properties, dissection propensity, and severity from sample to sample and along the aorta (which is highest in the ascending thoracic aorta and lower in the abdominal aorta). Yu et al. [309] further showed numerically and experimen-

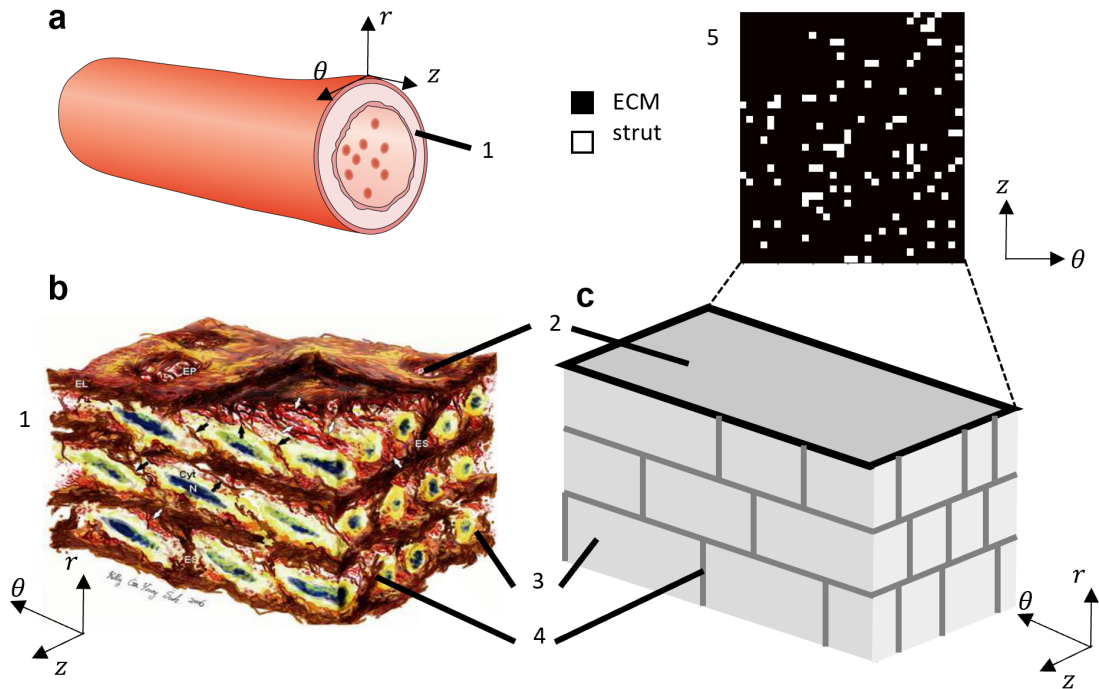


Figure 2.1: Sketch of the aortic wall and its microstructure. (a) The aortic wall consists of three layers: intima, media, and adventitia. (b) Artistic rendering of the lamellar microstructure of the aortic media by O'Connell et al. [191], reproduced with permission, plus (c) its representative volume element for the finite element simulation. 1: medial layer, 2: elastic lamellae, 3: lamellar media, 4: elastic (interlamellar) strut, 5: top view of the interlamellar struts (in white) shown as a strut map. z : axial, θ : circumferential, r : radial.

tally that the pressure drops induced by interlamellar damage to collagen follow a power-law behavior, indicating a possibility of predicting dissection based on the microstructure of the extracellular matrix. The power-law behavior was confirmed by Ban et al [11]. Conversely, Holzapfel and colleagues used a phase-field modeling approach to investigate the propagation of dissection in mode I tearing [256, 74, 243]. The computational cost associated with performing enough finite element simulations to empirically uncover precise relationships between tissue microstructure and dissection propagation is prohibitive, however. Hence, there is a critical need to develop a reliable surrogate model of the process by which dissection propagates within the aortic wall. Such a model will greatly facilitate further *in silico* investigations into this pathology, with the ultimate goal of developing computational tools to clinically assess and predict dissection propagation on a patient-specific basis, while also quantifying uncertainties associated with the resulting prognoses.

The potential of machine learning has grown rapidly in recent years, and the development of machine learning-based surrogate models for fast and accurate assessments has emerged as a key application of interest [27, 136, 210]. Importantly, physics-informed neural networks (PINNs) [133, 213] serve as one of the most promising models in scientific machine learning. PINNs penalize the residual of governing physics-based relations for the system in the loss function, where the partial derivatives are computed through automatic differentiation. This framework has been applied to predict flow fields [112, 166] and their variants [108], fracture progression [82], risk assessment [156, 143], and material property inference [305, 317, 154, 315] among many other applications [29, 118]. In biomedical engineering, PINNs have shown potential in cardiac activation mapping [232], inferring arterial boundary conditions [128], and inferring thrombus material prop-

erties [305] as summarized in [5]. Yet, for a system with different boundary/initial conditions, one has to retrain the network for each case, making the algorithm time-consuming. Hence, there is a pressing need to develop models that can learn the operator level of mapping between functions; that is, predicting the physical system under diverse boundary/initial conditions. Lu et al. [159] developed the data-driven framework “DeepONet,” an operator–regression neural network that learns the mapping between functions with theoretical guarantees. DeepONet has shown its effectiveness in predicting multiscale and multiphysics problems [31], bubble growth [145], and crack prediction in brittle materials [83]. Li et al. [139] developed a Fourier neural operator for learning parametric partial differential equations.

In this chapter, we consider the phase-field intralamellar damage model developed in [11] as a prototypical model to investigate dissection for a heterogeneous arterial wall. The mechanical process in the fluid-injection initiated arterial damage problem is well-characterized by its underlying P-V curve and damage progression. To the best of our knowledge, this is the first attempt to predict dissection progression and mechanical behavior in a heterogeneous aortic wall using scientific machine learning. We will demonstrate that details of the characteristic pressure drops can be well-characterized by incorporating the current damage field; otherwise, the model prediction will lead to a mean-field average without detailed pressure drops. Moreover, we investigate model performance in predicting the damage field given the observable displacement field.

The chapter is organized as follows. In section 2.3, we introduce the phase-field model that generates synthetic data, summarize a linear regression-based model used as a baseline comparison, and provide details of the DeepONet approach. In section 2.4, we introduce the data preparation, followed by the prediction results

of DeepONet for the corresponding P-V curve and damage field. Moreover, we evaluate the effect of network structure on testing errors. In section 2.5, we conclude by interpreting the results as well as discussing further applications of the presented model.

2.3 Methods

2.3.1 Phase Field Model

We employ a validated phase-field finite-element method [86] to simulate progressive damage in the native aortic wall. Unlike previous particle methods [221, 3], the phase-field model gives a continuum description of a discontinuous tear. The phase field variable $\phi(\mathbf{x}, t) \in [0, 1]$ satisfies

$$\phi(\mathbf{x}, t) = \begin{cases} 0, & \text{intact region,} \\ 1, & \text{fully torn region.} \end{cases} \quad (2.1)$$

We adopt the same phase-field model developed in [11] to describe the progression of dissection within the heterogeneous aortic media driven by a quasi-static injection of fluid. In modeling the delamination, we sought to find the displacement $\mathbf{u}$, a pressure-like Lagrange multiplier p that enforces near incompressibility of the normal wall tissue, a Lagrange multiplier m that controls fluid injection, and phase field ϕ by the minimization of total energy E_{total} ,

$$(\mathbf{u}_i, p_i, m_i, \phi_i) = \underset{\mathbf{u}, p, m, \phi}{\operatorname{argmin}} E_{\text{total}}(\mathbf{u}, p, m, \phi), \quad (2.2)$$

at each increment of injection, step i . The total energy E_{total} is expressed as

$$E_{\text{total}} = E_{\text{deform}} + E_{\text{tear}} + E_{\text{pressurized-fluid}}, \quad (2.3)$$

where these three terms represent the energy contributions from elastic deformation, tearing, and forced fluid injection within the medial layer.

The native aortic wall is modeled as a hyperelastic material, with the strain energy density function W_{wall} used to calculate effects of elastic wall displacement. Specifically, W_{wall} is modeled as a Fung exponential stiffening material [96, 227]:

$$W_{\text{wall}} = \frac{c}{2}(\lambda_1^2 + \lambda_2^2 + \lambda_3^2 - 3) + \sum_{i=1}^4 \frac{c_1^i}{4c_2^i} (e^{c_2^i[(\lambda^i)^2 - 1]^2} - 1), \quad (2.4)$$

where c , c_1^i , and c_2^i are material constants. Here, λ_1 , λ_2 , and λ_3 are principal stretch ratios and λ^i are stretch ratios in one of the four primary directions associated with the intramural fibrous constituents, mainly collagens ($i = 1, 2, 3$, and 4 represent axial, circumferential, and two diagonal directions, respectively). We use material properties reported in Ref. [228] based on previous biaxial experiments on human aortas [76]. The stiffness of the struts and the matrix differ by a factor of 20 such that their arithmetic mean yields the properties of the homogeneous wall.

An approach similar to the perturbed Lagrangian method is used to constrain the nearly incompressible condition of the vascular wall:

$$W_{\text{vol}} = -(1 - \phi)^3 p (\lambda_1 \lambda_2 \lambda_3 - 1) + \frac{p^2}{2\epsilon}, \quad (2.5)$$

where $p^2/(2\epsilon)$ serves as a regularization term with $\epsilon \gg 1$, and p is a Lagrange multiplier that enforces the incompressibility constraint that the determinant of

the deformation gradient equals unity in the non-damaged tissue. The cubic expression, $(1 - \phi)^3$, causes a rapid relaxation of the incompressibility condition in the damaged tissue [138], which ensures that incompressibility is not enforced over fully damaged tissue.

The total energy of deformation of the wall is:

$$E_{\text{deform}} = \int_V [((1 - \phi)^2 + \varepsilon)W_{\text{wall}} + W_{\text{vol}}] dV, \quad (2.6)$$

where the term $(1 - \phi)^2$, proposed in the [24], serves as a degrading function for the damage region, and $\varepsilon \ll 1$ serves as a regularization term.

The tearing energy can be expressed as [7, 24]:

$$E_{\text{tear}} = \frac{3G_c}{8} \int_V \left(\frac{\phi^2}{l} + l |\nabla \phi|^2 \right) dV, \quad (2.7)$$

with the critical energy release rate of tearing G_c set, respectively, as 5.77 or 0.20 J/m² for the radial struts and surrounding matrix to reproduce the pressure–volume curve in [225]. V is the volume of tissue. This approach enables a representation of the two-dimensional tear by the three-dimensional damage field, ϕ . The damage variable (ϕ) decays over a small length-scale, l , on the same order of magnitude as the mesh cell size. l was set as 0.1 μm in the model.

Following an initiation event, the dissection advances with the continued injection of pressurized fluid. We constrain over the total volume of the injected fluid by using a Lagrange multiplier m to prescribe the total volume of fluid, $V_{\text{injection}}$, with

$$E_{\text{pressurized-fluid}} = m \left(\int_V \phi \lambda_1 \lambda_2 \lambda_3 dV - V_{\text{injection}} \right). \quad (2.8)$$

Note that prescribing the pressure-load during the increase of the volume of injection would present a computational challenge. If pressure were prescribed as traction boundary conditions, then the oscillations of pressure after tearing may not be captured because of instabilities, that is, multiple volumes have the same pressure. Therefore, we introduce Eq. 2.8 to prescribe the volume of injection, $V_{\text{injection}}$, incrementally with m , a global Lagrange multiplier. The minimization of energy with respect to m allows the prescribed value of $V_{\text{injection}}$ to represent the damaged volume of the injection pool in the deformed configuration. Eq. 2.8 is a novel method to control the injection volume during the injection process.

We iteratively minimize the total energy at each loading step by taking the variational derivative in the direction of $\mathbf{u}$, p , m , and ϕ . A numerical viscous-like damping term was used to facilitate the numerical solution. We use the Bubnov–Galerkin method and the FEniCS [6] finite element package to solve the resulting nonlinear weak formulation.

2.3.2 Histologically Motivated Microstructure

Microstructures generated for the phase-field model are motivated by the realistic histology of the murine aortic wall. We present a 3D image of the wall acquired by multiphoton microscopy [38] to illustrate the sandwich-like structure wherein the medial layer is composed, on average, of 5-6 layers of elastic lamellae indicated by the green pixels in Fig. 2.2. In Fig.2.2(a), we show a subvolume of reconstructed murine aorta under *ex vivo* biaxial loading equivalent to physiological conditions. We compare the cross-section of an artificial-generated microstructure used in the phase-field simulations with a real aortic wall in (b), showing a qualitative consistency between the two geometries. Different components of the arterial

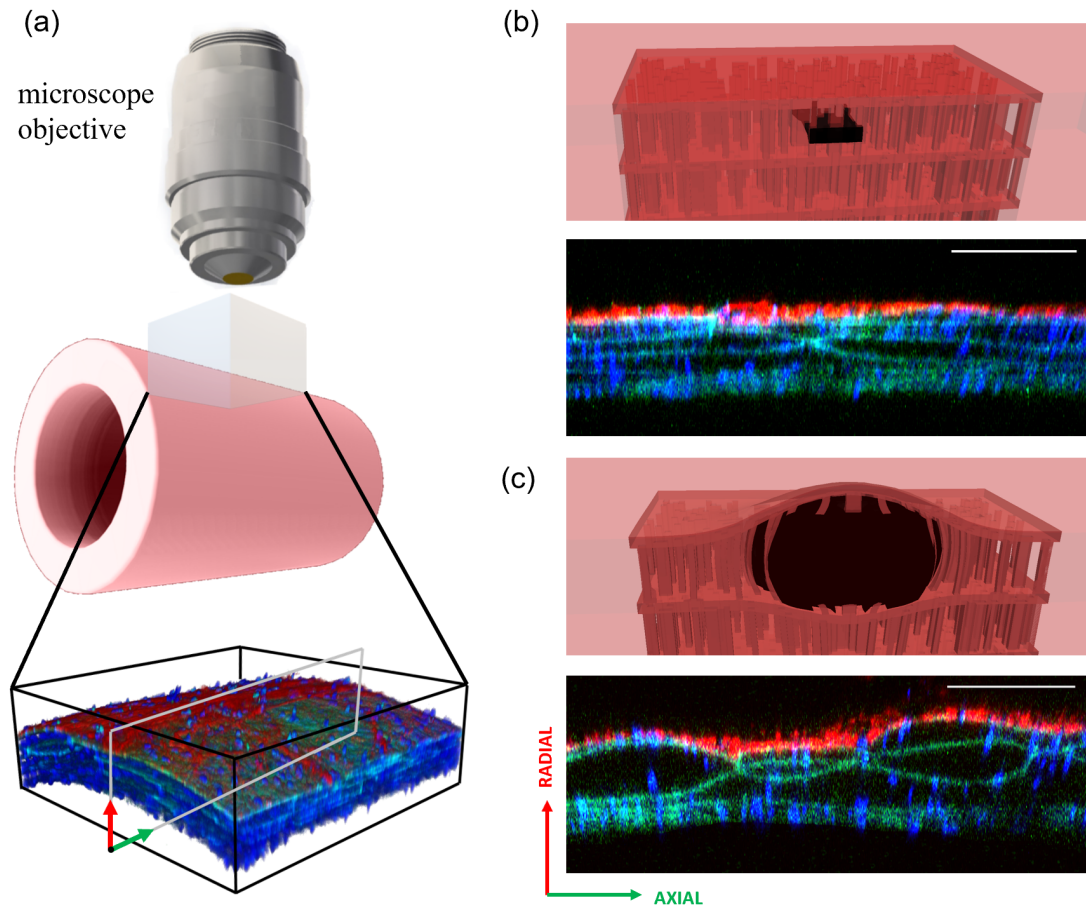


Figure 2.2: Histologically motivated microstructure of the murine aortic wall under biaxial loading. (a) 3D imaging of the murine arterial wall by multiphoton microscopy. Aortic wall (b) initiating and (c) propagating intralamellar delaminations, including histologically motivated microstructure for the phase-field model. Green: elastic lamellae. Blue: Smooth muscle cell nuclei. Red: Collagen in the adventitia. Scale bar: 100 μm .

wall are distinguished by colors: adventitial collagen is denoted by red, elastic lamellae by green, and smooth muscle cell nuclei by blue.

Admittedly, we notice that the real geometry is not homogeneous in the sense that the elastic lamellae in the phase-field geometry can be clearly identified. Nonetheless, it is still representative in that the strut, matrix, and lamellae structures are preserved. In Fig. 2.2(c), we compare an ongoing aortic dissection between the phase-field model and real arterial wall, where a visible false lumen can be observed from both images.

2.3.3 Linear Model

First, we consider a linear regression model to predict P-V data from an elastin strut map, which will serve as a (low-fidelity) baseline model for comparison with our higher-fidelity DeepONet results. In this approach, the inner product between the array of tissue microstructure $\mathbf{s}$, where $s_{ij} = 0$ indicates non-elastin matrix and $s_{ij} = 1$ indicates the presence of an elastin strut, and a coefficient array a_{ijk} was used to predict the observed pressure P as a function of injection volume V . Specifically, at a particular injection volume V_k ,

$$\begin{aligned}
 P_k - \bar{P}_k &= \sum_i \sum_j a_{ijk} (s_{ij} - \bar{s}_{ij}) + \varepsilon_k \\
 \text{for } \bar{P}_k &= \mathbb{E}[P_k] \approx \frac{1}{N} \sum_{n=1}^N (P_k)_n \\
 \text{and } \bar{s}_{ij} &= \mathbb{E}[s_{ij}] \approx \frac{1}{N} \sum_{n=1}^N (s_{ij})_n,
 \end{aligned} \tag{2.9}$$

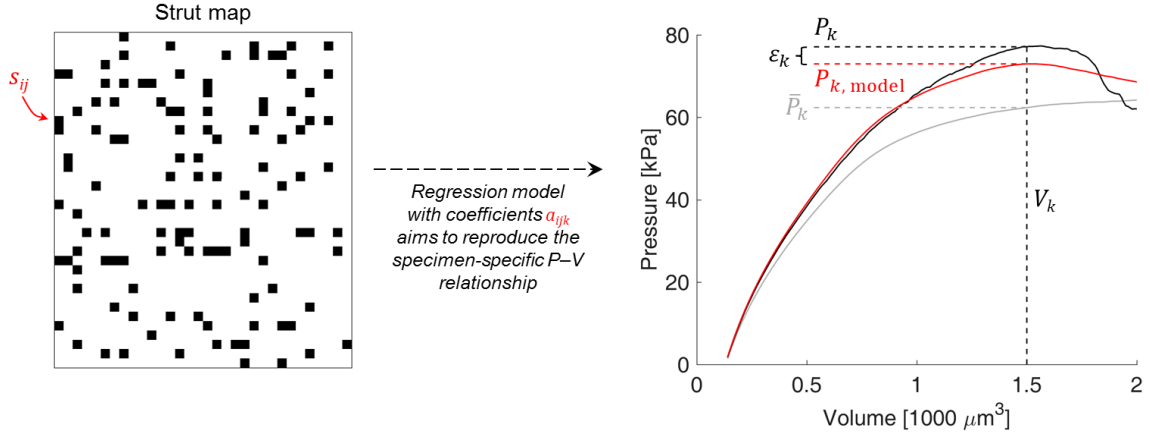


Figure 2.3: Summary of the linear P-V model. At each volume V_k , the corresponding pressure P_k is predicted from the elastin strut map s_{ij} and a set of volume-specific linear regression coefficients a_{ijk} . Elastin struts ($s_{ij} = 1$) are displayed in black, while non-elastin matrix ($s_{ij} = 0$) pixels are displayed in white.

where $\mathbb{E}[\cdot]$ denotes the expected (mean) value. $\bar{P}_k$ and $\bar{s}_{ij}$ are the expected values of the pressure and the tissue microstructure indicator function, respectively, each estimated from the sample means of the training data; n is the training sample index, N is the training data sample size, and $\varepsilon_k \sim \mathcal{N}(0, \sigma_k^2)$ is the pressure residual unexplained by a linear function of the strut map input (Fig. 2.3). The model coefficients a_{ijk} were determined directly via least-squares linear regressions of the volume-specific P-V data generated from the phase-field model. Detailed statistical analysis to establish the approximate normality of $P_k(V_k)$ as well as examination of model performance are presented in Supplemental Material, where a second low-fidelity (logistic regression) model is found as well.

2.3.4 DeepONet

Next, consider the general architecture of DeepONet; we refer the reader to [159] for more details. DeepONet was proposed for learning nonlinear operators by mapping input functions into corresponding output functions. Let G be a non-

linear operator taking an input function u and yielding the output function $G(u)$. Evaluation of the function $G(u)$ at $y \in \mathbb{R}^n$ is a real number and can be denoted as $G(u)(y)$, where y is typically a vector containing coordinates and other information. In practice, the input function u is represented discretely by its values $[u(x_1), u(x_2), \dots, u(x_m)]^T$ at a finite set of locations $\{x_i\}_{i=1}^m$. The architecture of the (unstacked) DeepONet is shown in Fig. 2.4(a). The branch net and the trunk net take the function u (in the form of $[u(x_1), u(x_2), \dots, u(x_m)]^T$) and the vector y as inputs, and yield intermediate outputs $[b_1, b_2, \dots, b_p]^T$ and $[t_1, t_2, \dots, t_p]^T$, respectively. In the last layer, DeepONet merges these two outputs by taking their pointwise multiplication:

$$G(u)(y) \approx \sum_{k=1}^p b_k t_k + b_0, \quad (2.10)$$

where we include an additional bias term $b_0 \in \mathbb{R}$ as a trainable variable as it may reduce the generalization error. With such a setup, the i -th sample from the training/testing dataset gives a triplet $[u^{(i)}, y, G(u^{(i)})]$ (branch input, trunk input, and output) with the following structure

$$\begin{bmatrix} u^{(i)}(x_1) & u^{(i)}(x_2) & \cdots & u^{(i)}(x_m) \\ u^{(i)}(x_1) & u^{(i)}(x_2) & \cdots & u^{(i)}(x_m) \\ \vdots & \vdots & \ddots & \vdots \\ u^{(i)}(x_1) & u^{(i)}(x_2) & \cdots & u^{(i)}(x_m) \end{bmatrix}, \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \begin{bmatrix} G(u^{(i)})(y_1) \\ G(u^{(i)})(y_2) \\ \vdots \\ G(u^{(i)})(y_N) \end{bmatrix}, \quad (2.11)$$

where evaluation of the function $u^{(i)}$ at $[x_1, x_2, \dots, x_m]$ is copied N times to match the evaluation of $G(u)$ at points $y_j \in \mathbb{R}^n$ ($j = 1, 2, \dots, N$). The dimension of each term on the right hand side is (N, m) , (N, n) , and $(N, 1)$.

We develop two DeepONets (*Net D* and *Net P-V* in Fig. 2.4(b)) for two different tasks: predicting damage progression and corresponding pressure–volume

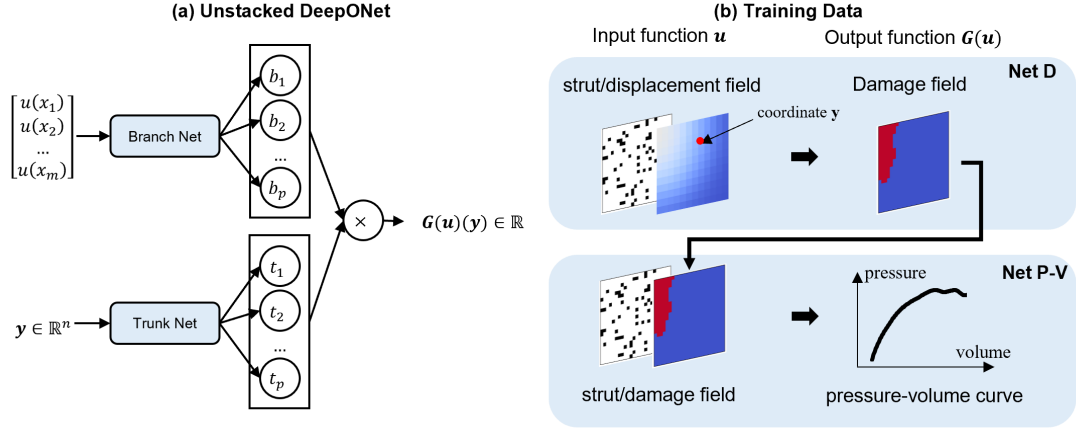


Figure 2.4: A schematic of DeepONet for fluid injection-induced dissection problems and its training data. (a) An unstacked DeepONet for learning an operator $G : u \rightarrow G(u)$ that takes two inputs, $[u(x_1), u(x_2), \dots, u(x_m)]$ and y , and returns the scalar $G(u)(y)$. The trunk net is a fully-connected neural network (FNN) whereas the architecture of the branch net varies depending on the problem. (b) Training data for the problem. First, we train *Net D* whose input u contains the elastic struts map and the current surface displacement map to predict the damage field at the current state. Next, the predicted damage field is taken as an input for *Net P-V* for predicting the corresponding pressure-volume curve for the injection fluid. Each network is trained separately and coupled for prediction.

curves. The architecture of the branch net and the trunk net is selected to adapt the structure of their input data. In this chapter, the inputs of the branch net ($[u(x_1), u(x_2), \dots, u(x_m)]^T$) are a stack of 2D images indicating in-plane fields of physical quantities, including the strut distributions, damage progression, and displacement profiles depending on specific tasks. In view of this, we mainly adopt a convolutional neural network (CNN), an efficient architecture for processing images, as the architecture of the branch net. A fully-connected neural network (FNN) is also considered for the branch net for comparison. For the trunk net, we always choose the FNN model due to the small dimensions (no more than two) of the input y , which can be the in-plane spatial coordinates or the injected fluid volume depending on specific tasks. We will provide more details regarding the definition of the tasks in section 2.4.

We adopt the mean squared error (MSE) to measure the discrepancy between

the model prediction and the data. To avoid overfitting and to improve the generalization capability of the model, we introduce a L_2 regularization term in the loss function, which penalizes large values of weights and biases in the trunk net and the final output layer. Therefore, the total loss $\mathcal{L}$ can be expressed as

$$\mathcal{L} = \frac{1}{MN} \sum_{i,j=1}^{M,N} (G_{\text{model}}(u^{(i)})(y_j) - G_{\text{data}}(u^{(i)})(y_j))^2 + \alpha \sum_{i=1}^K \xi_i^2, \quad (2.12)$$

where M is the size of the dataset, G_{model} is the operator learned by the DeepONet, G_{data} is the target operator, ξ_i ($i = 1, 2, \dots, K$) is the i -th trainable parameter of the trunk net and the output layer, and α is the weight of regularization (10^{-7} or 10^{-6} in this chapter [324, 44]; see also sec. S1).

2.4 Results

2.4.1 Data Preparation

The phase-field finite element model provided synthetic training data for the DeepONet. Motivated by the multiphoton images, the phase-field model consisted of several elastic lamellae with intralamellar material to model the medial layer (Fig. 2.5). Medial delamination was driven by injecting pressurized fluid between two consecutive elastic sheets, with the fluid injected quasi-statically. Initially, increments of fluid volume elastically deform the wall, causing the top-most elastin sheet to bulge. At a critical pressure, the matrix began to tear while the radial struts sustained the fluid pressure. The struts also resisted fluid propagation both by keeping the adjacent sheets together and by acting as strong barriers. As the injected fluid volume increased, however, a few of the struts snapped, result-

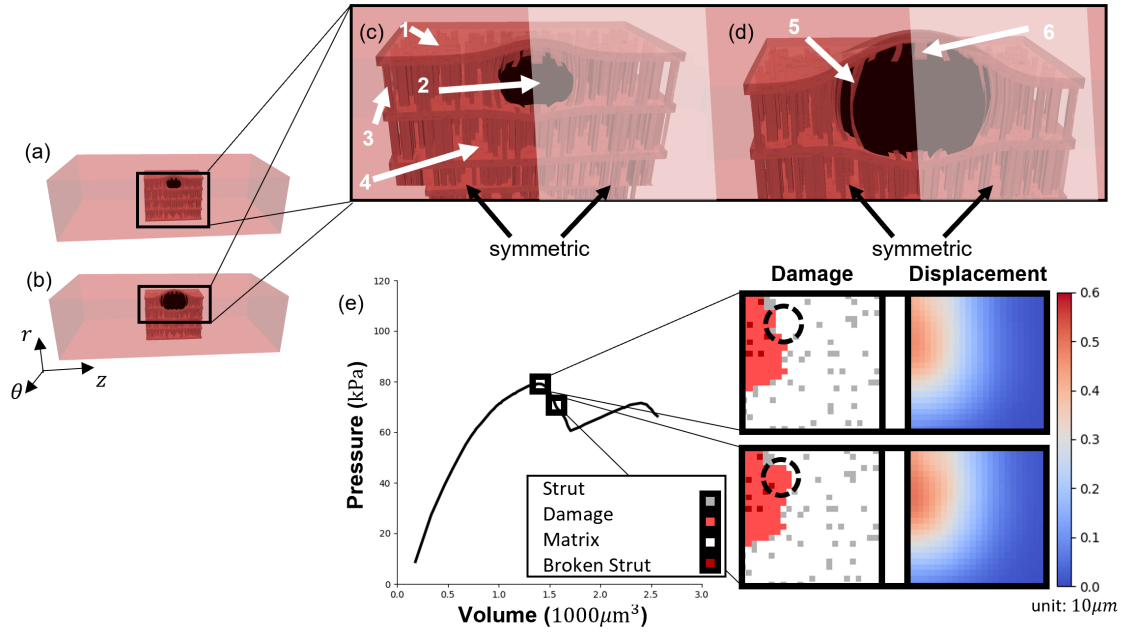


Figure 2.5: 3D snapshots for injection-induced delamination in the phase-field model at (a) an early stage and (b) a progressed stage. Note that (c, d) show enlarged views with details: (1) elastic lamellae, (2) damage region, (3) elastic struts, (4) intralamellar media, (5) a strut being torn, and (6) a broken strut. (e) Illustration of a pressure-volume response and its corresponding dissection state. A pressure drop results from irreversible tearing of the elastic struts and matrix indicated in the circle, with their corresponding displacement fields on the right. The computational geometry is symmetric with respect to the transparent part in (c) and (d), hence only half of the damage and displacement fields (for the transparent right side) are shown for illustration, with grey pixels showing struts and red indicating damage or high displacement.

ing in a sudden drop of pressure shown in Fig. 2.5(e). Given the experimentally observed stochastic distribution of the position of struts, this process results in a stochastic variation of pressure in response to the injection of intramural fluid [11].

To acquire the requisite training and testing datasets for DeepONet, we generated 2,100 phase-field solutions, each with a different stochastic distribution of strut position. In each case, the outputs from the finite element model include the P-V curve and two planar fields; the displacement field records the rising of the innermost medial elastic lamina and the damage field marks the tearing of tissue as the fluid progresses. Averaging over a section in the intralamellar space, where the fluid is injected, produces pixel-wise maps based on the finite element solution (Fig. 2.5). The injected volume in each case varies from $1.3 \times 10^3 \mu\text{m}^3$ to $3.5 \times 10^3 \mu\text{m}^3$.

The computational cost for training a DeepONet with 1900 training samples is approximately 620 min/1000 epochs on a NVIDIA V100 GPU. The prediction cost is trivial compared with the cost of training, which is less than 1s. Each FEA simulation was run over about 12 hours (wall-clock) using a single computing node with 20 processors. To ensure robustness and reliability, we routinely train a network 5 times for each case with different parameter initialization.

2.4.2 Damage Field Prediction (*Net D*)

Although the displacement field can be experimentally observed *in vitro* [17], the damage field is a hidden quantity that is difficult to observe directly. In this section, we infer the current damage field based on the displacement field using DeepONet (see also Fig. S2 - S5 in Supplemental Material). The triplet of the training dataset

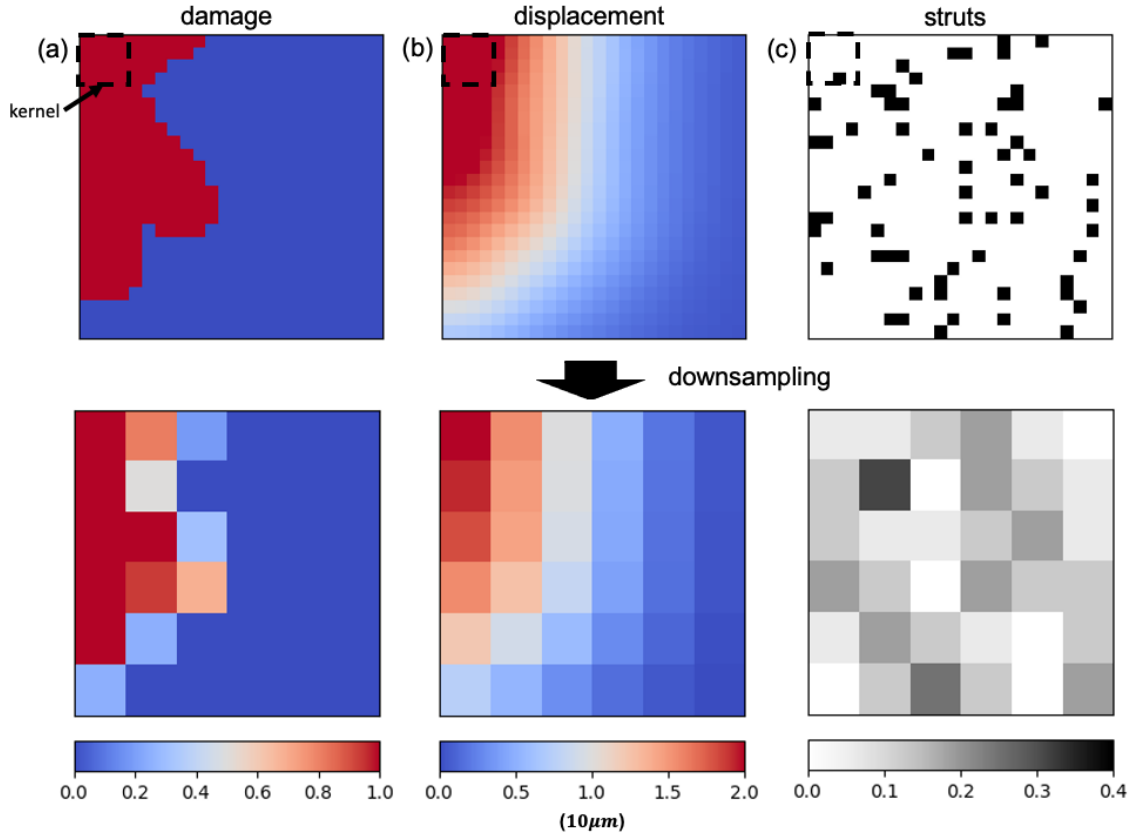


Figure 2.6: Image downsampling. Damage, displacement, and strut images can be downsampled by taking the average within a 4×4 kernel (yielding 2×2 image domains for the linear regression model, 6×6 image domains for the DeepONet model). Note that the physical meaning for each field after downsampling is: (a) the percentage of damage within a pixel, (b) mean displacement, and (c) percentage of strut within a pixel.

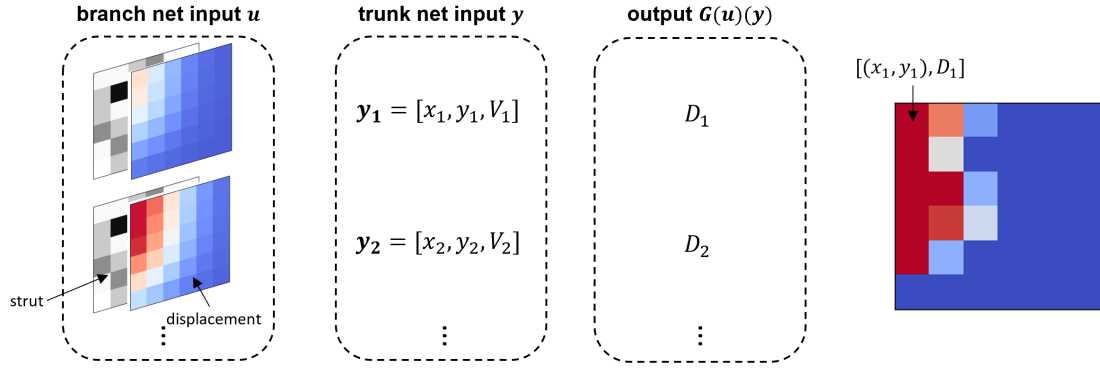


Figure 2.7: Triplets of a single training case for predicting damage progression. A dataset triplet is composed of images $\mathbf{u}$ (strut and displacement fields), a vector $\mathbf{y}$ containing pixel coordinates x_i, y_j and volume of injection V_k , and the damage percentage D_i .

Branch Net	Channels	Trunk Net	Width
CNN (small)	[4, 8, 12]	FNN (small)	[50, 50, 50]
CNN (large)	[10, 25, 50]	FNN (large)	[100, 100, 100]

Table 2.1: Model architectures for predicting damage progression. The trunk net is chosen as a fully-connected neural network (FNN) with three hidden layers while the branch net is a convolutional neural network (CNN) with three blocks. Each block contains a convolutional layer, a LeakyReLU layer, and a batch normalization layer.

is presented in Fig. 2.7 where the input and output are listed within. The trunk net input y is a concatenation of pixel coordinates (x_i, y_j) and injection volume V_k at the k^{th} injection step with the output $G(u)(y)$, the percentage of damage at that pixel. To evaluate model performance, we use 1,900 finite-element simulations as the training data and 200 simulations for testing, where each case contains 20 to 100 injection steps. Each simulation contains a different microstructure with initial damage imposed at one boundary. As fluid is quasi-statically injected into the medial layer, the corresponding damage progression is distinctive due to the difference in the strut map. We list the model architecture in Table 2.1.

First, we qualitatively show the ability of DeepONet to predict the damage progression in Fig. 2.8(a); compare the model prediction and true damage data

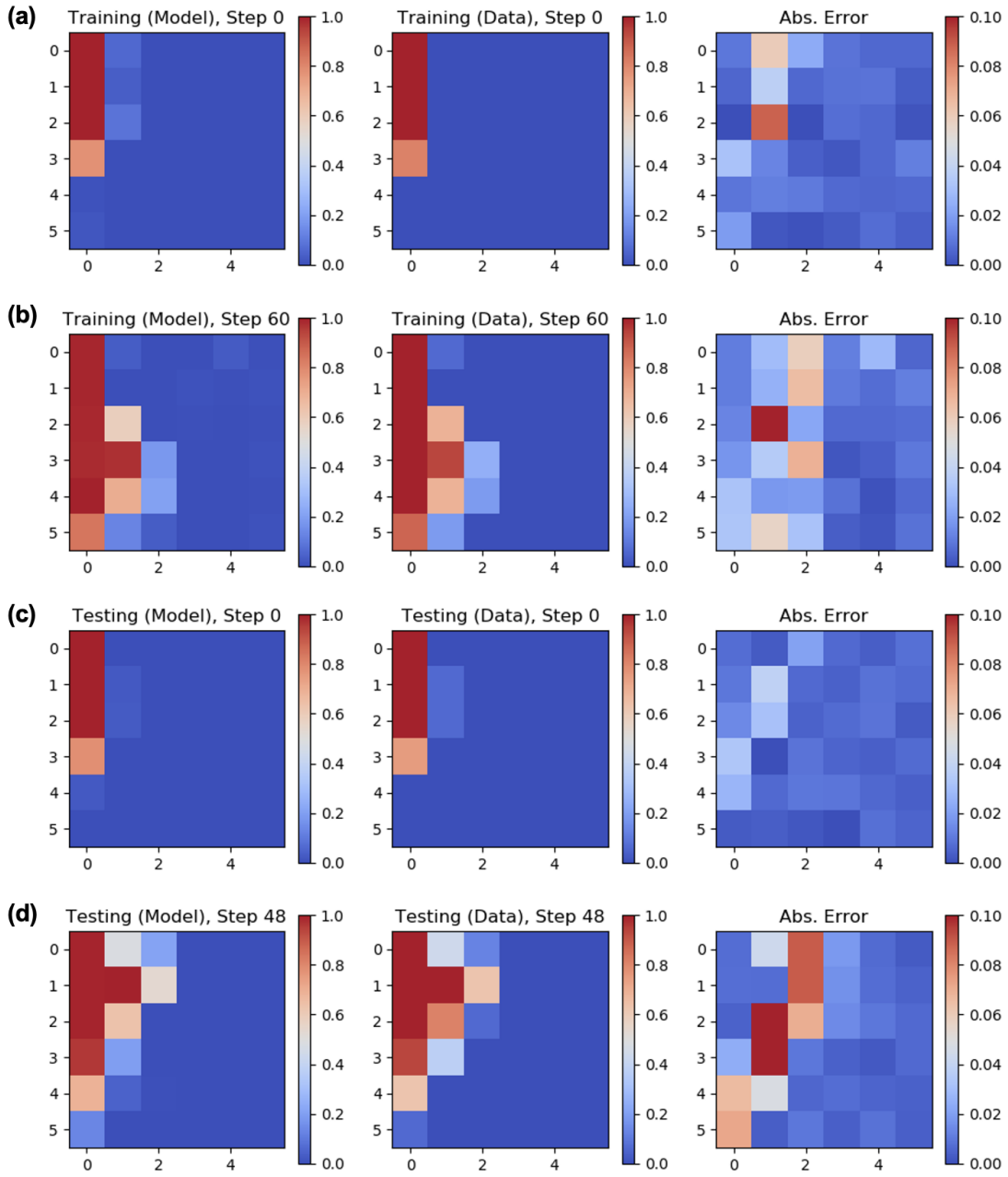


Figure 2.8: Training/Testing results of DeepONet for predicting damage progression. Model predictions and true damage fields are shown in the first and second columns, while the third column presents their absolute difference. (a, b) show the results for a training case at steps 0 and 60, while (c, d) represent a testing case at steps 0 and 48. The damage field indicates the percentage of torn material in the region at a particular pixel.

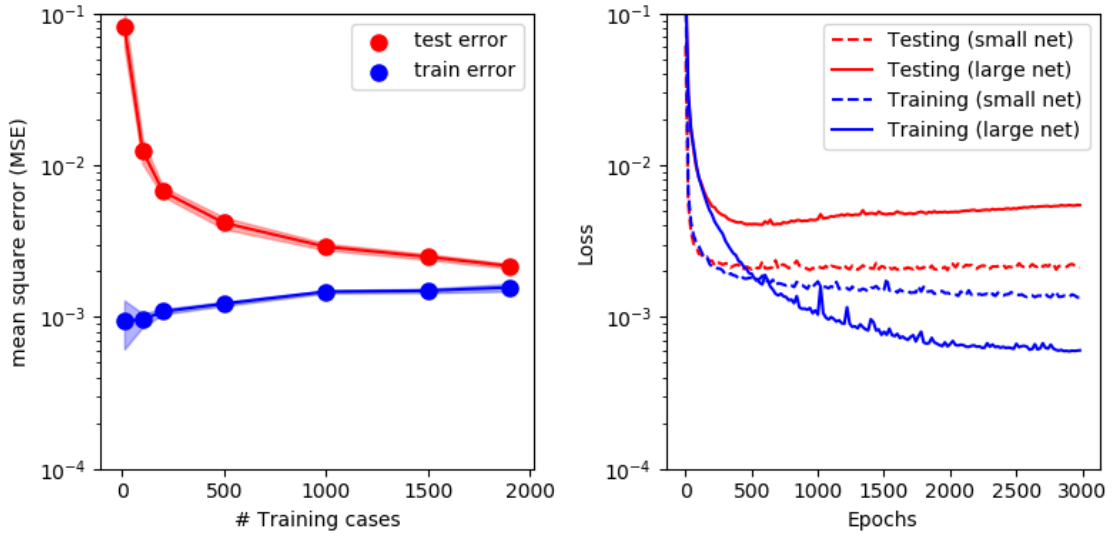


Figure 2.9: Errors for predicting damage fields against the number of training cases and epochs. The network architecture is shown in Table 2.1. The shaded area indicates the one standard deviation averaged from 5 runs with different initiation.

at the initial stage with the absolute error plotted in the third column. A later injection stage is plotted in Fig. 2.8(b). The largest absolute error for damage field prediction is below 10% at the damage tip. In (c, d), we compare the inference results for the damage field from a testing case. We again find an overall good agreement between model prediction and the true damage field at the initial step and last injection step (48). In general, the model predictions agree well with the simulations, with smaller errors at the initial loading step and larger errors at later steps.

Furthermore, we investigate the influence of the number of training cases on the prediction errors by training a network with a fixed architecture with different data. The number of training cases increases from 10 to 1,900, causing the testing error to drop from $O(10^{-1})$ to $O(10^{-3})$ with the training error increasing slightly as shown in Fig. 2.9(left). The values of testing error quickly converge to that of the training, indicating an improvement in performance with increasing training cases. Fig. 2.9(b) plots the errors against training epochs. We compare a

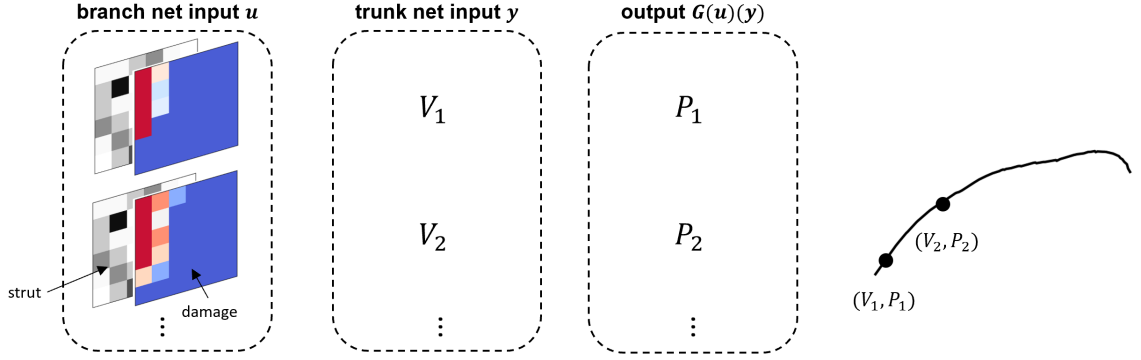


Figure 2.10: Triplets of a single training case for the P-V response. A dataset triplet is composed of images $\mathbf{u}$ (strut and damage fields), volume of injection V_k , and its corresponding pressure P_k .

large network structure with three hidden layers, each with 100 neurons, and a small network structure with three hidden layers and 50 neurons (Table 2.1). By employing the small network, the testing error ends up at around 0.001 compared with 0.004 from the larger network, indicating that the small network is better than the large network.

2.4.3 Pressure–Volume Curve (*Net P-V*)

In this section, we investigate the performance of DeepONet in predicting the associated P-V curves generated by the phase-field finite element model for many different microstructures (see Fig. S6 - S7 in Supplemental Material for an analysis of these synthetic data), which serves as the second part of the overall framework, based on model predictions of *Net D*. To train such a network, however, we use the true damage field from the data as training data while incorporating the predicted damage field in our previous experiments (see Sec. 2.4.2). The training dataset is a triplet with a structure listed in Fig. 2.10. For computational convenience, the microstructure of a heterogeneous wall and current damage fields are represented by down-sampled 6×6 images as the input of the branch net. The trunk net input

y is the injected volume V_k with the output $G(u)(y)$ for the corresponding fluid pressure P_k at k^{th} injection step. We use the same training and testing data in Net D to evaluate the model performance.

First, we quantitatively investigate how the performance is influenced by the branch net architecture. We choose the branch net as either a FNN or a CNN trained with/without the damage field, respectively, with the detailed structure listed in Table 2.2. We fix the trunk net with a structure $[1, 50, 50, 50]$. We plot the errors and deviations for the different network architectures for the P-V curve in Fig. 2.11(a) and observe that errors associated with either CNN or FNN without the damage field are significantly larger than those with the damage field. In particular, the testing error is almost an order lower than for the case without damage data, at $O(10^{-3})$. Figs. 2.11(b) and (c) show the training loss and testing loss versus the number of training cases. The testing error reduces from $O(10^{-2})$ to $O(10^{-4})$ with the number of training cases increasing from 10 to 1,900. The error bars show one standard deviation from 5 runs with different initiation. The results indicate that the DeepONet with FNN as the branch net has a slightly smaller training and testing error than that of CNN. Moreover, we plot the training and testing loss for FNN and CNN trained with/without information on the damage fields. The number of training and testing cases is again 1900 and 200, respectively. Both the training and testing errors with damage fields are smaller than the training/testing errors without the damage field. In addition, incorporating damage progression can avoid overfitting, which is pronounced in the testing error history shown in (d).

Then, we compare the model prediction and the phase-field model results (black lines) in Fig. 2.12. The branch net is a FNN with structure $[64, 50, 50, 50]$ while the trunk net is a FNN with network structure $[1, 50, 50, 50]$. We

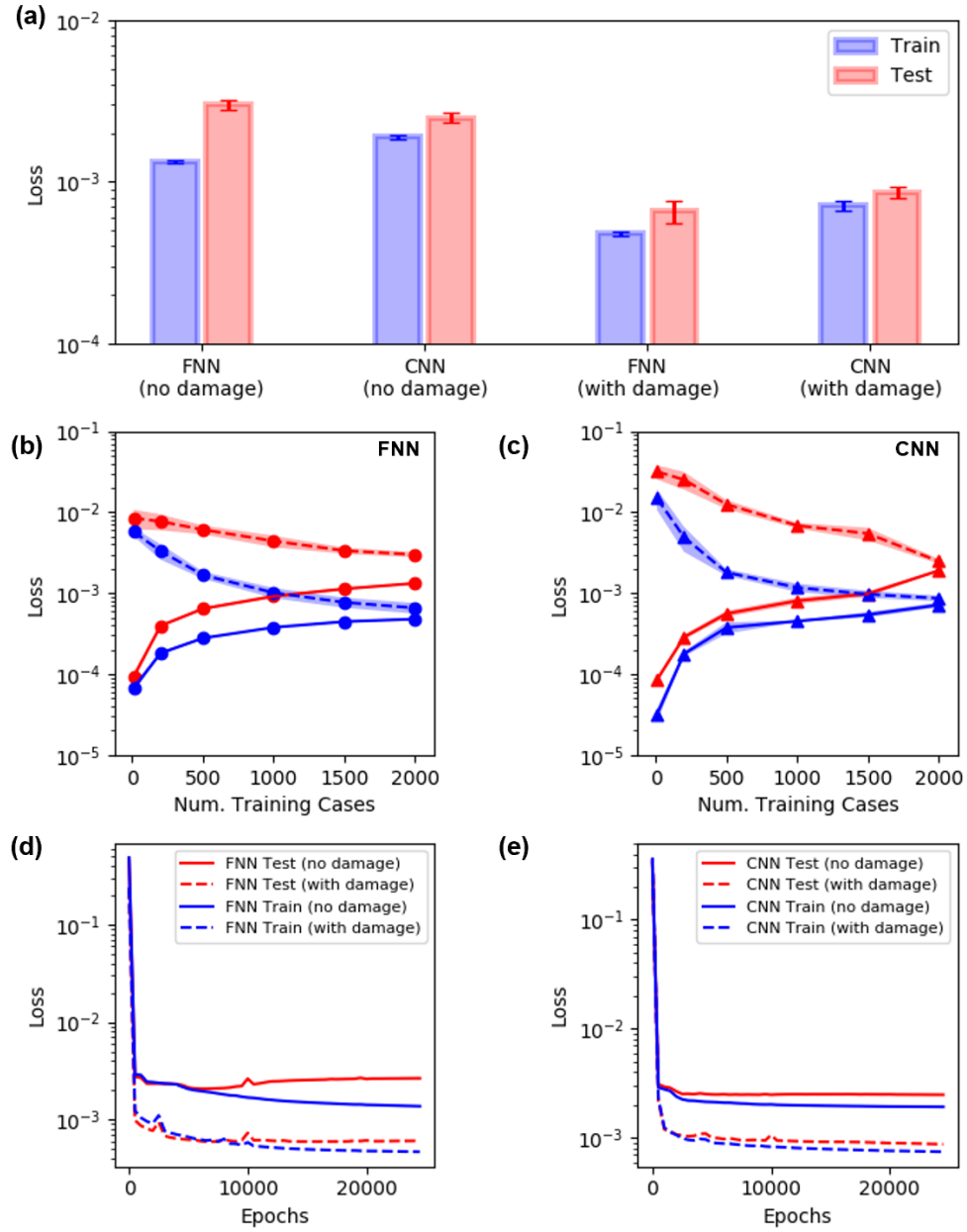


Figure 2.11: Learning the pressure-volume curve using different input data and trunk network architectures. (a) Errors and deviations of different network architectures trained to learn the pressure-volume curve. The trunk net is chosen as a fully-connected network (FNN) for each run, while the branch net is chosen as a convolutional neural network (CNN) or a FNN, respectively. The networks trained with damage fields have a lower testing error than those without, given the same training cases. The error bars show one standard deviation from 5 runs with different initiation. (b, c) The training and testing errors are plotted against the number of training cases. FNN and CNN (trunk net) trained with damage fields are plotted in blue, while those without damage fields are plotted in red. (d, e) Using the same color style, the losses vs training epochs are plotted for FNN and CNN, respectively.

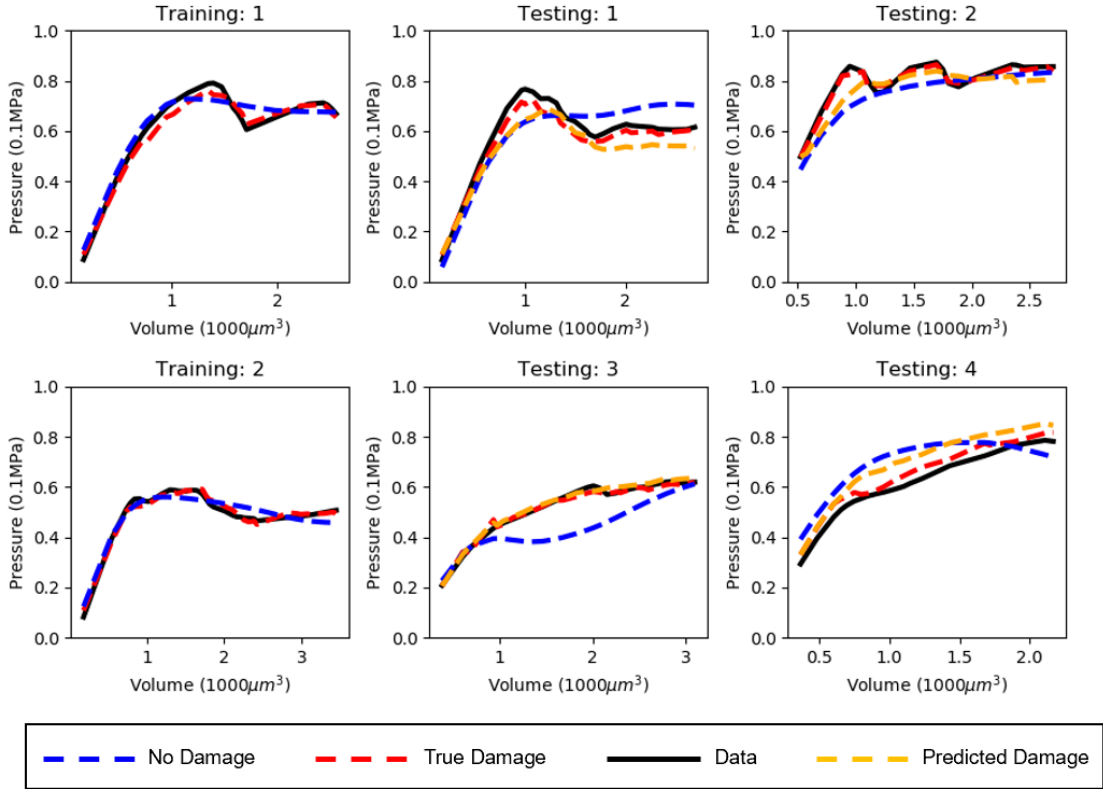


Figure 2.12: DeepONet for predicting P-V responses. Model predictions (dashed lines) for training cases 1 and 2 are compared with the true data (black line, first column). In particular, model predictions with and without damage fields are plotted in red and blue, respectively. The middle and right columns compare the testing data with models trained with (red)/without (blue) the current damage field. Notice that the orange line represents the predicted P-V curves based on the prediction of *Net D*. The branch net is a FNN trained with 1900 training cases. The true data were generated from the phase-field finite element solver. Note that the pressure values are normalized to $O(1)$ by dividing by 100.

Branch Net	Width	Trunk Net	Width
FNN	[50, 50, 50]	FNN	[50, 50, 50]
CNN	[4, 8, 12]	FNN	[50, 50, 50]

Table 2.2: Model architecture for the P-V curve. The trunk net is a fully-connected neural network (FNN) with three hidden layers. The branch net can be a FNN or a convolutional neural network (CNN) with three blocks, where each block contains a convolutional layer, a LeakyReLU layer, and a batch normalization layer. [4, 8, 12] indicates the number of channels in each block.

compare the model predictions from DeepONet models trained with different types of data: one with the true/predicted damage field and the other one without. It is not practical to have the true damage field as input since it is a hidden quantity, thus we employ the predicted damage fields from *Net D* as the input in this model. Qualitatively, the one without the damage field (blue line) can only learn the overall trend of a P-V curve both for training and testing cases, whereas the detailed pressure drop feature cannot be captured in the absence of the damage field in training. However, with information from the damage field, the overall performance improves significantly, shown via the orange and red dash lines. As mentioned above, the true damage is currently a hidden quantity whereas only the predicted damage progression is available in practice. Hence, we plot the predicted P-V curve based on the true and predicted damage fields for further comparison. Although predictions based on the true data slightly outperform the predicted damage field, both capture the detailed underlying mechanics by matching better with the phase-field data, which implicitly indicates that a learned relationship exists between the structure and its corresponding mechanical property.

In Table 2.3 we present all errors for the different models, including the low-fidelity linear regression model (see Supplemental Material). In training, the L_2^2 error for DeepONet with the damage field in the input is 5.22×10^{-4} and the

Training Errors	L_2^2 Error	Relative Error
DeepONet(Damage)	5.22×10^{-4}	3.10%
DeepONet(No Damage)	1.08×10^{-3}	4.15%
Linear Model	2.26×10^{-3}	6.48%
Testing Errors		
DeepONet(True Damage)	7.55×10^{-4}	3.70%
DeepONet(Predicted Damage)	1.37×10^{-3}	4.88%
DeepONet(No Damage)	4.36×10^{-3}	8.62%
Linear Model	2.47×10^{-3}	6.67%

Table 2.3: Error Table for predicting P-V curves using different models, both high (DeepONet) and low (linear regression) fidelity. See also Fig. S8 - S11.

relative error is approximately at 3%. As baseline models, the DeepONet model without damage and the linear model are trained using the microstructure alone and P-V curves. Their L_2^2 error is significantly higher, on the order of $O(10^{-3})$, with relative errors of 4% and 6%, respectively. As for the testing errors, the DeepONet model using the true damage field has the lowest error at 7.55×10^{-4} (3.7%). The DeepONet model using predicted damage fields has a relatively low error at 1.37×10^{-3} (4.88%), exhibiting its practicality in a more realistic scenario. The baseline models, however, have the largest error with the relative error reaching 8.62% and 6.67%.

2.5 Discussion

In this chapter, we demonstrate the potential of DeepONet for predicting the mechanical behavior of a heterogeneous aortic wall under dissection, here as reflected by the P-V curve and damage field in the case of a pressure-driven

injection of fluid within the medial layer; we use over 2000 phase-field finite element simulations as synthetic data. The model leverages recent advances in the operator learning model DeepONet. The predicted P-V curves agree well with the reference data produced by the phase-field finite element model. The damage field can be inferred based on the observable displacement field and the initial distribution of interlamellar radial struts, suggesting a potential approach to estimate the damage field from limited imaging data. When including the inferred damage field, the predicted P-V curve has a smaller error than the baseline models. We also investigated the network structure and its impact on model generalization: a network with a smaller size tends to have a lower testing error.

Practically, due to the extremely complicated mechanics, pure data-driven modeling without information on intralamellar damage progression would lead to an inaccurate inference of the P-V curve. A possible mitigation option will be incorporating the underlying physics into the network, which transforms the framework to physics-informed DeepONet [275, 83]. In addition, our *in silico* simulations only show dissection progression within the medial layer, without developing in the radial direction. Although not done, we could also adopt an encoder-decoder framework [9, 106] as a potential replacement of Net D. However, the advantage of DeepONet is the introduction of a trunk net, which makes the network capable of predicting continuous P-V curves. Moreover, DeepONet is more flexible in that its prediction can be an image or a 1D function as shown in Net P-V and Net D.

An actual aortic delamination may yield three distinct outcomes: it may turn inward to form a false lumen or re-entry site, turn outward resulting in rupture, or stabilize, with possible subsequent healing or reinitiation. One reason that our model did not yield these different scenarios is our oversimplification of the

real elastic lamellae structure: at this stage, our computational model only includes a minimal set of essential features required for reproducing the stepwise progressive tearing behavior, illustrating the possibility of reproducing the experimentally observed stepwise increases of pressure by the presence of radial struts [225]. Nonetheless, in future works, more detailed models may be developed, for instance, by accounting for the angular distribution of the orientation of fibers [268] or introducing actual heterogeneities in mural structure and properties [310], which may be inferred from imaging data, or by computationally accounting for tissue degeneration [78].

Some multiscale models represent the crack tip region using a finely discretized model with a coarse macroscopic model for the bulk region [239, 313]. Nonetheless, few works use multiscale modeling to study hyperelastic material tearing by injection. Certainly, multiscale modeling is useful for elucidating microstructural damage and deformation for dissection. Another possible improvement would be to incorporate this model into a probabilistic framework, where uncertainty is modeled by predicting the mean and variance of the quantity of interest while also accounting for uncertainties in the tissue mechanical properties [286, 223]. It is a natural framework in terms of describing a complicated bio-system with variability. Another promising extension of this work would be feature extraction and identification enabled by machine learning. From the phase-field simulations, we can generate a correspondence between dissection progression and microstructure. It would then be possible to learn the relation between local structures and their contribution to the mechanical properties, which enables a more interpretable machine learning model.

Author Contributions

J.D.H. and G.E.K. designed the research; M.Y., E.B., B.V.R., E.Z., and C.C. performed research; M.Y., E.B., B.V.R., and E.Z. analysed data; M.Y., E.B., B.V.R., E.Z., C.C., J.D.H., and G.E.K. wrote the paper

Ethics

This article does not present research with ethical considerations

Data Accessibility

Code is available from authors upon request.

Funding Statement

The work was supported by grant U01 HL142518 from the National Institutes of Health.

Acknowledgment

M.Y. would like to give a special thank to Ms. Xue-Fei Lin for her artistic support.

2.A Additional Details

We trained both Net P-V and Net D with different regularization parameters. As shown in Fig. 2.13, we found that Net P-V and Net D have a lower testing error and higher generalization when $\alpha = 10^{-6}$ and 10^{-7} , respectively (see Eq. 12).

2.B Prediction of Displacement Fields

In this section, we present a network for predicting the displacement field based on the damage field at the current step. We aim to show the capability of the network to infer the displacement field given experimental measurements. The triplet of the training dataset is similar to that in Fig. 7 (main text) whereas the displacement and damage fields are taken as the output and input, respectively. We use the same simulations as in Sec. 2.4.3 as training data.

We show the performance of DeepONet in predicting the displacement fields in Fig. 2.14, where we plot the predictions in a similar fashion. The training data contains 1,900 cases with 200 additional cases for testing. We trained the model for 3,000 epochs on a single NVIDIA V100 GPU; (a) and (b) show the model prediction and true damage data at the initial and a later stage with the absolute

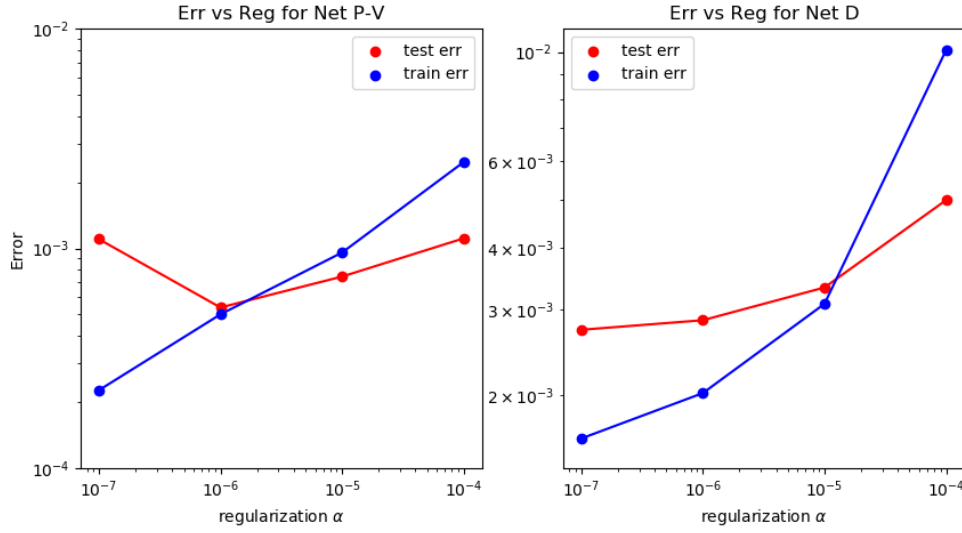


Figure 2.13: L_2^2 errors with different regularization parameters for Net P-V and Net D. We train a Net P-V and a Net D with 1,000 training cases using different regularization parameter α .

error plotted in the third column. In (c, d), we compare the inference results for the displacement field from a testing case. The inferred displacement field matches the true data well, with the maximum pixel-wise relative error less than 5%.

Next, we investigate the number of training cases on the influence of prediction errors by training a network with a fixed architecture with differing amounts of data. With the increase of training data, the training error and testing error drop to 2×10^{-4} and 6×10^{-5} . The figure on the right shows the training/testing error against epochs for a large and small network. The large one contains three hidden layers each with 100 neurons, whereas the small one contains three hidden layers and 50 neurons. By employing the small network, the testing error reduces from 0.08 to 0.001 while the training cases increase from 10 to 1,900. The performance of the small network, indicated by the distance between training and testing error, is better than that of the large network.

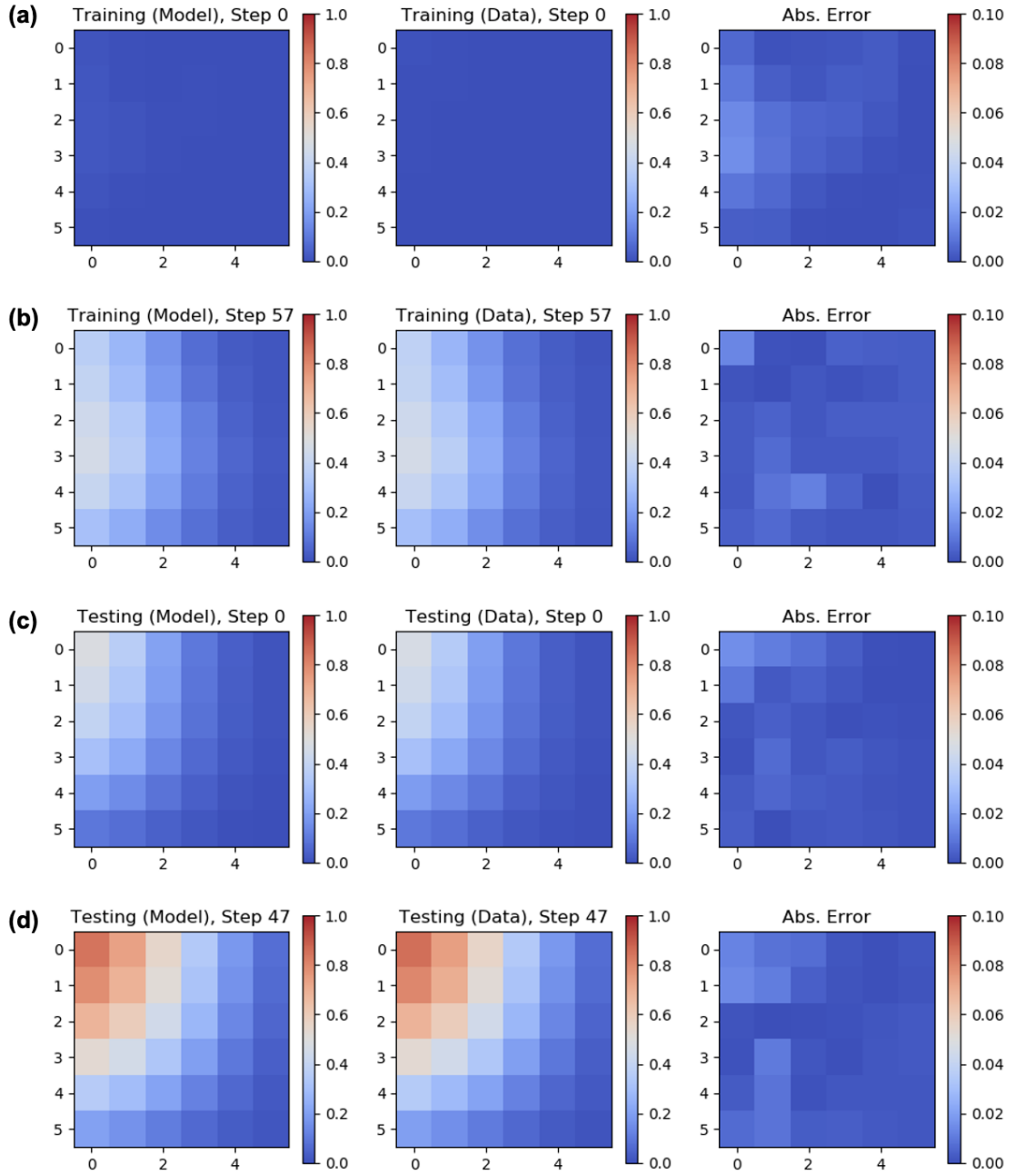


Figure 2.14: Training/Testing results of DeepONet for predicting displacement fields. Model predictions and true displacement fields are shown in the first and second columns, while the third column presents their absolute difference. (a, b) show the results for a training case at steps 0 and 57, while (c, d) represent for a testing case at steps 0 and 47. The length unit is μm .

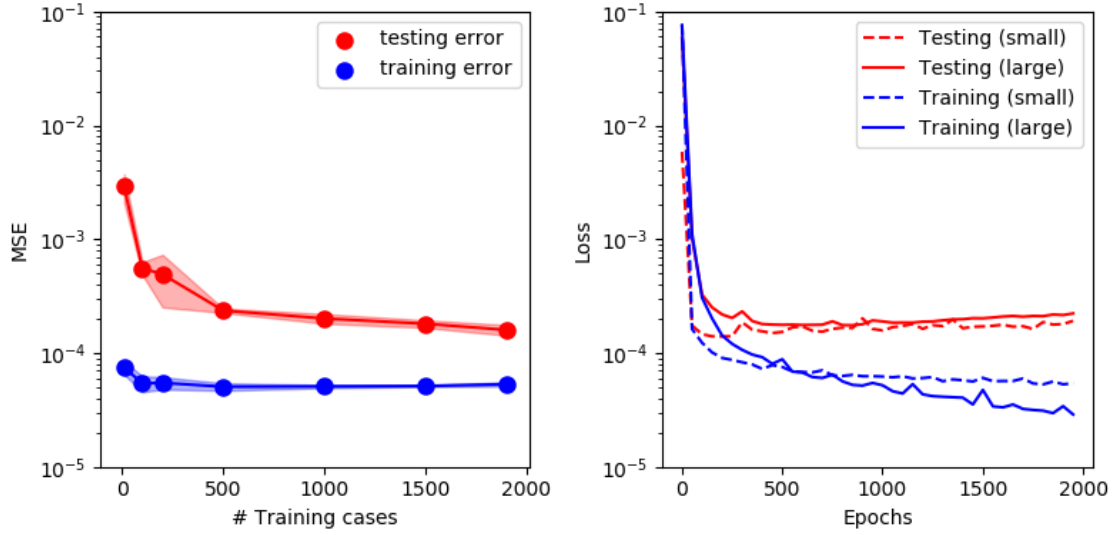


Figure 2.15: Mean square errors (MSE) for predicting displacement fields against the number of training cases and epochs. The network architectures are the same in Table 1. The shaded area represents one standard deviation from 5 runs with different initiation.

2.C Field Predictions Based on Strut Maps

The ultimate goal is to develop a reliable surrogate model of dissection only with the information of strut maps for each subject. Due to the nature of this complex system, however, our data-driven modeling renders a satisfactory and reliable result only with the guidance of additional information, e.g., the damage or displacement field. In this section, we conduct a quantitative comparison between models with partial (strut) and additional (displacement/damage) information. In Fig. 2.16, we show the testing results for predicting damage fields based on strut maps. Fig. 2.16 (a) shows the current damage field at steps 0, 20, and 40 from a model with strut maps. These results can qualitatively match the true damage progression at an early injection stage shown by a small injection volume, but the error will drastically grow at later stages (step 40 or later). A more quantitative result is presented in (c) where testing errors for the two models (with an error bar) show the overall performance over different injection stages. In general, both the

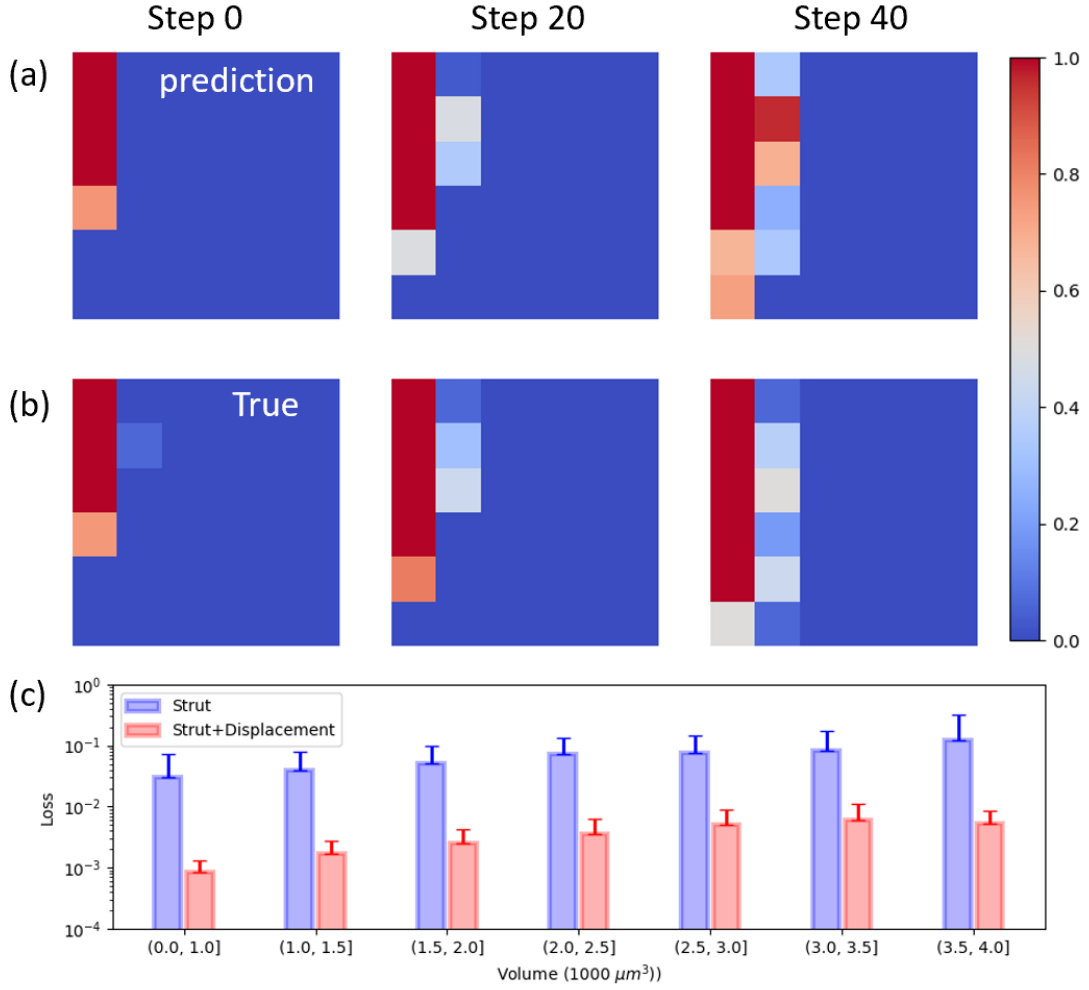


Figure 2.16: Testing results of DeepONet for predicting damage fields based on partial information. (a) Model predictions and (b) true displacement fields at steps 0, 20, and 40 are shown, whereas the predictions are only based on strut maps. (c) Testing loss of models with different inputs versus injection volume. The red bars indicate testing loss (L_2 error) of model predictions with strut and damage field as input whereas the blue bars are predictions based only on strut maps. The error bars show the one standard deviation from 5 runs with different initialization.

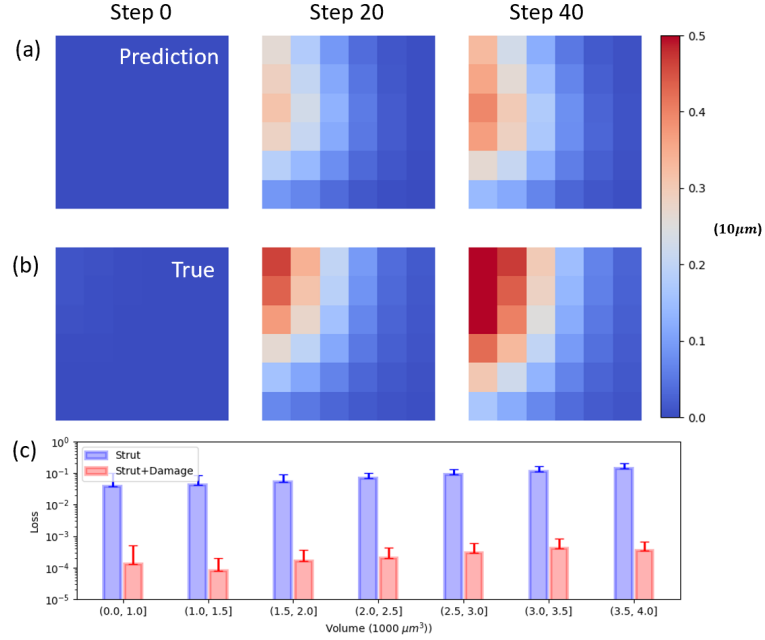


Figure 2.17: Testing results of DeepONet for predicting displacement fields based on partial information. (a) Model predictions and (b) true damage fields at steps 0, 20, and 40 are shown, whereas the predictions are based only on strut maps. (c) Testing loss of models with different inputs versus injection volume. The red bars indicate testing loss (L_2 error) of model predictions with strut and damage fields as input whereas the blue bars are predictions only based on strut maps. The error bars show the one standard deviation from 5 runs with different initializations.

blue bars, representing predictions based on the strut map, and the red bars show growth from a small to a large injection volume. However, with the guidance of additional information, i.e., displacement fields, the network achieves a much better prediction with orders-of-magnitude lowering of testing errors. A similar trend is observed in Fig. 2.17 for predicting displacement fields.

2.D Exploratory Analysis of Pressure–Volume Data

Because the data-generating finite element simulations of dissection were executed until failure, the training and testing data used to develop and validate the *Net* P - V presented herein covered disparate ranges of injection volume and pressure

values. To examine the statistical characteristics of the resulting training and testing data, we quantified the empirical distributions of the observed volume and pressure values, as well as several moments of those distributions. Hence, we generate another dataset with 1,000 cases using the same approach as that for the main text. However, the size of strut maps is now 36×32 .

Diffusion-based kernel density estimation [23] (optimal bandwidth $h = 52 \mu\text{m}^3$) of the maximum injection volume attained by the finite element simulations yielded two primary sub-populations of samples: one tightly concentrated around $\sim 3500 \mu\text{m}^3$ and another more variable group of samples that mostly experienced failure at lower injection volumes (Fig. 2.18(a)). We found that the overall distribution of maximum volume values was well described by a Gaussian mixture model with $K = 2$ components. In contrast, the distribution of maximum pressure values was unimodal and nearly Gaussian (Fig. 2.18(b)) with mean value at 68 kPa.

To further quantify the distribution of observed injection pressures, we examined various percentiles and correlations of the pressure data as a function of the injection volume (Fig. 2.19(a,b)). We found that the effective width of the volume-specific pressure distribution increased monotonically with injection volume, which can be summarized through a corresponding monotonic increase in the standard deviation (Fig. 2.19(c)). Moreover, we found that the standard deviation of the pressure distribution was linearly proportional to the corresponding mean observed pressure at the same injection volume (Fig. 2.19(d)). The pressure data were positively skewed especially at low injection volumes, though more symmetric as volume (and mean pressure) increased (Fig. 2.19(e)). Similarly, pressure distributions were leptokurtic at low volumes, but approached a kurtosis of 3 at higher volumes. Together, these metrics suggest that the pressure distributions

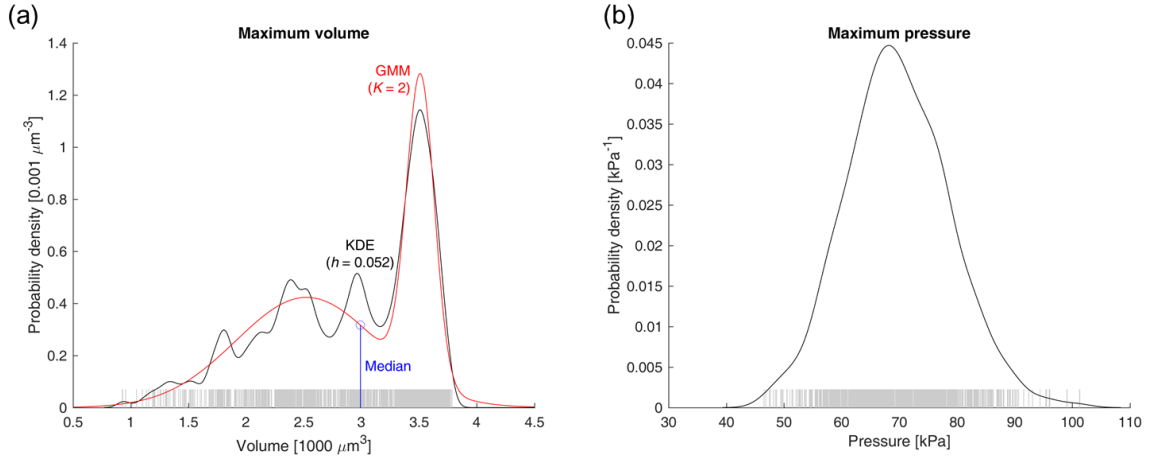


Figure 2.18: Maximum volume and pressure distributions. Because the data-generating finite element simulations of dissection were executed until failure, the training and test data used to develop and validate the P-V predictive models presented herein exhibited a large range of maximum injection volume and pressure values. (a) Kernel density estimation of the maximum volume data yielded two primary sub-populations of samples: one tightly concentrated around $\sim 3500 \mu\text{m}^3$ and another more variable group of samples that failed earlier. The overall distribution is well described by a two-component Gaussian mixture. (b) The distribution of maximum pressure values was unimodal and nearly Gaussian. GMM, Gaussian mixture model; KDE, kernel density estimation.

at high volumes are nearly Gaussian (Fig. 2.19(f)), consistent with our qualitative examination of the maximum pressure distribution (Fig. 2.18(b)).

2.E Linear Model Prediction of the Pressure–Volume Curve

To examine the effects of varying complexity in the input space, we applied the aforementioned linear model to two preprocessed versions of the elastin strut map: (1) a version of the strut map that was downsampled to half its size along each dimension (Fig. 2.20(a)), and (2) a version of the strut map that was first cropped to only the 20×10 top-left region surrounding the initial injection site and then downsampled as in the first version (Fig. 2.20(b)). In downsampling,

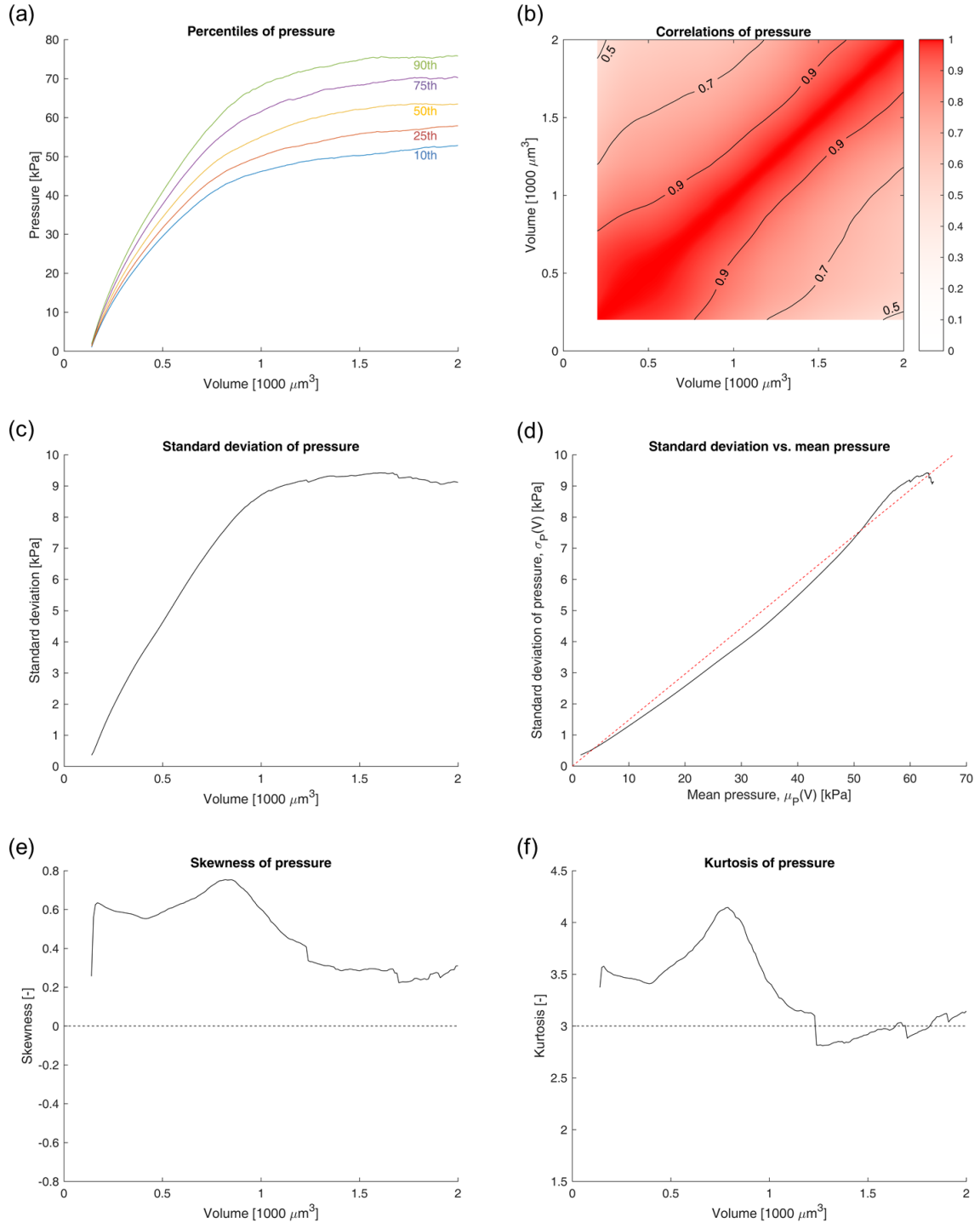


Figure 2.19: Statistical analysis of the pressure-volume data. (a) Percentiles and (b) correlations of observed pressure as a function of the injection volume. (c) Standard deviation of observed pressure as a function of the injection volume. (d) Standard deviation of volume-specific pressure plotted against the mean pressure observed at the same volume, showing a nearly linear relation. (e) While at low volumes, the pressure data is positively skewed, the skewness is closer to zero at higher volumes. (f) Similarly, pressure distributions are leptokurtic at low volumes, but approach a kurtosis of 3 at higher volumes. Together, these metrics suggest that the pressure distributions at high volumes are nearly Gaussian (cf Fig. 2.18(b)).

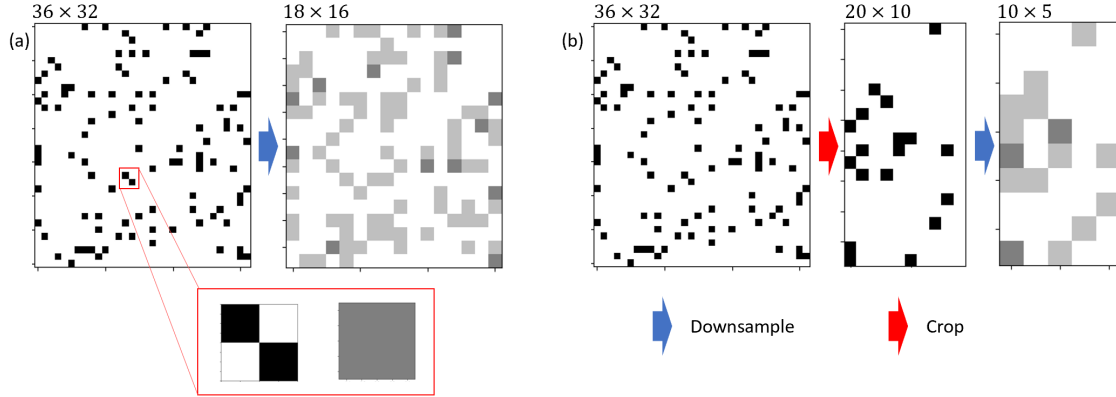


Figure 2.20: Preprocessing of the strut map for linear regression. (a) In the first version of the model, the input elastin strut map data is downsampled to half of its size along each dimension. In downsampling, each adjacent 2×2 block of pixels in the strut map is reduced to one pixel, whose value is set to the mean in the original pixel block. (b) In the second version of the model, the original strut map is first cropped to include only the top-left region surrounding the injection, and then the cropped strut map is downsampled, leading to a substantially reduced input space.

each adjacent 2×2 block of pixels in the strut map is reduced to one pixel, whose value is set to the average fraction of elastin in the original pixel block (this is equivalent to the application of a box filter to the original strut map). The total input size was thus 288 in the first version of the model but only 50 in the second version. The additional cropping step served to substantially reduce the input space, thus limiting the complexity of the model and reducing the potential for the model to overfit the training data while performing worse on the testing data. Alternatively, a regularization scheme could have been incorporated (e.g. via ridge or lasso regression); however, such approaches typically penalize coefficient magnitudes even in regions where the optimal coefficients are legitimately (i.e. non-spuriously) high. In contrast, cropping leaves those coefficients unpenalized while still removing spurious coefficients from distant tissue regions that should not be expected to contribute to the delamination response.

Regression results from both versions of the linear model yielded consistent trends in the corresponding optimal coefficient maps (Fig. 2.21). In both models,

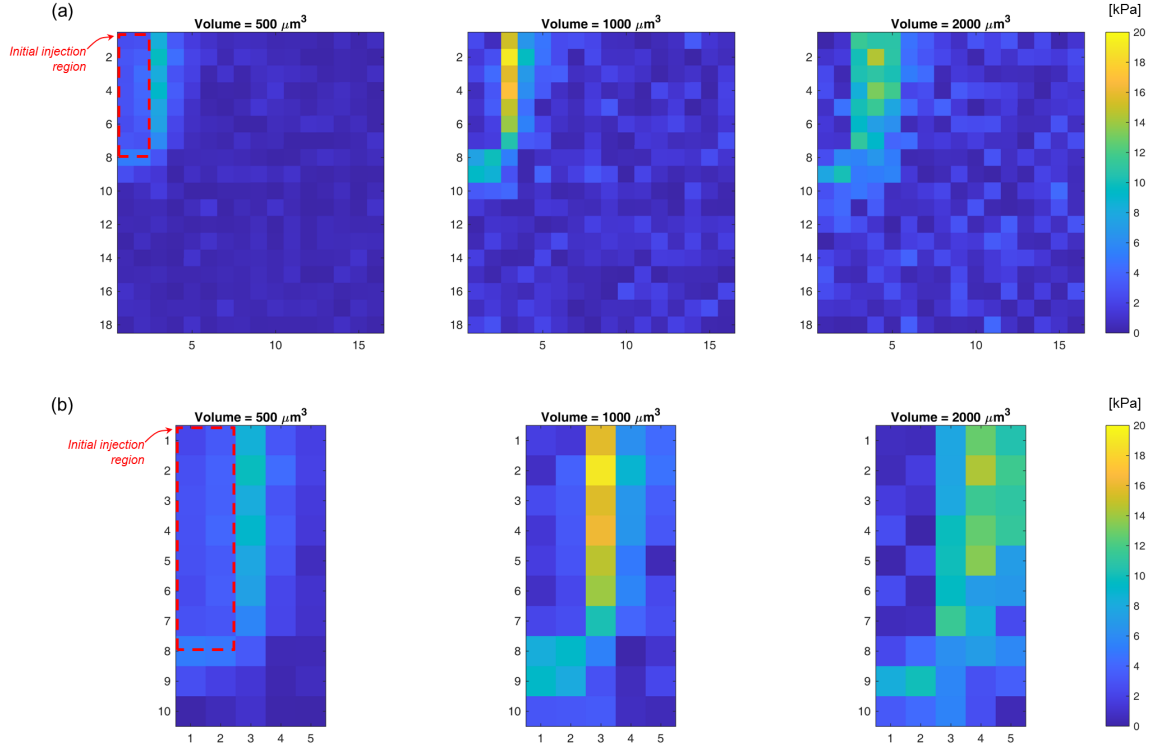


Figure 2.21: Absolute values of the least-squares linear P-V model coefficients. (a) Due to the large input space in the first version of the model, the regression results in spurious coefficient values in regions far from the injection site, suggesting possible overfitting. (b) The reduced input space yielded similar results in the remaining coefficients. In both models, pressure predictions at low injection volumes (e.g. $\leq 1000 \mu\text{m}^3$) are dominated by the elastin microstructure along the immediate boundary of the initial injection region, but broader regions of the tissue contribute to the model at higher volumes ($\sim 2000 \mu\text{m}^3$).

pressure predictions at low injection volumes (e.g. $\leq 1000 \mu\text{m}^3$) were dominated by the elastin microstructure along the immediate boundary of the initial injection region. At higher volumes ($\sim 2000 \mu\text{m}^3$), after the injection region has enlarged in a sample-specific manner, broader regions of the tissue contribute to the model prediction, though the dominant coefficients still neighbor the (current) injection region. Due to the large input space in the first version of the model (downsampled only), the regression also results in spurious coefficient values in regions far from the injection site, suggesting possible overfitting of the training data.

To assess the performance and robustness of each version of the linear model, we examined the discrepancy in the distribution of model residuals for both the

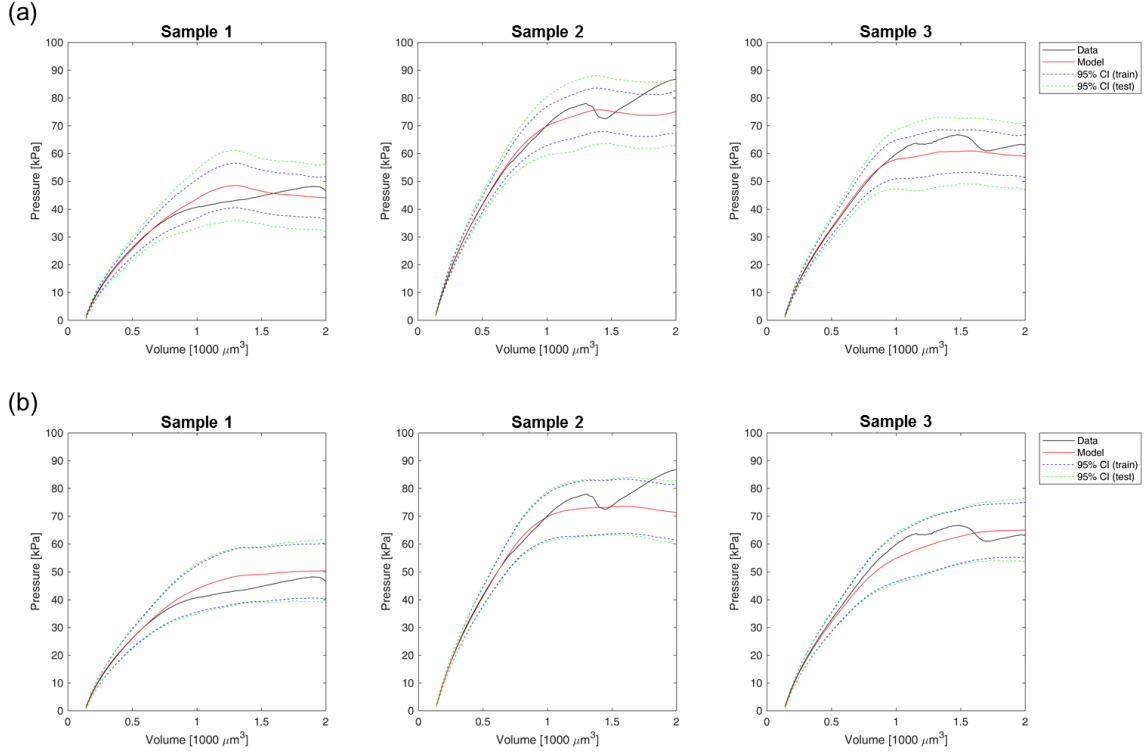


Figure 2.22: Performance of the linear P-V model for three representative samples. (a) In the first version of the model, the uncertainties in predicted pressure values estimated from the training data are substantially smaller than the errors observed in validation against the test data, indicating that the model overfits the training data. (b) In the second version of the model, which utilizes a cropped and downsampled strut map as its input, the uncertainties estimated from the training and test data are approximately equal. CI, credible interval.

training and testing data. In the first version of the model, where the input strut map data was only downsampled, the residuals in pressure prediction were substantially larger in the testing data compared to the training data (Fig. 2.22(a)), indicating that the model overfits the training data. In contrast, training and testing residuals were almost identical in the second version of the model, where the input strut map data was both cropped and downsampled (Fig. 2.22(b)). This improved performance was especially pronounced at higher injection volumes, where the average ratio between the root-mean-square testing and training residuals was ~ 1.6 in the downsampled-only model, but only ~ 1.1 in the downsampled and cropped model.

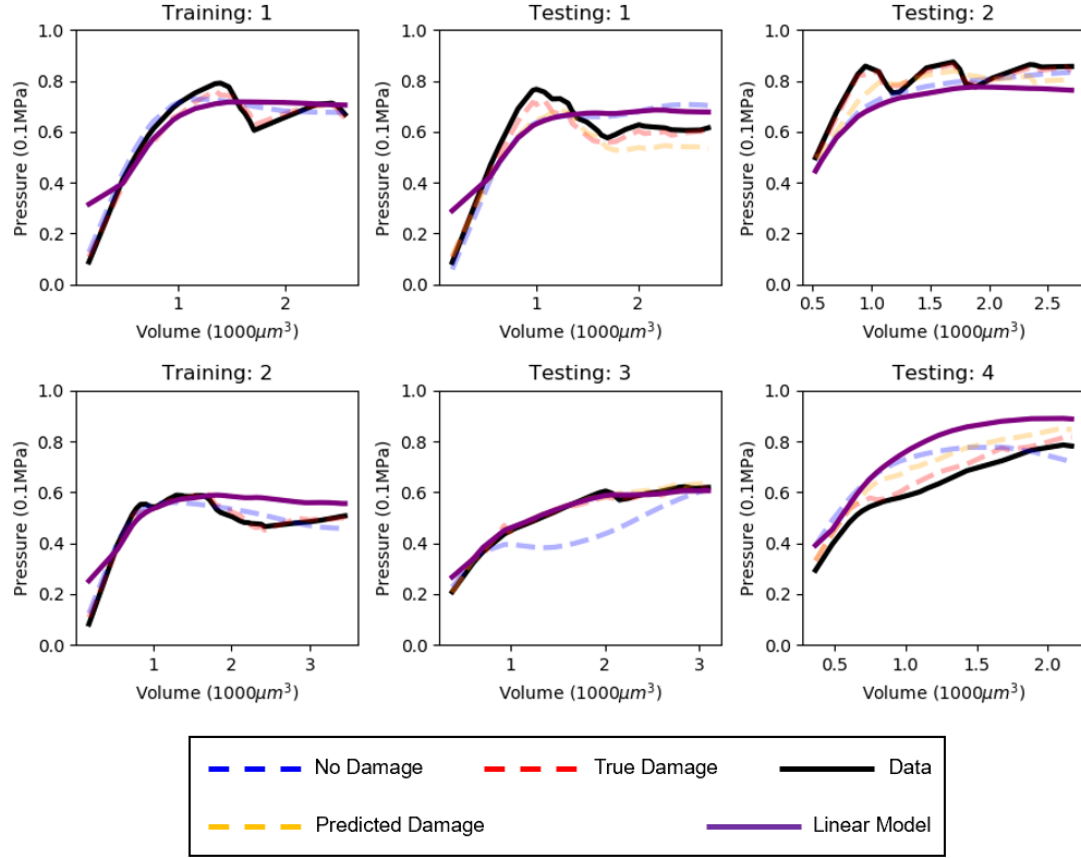


Figure 2.23: Results of the linear model for predicting P-V responses. Model predictions (dash lines) for training cases 1 and 2 are compared with the true data (black line, first column). In particular, model predictions with and without damage fields are plotted in red and blue, respectively. The other columns compare the testing data with models trained with (red)/without (blue) the current damage field. The branch net is FNN trained with 1900 training cases. The true data were generated from the phase-field finite element solver.

Fig. 2.23 compares predictions of the linear model and the DeepONet model proposed in the main text. The black solid lines represent the P-V curves from the finite-element simulations while the purple line denotes the mean linear model prediction. As mentioned, the predictions from the linear model can only capture the overall trend of the P-V curve, failing to characterize the detailed pressure drop induced by struts that break. Results from other models are also plotted as a comparison. A more quantitative comparison of errors for the linear model is in Table 3.

2.F Iterative Prediction of the Damage Field Using Logistic Regression

Consider, too, a logistic regression model to iteratively predict the damage field data from the elastin strut map and predicted damage values at preceding injection volumes. This model was developed to serve as a further baseline/proof-of-concept model for comparison with our higher-fidelity DeepONet results. In this approach, given a design matrix $\mathbf{X}$ describing the current tissue microstructure and a set of model parameters β , the probability of a particular pixel being the next to experience damage is modeled by a logistic function using

$$\Pr(\text{tear}) = \frac{1}{1 + e^{-\mathbf{X}\beta}}. \quad (2.13)$$

From our mechanistic understanding of the underlying tear propagation, we included in $\mathbf{X}$ parameters related to the current damage state of the tissue surrounding the pixel of interest as well as a quantification of the local tissue stiffness, both of which are known to influence the instantaneous tearing location and direction [11]. The local damage state was described using one-hot encodings of the number of lateral and diagonal neighbors that were currently damaged, while the local stiffness was described using 1×1 , 3×3 , and 5×5 neighborhood averages of the underlying tissue constituent stiffness moduli, to account for the distance-dependent effect of local stiffness. Including a constant term, this results in 10

model parameters β_i corresponding to the design matrix

$$\mathbf{X} = \begin{bmatrix} 1 \\ \mathbb{1}_1(n_{\text{dam, lat}}) \\ \mathbb{1}_2(n_{\text{dam, lat}}) \\ \mathbb{1}_3(n_{\text{dam, lat}}) \\ \mathbb{1}_1(n_{\text{dam, diag}}) \\ \mathbb{1}_2(n_{\text{dam, diag}}) \\ \mathbb{1}_3(n_{\text{dam, diag}}) \\ \mu_{1 \times 1} \\ \mu_{3 \times 3} \\ \mu_{5 \times 5} \end{bmatrix}^T, \quad (2.14)$$

where $\mathbb{1}_a(x)$ is the indicator function (equal to 1 if $x = a$ and 0 otherwise), $\{n_{\text{dam, lat}}, n_{\text{dam, diag}}\}$ are the number of damaged lateral and diagonal neighbors, and $\{\mu_{1 \times 1}, \mu_{3 \times 3}, \mu_{5 \times 5}\}$ are the local tissue stiffness values averaged over the corresponding neighborhood sizes. The stiffness values were normalized to the stiffness of elastin and the stiffness of the non-elastin matrix was prescribed as $1/20^{\text{th}}$ of the elastin stiffness following [11] (Fig. 2.24).

The resulting best-fit coefficients showed that the number of currently damaged neighbors and the local tissue stiffness both contribute substantially to the probability that a given pixel will be the next to experience damage—that is, to tear (Fig. 2.25(a)). An increase in the number of currently damaged neighbors, whether lateral or diagonal, was associated with an increased probability of tearing pressure $\text{Pr}(\text{tear})$. In contrast, greater local tissue stiffness was associated with a smaller $\text{Pr}(\text{tear})$. Note that while the absolute value of β_{10} is largest among the stiffness-associated parameters, this does not imply that the average 5×5 stiffness

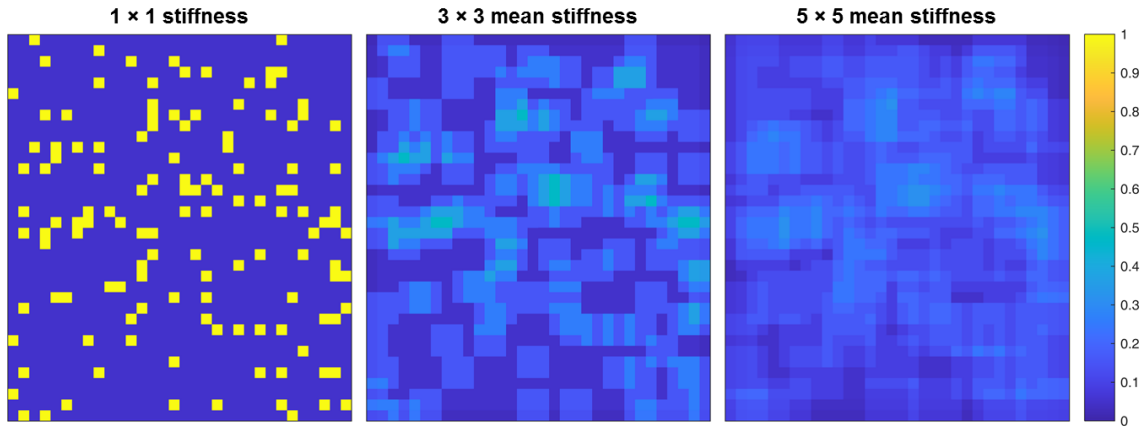


Figure 2.24: Neighborhood normalized stiffness maps. Maps of the tissue stiffness averaged over 1×1, 3×3, and 5×5 pixel neighborhoods, for a representative sample. The stiffness values were computed directly from the elastin strut map, and normalized to the prescribed elastin stiffness, which is 20× greater than that of the non-elastin matrix.

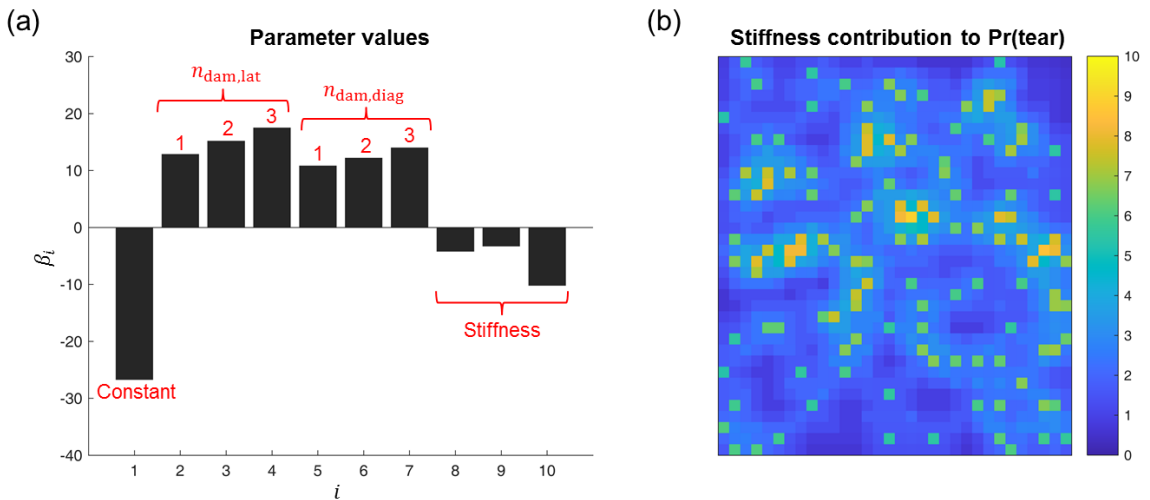


Figure 2.25: Logistic regression results. (a) Best-fit coefficients for the logistic regression model, showing contributions of damaged neighboring pixels and effective local tissue stiffness. (b) Corresponding map of the contribution of local tissue stiffness to the probability of tearing.

is most predictive of tearing, since the contribution of this parameter is spread over 25 total pixels. Rather, the relative contribution of each neighborhood to $\text{Pr}(\text{tear})$ is computed by the product of the corresponding parameter and the neighborhood size. The resulting map of the effective contribution of stiffness to the probability of tearing is heterogeneous but fairly diffuse (Fig. 2.25(b)), highlighting the length scale of which the differential stiffness acts. For an elastin strut surrounded by a non-elastin matrix, the normalized stiffness contribution to decreasing $\text{Pr}(\text{tear})$ from the elastin pixel itself is 4.944, while the remaining 3×3 neighborhood pixels contribute 0.767 each and the still-remaining 5×5 neighborhood pixels contribute 0.406 each, consistent with the stiffness effect decreasing monotonically with distance.

To evaluate the predictive capability of the model, we numerically simulated the iterative propagation of damage, starting from the initial injection region. In each time step, $\text{Pr}(\text{tear})$ was computed for every yet-undamaged pixel, using the elastin strut map and the *predicted* damage field from the previous time step as predictors, then the damage was assigned to the pixel with the highest $\text{Pr}(\text{tear})$. While the order in which pixels experienced damage often differed from the “ground-truth” finite element results, the evolution of the predicted damaged region qualitatively matched the shape of the true damaged region (Fig. 2.26). Note especially that this model, while simple in its construction, predicted correctly that damage is substantially less likely—and thus delayed—in areas where there is a high density of elastin struts, particularly when those struts are spatially aligned and form a “barrier” to tear propagation.

On the whole, the results from this logistic regression-based predictive model suggest that the damage field can be well described by accounting primarily for the state of the tissue microstructure along the immediate boundary of the current

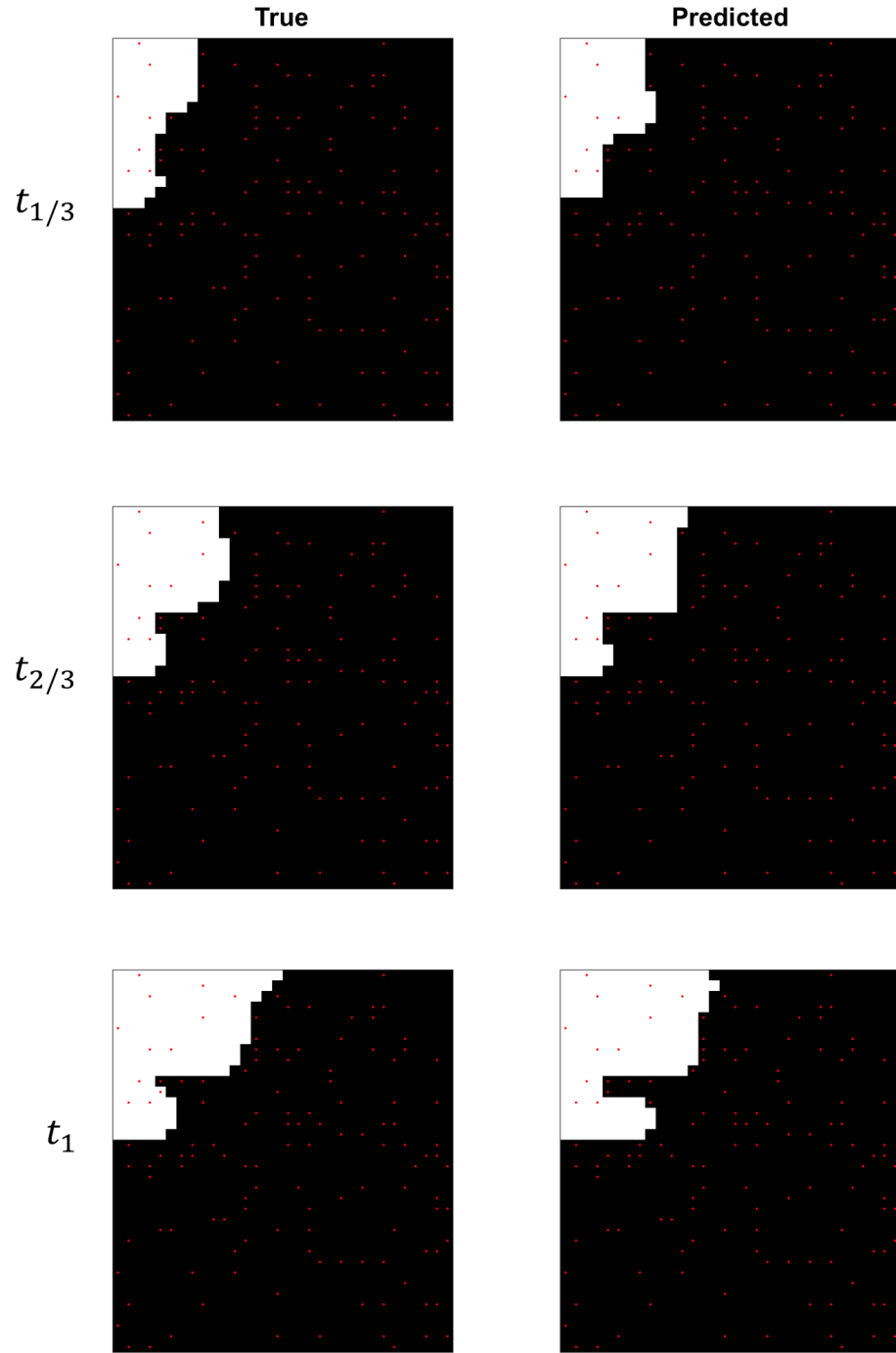


Figure 2.26: Logistic regression-based damage prediction. True (i.e. finite element-derived) and predicted (by the logistic regression model) maps of damage, shown at frames one-third ($t_{1/3}$) and two-thirds ($t_{2/3}$) of the way through the simulation as well as at the final simulation time (t_1). The damaged region is displayed in white, while the locations of elastin struts are shown as red dots.

damaged region, though a more complex modeling framework is needed to improve accuracy in predicting the nuanced patterns in the evolving tear propagation and the corresponding pressure–volume relation. In the DeepONet approach, we thus prescribed a convolutional neural network (CNN) architecture to the trunk net, since this architecture similarly relies disproportionately on interactive effects between small neighborhoods of pixels (section 3.3).

CHAPTER THREE

Computational Modeling of Aortic Mechanics Based on Microimaging

3.1 abstract

The aortic wall experiences dissection and failure as a result of injury, age, and disease, which are closely related to variations in the microstructures of the medial layer. The variations dramatically affect the mechanical properties of the aortic wall and its potential pathophysiological outcomes, including dissection progression. Given the complexity of the aortic wall, we develop a particle-based computational model for investigating the underlying aortic biomechanics based on ex vivo microimaging that reveals realistic microstructure of the subject. The computational geometries are collected and reconstructed from two-photon imaging data of a wild-type and a mgR (Marfan) mouse, based on which we perform a series of in-silico mechanical tests. The averaged intramural stresses predicted by the particle model shows an agreement with the analytical solution, whereas the Marfan mouse aorta exhibits a more disrupted radial stress distribution led by the structural deterioration by fenestration on the elastin in the media. The results further indicate that structural integrity may also contribute in vessel delamination since we observe a more distributed tensile radial stresses in the Marfan model than the wild-type model. This work is the first attempt of utilizing imaging-driven modeling based on two-photon imaging for investigating underlying aortic mechanics. The proposed computational model provides a novel framework for quantitatively analyzing complex dynamics and elucidating mechanisms of each constituents based on multimodality data.

keywords: Smoothed Particle Hydrodynamics, Two-Photon Imaging, Soft Tissue, Aortic Dissection, Four-Fiber Family

3.2 Introduction

Aortic dissection is a life-threatening disease that manifests within the medial layer, leading to delamination and mechanical failure of the aortic wall along with other severe clinical conditions [189]. In particular, acute aortic dissection in the thoracic aorta propagates within a few seconds, which poses a challenge to randomized trials, rapid diagnosis, and prognosis in clinics to prevent the patients from high mortality rate [79, 130]. Various imaging modalities – including MRI, CT, and echocardiogram – are among the standard investigations in clinics [79].

The aortic wall is, in essence, a complex system, which is composed of three primary layers: tunica intima, tunica media, and tunica adventitia. The innermost intima layer does not contribute significantly to the mechanical behavior of the aortic wall, except for the subendothelial layer in coronary arteries [98]. The medial and adventitial layer, on the other hand, are fundamental to the load-bearing capacity at lower and higher pressure, respectively [15]. The medial layer consists of laminar units, which are mainly composed of collagen fibers (Type I and III, in the circumferential direction), elastin, and smooth muscle cells [270]. The adventitia, the outermost layer, is mainly made up of fibroblasts, fibrocytes, collagen fibers (Type I [102]), and other ground substances [270]. Although tremendous progress has been made in delineating and analyzing the various constituents and their functions in this complex system, the fact that the underlying mechanisms of dissection initiation and propagation are still poorly understood is partially contributed by the complexity of the internal microstructure and the myriad associations and interactions between proteins, proteoglycans (PGs), and growth factors of the vascular matrix [121].

Notwithstanding the majority of aortic dissection cases observed in clinics arises from an intimal tear [253, 189, 194], there are, yet, a relatively significant portion of dissection cases occurred without structure damage in the endothelial. Apart from the hematoma hypothesis proposed by [194, 271], aortic dissection is embodied in another hypothesis proposed by Humphrey [102], which illustrates the mechanical contributions of glycosaminoglycans/glycoproteins (GAGs/GPs) in the artery (thoracic). By Gibbs-Donnan swelling pressure, loss of tensile stiffness, and development of local stress concentration, GAGs/PGs could contribute to the delamination of elastic lamellae via one of the aforementioned approaches.

There exist many experimental findings that shed light on the new hypothesis. Seminal work from Roach and colleagues investigated pressure-induced delamination of the aortic wall at various locations [91, 163, 225], which qualitatively described some mechanical characteristics of GAG-induced dissection. More works further identified that abnormal accumulation of GAGs – along with loss of smooth muscle, fragmentation of elastic fiber, and remodeling of fibrillar collagens – are among the main histological features in the aortic tissues from patients with thoracic aortic aneurysm and dissection (TAAD) [45, 102, 306].

A series of computational investigations that account for nonlinearity, heterogeneity, and anisotropy in the aorta have been conducted to rigorously understand the wall mechanics with GAGs/PGs presented. In [227, 226], the authors developed two complementary models, a finite-element model and a continuum-based model, whose results suggested that (1) the important roles of GAGs/PGs in mechanosensing and integrity of the aorta and (2) abnormal aggregates may lead to pathological conditions that would reduce the overall integrity of the aortic wall. Ehsan et al. [11, 12] introduced microstructure-motivated phase-field finite element model to mimic an injection-driven dissection. The model represents the

crack propagation by phase-field evolution, without adopting the expensive mesh enrichment required in extended FEM [178, 322]. Another way of modeling dissection propagation is to adopt particle methods, represented by smoothed particle hydrodynamics ([157, 3, 221, 4]), peridynamics ([240, 312]), or dissipative particle dynamics ([284]). In particular, the SPH with the four-fiber family model ([3, 4]) provides further insights into the complicated interaction between lamellar units and GAG pools at the progression stage of aortic dissection. One interesting finding is that, unlike in a real scenario, dissection progresses only within intralamellar space (circumferential direction) based on all simulations represented without turning directions. It is assumed that the heterogeneity of local microstructure may induce the turning of delamination, which poses a pressing need to develop an imaging-driven, not just microstructure-motivated, computational model based on realistic microstructure provided by imaging modalities.

Recently, the development of modern imaging techniques, namely, two-photon microscopy (TPM), has greatly facilitated the understanding of vascular mechanics [90, 262, 310, 37]. TPM is an advanced modality suitable for investigations of regional microstructural properties for biomaterial due to its non-destructive nature and good penetration depth [92]. Cavinato et al. [38] combined ex vivo biomechanical testing with TPM to quantify layer-specific microstructural properties in murine aortas. In particular, primary extracellular matrix components, such as fibrillar collagen and elastic lamellae, can be acquired and reconstructed, providing a viable approach to developing quantitative modeling based on realistic geometries.

In this work, we address investigating the GAG-induced dissection hypothesis by developing an imaging-driven particle model whose geometry is acquired based on realistic microimaging data. The particle model is adopted based on

smoothed particle hydrodynamics (SPH) with the four-fiber family model for describing the nonlinear behavior of the multi-constituent arterial wall [10, 74]. Herein, the geometry adopted was acquired using TPM acquired from a loading state in the biaxial mechanical testing.

We organize the paper as follows: In Sec. 3.3, we introduce the methodology, including TPM and a computational model. In Sec. 3.4, we report the results that process the TPM image stacks, approaches to construct the 3D geometry, and simulations on which we perform detailed simulations based. At last, we conclude with a short discussion at the end.

3.3 Methodology

We present the models needed to develop the computational framework in this section. The overall workflow is shown in Fig. 3.1. A segment of the descending thoracic aorta was isolated and cannulated on a biaxial loading device. Next, mechanical forces from the device and microstructural imaging from TPM were collected simultaneously while pressurizing and stretching the DTA segment to its in vivo states. The collected data was analyzed and postprocessed for detailed microstructural reconstruction and parameter fitting for the constitutive model, based on which the particle simulations for revealing detailed local mechanics were performed. Experimental details of vessel isolation and biaxial loading were elaborated in [38, 62]. We will briefly introduce the basic formulation of the particle model, the constitutive model for arterial mechanics, and the imaging processing pipeline for imaging reconstructions.

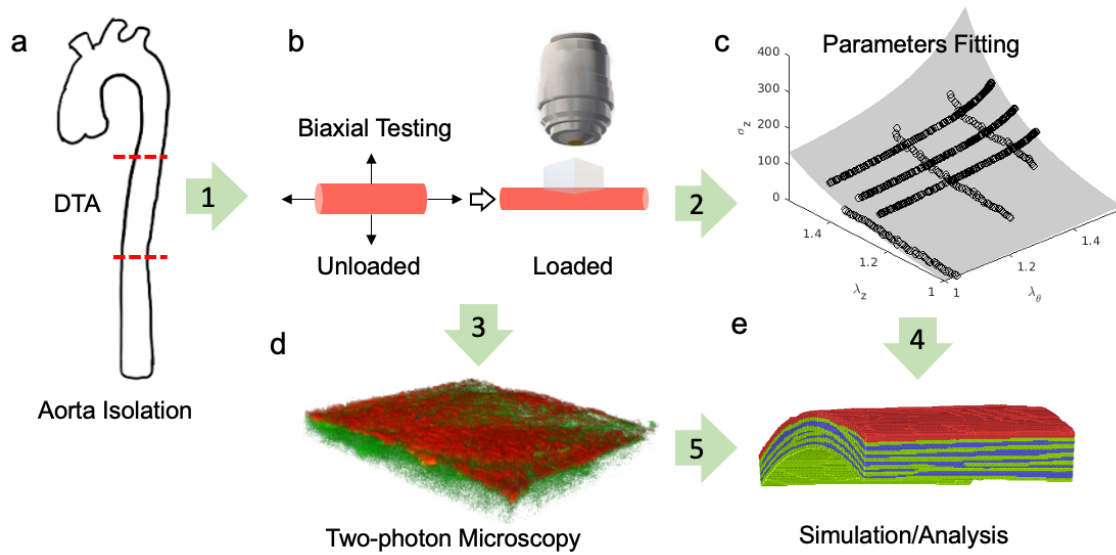


Figure 3.1: Schematic of imaging-driven modeling of aortic mechanics. (a) Aorta isolation from mouse models. A segment of the descending thoracic aorta (DTA) was isolated from euthanized mice and was (1) cannulated and mounted on a biaxial loading device. (b) The segmented aorta was pressurized and loaded to in vivo states where two-photon imaging and mechanical data were collected. The collected data were analyzed separately: (c) the mechanical data were used to quantify mechanical parameters in the four-fiber model and (d) the imaging data were postprocess in a specific pipeline for geometry reconstruction. (e) Detailed computational modeling was developed using parameters fitted from (c) and particle representation of the microstructure from (d).

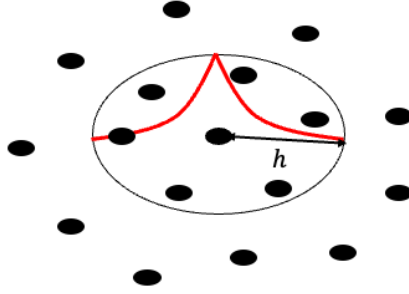


Figure 3.2: “spiky” kernel in SPH with the kernel radius h .

3.3.1 Smoothed Particle Hydrodynamics

In Sec. 3.3.1, we introduce the framework of total lagrangian smoothed particle hydrodynamics (TLSPH), which refers to the initial configuration as the reference state. The framework is suitable for simulating vascular biomechanical problems shown by [3, 4]. The methods of discretizing the equilibrium equation with TLSPH and pre-loading the configuration to its in vivo states will be discussed in Sec. 3.3.1 and 3.3.1.

TLSPH

In TLSPH, physical quantities are approximated based on kernel methods. Let us consider a function $f(X)$ and its approximation $\hat{g}(X_i)$ at point X_i

$$\hat{g}(X_i) = \int f(X)W(X - X_i)dX, \quad (3.1)$$

We define W as a weighting kernel:

$$W(R_j, h) = A \begin{cases} (h - R_j)^3, & R_j < h, \\ 0, & R_j \geq h, \end{cases} \quad (3.2)$$

where h is the radius of the kernel illustrated in Fig. 3.2. The equation above denotes that one integrates the information from a discrete point if the distance (R_j) between X_j and the reference point is within the kernel h and neglect the point when $R_j \geq h$. Coefficient A is defined as $A = 10/\pi h^5$ in two dimensions or $A = 10/\pi h^6$ in three dimensions.

In TLSPH, $g(X_i)$ is numerically approximated as a function of particle volume V_j and kernel $W(R_j, h)$ as

$$g(X_i) = \sum_{j \in S} f_j V_j W(R_j, h), \quad (3.3)$$

Taking the gradient of $g(X_i)$ with respect to its reference coordinate X

$$\begin{aligned} \nabla_X g(X_i) &= \sum_{j \in S} f_j V_j \nabla_X W(R_j, h), \\ \nabla_X W(R_j, h) &= \left(\frac{\partial W(R_j, h)}{\partial R_j} \right) \frac{R_j}{|R_j|}, \end{aligned}$$

If $f(X)$ is a vector, then the integration rule rewrites as

$$\nabla_X g(X_i) = \sum_{j \in S} f_j \otimes V_j \nabla_X W(R_j, h) \quad (3.4)$$

Then, we introduce two *ad-hoc* numerical corrections. Introducing the following equation guarantees the symmetrization of force [180]:

$$\nabla_X g(X_i) = \sum_{j \in S} (f_j - f_i) V_j \nabla_X W(R_j, h). \quad (3.5)$$

Modifying kernel W as following preserves the first-order completeness [217]:

$$\tilde{\nabla}_X W(R_j, h) = A^{-1} \nabla_X W(R_j, h), \quad (3.6)$$

$$A_i = \sum_{j \in S} V_j \nabla_X W(R_j, h) \otimes R_j \quad (3.7)$$

Equilibrium equation and Constitutive Model

In this section, we briefly discuss the equilibrium equation and its form in TLSPH.

The governing equation of solid mechanics writes:

$$\nabla \cdot P + \rho_0 \mathbf{b} = \rho_0 \ddot{\mathbf{x}}, \quad (3.8)$$

P is the first Piola-Kirchhoff stress (PK1) and $\mathbf{b}$ denotes the body force. ρ_0 is the referential mass density. The PK1 stress can be expressed as a function of deformation gradient F and strain energy density $\mathcal{W}$.

$$P_i = F_i S_i = 2F_i \frac{\partial \mathcal{W}}{\partial C}, \quad (3.9)$$

where the deformation gradient F in TLSPH is defined as:

$$F_i = \frac{\partial x_i}{\partial X_i} = \sum_{j \in S} r_j \otimes V_j \tilde{\nabla}_X W(R_j, h) \quad (3.10)$$

As shown in [22], the equilibrium equation for each particle is equivalent as the Newton's law:

$$m_i \ddot{x}_i = f^{int} + f^{ext} + f^{hg}, \quad (3.11)$$

where the internal forces is

$$f_i^{int} = \sum_{j \in S} V_i V_j (P_i \tilde{\nabla}_X W(R_i, h) - P_j \tilde{\nabla}_X W(R_i, h)). \quad (3.12)$$

f_{ext} denotes the external forces acting on each of the particles where, herein, f_{ext} is 0. To suppress spurious hourglassing mode, we adopt an addition term f^{hg} from [71]

$$f_i^{hg} = \sum_{j \in S} -\alpha \frac{EV_i V_j W(R_j, h)}{2R_j^2} (\delta_i + \delta_j) \frac{r_j}{r_j} \quad (3.13)$$

For updating the displacement field, particle velocity, and acceleration, we adopt the forward Euler method. We refer the reader to [221, 71] for more details.

Deposition Prestretch

The total Lagrangian framework refers to the initial configuration as the reference state. However, the ex-vivo imaging data were collected at a pressurized state with axial deformation being stretched to its in-vivo state. That being said, the internal stresses were not zero at the initial configuration [62, 56, 13]. In addition, there exist residual stresses induced by cell-mediated deposition and developmental pre-stretches even at an unloaded state (the inner pressure is zero). Simulating vascular mechanics at in vivo states requires further consideration that incorporate the preloadings and prestresses [34]. Thus, we adopt a different approach by introducing deposition prestretches to the deformation gradient [34, 15].

Imposing deposition prestretch to a static system has been discussed extensively in previous works. Herein, we consider imposing prestretches to a dynamical system in two stages, illustrated in Fig. 3.3. Consider a quarter-circle aorta

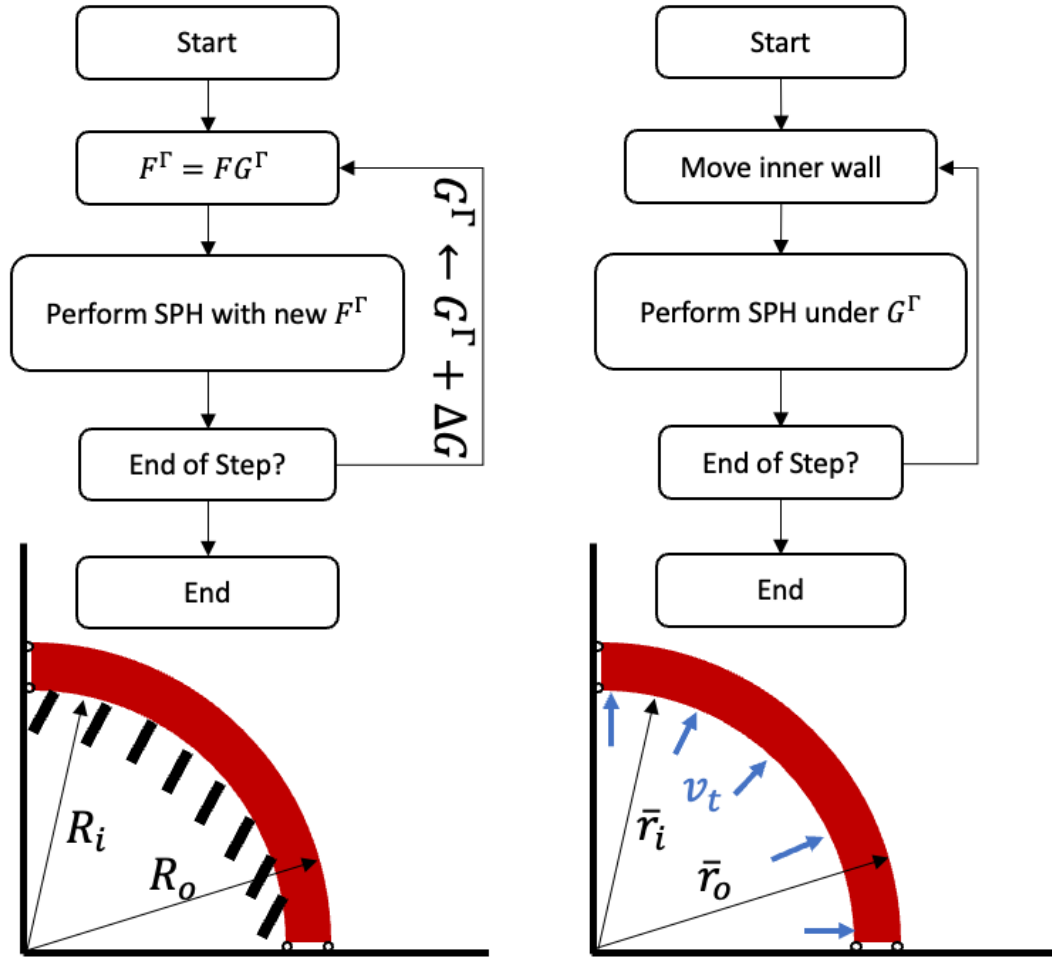


Figure 3.3: Algorithms that introduce deposition prestretch in SPH. Left: The inner wall is fixed with incremental deposition prestretch (ΔG). The wall is free to move in the radial direction while being bounded in the circumferential direction. Right: Once the deposition prestretch is imposed, the inner wall is moved in the radial direction until the end of the moving step.

with particles at 0° and 90° being free to move in the radial direction. In stage I, we start our simulations from the in-vivo configuration with the total deformation gradient F as the identity matrix. We fix the displacement of the particles on the inner vessel wall while setting the outer surface as traction-free. Next, we stretch each constituent in their principal directions by adding an incremental stretch ΔG to the original stretch tensor G for constituent Γ , which represents elastin (*e*), collagen (*col*), and smooth muscles (*smc*). The overall deformation gradient is then modified as $F^\Gamma = F^\Gamma G^\Gamma \neq \mathbf{I}$ and accordingly, the Cauchy stress $\sigma \neq 0$. In an ideal computational setup where the material is incompressible and homogeneous, the particles should remain at their original locations after imposing prestretches. However, due to the nearly incompressible assumption, structural heterogeneity, and the numerical deficiency of SPH at boundaries [80, 279], the particles will slightly deviate from their reference state in our simulations. The incremental prestretch is set as very small to avoid a sudden acceleration of particles. Once, the stretch tensor G^Γ for each constituent is loaded to their in-vivo state, we unlock the displacement of the inner wall in stage II by moving the inner wall along the radial direction. We record the pressure and stresses in each direction during stage II to reproduce the biaxial loading state in experiments.

3.3.2 Four-fiber Model

We adopt the four-fiber family model, a widely accepted constitutive law, as the strain energy function $\mathcal{W}(\mathbf{C}, \mathbf{M})$ for modeling mechanical behavior of the aorta [10, 107, 62]. The four-fiber model describes the aorta as hyperelastic and anisotropic with fiber-reinforcement in four directions. Its strain energy density

writes as

$$\mathcal{W}(C, M) = \phi_F^e \left[\frac{\mu}{2} (I_1 - 3) - \mu \ln(J) + \frac{K_0}{2} \left(\frac{J^2 - 1}{2} - \ln J \right) \right] + \sum_{i=1}^4 \phi_F^i \left[\frac{k_1^i}{4k_2^i} (\exp\{k_2^i (I_{4i} - 1)^2\} - 1) \right], \quad (3.14)$$

where principal invariants I_1 and I_{4i} are defined based on the right Cauchy Green strain C , deformation gradient F , and fiber directions M_i ,

$$I_1 = C : I, J = \det F, I_{4i} = C : \mathbf{M}_i \otimes \mathbf{M}_i,$$

$$\mathbf{M}_1 = [0, 0, 1],$$

$$\mathbf{M}_2 = [0, 1, 0],$$

$$\mathbf{M}_3 = [0, \cos(\alpha), \sin(\alpha)],$$

$$\mathbf{M}_4 = [0, \cos(\alpha), -\sin(\alpha)]$$

α is the fiber orientation with respect to the circumferential direction. μ is the shear modulus and $k_{1,2}^i$, $i = 1, \dots, 4$ are the fiber modulus and exponential coefficients in the axial ($i = 1$), circumferential ($i = 2$), and two symmetric diagonal directions ($i = 3, 4$). In addition, constituents in the aorta are inherently differentiated by microstructures that contain different components. To model such heterogeneity, we introduce mass fraction ϕ for different groups Γ (elastin *eln*, media *med*, and adventitia *adv*) in directions i [4]. The above parameters are usually calibrated using nonlinear regression methods based on experimental data [62].

3.3.3 Two-photon microscopy

We use signals in different TPM channels to delineate the main constituents of the descending thoracic aortic wall: fibrillar collagens, elastic fibers (main com-

ponents of elastic lamellae), and cell nuclei. Each constituent was detected at a different wavelength: we use the second harmonic generation (SHG) signal at wavelength 390-425 nm to detect collagen structures and collect the two-photon auto-fluorescent signal of elastic fibers at 500-550 nm. Elastic fibers and fibrillar collagens are represented in the green and red channels as shown in Fig. 3.1. Nuclei were detected using fluorescent signals with a wavelength higher than 550 nm. However, nuclei distributions (blue channel) are not utilized when reconstructing the computational geometry since the nuclei signal is highly aligned with the elastins, leading to difficulties in constituent delineation. The in-plane (axial-circumferential) view of field ranges $500\ \mu\text{m} \times 500\ \mu\text{m}$ with imaging resolution and out-of-plane step size at 0.48 and $1\ \mu\text{m}/\text{pixel}$. The samples were collected at diastole with inner pressure at 80 mmHg and axial stretch at 1.45. We post-process the collected data following the procedures adopted in [38]. The images are transformed from a circumferential-axial into a polar coordinate using a custom MATLAB algorithm for facilitating microstructural analysis. The final geometry is transformed back to Cartesian coordinate for considering curvature. We refer the reader to find more details in [38].

3.4 Results

3.4.1 Benchmark testing

To verify the implementation of the hyperelastic SPH model, we first consider a benchmark problem: inflation test for a quarter-circle aorta. Particles on the inner wall are fixed at stage I and then set to move in the radial direction for

Parameter	Symbol	Value
Initial Radius (inner/outer)	R_i, R_o	8.0 mm, 8.5 mm
Elastin-dominated material constant	μ	31.466 kPa
Axial collagen parameters	k_1^1, k_2^1	1.417 kPa, 0.541
Circumferential collagen parameters	k_1^2, k_2^2	12.914 kPa, 0.237
Diagonal collagen parameters	$k_1^{3,4}, k_2^{3,4}$	4.584 kPa, 0.843
Fiber orientation (circumferential)	α	0.7941
Deposition Pre-stretch	$G_\theta^e, G_z^e, G_{1,2,3,4}^c, G^{smc}$	1.4, 1.59, 1.2, 1.0
Mass fraction	$\phi^e, \phi^{1,2,3,4}$	1.0, 1.0

Table 3.1: Parameters used in the benchmark mean-behavior model. The parameters are fitted based on biaxial data of a WT model.

mimicking inflation. We assume the deposition prestretches of the elastin are 1.4 (circumferential) and 1.59 (axial). The prestretches for the fiber families are all set to 1.2. We employ the algorithm presented in Fig. 3.3 to impose the deposition prestretches before moving the inner wall. The inner radius at the initial state is 8 mm with the wall thickness at 0.5 mm (shown as blue particles in 3.4). We tabulate parameter values of the constitutive model in Table 3.1. Fig. 3.4 shows that the passive mean-behavior SPH models describe well the mean circumferential stresses (σ_θ) during inflation, agreeing with the thick-wall analytical model. The circumferential stresses are averaged from the inner particles with the exclusion of the corner ones.

3.4.2 Imaging reconstruction

In this section, we discuss a customized pipeline in ImageJ and Python for post-processing the transformed raw images from TPM. As shown in Fig. 3.5(a), a stack of transformed raw images in the green channel represents the microstructure

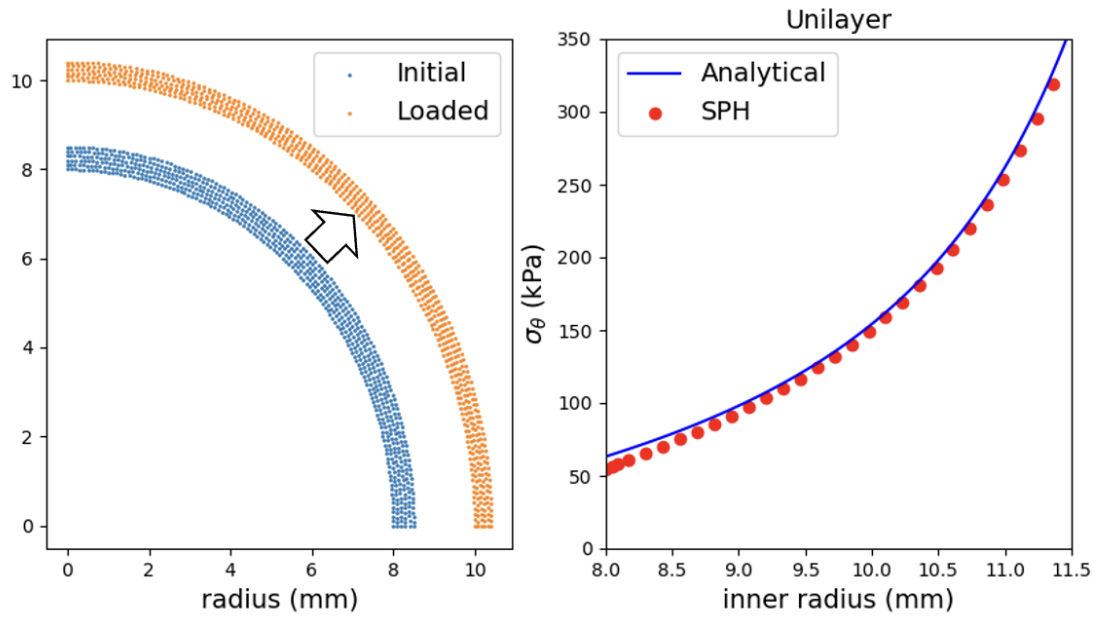


Figure 3.4: Inflation test and wall stresses predicted by the mean-behavior SPH model. Left: SPH particles at the initial (blue) and loaded (orange) configuration. Right: Circumferential stresses predicted from the SPH (red dots) and an analytical thick-wall model during inflation.

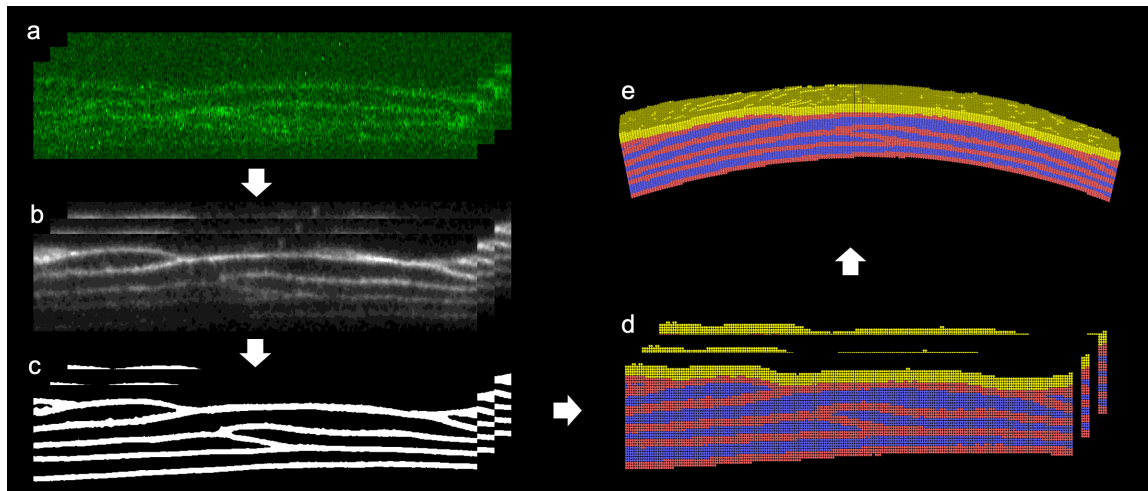


Figure 3.5: Procedures of imaging processing and microstructure reconstruction. (a) A stack of raw images that represent elastic lamellae was processed, (b) filtered, and (c) thresholded through a customized pipeline. (d) The thresholded images are converted into particles with collagen (yellow) coating on the outer and smooth muscle filled in the intralamellar space (blue). (e) The reconstructed geometry is transformed in a cylindrical coordinate with curvature.

Constituent	WT Histo. (%)	mgR Histo. (%)
(Media) Elastin	54	38
(Media) SMC	24	20
(Media) Collagen	15	17
(Media) Others	7	25
(Adventitia) Elastin	0.5	1
(Adventitia) SMC	10	5
(Adventitia) Collagen	87	82
(Adventitia) Others	3	12
Media Volume Fraction	76	68

Table 3.2: Mass fractions of each constituent for the WT/mgR model. The parameters are collected from histological analysis.

of elastic fibers. In (b), we employ a series of filters to denoise the raw images, then take pixel average in the axial direction to further enhance the contrast of elastic fibers. Next, we adopt a local thresholding filter to extract elastic fibers after manual labeling (Fig. 3.5(c)). Following the previous step, we design a customized python pipeline to transform the thresholded structure into a particle representation of the subject-specific microstructure, where main constituents are denoted and identified with different colors. Smooth muscle cells are filled within the intralamellar space represented by blue particles. In particular, images from TPM does not provide enough resolution to resolve fibrous level microstructure, i.e., single fiber orientation and its distribution. Hence, we coat a layer of adventitia particles on the top of the media while keeping the mass fraction of the media consistent with our histological analysis (Table 3.2). Then, our customized pipeline processes each slice of the image stack and transforms the 3D structure to the Cartesian coordinate for preserving bending moments (Fig. 3.5 (d-e)).

Fig. 3.6 shows a slice of raw images and its particle representation after being

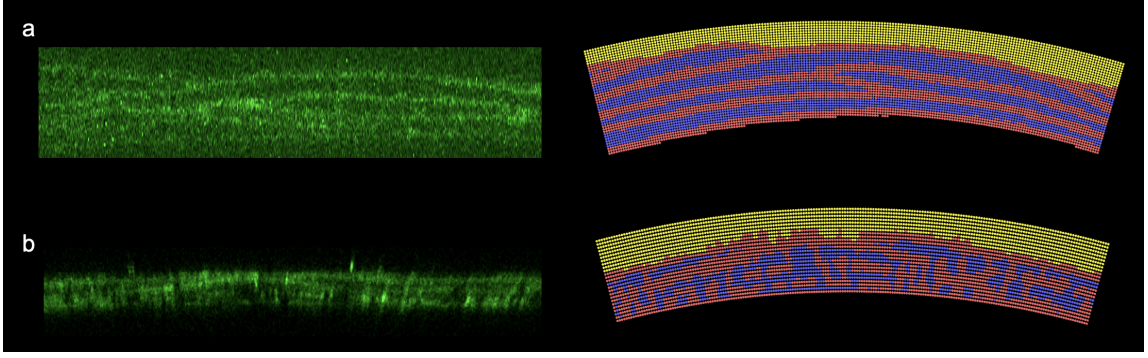


Figure 3.6: Reconstructed microstructure for the (a) mgR and (b) WT model. Mass fraction of each constituent: (Elastin, WT) 44.58%, (SMC, WT) 32.12%, (Adventitia, WT) 23.30%; (Elastin, mgR) 45.99%, (SMC, mgR) 21.68%, (Adventitia, mgR) 32.33%.

Parameter	Symbol	WT	mgR
Initial Radius (inner/outer, at 80 mmHg)	R_i, R_o	473.1 mm, 545.1 mm	386.4 mm, 432.0 mm
Elastin-dominated material constant	μ	165.051 kPa	113.536 kPa
Mass fraction (elastin)	$\phi_{eln}^e, \phi^{1,2,3,4}$	0.395, 0	0.38, 0
medial collagen parameters (tensile)	$k_1^{1,2,3,4+}, k_2^{1,2,3,4+}$	136.996 kPa, 7.7326	85.324 kPa, 33.2863
medial collagen parameters (compression)	$k_2^{1,2,3,4-}$	0.01	0.001
Mass fraction (intralamellar)	$\phi_{int}^e, \phi_{int}^1, \phi_{int}^2, \phi_{int}^{3,4}$	0.395, 0.0366, 0.4010, 0.0702	0.38, 0.0007, 0.2140, 0.0776
adventitia collagen parameters	$k_1^{1,2,3,4+}, k_2^{1,2,3,4+}$	277.140 kPa, 21.0304	424.112 kPa, 23.734
Mass fraction	$\phi_{adv}^e, \phi_{adv}^1, \phi_{adv}^2, \phi_{adv}^{3,4}$	0.007, 0.2052, 0, 0.3939	0.01, 0.0038, 0.1176, 0.3743
Fiber orientation (circumferential)	α	1.004	0.7225
Deposition Pre-stretch	$G_{\theta}^e, G_z^e, G_{1,2,3,4}^c$	1.6, 1.45, 1.1	1.8, 1.6, 1.1

Table 3.3: Parameters used in the WT/mgR multilayer model. The parameters are fitted and collected based on biaxial fitting and histological imaging processing [62]. *eln*: elastin, *int*: intralamellar substances, *adv*: adventitia.

post-processed. It is noticeable that the WT model is thicker than the mgR model with more continuous elastic layers in the media. The mgR model, however, has rather discontinuous elastic lamellae due to fenestrations on the elastin, leading to a reduced load-bearing capacity. The first and last layers in the circumferential direction are duplicated and extended circumferentially so that a quarter-circle shape is formed (Fig. 3.7 left column).

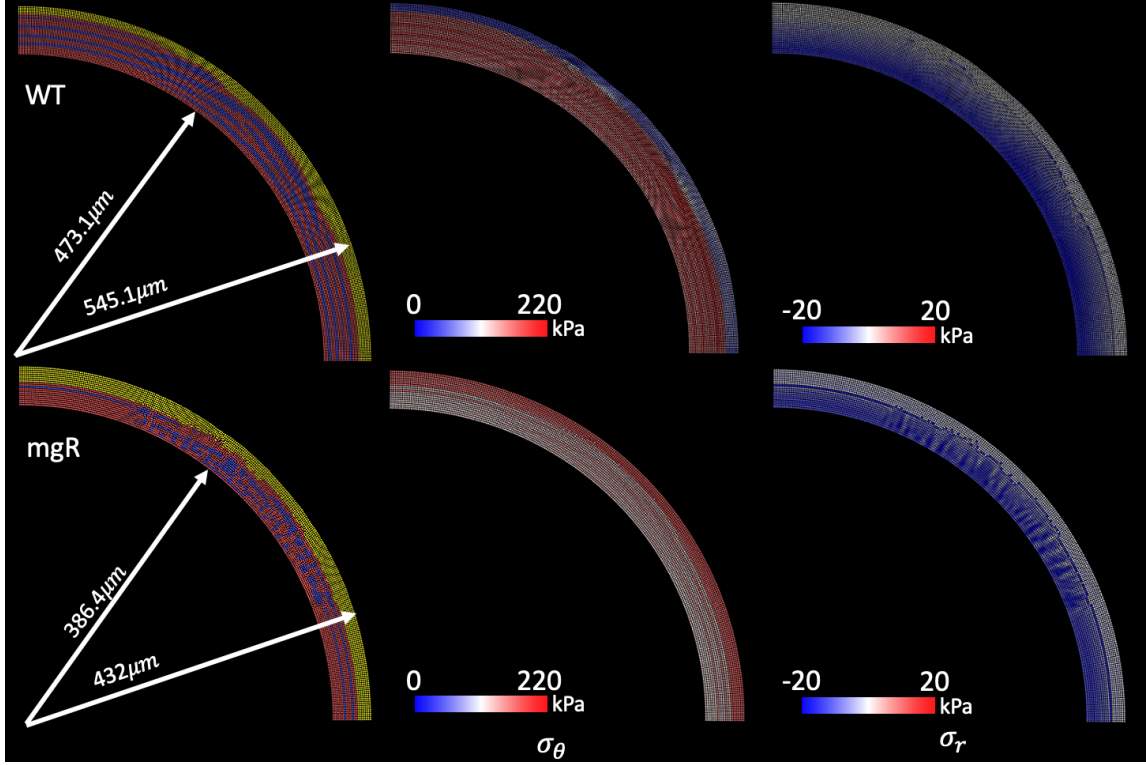


Figure 3.7: Microstructure and stresses distribution in the WT (top) and mgR (bottom) predicted from SPH simulations. The stresses are simulated at diastole (80 mmHg).

3.4.3 Imaging-driven simulations

We perform numerical simulations for investigating the internal stresses of the aorta with detailed microstructures. First, we perform parameter fittings for each model following the procedures in [62]. Combining the mass fractions collected from histological analysis, we tabulate the final fitted parameters in Table 3.1. Notice that the SMC mass fraction in the mgR model (ϕ_{int}^2) is dramatically reduced from 40% (WT) to 20%, substituted by other ground substances, such as GAGs. We present more comparison of fitted results with biaxial loading data in 3.B.

We present stress distributions of the two models at diastole in Fig. 3.7. Comparing transmural distributions of stresses reveals a fundamental difference in the role of load-bearing between the WT and mgR model. Regarding the circumfer-

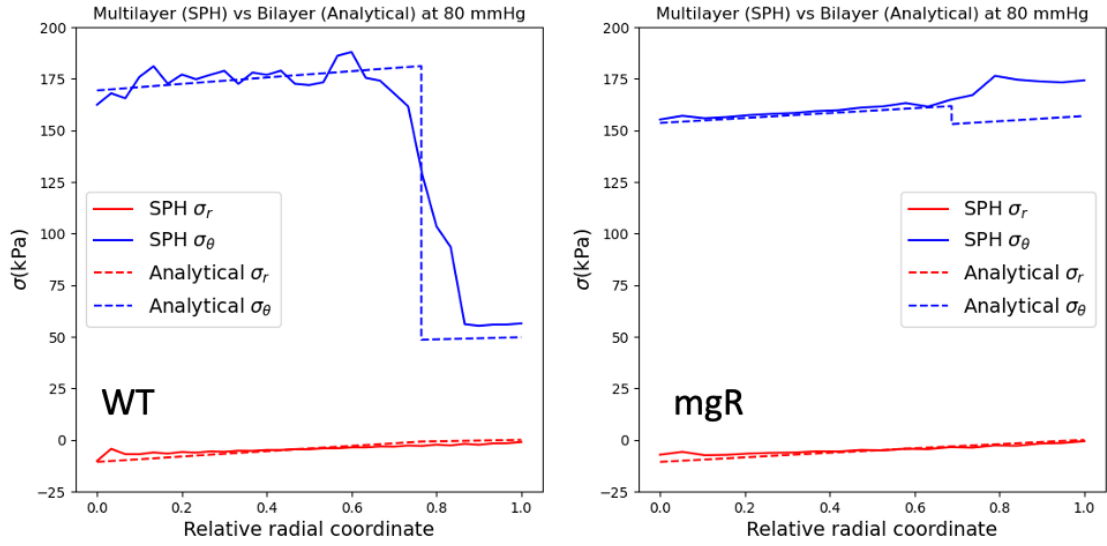


Figure 3.8: Distribution of the radial (σ_r) and circumferential (σ_θ) stresses in the WT (left) and mgR (right) model. The predicted stresses from SPH are plotted with the blue solid lines with the dashed lines representing analytical solutions.

ential stress σ_θ , the medial layer in the WT model accounts for significantly more stresses than the adventitia layer at diastole. The relatively continuous elastic layers guarantee more uniform distribution of σ_θ through the aortic wall. The media in the mgR model, however, partially shifts its loading to the adventitia, especially when the microstructure beneath is more disrupted (shown by an abrupt increase in σ_θ at 60°). In the radial direction, the disruption in σ_r of the mgR model is more pronounced than in the WT model due to the discontinuous elastin in the media. We observed a qualitative correspondence between the fenestration on the elastin and tensile radial stresses, indicating that abnormal geometric factors, such as fenestrations or GAG pools, could alter the local stress distributions. Given that dissection progresses due to tensile, not compression stress, the distinctive microstructure of mgR models may also contribute to the failure of the aorta.

Next, we compare the mean transmural stresses in the radial and circumferential directions and present the averaged results in Fig. 3.8. The dashed curves that represent the analytical solutions at 80 mmHg are compared with the averaged

SPH predictions through the wall. Although being locally disrupted in the mgR model, the mean σ_r from SPH corresponds well with the analytical solutions. We also observe an agreement for the mean σ_θ in the WT model: σ_θ in the media is approximately at 175 kPa whereas σ_θ in the adventitia quickly drops to around 50 kPa. The statistical results from SPH can capture this drop with satisfactory accuracy. For the mgR model, however, σ_θ in the media from SPH and analytical solutions corresponds well with a little discrepancy in the adventitia. We also observe the load-shifting effect in this stress comparison where the magnitude of increase in σ_θ is about 100 kPa in the mgR adventitia. Such load-shifting effect is consistent with the observation in [15] where the authors also found that the effect is more pronounced at supranormal pressure conditions. The reason why SPH simulations can only qualitatively match the transmural stresses of the analytical solution comes from that the SPH takes underlying microstructure into consideration whereas the analytical results are computed based on a homogenized assumption in each layer. Hence, heterogeneity of realistic microstructures may lead to variations and even disruptions reflected in mean stresses.

3.5 Discussion

In this work, we develop a particle modeling framework for investigating aortic mechanics based on TPM data. Particles representing aortic microstructures are postprocessed and reconstructed by a customized pipeline using ImageJ and python. We employ a constituent-specific four-fiber model for describing stress-stretch relationships where the model parameters are fitted using biaxial loading data. Due to the imaging data being collected at loaded states, we also introduce “deposition prestretch” to preload the reconstructed geometry to its in-vivo

states. To verify the accuracy of our SPH implementation, we perform an inflation test where the predicted stresses match well with its analytical solutions. Then, we utilize the model to simulate the internal stresses of a WT and mgR model. The averaged transmural stresses are in agreement with the analytical solutions with detailed local stress distributions revealed by SPH. We further demonstrate that discontinuous elastic layers led by fenestrations could deteriorate the local microstructure by turning compressive radial stresses to tensile.

This is the first attempt at utilizing realistic microstructures for investigating arterial biomechanics. Such a framework can be extended to investigate biomechanical problems for other arteries when microstructure plays an important role in local biomechanics. Certainly, more improvements and directions can be adopted and considered in the future. Firstly, we will incorporate active stresses as it can protect aortas from delamination by reducing vessel deformation [211, 184]. Another important direction is to implement GAGs into the mgR model since GAGs account for 21% of the mass fraction in the mgR model. In addition, performing 3D simulations is also interesting to look into since one could track the preferential orientations of dissection progression once delamination occurs [244].

There exist challenges and limitations for this imaging-driven modeling. One interesting observation from our numerical fitting procedures: the more parameters to fit, the less confident the results are. Developing a detailed computational model requires as much information on material properties as possible, whereas some of them are challenging to estimate from our current constitutive model and fitting framework. Ad-hoc assumptions on deposition prestretches and collagen fiber parameters are usually made to simplify parameter fitting procedures, but these assumptions may fail in real scenarios. Certainly, if more features are included in the computational model, one has to ask the question that if their

material properties can be inferred based on measurable data. Modeling damage of soft material is another important part to consider when modeling aortic dissection since choosing different damage models may lead to different, or even contradicting results. In addition, deposition prestretch works well for a relatively homogenized aorta, however, introducing such a technique into a non-uniform, heterogeneous microstructure may lead to numerical instability.

3.6 Acknowledgments

MY would like to thank Enrui Zhang (Brown), Dr. Yue Yu (Lehigh) and Bart Spronck (Masschritch) for meaningful discussion and help. We acknowledge computational resources from Center for Computation and Visualization (CCV) at Brown University and Frontera at Texas advanced computing center (TACC) BCS21009. We acknowledge the support by grant U01 HL142518 from the National Institutes of Health.

3.A Parallel efficiency

Our code is developed based on LAMMPS SMD, a particle simulation package for solving problems in continuum mechanics [72, 71]. We implement constitutive models for hyperelastic materials, such as Neo-Hookean, HGO, and four-fiber models, into the existing framework while enabling heterogeneous material simulations. Unlike previous SPH implementation for hyperelastic materials [4, 221], the package inherits the parallel framework of LAMMPS [205], making our new

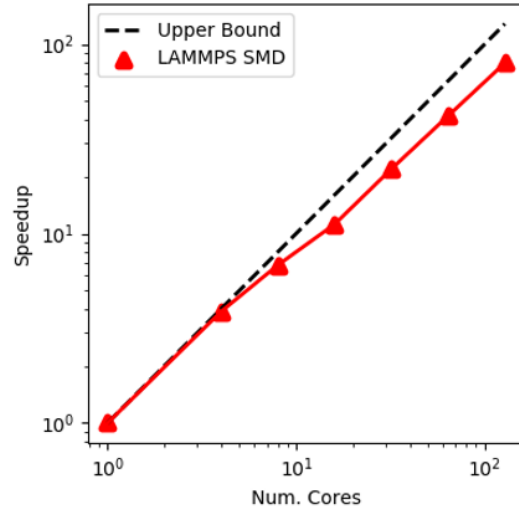


Figure 3.9: Benchmark testing for parallel efficiency of LAMMPS SMD. We test the code efficiency for a fixed size problem run on different numbers of processors. Speedup factors of LAMMPS SMD (out implementation) are bounded by the theoretical speedup.

implementation readily for large scale simulations.

Fig. 3.9 shows computational speedup of LAMMPS SMD for a fixed-size uniaxial tension problem with 10,000 particles. The simulations are performed on Frontera, TACC with Intel Xeon Platinum 8280. We define the computational speedup as the ratio of wall time on single processor (T_1) over multiple processors (T_n). Theoretically, if the code scales perfectly, the speedup increases linearly with the increase in the numbers of processors. We perform the fixed-scale problem on various processors, ranging from 1, 4, 8, and up to 128. The realistic speedup, albeit limited by other time consumption such as message passing, presents a nearly linear increase with the increasing numbers of processors. The overall performance represents a good parallel efficiency of the framework.

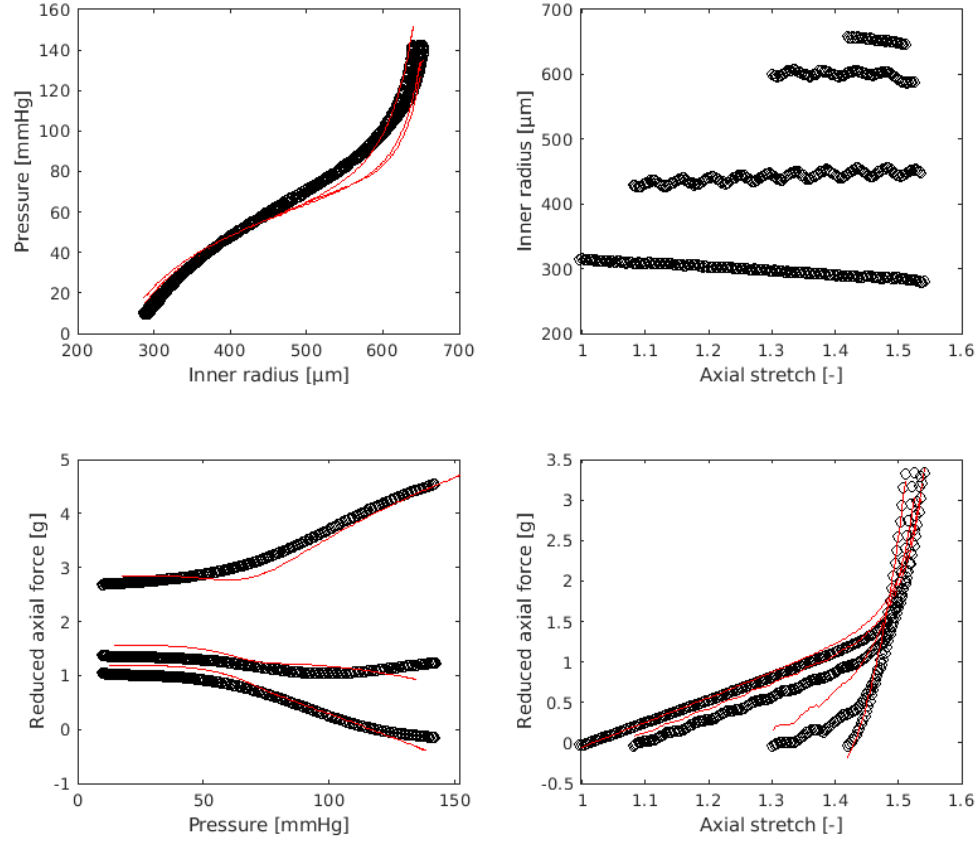


Figure 3.10: Fitting from the four-fiber model (red lines) with best-fit parameters and biaxial testing data (dots) for the WT mouse.

3.B Bilayer model parameter fitting

We follow the parameter fitting algorithm proposed in [62] to find a set of optimal parameters that minimize the difference in axial force and intramural pressure between four-fiber models and experiments. The fitted parameters are shown in Table 3.3. We compare four-fiber model predictions (red lines) and experimental measurements from biaxial tests (dots) for the WT (Fig. 3.10) and mgR model (Fig. 3.11).

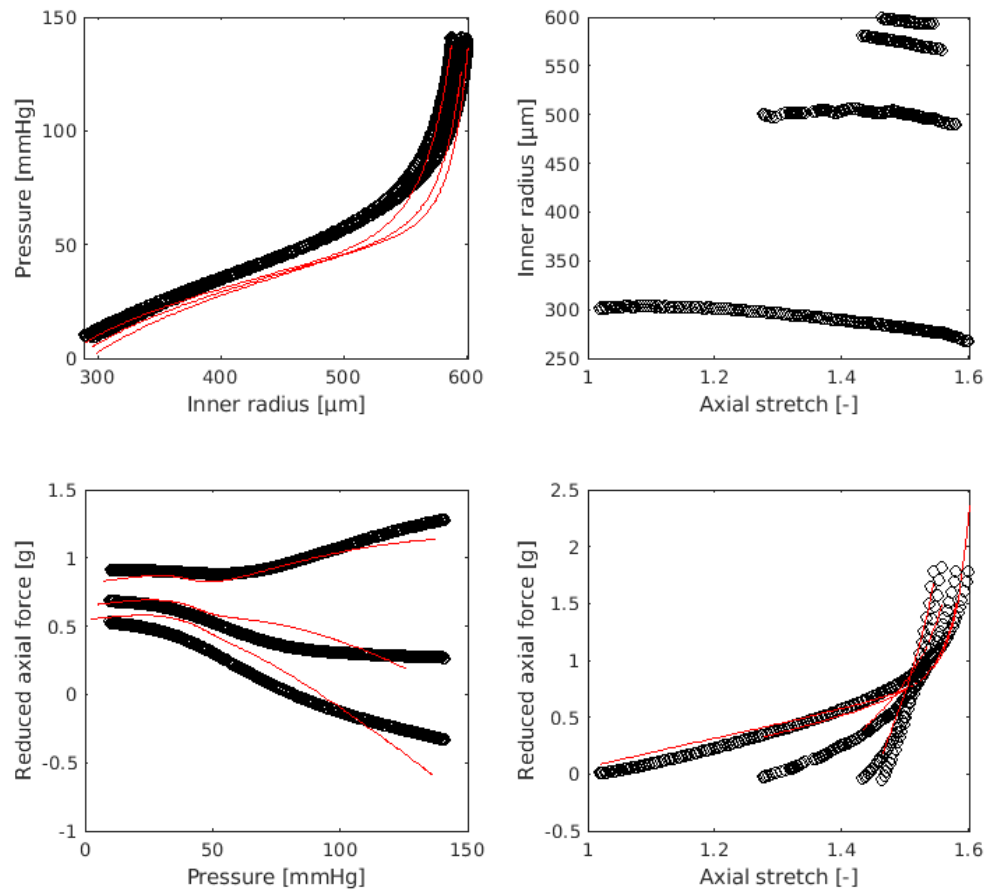


Figure 3.11: Fitting from the four-fiber model (red lines) with best-fit parameters and biaxial testing data (dots) for the mgR (Marfan) mouse.

CHAPTER FOUR

Non-invasive Inference of Thrombus Material Properties with Physics-Informed Neural Networks

4.1 Abstract

We employ physics-informed neural networks (PINNs) to infer properties of biological materials using synthetic data. In particular, we successfully apply PINNs on inferring permeability and viscoelastic modulus from thrombus deformation data, which can be described by the fourth-order Cahn-Hilliard and Navier-Stokes Equations. In PINNs, the partial differential equations are encoded into a loss function, where partial derivatives can be obtained through automatic differentiation (AD). In addition to tackling the challenge of calculating the fourth-order derivative in the Cahn-Hilliard equation with AD, we introduce an auxiliary network along with the main neural network to approximate the second-derivative of the energy potential term. Our model can simultaneously predict unknown material parameters and velocity, pressure, and deformation gradient fields by merely training with partial information among all data, i.e., phase field and pressure measurements, while remaining highly flexible in sampling within the spatio-temporal domain for data acquisition. We validate our model by numerical solutions from the spectral/*hp* element method (SEM) and demonstrate its robustness by training it with noisy measurements. Our results show that PINNs can infer the material properties from noisy synthetic data, and thus they have great potential for inferring these properties from experimental multi-modality and multi-fidelity data.

4.2 Introduction

Thrombus deformation and failure [220, 219] are important in deep vein thrombosis [283, 137, 132], pulmonary embolism [237, 164], and atherothrombosis [299, 59], where a key concern is thrombus failure and subsequent shedding of emboli, which can cause life-threatening complications under certain conditions. If we model a thrombus as a porous medium, where fibrin is loosely connected around the core area [287], we can study its interaction with blood flow using mathematical models, i.e., the Cahn-Hilliard and Navier-Stokes Equations [254]. Phase field modeling gives a macroscopic representation of thrombus deformation whereas the microscopic details such as the fibrin network and platelets are neglected. This system of governing equations can still reflect the mechanical interactions of a thrombus and its ambient blood flow [321, 254, 148]. Material properties, e.g., the permeability and viscoelastic modulus, in the governing equations play an important role in thrombus mechanics. Specifically, they can be indicators for the possibility of vessel occlusion and thromboembolism, since pieces of thrombus could be detached by the local shear stress and be transported to distal vessels [51]. Therefore, there is a pressing need to infer material properties from measurements, which is central to predicting thrombus shape and deformation under diverse hemodynamic conditions and assessing the risk of thromboembolism and other clinical consequences. Similar estimation of unknown parameters from data is also a central problem in electrocardiology and medical image reconstruction [162, 111, 89, 231], geophysics [175, 259, 233, 297], and many other fields [249, 288, 20, 46, 144].

However, the values of permeability and viscoelasticity are patient-specific and difficult to quantify from either experimental measurements or traditional

numerical simulations using the finite element or finite volume method. A variety of numerical methods for inverse problems, i.e., Bayesian approaches [43, 293, 33], smoothing approaches [216, 142], and adjoint methods [203, 171, 187], have been developed to infer PDE parameters from data.

Recent advances in solving inverse problems using deep learning techniques provide us with a promising alternative to identify PDE parameters [176]. In particular, physics-informed neural networks (PINNs) [213, 214] is a relatively simple framework, which encodes the information from governing equations describing physical conservation laws. Specifically, the residuals of physics equations are encoded into the loss function of the neural network as constraints such that the network outputs satisfy the PDE equations as well as the initial and boundary conditions. Conceptually, adding these physical constraints restricts the optimizing weights and biases in a constrained space. Also, PINNs is a mesh-less framework since partial derivatives can be computed with automatic differentiation (AD) in most neural network packages. As a result, the residuals of PDEs can be evaluated at random points in the spatio-temporal domain for training effectively and efficiently. Additionally, for forward problems, the training data is unpaired; PINNs does not require any data other than the spatio-temporal coordinates of training points (and the initial/boundary conditions). Successful applications of PINNs range from flow visualization [214] to high-speed flows [166], stochastic PDEs [314], fractional PDEs [195], and cardiac flows [128], to name a few. For inverse problems with unknown parameters in PDEs, PINNs can infer even hundreds of parameters based only on measurements with a limited number of training points and without any prior knowledge of the unknown parameters [214]. In PINNs, solving inverse problems follows the same workflow as forward problems only by penalizing the difference between point measurements and model

predictions to the loss function. Unknown values of the parameters are set as model variables such that they can be optimized based on the gradients of the loss function with respect to their value. The potential of PINNs to infer parameters or their distributions has been explored for highly-nonlinear [213], stochastic [314], ill-posed [214, 213], multi-fidelity problems [173], non-homogeneous material properties inference [317], and other cases [251, 42].

In this chapter, we apply PINNs to identify values of two parameters, namely, the permeability and viscoelastic modulus in the Cahn-Hilliard and Navier-Stokes equations. This is perhaps the first attempt to leverage the power of PINNs as a new method to infer physiological parameters using high-order multi-physics and multi-field nonlinear PDEs. In addition, to tackling the challenge of calculating the fourth-order derivative in the Cahn-Hilliard equation with AD, we introduce an auxiliary network along with the main neural network to approximate the second-derivative of the energy potential term. Moreover, we investigate the effects of the number of training points, the influence of noisy data, and different types of data on the accuracy of our inferred results.

The remainder of the chapter is organized as follows: In section 4.3, we present the Cahn-Hilliard and Navier-Stokes system of equations as well as the PINN model. In section 4.4, we present the fields construction and parameter inference results for representative cases such as a thrombus and a biofilm in a channel. We also explore the sensitivity of the PINN predictions by reducing the number of training data, adding noise, and using partial data from some of the fields. We conclude in section 4.5 with a brief summary.

4.3 Methods

4.3.1 Cahn-Hilliard and Navier-Stokes Equations

Mechanical interaction between thrombus and blood flow as a fluid-structure interaction (FSI) problem can be modeled by the Cahn-Hilliard and phase field coupled Navier-Stokes equations (referred as Navier-Stokes equations) in fully-Eulerian coordinates [321]:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}\right) + \nabla p = \nabla \cdot (\boldsymbol{\sigma}_{vis} + \boldsymbol{\sigma}_{coh} + \boldsymbol{\sigma}_{ela}) - \mu \frac{(1 - \phi)\mathbf{u}}{2\kappa(\phi)}, \quad (4.1)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (4.2)$$

$$\frac{\partial \psi}{\partial t} + \mathbf{u} \cdot \nabla \psi = 0, \quad (4.3)$$

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = \tau \Delta \omega, \quad (4.4)$$

$$\omega = \Delta \phi + \gamma g(\phi), \quad (4.5)$$

where $\mathbf{u}(\mathbf{x}, t)$, $p(\mathbf{x}, t)$, $\boldsymbol{\sigma}(\mathbf{x}, t)$, and $\phi(\mathbf{x}, t)$ represent the velocity vector, scalar pressure, stress tensor, and scalar phase field; $g(\phi)$ equals the derivative of the double-well potential $(\phi^2 - 1)^2/4h^2$, where h is the interfacial length; $\psi = [\psi_1, \psi_2]$ denotes the auxiliary vector whose gradients are the components of the deformation gradient tensor $\mathbf{F}$ as follows:

$$\mathbf{F} := \begin{bmatrix} -\frac{\partial \psi_1}{\partial y} & -\frac{\psi_2}{\partial y} \\ \frac{\partial \psi_1}{\partial x} & \frac{\partial \psi_2}{\partial x} \end{bmatrix}.$$

Equation (1) is the Navier-Stokes equation with viscous, elastic, and cohesive stresses, respectively, which can be written as:

$$\boldsymbol{\sigma}_{vis} = \mu \nabla u, \quad (4.6)$$

$$\boldsymbol{\sigma}_{ela} = \nabla \cdot \left(\lambda_e \frac{(1 - \phi)}{2} (\mathbf{F}\mathbf{F}^T - \mathbf{I}) \right), \quad (4.7)$$

$$\boldsymbol{\sigma}_{coh} = \lambda \nabla \cdot (\nabla \phi \otimes \phi). \quad (4.8)$$

Equation (2) is the continuity equation and equation (3) denotes the transport of ψ . The fourth-order Cahn-Hilliard equation is decoupled into two second-order equations in equations (4) and (5) for formulating the weak form; γ , τ , and λ are the interfacial mobility, relaxation parameter, and mixing energy density, respectively. Note that the material parameters of interest are viscoelastic modulus λ_e and permeability $\kappa(\phi)$, which are to be determined from the data by PINNs. Other PDE parameters are assumed as known.

We impose Dirichlet boundary conditions $\mathbf{u} = g$, $(\mathbf{x}, t) \in \Gamma_i \times (0, T)$ for velocity at the inlet Γ_i , and no-slip boundary on the wall Γ_w . Neumann boundary conditions, i.e., $\frac{\partial \phi}{\partial \mathbf{n}} = \frac{\partial \omega}{\partial \mathbf{n}} = \frac{\partial \psi}{\partial \mathbf{n}} = 0$, $\mathbf{x} \in \Gamma_w \cup \Gamma_i \cup \Gamma_o$ are imposed for ψ , ϕ , and ω on all boundaries, and for pressure and velocity at the outlet Γ_o . This model is feasible for both 2D and 3D but we only consider two-dimensional (2D) physical domain in this chapter for proof of concept demonstration.

4.3.2 Physics-Informed Neural Networks (PINNs)

In Fig. 4.1 we show a schematic of PINNs. Given the time t and coordinates x , y of training points as inputs, we construct two fully-connected neural networks,

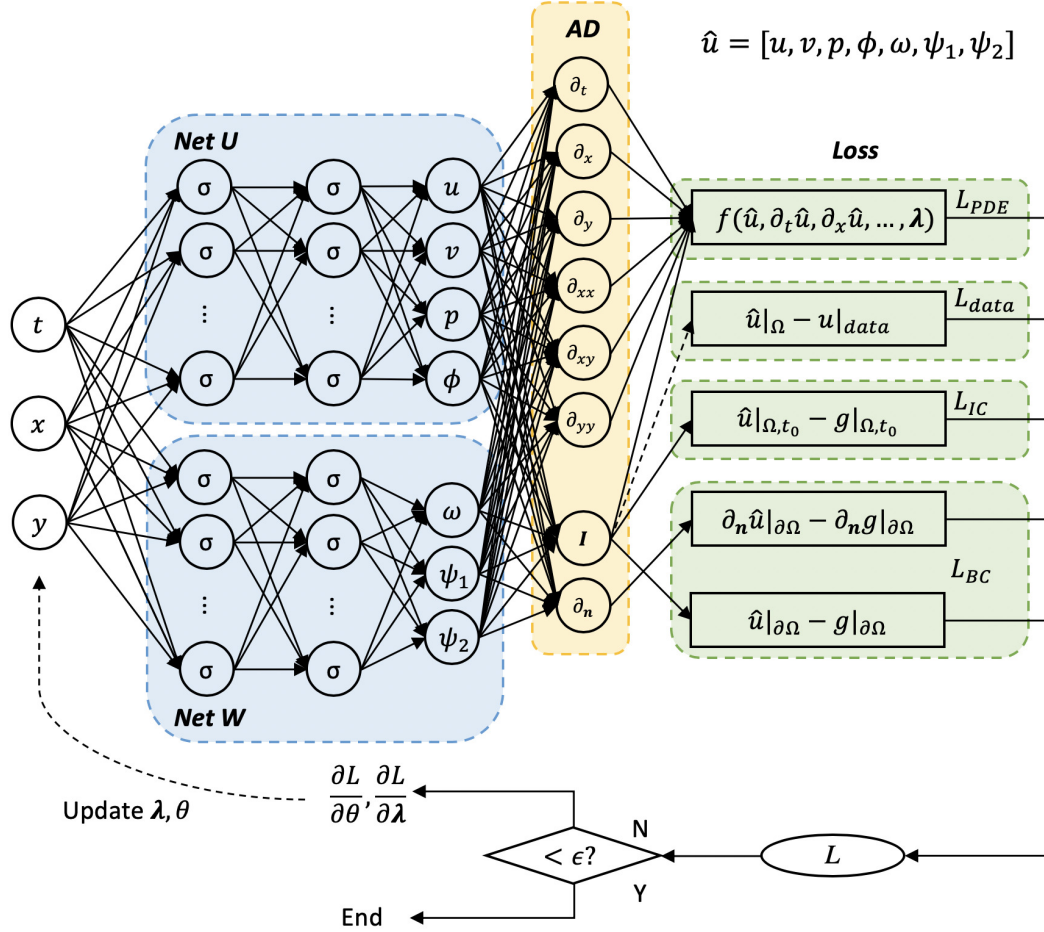


Figure 4.1: Schematic of a PINN for solving inverse problem for Cahn-Hilliard and Navier-Stokes PDEs. Circled by blue boxes, *Net U* and *Net W* represent surrogate models for the PDEs solution whose derivatives can be computed with automatic differentiation (AD). The computed derivatives are used in the loss function to restrict model outputs such that they satisfy the system of PDEs in Ω . For inverse problems, the residual between sensor measurements $u|_{data}$ and model outputs $\hat{u}_{\Omega}$ are included in the loss function. We use ADAM to optimize the model parameters θ (weights and biases) and search the unknown values of the material parameters from the PDEs λ to minimize the loss function.

Net U and *Net W*, where the outputs of *Net U* represents a surrogate model for the PDE solutions u , v , p , and ϕ and the outputs of *Net W* are PDE solutions ω , ψ_1 , and ψ_2 . We denote the PDE solutions as $\hat{\mathbf{u}}$ concatenated by the outputs from *Net U* and *Net W*, whose derivatives with respect to the inputs are calculated using AD. Then, we formulate the total loss L as the combination of PDEs residual loss (L_{PDE}), initial and boundary condition loss (L_{IC} , L_{BC}), and data loss L_{data} as follows:

$$L = \omega_1 L_{PDE} + \omega_2 L_{IC} + \omega_3 L_{BC} + \omega_4 L_{data}, \quad (4.9)$$

and

$$L_{PDE}(\theta, \boldsymbol{\lambda}; X_{PDE}) = \frac{1}{|X_{PDE}|} \sum_{\mathbf{x} \in X_{PDE}} \|f(\mathbf{x}, \partial_t \hat{\mathbf{u}}, \partial_x \hat{\mathbf{u}}, \dots, \partial_{xx} \hat{\mathbf{u}}, \dots; \boldsymbol{\lambda})\|_2^2, \quad (4.10)$$

$$L_{BC}(\theta, \boldsymbol{\lambda}; X_{BC}) = \frac{1}{|X_{BC}|} \sum_{\mathbf{x} \in X_{BC}} \|\mathfrak{B}(\hat{\mathbf{u}}, \mathbf{x})\|_2^2, \quad (4.11)$$

$$L_{IC}(\theta, \boldsymbol{\lambda}; X_{IC}) = \frac{1}{|X_{IC}|} \sum_{\mathbf{x} \in X_{IC}} \|\hat{\mathbf{u}} - \mathbf{u}_{t_0}\|_2^2, \quad (4.12)$$

$$L_{data}(\theta, \boldsymbol{\lambda}; X_{data}) = \frac{1}{|X_{data}|} \sum_{\mathbf{x} \in X_{data}} \|\hat{\mathbf{u}} - \mathbf{u}_{data}\|_2^2, \quad (4.13)$$

where ω_1 , ω_2 , ω_3 , and ω_4 are the weights of each term. The training sets X_{PDE} , X_{BC} , and X_{IC} are sampled from the inner spatio-temporal domain, boundaries, and initial snapshot, respectively. X_{data} is the set that contains sensor coordinates and point measurements; $|\cdot|$ denotes the number of training data in the training set. In particular, $\mathfrak{B}$ represents a combination of the Dirichlet and Neumann residuals at boundaries. Finally, we optimize the model parameters θ and the PDE parameters $\boldsymbol{\lambda} = [\lambda_e, \kappa]$ by minimizing the total loss $L(\theta, \boldsymbol{\lambda})$ iteratively until the loss satisfies the stopping criteria. Optimizing the total loss is a searching process for $\boldsymbol{\lambda}$ such that the outputs of the PINN satisfy the PDE system, initial/boundary conditions, and point measurements. We use the mean relative L_2 error (ϵ), same as in [214], to

quantify errors between reference data and model predictions:

$$\epsilon := (\frac{1}{N} \sum_i^N [\hat{u}(\mathbf{x}_i) - u(\mathbf{x}_i)]^2) / (\frac{1}{N} \sum_i^N [u(\mathbf{x}_i) - \frac{1}{N} \sum_i^N u(\mathbf{x}_i)]^2) \quad (4.14)$$

4.4 Results

To demonstrate the inference ability of the PINN model, we adopt four representative cases for parameters inference. The high-resolution training datasets are generated from the spectral/*hp* element solver *NEKTAR* [117] coupled with the Cahn-Hilliard equations with 3rd-order Jacobi polynomials. For the neural network architecture, our preliminary results suggested that using 9 hidden layers with 20 neurons per layer for *Net U* and *Net W* could be a good balance between the network representation capacity and computational costs. We use the ADAM optimizer [126] with learning rate 0.001 to train the model for a number of epochs, which is defined as the number of complete passes through the full training dataset.

The two material parameters of interest are normalized as follows. For permeability κ , we follow the normalization in [287], where $\kappa = \frac{k_f}{a_f^2}$, k_f is the true permeability and a_f is the fibrin radius. The elastic modulus λ_e of the thrombus is normalized by the magnitude of a characteristic velocity ($3.33 \times 10^{-3} m/s$) and length ($3 \times 10^{-4} m$), and it is typically of $O(10 - 10^{-1})$ with units in *Pa* [321, 290].

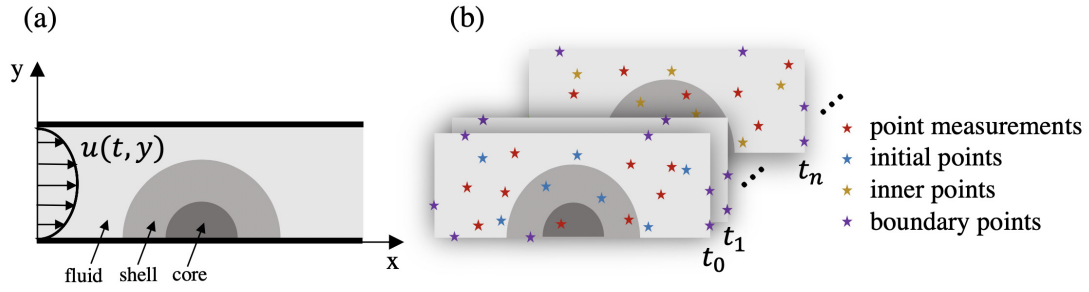


Figure 4.2: 2D flow past a thrombus. (a) The computational domain is a channel with walls on the top and bottom boundaries with the inlet flow $u(t, y)$ entering from the left side.; $\phi=1$ corresponds to fluid. A thrombus with a permeable core $\phi = -1$ and shell $\phi = 0$ is present at the bottom boundary. (b) Sampling points for inferring permeability include initial points (★) at the time t_0 , inner points (★) from t_1 to t_n , boundary points (★) on boundaries, and point measurements (★) with PDE solutions.

4.4.1 Inference of Permeability

Thrombus in a channel with uniform permeability

To infer the permeability κ in the Cahn-Hilliard and Navier-Stokes equations, we perform simulations for a semi-circle permeable thrombus in a channel with a steady parabolic flow coming from the left. This setup represents an idealized thrombus with an impermeable core consisting of a fibrin clot and a permeable shell region consisting of loosely-packed and partially-activated platelets. This model has been adapted as an idealized thrombus in previous works [321, 292, 300]. We impose a Neumann type boundary condition for ϕ and ψ as $\nabla\phi \cdot \mathbf{n} = \nabla\psi \cdot \mathbf{n} = 0$, where $\mathbf{n}$ is the unit vector perpendicular to the boundaries. We set the density $\rho = 1$, viscosity $\mu = 0.1$, $\lambda = 4.2428 \times 10^{-5}$, $\tau = 10^{-6}$, viscoelastic modulus $\lambda_e = 0$, and the interface length $h = 0.05$. These parameters in PINNs are non-dimensionalized values so as to be consistent with the CFD solver. The thrombus is present in the middle of the channel as shown in Fig. 2(a) with a uniform permeability in the core ($\phi = -1$) and in the outer shell layer ($\phi = 0$). In general, the inlet velocity

$u(t, y)$ can be time-dependent, but in this case it is set as steady $0.3(y - 2)y$. In plot (b), we sample coordinates of training data in the initial snapshot t_0 ($\star$), inner spatio-temporal domain from t_1 to t_n ($\star$), and at boundaries ($\star$). Moreover, we also sample point measurements ($\star$) including their coordinates and PDEs solutions in the spatio-temporal domain to calculate the data loss term in the total loss. In this section, the point measurements only contain scalar data from the phase field ϕ as the data source to recover the velocity field and the value of κ . In this chapter, we refer to the phase field point measurements as training data, which are randomly drawn from all points in the inner spatio-temporal domain. For each case, 1,000 points are drawn from an initial snapshot, each containing information on the phase field, velocity field, and pressure field as initial conditions. Since we want the network to satisfy the governing equations in the spatio-temporal domain and satisfy the boundary conditions, we randomly sample 25,000 inner points to compute the PDE residuals and 1,000 boundary points to compute the residuals at boundaries. Noting in particular that point measurements and inner points are drawn from the inner spatio-temporal domain, the former contains the PDE solutions whereas the latter does not. The number of inner points, initial points, and boundary points will remain the same unless specifically mentioned for all the cases in the following context.

In Fig. 4.3, we first show four simulation cases with permeability values from 10^{-3} to 10^2 . We first consider a simple case with the value of κ uniform across the core and shell areas as proof of concept. We randomly sample 10,000 phase field point measurements in the spatio-temporal domain to infer κ for each case. The first column shows the reference data of phase and velocity fields at $t = 0.63$ as ground truth. As shown in Fig. 4.3(a), the thrombus is permeable, thus fluid can penetrate the core and shell of the thrombus. In Fig. 4.3(b)-(d),

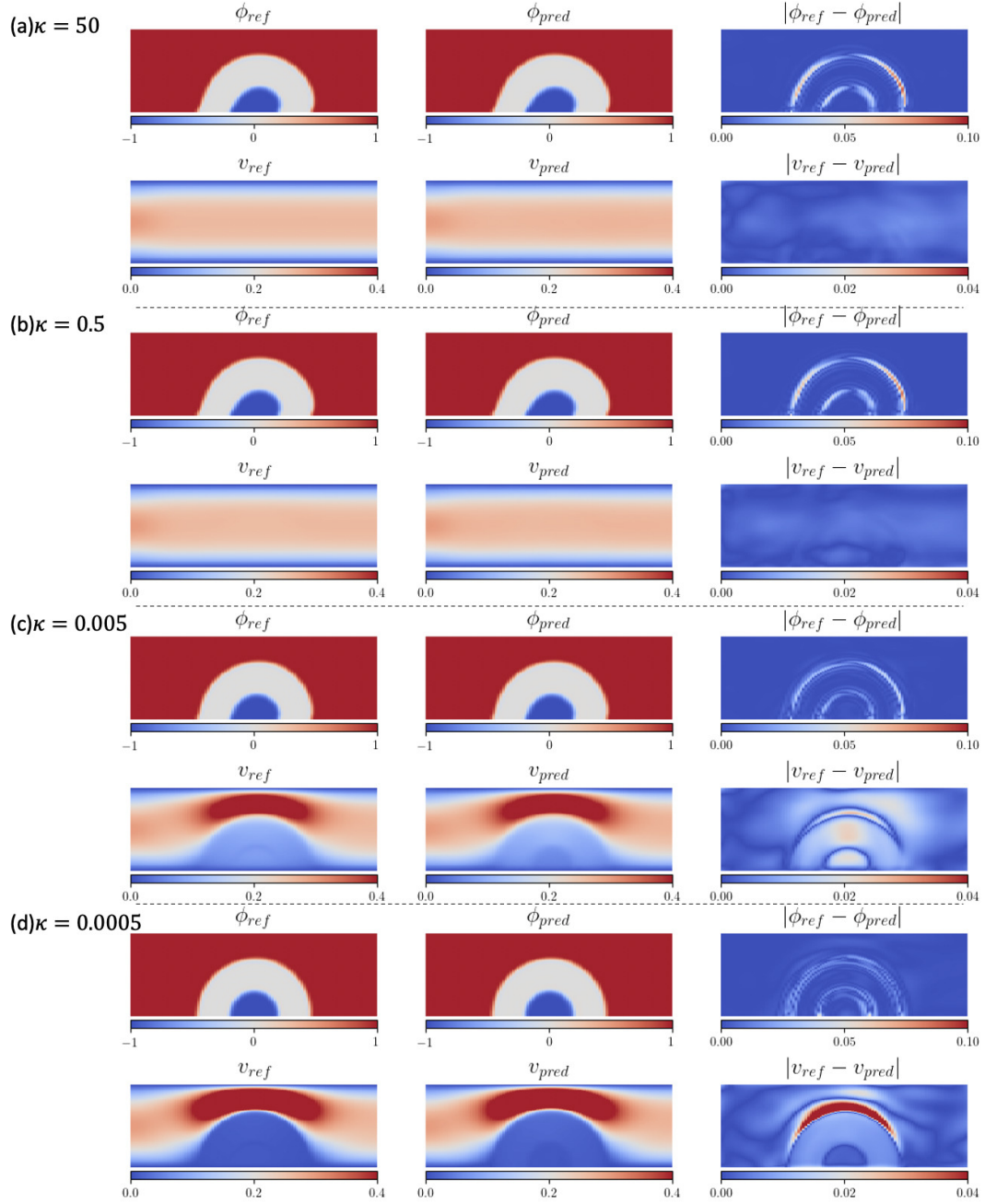


Figure 4.3: Prediction and error of 2D flow past a thrombus for various permeability values at $t = 0.63$. Representative snapshots of the reference data for permeability value κ (a) 50 (b) 0.5 (c) 0.005 (d) 0.0005 are shown against the predicted phase and velocity field. The first column shows the reference simulation results while the second shows the results from the PINNs. The third column shows the absolute value of the difference between references and model predictions. The network is informed by 10,000 point measurements drawn from $t \in [0.33, 0.63]$.

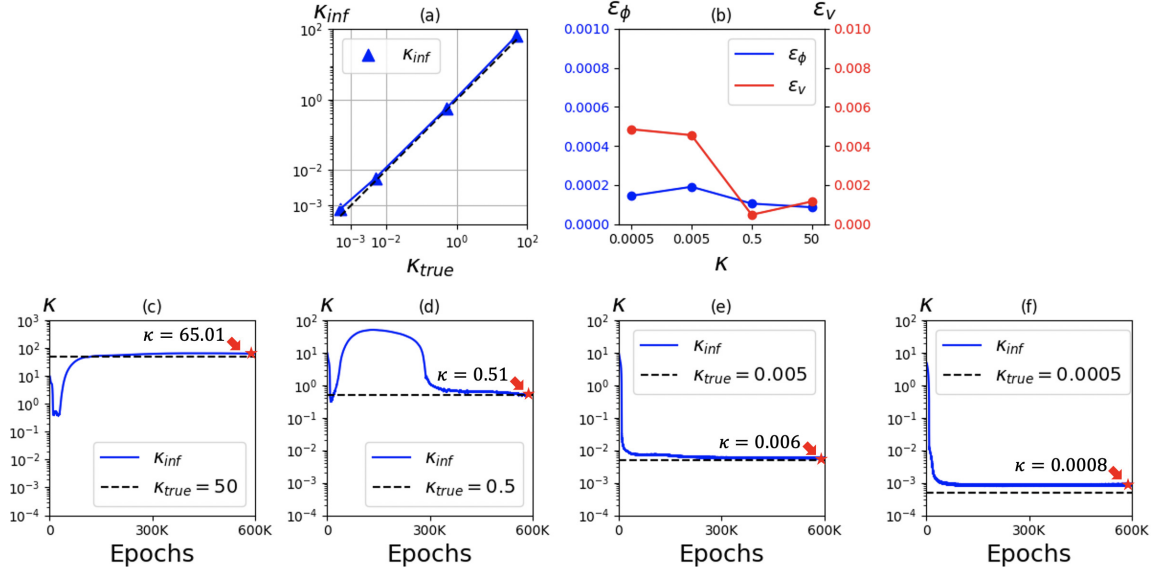


Figure 4.4: Performance and history of PINN on predicting permeability of 2D flow past a thrombus with κ ranging from 0.0005 to 50. (a) Comparison of the inferred values with the reference values. (b) Mean relative L_2 errors for phase and velocity field when κ varies from 0.0005 to 50. The plots in (c, d, e, f) illustrate the inference of the permeability with respect to the number of iterations of PINN when $\kappa = 50$, $\kappa = 0.5$, $\kappa = 0.005$, and $\kappa = 0.0005$. The number of training points is 10,000.

the fraction of fluid in the thrombus falls off dramatically as a result of a decrease of the thrombus permeability and an accelerated area is formed on the top of the thrombus. Unlike the thrombus with large κ , fluid can hardly flow through the impermeable thrombus and the phase deformation becomes relatively small. In other words, the deformation of phase fields is noticeable when κ is large, but is minimal for permeability 0.0005. In Fig. 4.3, the second column depicts the predicted phase and velocity fields, and the third column shows the absolute error. The model predictions agree well with the reference data for both phase and velocity fields. The error in the phase field is mainly distributed on the thrombus interface with a maximum absolute error smaller than 10% for all cases. However, the velocity error grows with the decrease of permeability with the largest local error reaching 10% under the bottleneck region. Overall, the results show that the PINN model can regress PDE solution fields and infer parameters from synthetic data.

We summarize the results of parameter inference, mean relative L_2 errors as well as the history of inference value in Fig. 4.4. Plot (a) shows that all the inferred κ values fall near the diagonal line, indicating a good agreement between the reference and inferred values. Plot (b) shows the mean relative L_2 error for each case: the maximum relative error for velocity prediction is below 0.6% and for the phase field is less than 0.02%. Figs. 4.4 (c-f) depict the convergence history of parameter retrieval in the training process to the true different values of κ . For $\kappa = 0.5$ and 0.005, the inferred network predictions converge to the true values with 2% and 20% error. For $\kappa = 50$, the system is insensitive to the variations in κ and *a posteriori* practical identifiability analysis is provided in the Appendix. For $\kappa = 0.0005$, the velocity field has a sharp interface on the top of the thrombus surface. These reasons may lead to a discrepancy between the true and inferred value, causing a relatively large error in this special case.

To investigate the effect of the number of training points on the model prediction ability, we retrain the model for the same cases as in Fig. 4.3 and Fig. 4.4 with a different number of training points from 0.1% up to 40% of the total number of points (100,000). To clarify, the training points in Fig. 4.5 only include point measurements in the inner spatio-temporal domain. Figs. 4.5 (a, d, g, j) show the trend of the inferred values changing with the spatio-temporal resolution of the training data. Generally, the inference results converge toward the true value with mild deviations if the training data is greater than 2,500 training points (2.5% of the total number of points). The second and the third column show the mean of relative L_2 errors for velocity and phase fields. The velocity errors for κ equals 0.5 and 50 are one order smaller than those of small permeability, indicating better predictions when κ is large. The phase field errors are all lower than 1% if the training data is above 2.5% of the total number of points. Hence, for this case, we

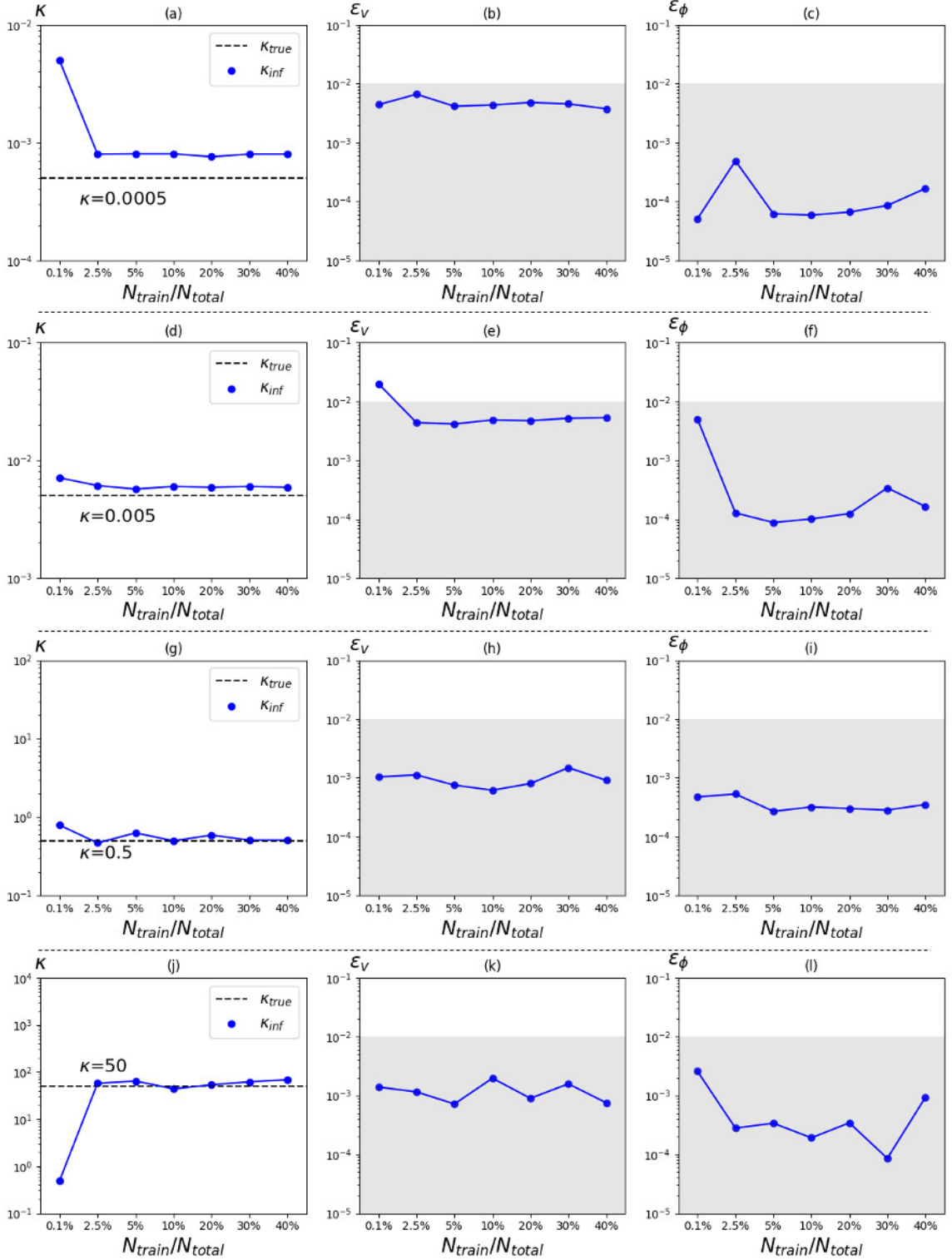


Figure 4.5: Effect of the number of training points on the accuracy of PINNs. $N_{total} = 100,000$. Plots (a, d, g, j) show comparison of reference and inferred permeability for a different number of training points from 0.1% of the total number of points to 40%. The second and the third column show the mean of relative L_2 error for (b, e, h, k) velocity field and (c, f, i, l) phase field over time for different numbers of training points. We shaded the area where the error is lower than 1%.

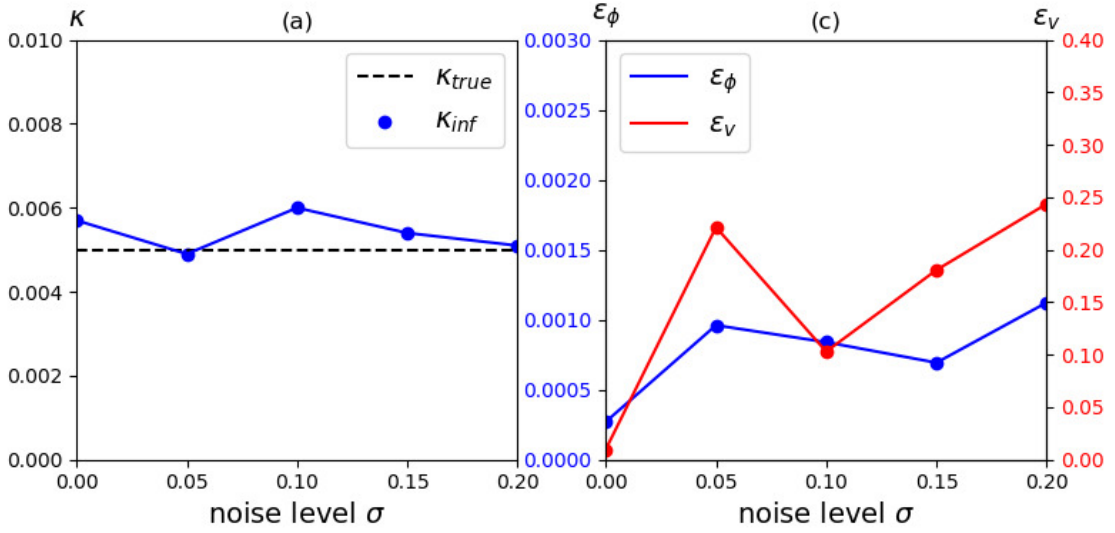


Figure 4.6: Predictions of 2D flow past a thrombus trained with noisy phase field measurements. (a) Inferred permeability and (c) mean relative L_2 errors between the model predictions and the reference for phase ϵ_ϕ and velocity field ϵ_v . Noise is added to the phase field with the noise level ranged from 0 to 20%. Here, 10,000 data points are scattered in the spatio-temporal domain as the training data to infer the permeability.

conclude that it is sufficient to guarantee convergence and good results if 2.5% of points are used to make inferences.

Furthermore, to validate the robustness of our model to noisy measurements, we add white noise $\mathcal{N}(0, 1)$ to the input data, i.e., phase field ϕ , in the following way:

$$\hat{\phi} = \text{clip}(\phi + \sigma \mathcal{N}(0, 1)), \hat{\phi} \in [-1, 1], \quad (4.15)$$

where σ is the noise level. We add normal-distributed white noise signals to the reference data ϕ and impose a clip function to restrain the value of ϕ within -1 and 1. We test the value of the noise level σ up to 20% of the variance and train the model with these noisy data. The reference permeability value in these cases is set as 0.01. Fig. 4.6 (a) plots the inferred κ against various noise levels, showing that the inferred value is not affected by the added noise. In plot (b), the relative L_2 errors for phase and velocity field show slight increase as the noise level increases.

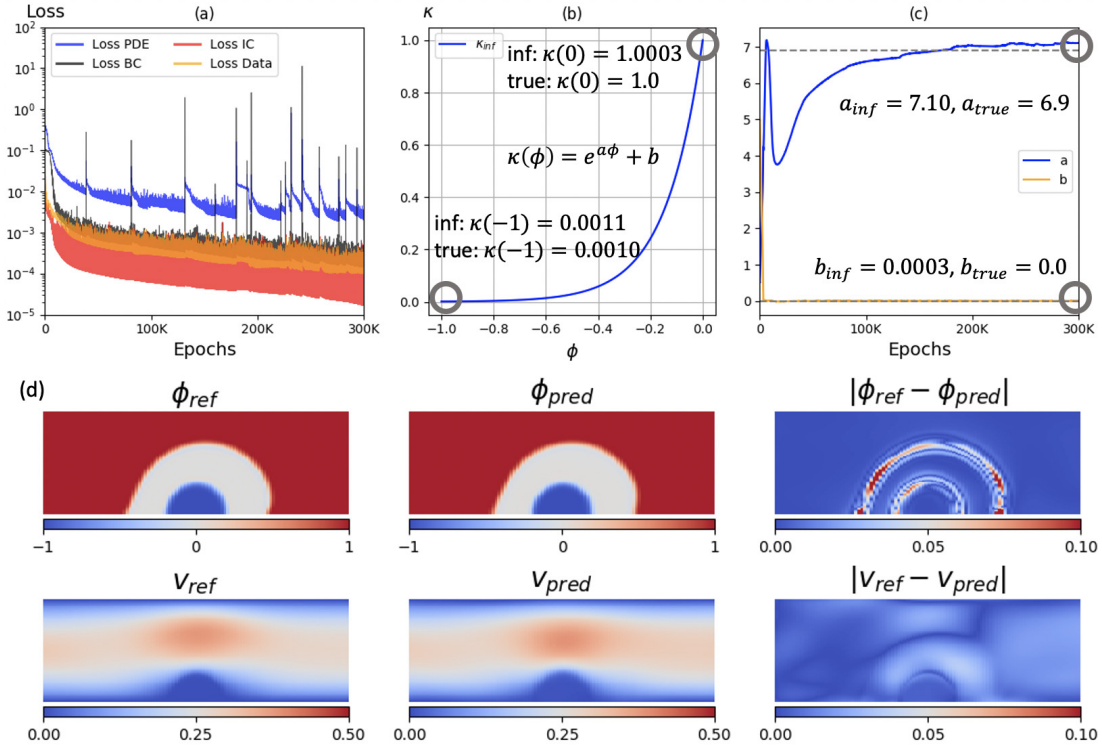


Figure 4.7: 2D flow past a thrombus with phase dependent permeability. (a) History of network losses (Loss PDE, Loss IC, Loss BC, and Loss Data) and (b) inferred the permeability κ as a function of ϕ . (c) The values of a and b at each epoch. (d) Comparison of phase field and velocity field for $\kappa(\phi)$ at $t = 0.78$ and their absolute error. The core permeability is 0.001 and the shell permeability is set 1 as the actual values. 10,000 data points are scattered in the spatio-temporal domain from 30 snapshots ($t \in [0.03, 0.93]$) as the training data to infer the permeability.

In particular, the predictions of the velocity field is more sensitive to the noise as ϵ_v increases from less than 1% at $\sigma = 0$ to 25% at $\sigma = 0.20$. In general, the good agreement in parameter inference and small increment in field predictions demonstrate strong robustness of the PINN model to noisy data.

Thrombus in a channel with space-dependent permeability

Unlike the aforementioned idealistic case, in a real thrombus, the permeability varies spatially depending on the volume fraction (ϕ), with the core area much less permeable than the outer shell. To validate the inference ability of the PINN

for a space-dependent permeability, we test another case with the $\kappa = 1$ for the shell area and $\kappa = 0.001$ for the core area as shown in Fig. 4.7. Since the core area is hardly permeable while the shell has a larger κ , we expect a non-uniform displacement from the thrombus core and shell as the outer layer moves with ambient flow and the inner layer stays still. To express such spatial variation explicitly, we consider a relation between ϕ and κ in this case:

$$\kappa(\phi) = e^{a\phi} + b, \quad (4.16)$$

where a and b are model parameters to be optimized in the PINN model and the true values of a and b are 6.90 and 0.0. Since the fluid phase ($\phi = 1$) does not have permeability, we only consider κ for the core ($\phi = -1$) and shell layer ($\phi = 0$).

In Fig. 4.7 we present the history of the different losses, namely PDE loss, boundary condition loss (Loss BC), initial condition loss (Loss IC), and data loss (Loss Data) in (a). Plot (b) shows the inference result for κ as a function of ϕ ; the permeability at core area ($\kappa(\phi = -1) = 0.0011$) and shell area ($\kappa(\phi = 0) = 1.0003$) match the reference values well. Plot (c) shows that the model parameter quickly converges to the true values.

We present the reference data ϕ_{ref} , v_{ref} and the model predictions ϕ_{pred} , v_{pred} and their difference at $t = 0.78$ in plot (d). The field predictions exhibit a good agreement compared with the ground truth data. The errors for the phase field are mainly distributed in and around the outlet layer of the thrombus and the errors in velocity field are mainly confined within the shell layer.

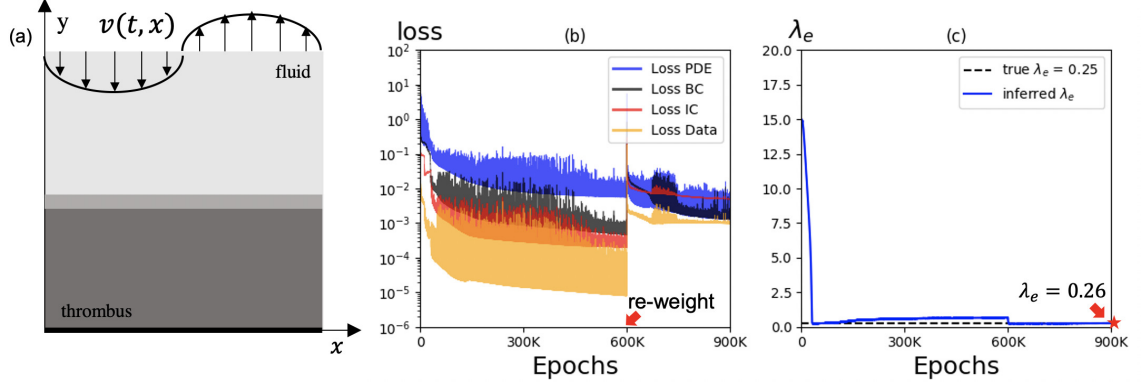


Figure 4.8: (a) Schematic for viscoelastic thrombus in a cavity, (b) history losses for each term, and (c) inference of λ_e . (a) We impose a time-dependent sinusoidal vertical velocity $v(t, x) = -(1 - \cos(2\pi t)) \sin(2\pi x)$ at the top boundary, and set the left and right sides as periodic. We changed the weight for each loss term at epoch 600,000. Training data points are scattered among the spatio-temporal domain to train the network, including 20,000 phase field points from the inner spatio-temporal domain, 2,000 pressure measurements at boundaries, and 1,000 initial points. **Loss PDE**: loss for the PDEs residuals, **Loss BC**: loss for boundary conditions, **Loss IC**: loss for initial conditions, **Loss Data**: loss for measurements data.

4.4.2 Inference of viscoelastic Modulus

The viscoelastic modulus is another important parameter that has to be estimated indirectly. There are few rheometry experiments that measure the viscoelastic shear modulus λ_e with oscillatory shear deformation. We assume homogeneity and isotropy of the thrombus for simplicity. To explore the viability of parameter inference from imaging data, we consider two typical setups as illustration: a thrombus in a cavity and a biofilm in a channel.

Viscoelastic thrombus in a cavity

We first consider a viscoelastic thrombus in a 1×1 cavity as shown in Fig. 4.8 (a). This example represents a benchmark test for simulating the deformation of a viscoelastic material. The top layer with the light gray color denotes the fluid phase while the bottom layer with the darker color indicates the initial state of

the viscoelastic thrombus. We impose a time-dependent sinusoidal vertical flow $v(t, x) = -(1 - \cos(2\pi t)) \sin(2\pi x)$ at the top boundary. At the left and right boundary, we set the Dirichlet boundary $\phi(y) = \tanh(y - 0.5) / \sqrt{2}h$ and the periodic boundary for velocity; we also set $\nabla\phi \cdot \mathbf{n} = 0$ at the top boundary and $\phi = -1$ on the bottom wall. We sample 20,000 phase field points and 2,000 pressure points at boundaries from 20 consecutive snapshots ($t \in [0.03, 0.63]$) in addition to 1,000 initial points. To train this neural network, we only utilize the phase field information and some pressure measurements at the boundaries as data sources. Such data acquisition does not require information other than the phase field from the inner spatio-temporal domain, such as the pressure or auxiliary vector field ψ , and hence it can potentially be used in a real experimental setup. The weights are chosen as followed:

$$\omega_1 = \omega_3 = 1, \omega_2 = \omega_4 = 5, \text{ epoch} \in [1, 600,000], \quad (4.17)$$

$$\omega_1 = 10, \omega_3 = \omega_2 = \omega_4 = 1, \text{ epoch} \in [600,001, 900,000], \quad (4.18)$$

We set $\rho = 1$, $\mu = 0.01$, $h = 0.02$, $\lambda = 2.5 \times 10^{-9}$, and $\tau = 10^{-4}$.

In Fig. 4.8 we present in plot (b) the history of the loss for each term and in plot (c) the inferred value of the viscoelastic modulus. In plot (b), the PDE loss (blue line) converges around 10^{-3} and the other losses balanced at the same order with the PDE loss after redistributing the weights at epoch 600,000. Another result of changing the weights is that the inferred value for λ_e converges closer to the reference value 0.25.

Fig. 4.9 compares the reference data and the model predictions at time $t = 0.48$. Phase, velocity, and pressure fields are plotted respectively on each row, and the last column plots the absolute difference between the data and predictions. We can

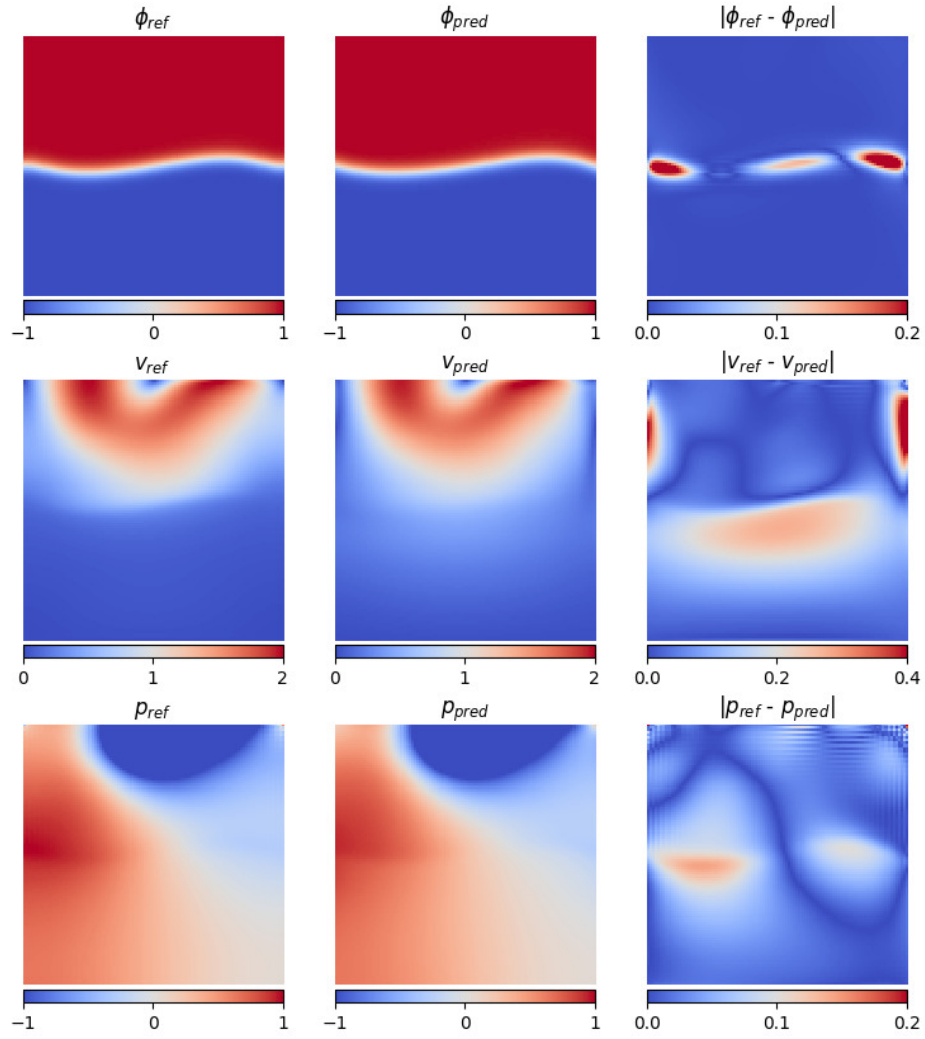


Figure 4.9: Viscoelastic thrombus in a cavity. The first column presents the phase field distribution ϕ , velocity field v , and pressure distribution p from reference data, respectively. The second column shows the same field predictions from the model at $t = 0.48$. The absolute difference between the data and the model predictions are plotted in the third column.

	ϵ_v	ϵ_ϕ	ϵ_p	ϵ_{ψ_1}	ϵ_{ψ_2}
Full	3.511×10^{-2}	1.157×10^{-3}	2.391×10^{-3}	2.636×10^{-2}	2.450×10^{-2}
Half	4.494×10^{-3}	4.129×10^{-4}	1.057×10^{-3}	2.516×10^{-2}	2.236×10^{-2}

Table 4.1: Summary of the mean relative L_2 error for the thrombus in a cavity over half and full time window. Training points in the cavity problem are sampled from $t \in [0.03, 0.63]$. The half window relative errors are defined as the errors from $t \in [0.03, 0.33]$, whereas the full window relative errors are defined as the errors from $t \in [0.03, 0.63]$

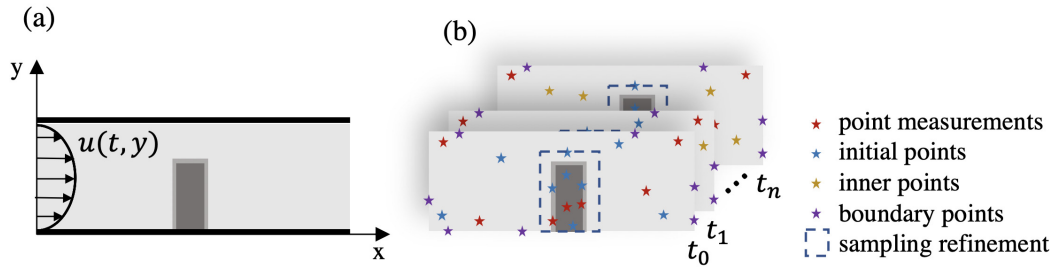


Figure 4.10: 2D flow past a viscoelastic biofilm setup. (a) The computational domain is a channel with wall boundaries on the top and bottom sides, and with flow $u(t)$ entering from the left side. A viscoelastic biofilm is present at the bottom boundary. (b) Training points include initial points ($\star$) at time t_0 , inner points ($\star$) from t_1 to t_n , boundary points ($\star$) on the boundaries, and point measurements ($\star$) with PDE solutions. The sampling points are refined within the dashed line area to improve the results of the PINN model.

observe excellent inferred results for ϕ , p , and v with some minor discrepancies at the interface layer and top periodic layer. Additionally, our model renders high-resolution results in fields construction as can be seen in the summary of the mean relative L_2 error in Table 4.1. The first row shows the mean errors for each field over the full snapshots, while the second row lists the mean errors over the first half among all snapshots. As the data indicate, the errors increase as the system develops since the full time window errors are all greater than those at the first half time window. But overall, we can conclude that the model infers the fields for each variable with satisfactory accuracy.

Biofilm in a channel

In this section, we consider a thin biofilm sitting in a channel. This case is used to mimic the oscillatory shear experimental setup [321] whereas the thrombus is detached from the top surface, unlike the original setup. It is also representative for studying the mechanics of thromboembolism caused by plaque formation and rupture. Other biological implications of biofilms can be found in [254], where biofilms are aggregations embedded in a self-produced matrix of extracellular polymeric substances. From a mechanical point of view, we expect to see a different mode of oscillation on a biofilm when changing λ_e [321]. However, this effect may not be distinguishable for a semi-circular thrombus and it will make parameter inference even more challenging. Hence, we choose the biofilm as a benchmark example.

A thin viscoelastic biofilm is present in the middle of a channel with oscillatory flow $u(t, y) = 0.9 \sin(2\pi t)(2y - y^2)$ coming from the left side of the domain. We expect to observe a swinging movement of the biofilm with the oscillatory flow. Similar to the sampling strategy on the inference of permeability, we sample the four types of points from this domain as shown in Fig. 4.10(b): initial points (★), inner points (✦), boundary points (✧), and points with measurements (✶). Since the dynamics is rich in the area indicated by the black dash line, we refine the density of sampling points from t_0 to t_n within the box for better accuracy. For this 2D flow, we set $\lambda = 4.2428 \times 10^{-5}$, $\tau = 0.5$, $\rho = 1$, and $\mu = 0.1$, and the interface width $h = 0.04$.

In the first test, with λ_e and κ unknown, we aim to infer the parameters and recover the whole field from the training data. The model is informed with 15,000

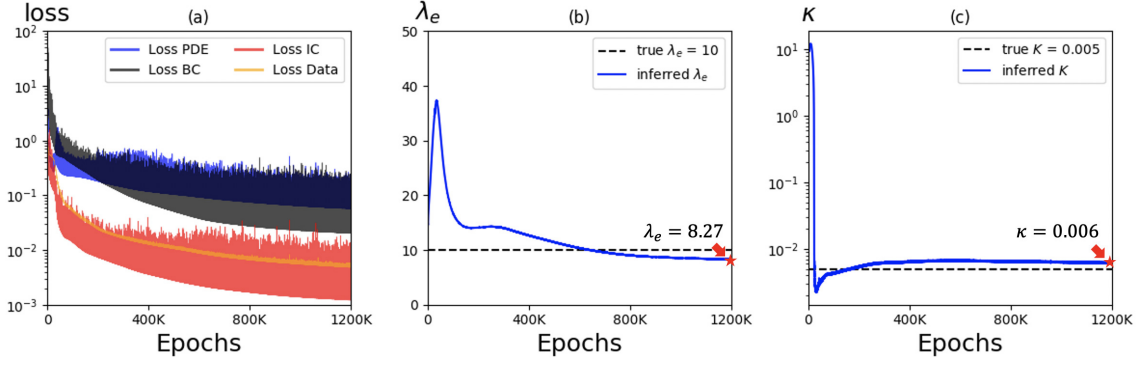


Figure 4.11: History of (a) network losses and inference for (b, c) λ_e and κ against the number of training epochs for the biofilm problem. The full time window is from 0.02 to 1.16.

phase field data in the spatio-temporal domain, 2,000 pressure measurements at boundaries, and 1,000 initial points. Fig. 4.11(a) shows the history of the training losses where the loss PDE is the largest among all losses and the loss for data measurements and initial conditions are the lowest. The inferred values for the two unknown parameters are in plot (b) and (c), indicating that the predicted λ_e and κ converge towards the actual values, at 8.27 and 0.006, as compared to the true value of $\lambda_e = 10$ and $\kappa = 0.005$. To show the regressed fields, we compare the reference data and model predictions for phase and velocity fields at times $t_0 = 0.02$, $t_1 = 0.44$, and $t_3 = 0.86$ in Fig. 4.12. The first column of Fig. 4.12 shows the actual distribution of phase and velocity fields from the reference data, and the second column shows the regressed fields from the PINN model. In plot (a), we observe overall good phase predictions in the second column with some minor smoothing effects around the sharp interface of the biofilm. Plot (b) presents the comparisons for the velocity field, and we observe that the fluid is forced to pass from the top of the biofilm, with local acceleration because of the impermeability of the biofilm. The model predictions v_{pred} show the capability of the PINN for capturing such effects and regressing the velocity field. The absolute error for the velocity field is generally below 10% with larger differences at the flow restricted area and close to the bottom boundary. We summarize the mean relative L_2 error

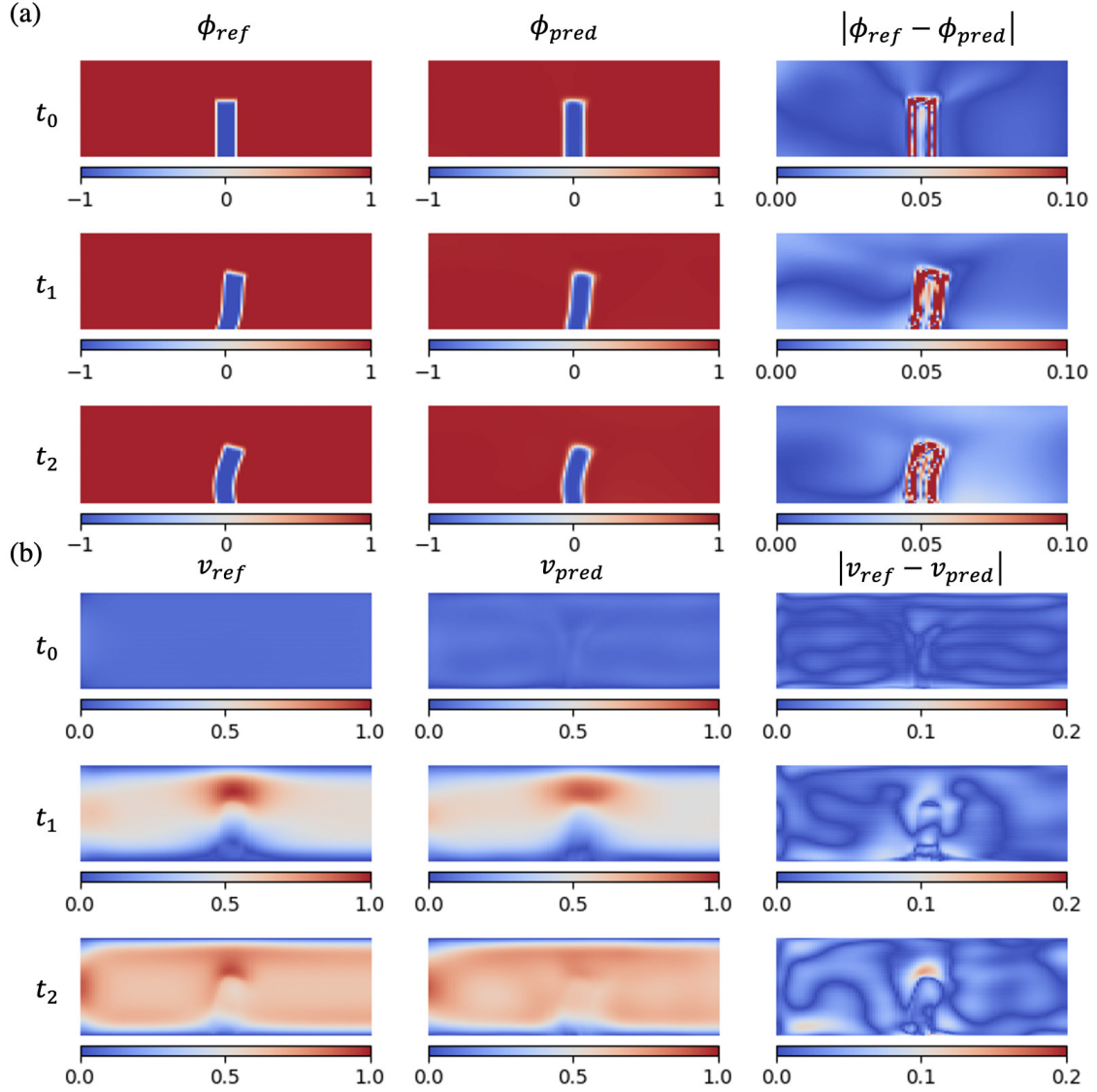


Figure 4.12: 2D flow past a viscoelastic biofilm. With λ_e and κ unknown at the same time, we sample 15,000 phase field data, 2,000 pressure points at boundaries, and 1,000 initial points as the training data for parameters inference and field regression. The time is from 0.02 to 1.16. Representative snapshots (at $t_0 = 0.02$, $t_1 = 0.44$, and $t_2 = 0.86$) of the reference (a) phase field and (b) velocity fields are shown against the predicted phase and velocity from the model. The first column shows the reference fields from simulation, the second shows the predicted results from PINNs, and the third column shows the absolute value of the difference between the references and the model predictions.

	ϵ_v	ϵ_ϕ	ϵ_p	ϵ_{ψ_1}	ϵ_{ψ_2}
Full	2.396×10^{-2}	9.700×10^{-3}	2.260×10^{-4}	2.665×10^{-3}	7.453×10^{-4}
Half	1.417×10^{-2}	8.893×10^{-3}	9.090×10^{-5}	1.490×10^{-3}	4.016×10^{-4}

Table 4.2: Summary of the mean relative L_2 error for the biofilm problem over half and full time window. Training points in the biofilm problem are sampled from $t \in [0.02, 1.16]$. The half window relative errors are defined as the errors from $t \in [0.02, 0.56]$, whereas the full window relative errors are defined as the errors from $t \in [0.02, 1.16]$.

in Table 4.2; the mean error for the first half time window is relatively smaller than the full time.

However, there still exists a relatively large error for inferred value of λ_e . There are three possible reasons. First, the dynamics are not very sensitive to the variation in λ_e . In the Appendix (Figure 15), we perform *a posteriori* sensitivity analysis and compared the phase and velocity field with λ_e at 15 and 0.1, demonstrating that the variation in elastic modulus mainly affects the shape and velocity around the biofilm interface. Hence, it is crucial to have an accurate prediction of the moving boundary of this fluid-structure system. Second, the interface around the biofilm is relatively sharp with rectangular corners, which poses additional difficulties in capturing the detailed but important interface dynamics. On the other hand, PINNs perform well when on inferring λ_e when the system with similar governing equations has a less sharp interface in Fig. 4.9. Lastly, the Cahn-Hilliard equation is essentially a fourth-order PDE system, hence, posing a challenge for neural networks to approximate such a high-order system.

Furthermore, we assess to the inference ability with different numbers and types of data. The number of training points (only includes phase field point measurements) is indicated by its ratio to the total number of points. We set κ as a known parameter to focus on the modulus λ_e . The first two rows in Fig.

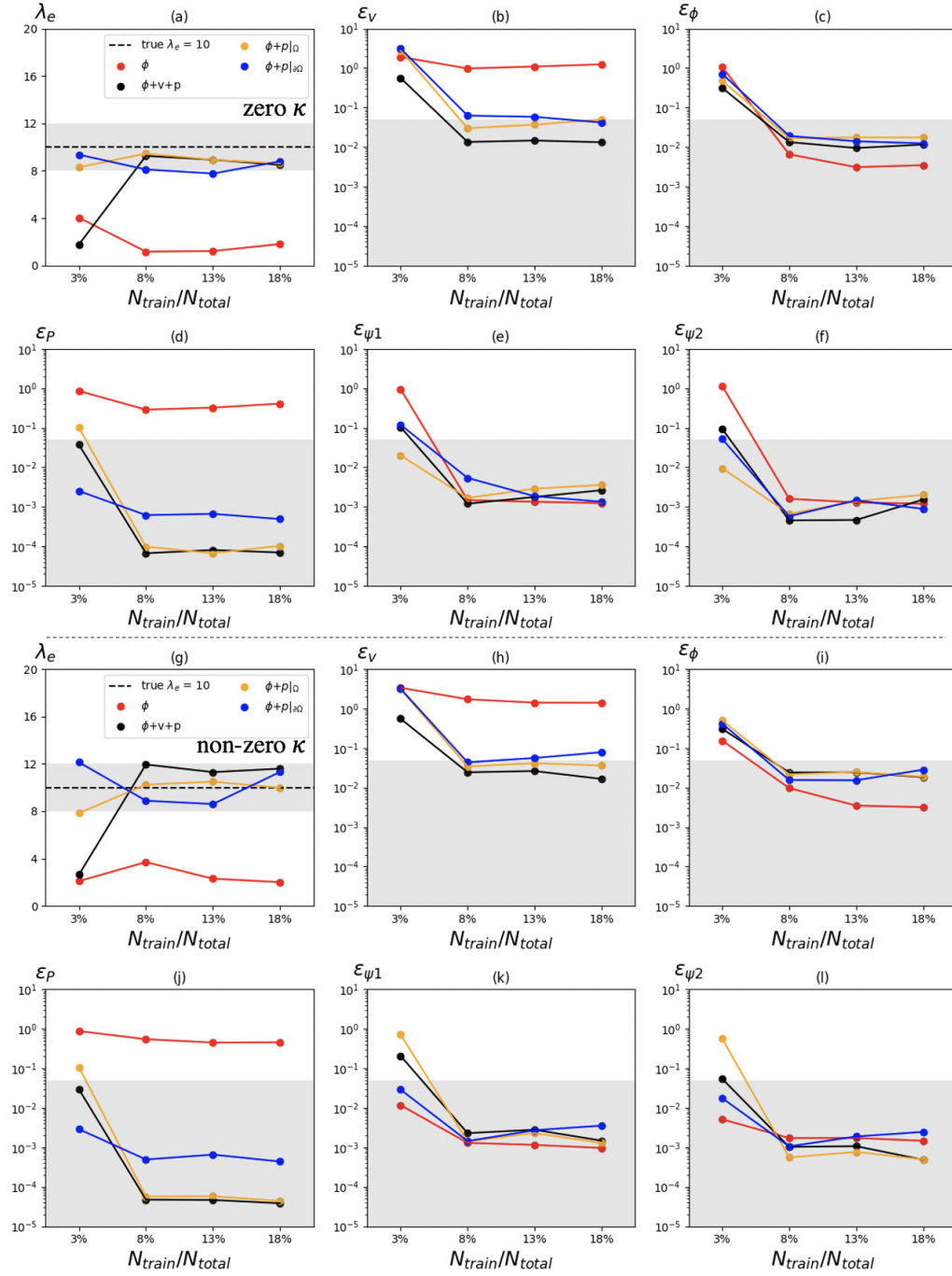


Figure 4.13: Effect of the number of training points on the prediction value of λ_e with κ value as known. (a, g) Predicted λ_e versus the number of training points for case with zero κ and non-zero κ . (b-f, h-l) The mean relative error L_2 versus the number of training points for each field when $\kappa = 10$ and $\kappa = 0.01$. We also plotted the inference results and L_2 error given various training data sources such as $\phi + u + p$, $\phi + p$, and only with ϕ . For the one with pressure information and phase field, inner (Ω) and boundary (∂) pressure measurements are used respectively to train the neural network. We shaded areas with lower than 5% of error. $t \in [0.02, 1.16]$, $N_{total} = 200,000$.

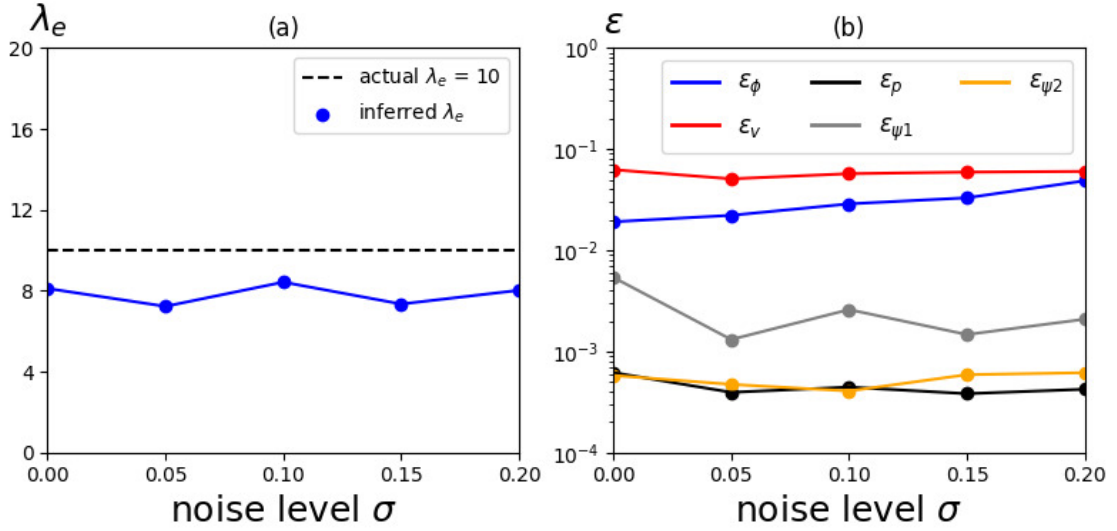


Figure 4.14: 2D flow past a viscoelastic biofilm trained with noisy measurements with λ_e unknown. (a) The inferred λ_e . (b) presents the mean relative L_2 errors between the model predictions and the references for each field. Noise is added to the phase field with the noise level ranged from 0 to 20%. Here, 16,000 (8%) data points are scattered in the spatio-temporal domain.

4.13 present the inferred λ_e and mean of relative L_2 error for each field for the biofilm problem. The third and fourth rows show the same results for $\kappa = 0.01$. The inferred modulus converges to the true value with an increasing amount of data used in training. Hence, the more training points and more data sources we use, the more accurate the results. Moreover, we shaded the 20% error region between $\lambda_e = 8$ and 12 in (a, g) which includes most of the points for yellow, black, and blue lines. By contrast, the errors in parameter inference exhibit a poor performance if only the phase field data are employed to train the neural network. Additionally, we also compare the change of the errors for each variable with the increasing number of training points in (b-f) and (h-l), and almost all points fall below or on the verge of the 5% error region shaded with gray except the training with too small training data or modality. These results demonstrate that it is sufficient to infer λ_e with a limited amount of data from the phase field and pressure measurements at the boundaries.

Finally, we investigate the inference ability on λ_e of the model to noisy measurements. We repeat a set of similar noisy tests as that in permeability inference. Results in Fig. 4.14 are obtained with 16,000 (8%) points measurements with the maximum noise level at 20%. The noise level is similar to that defined in equation 4.15 where we denote the percentage with respect to the noise variance. Fig. 4.14 (a) shows the inferred viscoelastic modulus at various noise levels. The inferred value is around 8 with some minor oscillations. Plot (b) illustrates that all the errors exhibit no or very little increase with the increasing intensity in noise level. Hence, we conclude that the PINN shows good robustness to noisy measurements given these findings.

4.5 Discussion

In this chapter, we demonstrate the potential of PINNs to infer material properties of thrombus in a flow field, i.e., permeability and viscoelastic modulus, from authentic data. Such modeling leverages the recent advances in deep learning algorithms for scientific machine computing by penalizing the Cahn-Hilliard and Navier-Stokes equations, which provides a mathematical description for thrombus deformation. Our findings agree well with permeability reference values over a wide range, i.e., from $O(10^{-4})$ to $O(10^1)$, and the model predictions match with the simulation results from the high-order spectral/ hp element method. In particular, only based on the phase field distribution, the PINN model inferred the value of permeability for a thrombus in a channel, suggesting a potential approach to directly estimate material properties from imaging data. For the inference of viscoelastic modulus, we show that it will be sufficient to make the inference given that the phase field data along with some pressure measurements at boundaries

serve as input to the model. We also demonstrated the robustness of model inference with noisy measurements for both parameters. In addition, we successfully use PINNs to address a thrombus with a heterogeneous permeability, i.e., different permeabilities at the core and shell layer.

Phase field modeling is a special case where the concentration field can be converted from gray-scale values of the image pixels through calibration, and material properties can be directly obtained from *in operando* imaging data [320]. In our system, the multiple components and the complex geometry of a thrombus can be represented by the gray values of each pixel, which increases the possibility of using experimental images to learn thrombus properties in the future.

In this research, we have demonstrated that PINNs can regress the entire fields and unknown parameters given only partial measurements (phase field and pressure at boundaries). This provides a novel and viable way to incorporate data from imaging techniques and multi-modality data to train physics-informed deep learning models for field regression and non-invasive parameter inference in biomedical systems. A possible future improvement will be to employ instead of the governing PDEs, the Gibbs energy functional that is minimized to derive the PDEs; this may be advantageous as lower-order derivatives as well as a smaller number of equations is involved. Another future improvement will be to consider a more realistic and representative setup such as inferring parameters for a 3D thrombus with more realistic shape and unsteady ambient flow. Since predictions on the interface have a substantial impact on parameter inference, a domain decomposition based PINN model [108] can be adopted to better capture the dynamics around the interface. Also, solving such PDEs in variational form [125, 124] can help to reduce the highest order.

In terms of current limitations, one foreseeable challenge may come from the low imaging resolution, which causes difficulties to delineate and convert the presence of a shell layer with partially-activated platelet to the gray-scale values of the images. Incorrect quantification of such area can lead to a mismatch between the true and inferred value. Another possible limitation of the current study for biomedical applications is that our approach requires that the system must be represented via explicit governing differential equations, whereas the governing equations for many biological processes are not known a priori. Promising remedies to overcome such challenge are data-driven system identification [28, 165, 230, 238], or variational system identification [277, 278] for discovering hidden governing equations from noisy and sparse data. Also, high dimensionality and lack of boundary/initial information could deteriorate the overall accuracy of the model predictions.

In ongoing work, we are extending our research by exploring the possibility of using imperfect PDE constraints and also noise-filtering techniques for realistic imaging data. Furthermore, although PINNs are less data-hungry compared to traditional data-driven models, the amount of data used in training, from an experimental point of view, is still quite intensive given the small dimensions and limited spatio-temporal resolution of imaging. In particular, the inference of biomechanical properties of a realistic thrombus depend on its internal properties, topology, and ambient flow (possibly surrounded by turbulent flow [242]), which requires considerable data to fully capture such dependencies. *In vivo* or *in vitro* experimental data combining multi-modality imaging techniques, such as ultrasound (US) and optical coherence tomography (OCT), or panoramic-digital imaging correlation (pDIC), can accurately reflect the real vessel geometries and material properties. However, the morphological diversity of a thrombus and

expensive experimental cost make acquiring sufficient experimental data impractical. Multi-fidelity frameworks [158, 173], leveraging data from different modalities, seem to be promising for addressing these issues given the insufficient amount of high fidelity data. Specifically, the high fidelity data can be realistic, but expensive experimental imaging data with actual material properties (i.e., viscoelastic modulus and permeability). On the other hand, considerable low-fidelity *in silico* data represented by computational results from finite-element simulations can be generated. The multi-fidelity machine-learning framework can learn to optimally combine the multi-modality experimental data with simulations to obtain a response surface for the biomechanical properties of a thrombus in response to the morphological and physiological variables. We are currently working on extending this multi-fidelity/multi-modality framework, and we will report our results in a future publication.

Acknowledgment

The work is supported by grant U01 HL142518 National Institute of Health.

4.A *A posteriori* practical identifiability analysis

We perform *a posteriori* sensitivity analysis to κ in Fig. 4.15. We rerun a set of simulations for a thrombus in a channel with $\kappa = 50 - 0.0005$ and plot the phase field difference when κ is set as the true value and inferred value. In (a) the maximum difference in the phase field for $\kappa = 50$ is at 2.5×10^{-5} while the

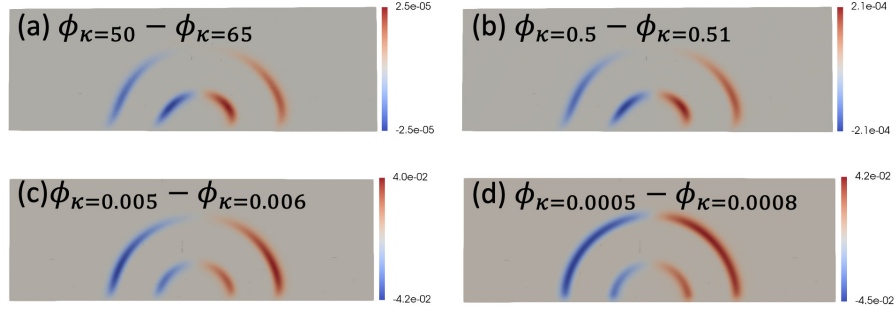


Figure 4.15: *A posteriori* practical identifiability analysis of κ to the phase field. (a-d) Phase field difference between κ is set as true values and inferred values.

maximum difference for $\kappa = 0.0005$ is almost 1,000 times larger, showing that the system is relatively insensitive to variations in κ when permeability is large. Therefore, the system is less “sensitive” when $\kappa = 50$ than κ is at $O(1 - 10^{-4})$ and it is harder to have an accurate inference results when $\kappa = 50$.

In Fig. 4.16 we present fields comparison when $\lambda_e = 0.1$ and 15 for the biofilm problem. In Fig. 4.16 (a), we observe small differences between the deformation of the biofilm at the same time where most of the differences appear at the interface. Plot (b) compares the velocity field where we observe an acceleration area on the top of the biofilm. However, the main inconsistency is caused by the velocity inside the biofilm. In plot (c) we show the pressure field comparison and we observe an overall similar pressure distribution for λ_e at different values with minor differences close to the center. Overall, the results in Fig. 4.16 and 4.15 demonstrate that the parameter sensitivity for κ and λ_e pose a difficulty on the inverse inference to the unknown parameters.

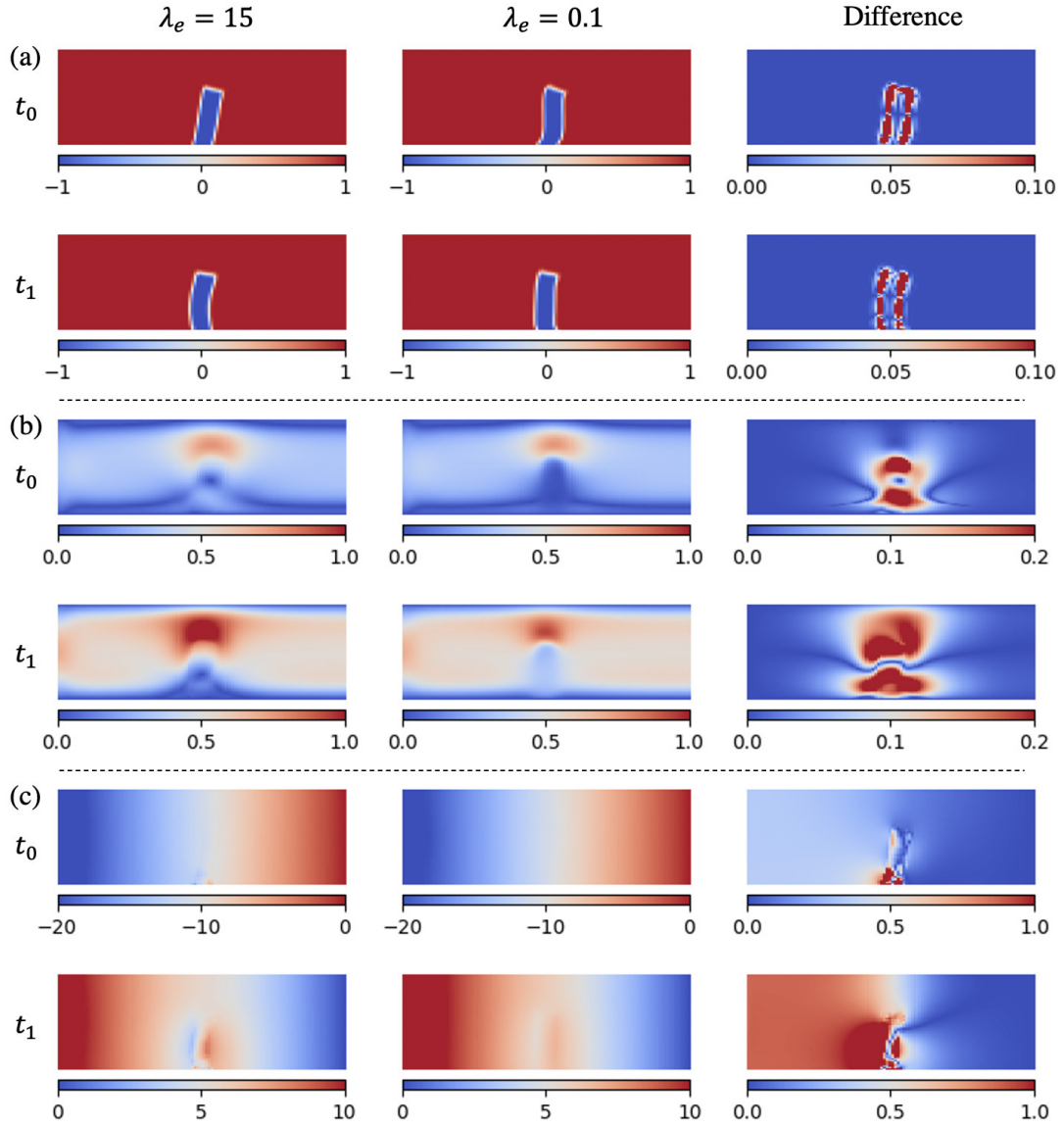


Figure 4.16: 2D flow past a viscoelastic biofilm for $\lambda_e = 15$ and 0.1. The first two lines show the phase field difference at $t_0 = 0.44$ and $t_1 = 0.86$. The middle two lines and the last two lines show the comparison of velocity and pressure at t_0 and t_1 . The last column presents the absolute difference between fields data when $\lambda_e = 15$ and 0.1.

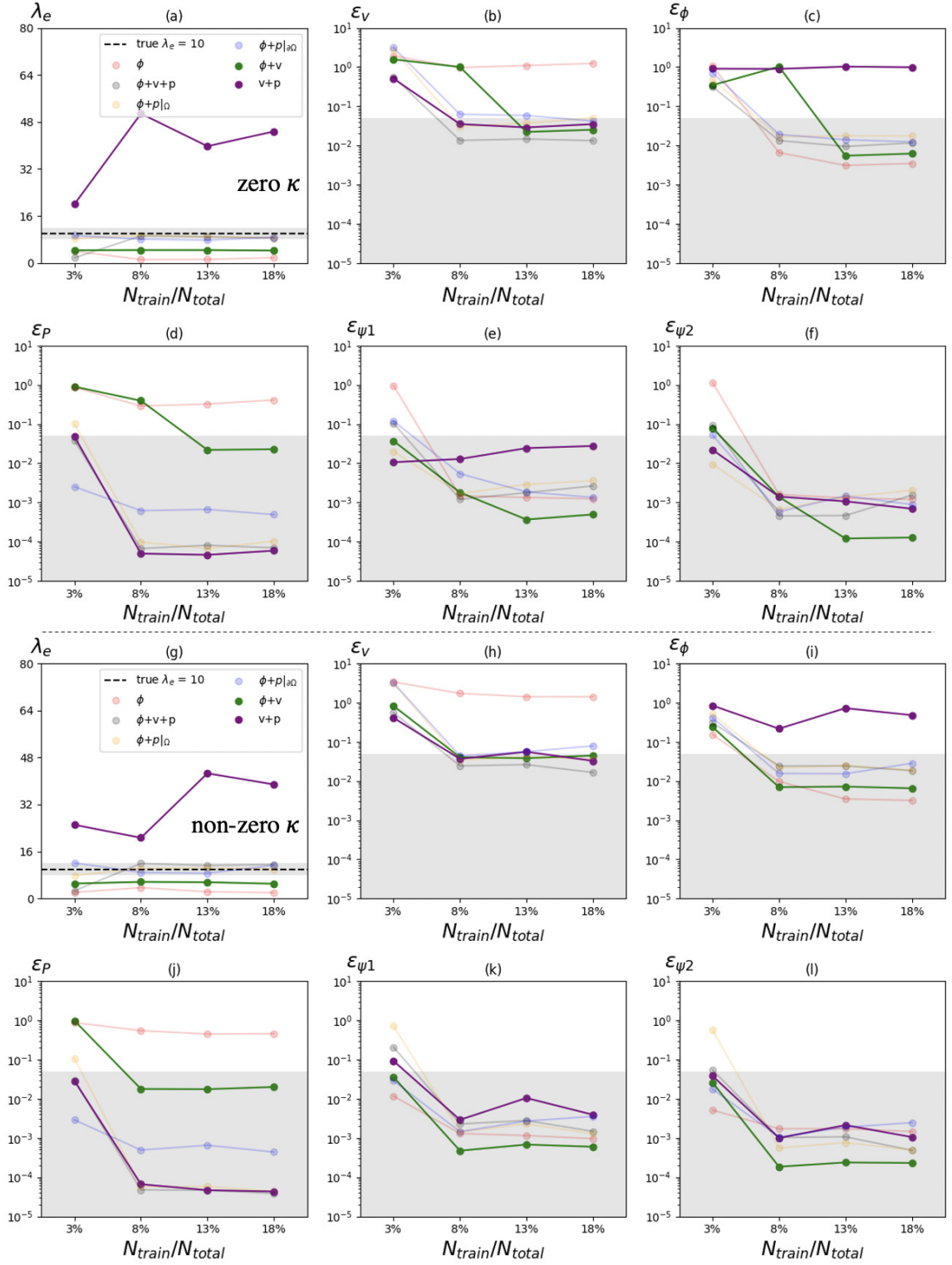


Figure 4.17: Results for the biofilm case with additional results. We train the PINN model with $p + u$ and $\phi + u$ as comparisons, and we plot the inference results and errors on the top of Fig. 4.13 with opaque lines indicating the new results. The shaded areas indicate that the error is lower than 5%.

4.B Supplementary results for the inference of viscoelasticity

In this section, we present two more additional tests result from the network trained with $\phi + u$ (green line) and $u + p$ (purple line) in Fig. 4.17 on the top of Fig. 4.13. The inferred parameter value on the purple line has the largest error among all lines, and the phase field error ended at the order of 1, indicating converged training results from the PINN model by only using information from velocity and pressure fields. The green line shows the results from training with ϕ and u . While the error in (e-i) shows a satisfactory agreement with the actual data, the inferred parameter value still cannot match the true value very well. We use these two tests as a supplementary proof to show the importance of pressure data on the inference of λ_e .

4.C Inferring permeability from a pre-trained model

In this section, we briefly discuss the possibility to infer parameter values from a pre-trained model. To be more specific, we use the weights and biases of a network for inferring $\kappa = 0.5$ as the initial state to infer $\kappa = 0.005$ and 0.0005 . Figure 4.18 plots the value of κ and total loss against training epochs for PINNs with a pre-trained model and randomized initialization. For $\kappa = 0.005$, (a) and (b) show that the model with pre-training shows a faster convergence on inferring κ and minimizing loss compared with the model without pre-training. However, pre-training may not be helpful for impermeable thrombus ($\kappa = 0.0005$) even if the initialization of κ is far from the true value. So the effect of using a pre-trained

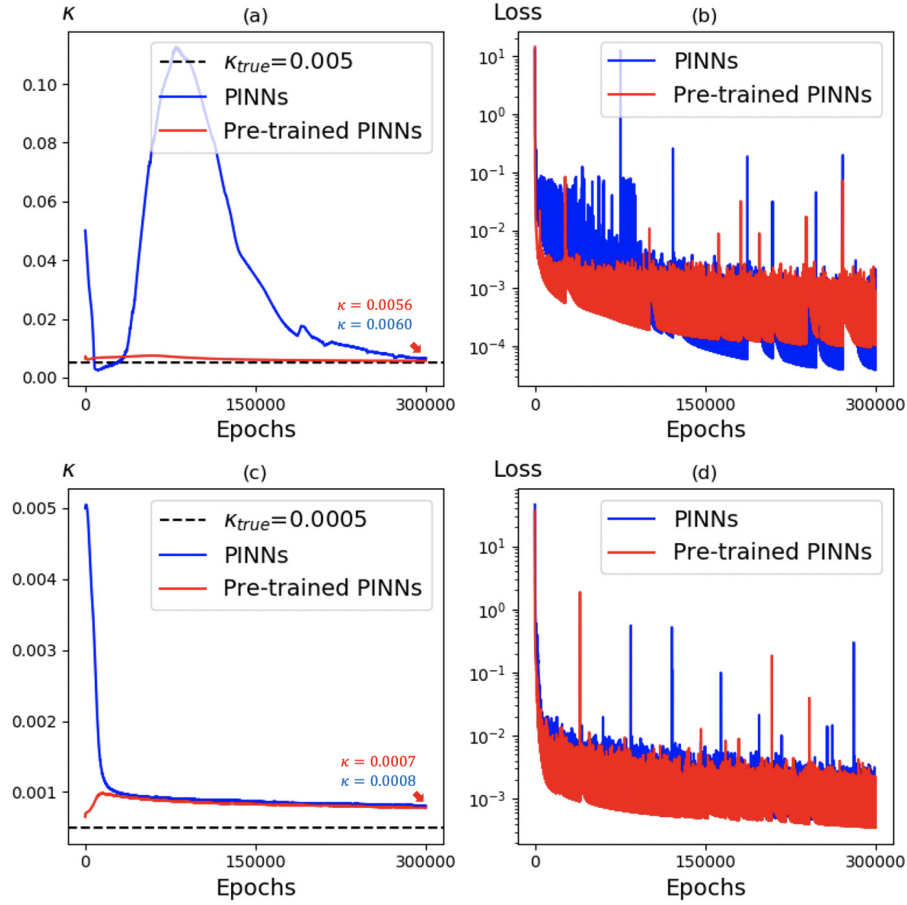


Figure 4.18: Effect of using a pre-trained model on the inference of $\kappa = 0.005$ and 0.0005 . (a, c) compare the history of inferred κ from the model with and without pre-training versus epochs. (b, d) shows the training loss versus epochs for each model. The pre-trained model is adopted as the weights and biases for $\kappa = 0.5$ at last epochs.

model as initialization for faster convergence is case dependent and needs a more detailed and careful discussion in our future work.

4.D Relative error over time

In this section, we plot the relative L_2 error over time for the cavity and biofilm example. The errors in the first half window are smaller than that in the full time window. The relative errors in every field increase over time, hence suggesting

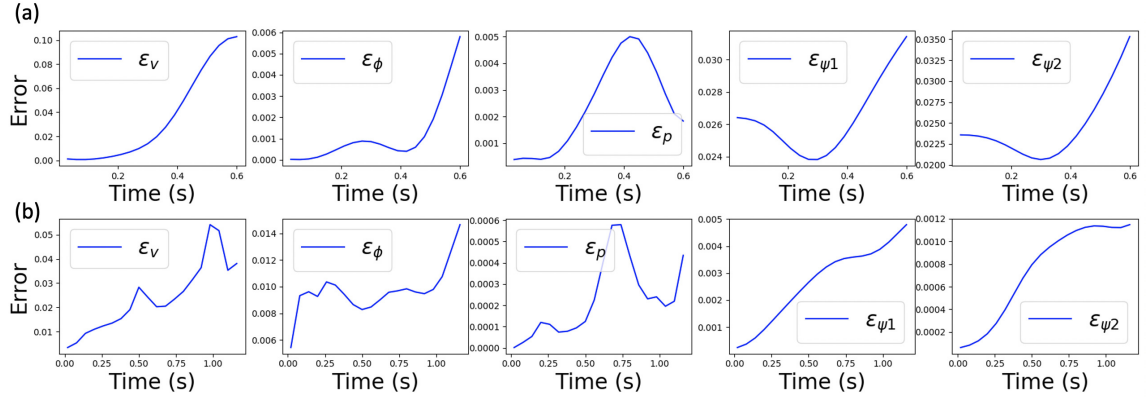


Figure 4.19: Relative L2 error over the full time window for the (a) cavity and (b) biofilm example.

that longer time integration may cause larger deviations on the field predictions.

CHAPTER FIVE

Uncertainty Quantification of Nonlinear Ex-vivo Aortic Biomechanics with Functional Priors Learning and Posterior Estimation

5.1 Abstract

Quantification of patient-specific biomechanical properties facilitates diagnosis and prognosis of cardiovascular diseases in clinics. Traditionally, quantifying parameter values of a constitutive model utilizes nonlinear regression algorithms based on extensively collected yet noisy data. In this chapter, we utilize a recently proposed Bayesian framework, functional prior, to accurately learn the priors and efficiently estimate the posteriors of the constitutive relationships. The model that incorporates physics-informed generative adversarial networks (PI-GANs) and a sampler, No-U-Turn (stems from Hamiltonian Monte Carlo, HMC), is trained with synthetic data, which are generated from a widely-accepted constitutive model, four-fiber family model, for informing the priors. Then, we combine the generator in PI-GANs with No-U-Turn to make posterior inference based on sparse data measurements that are collected from either synthetic data or ex-vivo biaxial mechanical tests for mouse with various genotypes. We have demonstrated that the estimated means can accurately recover the subject-specific constitutive relationships with standard deviations indicating the confidence of predictions. In future, the framework has great implications to estimate patient-specific mechanical properties of arteries with efficiency and accuracy.

5.2 Introduction

To quantitatively describe the biomechanics, mechanobiology, and physiopathology of the aorta, accurate constitutive modeling is an essential problem that has been investigated over decades. Considerable progress has been made in model-

ing the mechanical behavior of the aorta [101, 70, 241, 265, 68, 69, 96, 10]. Most of the models are phenomenological: each term in the constitutive models is motivated to mimic the macroscopic behavior of underlying microstructure.

Next, we will briefly review the internal microstructure of the arterial wall. The aorta consists of three main layers: intima, media, and adventitia. Intima is the innermost layer of the aorta, which is made of a single layer of endothelial cells. The intima does not play an important role in load bearing, whereas the other two layers take most of the responsibility. The medial layer consists of a complex structure of smooth muscle cells, elastin, and collagen fibrils. Elastic lamellae delimit intralamellar media that are made of smooth muscle cells and extracellular matrix. The collagen fibrils are mostly oriented in the circumferential direction. The adventitia is the outer layer of the aorta and consists mainly of collagen fibers, fibroblasts, and other ground substances. The adventitia serves to provide support to the aorta and enhance the stability and strength of the overall structure, particularly at high pressure. The interaction between many constituents, glycoproteins, and other components in the aorta together formulate a highly complex and nonlinear system.

Based on the microstructure of the aorta, a family of exponential models that account for fibers alignment and orientation, has received much success in describing the mechanical behavior of healthy aorta. Two famous models among of them include the Holzapfel-Gasser-Odgen (HGO or two-fiber family) model [96] and the four-fiber family model [10]. Specifically, the models contain an isotropic term described by the Neo-Hookean strain energy density for the ground substances and anisotropic terms (several exponential terms) for characterizing the nonlinear support by fibrous extracellular matrix in diverse directions [62, 170, 107]. Interestingly, it is found that the four-fiber model performs better in predicting

biomechanics than the two-fiber model [107]. The model parameters are usually quantified by nonlinear regression algorithms such as Levenberg-Marquardt (LM) [75, 170].

However, traditional nonlinear regression algorithms are often suffered from converging into local minimum despite that a sufficient amount of data from ex-vivo is required. Such an issue may cause the algorithm renders the parameter with extreme values, which are away from the physiological values. In addition, acquisition of ex-vivo experimental data is subject to noise that is sourced from experimental techniques, subject-specific internal microstructures, human errors, and etc. Noisy measurements may deteriorate the accuracy of nonlinear regression for quantifying both healthy and diseased artery parameters. In addition, nonlinear regression algorithms alone cannot show the confidence of predictions. Hence, there is a pressing need to develop a model that performs uncertainty quantification (UQ) of the aortic material properties given noisy and sparse measurements.

Recently, the field of biomechanics has drawn more attention to UQ as biological systems are usually subject to a great variability [190, 57], such as in cardiovascular system [303, 257, 235, 234], electrocardiology [49], cerebral systems [236], or aortic vessels [19, 246]. In particular, Rego et al. [223] have addressed this issue by developing a Bayesian framework for quantifying the local material properties of ascending thoracic aorta. The above UQ works are developed based on polynomial chaos expansion or Monte Carlo methods, which are standard methods in UQ.

In recent years, UQ with machine learning is a fast developing area [208, 114] with successful applications in high-dimensional problems [260], solving

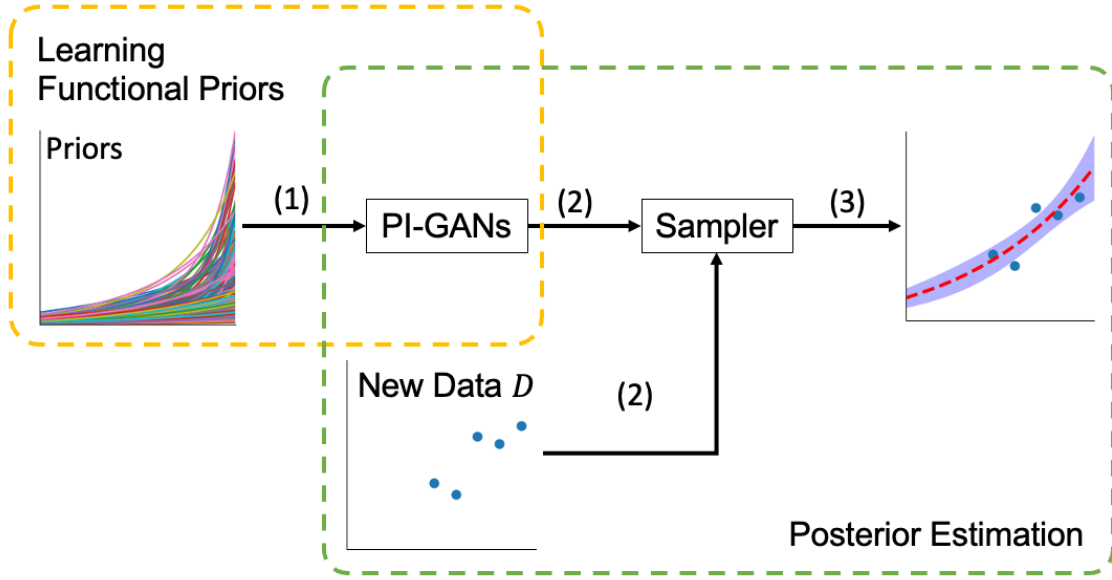


Figure 5.1: Schematic of learning functional priors (yellow box) and posterior estimation (green box). (1) Prior knowledge of the function family will inform a physics-informed generative adversarial network (PI-GAN) in the training stage. (2) Upon the completion of training PI-GAN, the model will be incorporated in a sampler to (3) estimate the posterior conditioned on new data D .

stochastic partial differential equations (PDEs) [295, 314, 298], among many other areas. In particular, a promising model, functional prior [174], utilizes physics-informed generative adversarial networks (PI-GANs) [296] for learning functional priors of the system and infers posteriors based on a Bayesian framework, such as Markov Chain Monte Carlo (MCMC) or Hamiltonian Monte Carlo (HMC). The model has been demonstrated its performance, accuracy, and flexibility with great potential to learn hidden knowledge thanks to its data-driven nature.

The present chapter aims to utilize the functional prior framework to infer constitutive relations of the aorta based on subject-specific data in a probabilistic approach. As a proof of concept, such a framework enables efficient posterior inference based on sparse measurements (Fig. 5.1). Prior knowledge of the constitutive laws is learned by a PI-GAN [296, 298, 81]. Physics-informed learning is a family of machine learning trained with the information of governing equa-

tions expressed in various forms. We refer the reader to these papers for more details [30, 112, 224, 317, 305, 155, 119]. In particular, learning priors are achieved in an adversarial training process where the generator aims at approximating the data distribution and the discriminator tries to distinguish if the input is from true data or is generated from the generator. Herein, we adopt the framework so that the generator predicts the energy of the arterial wall whose derivatives are used to compute Cauchy stresses, measurable quantities from experiments [62]. This physics-informed feature provides a possibility of network training using experimental data. After training, a Bayesian framework based on MCMC or HMC will incorporate the generator for estimating the likelihood. As the output of Bayesian sampling algorithm, a distribution of noise conditioned on new data is generated and transmitted to the generator again for estimating the means and standard deviations of stresses.

The chapter structure is listed below: In Sec. 5.3, we briefly describe the functional prior framework and its two components: PI-GAN and Bayesian inference. In Sec. 5.4, we report the inference accuracy for benchmarks in 1D and 2D, followed by performing posterior inference based on ex-vivo experimental data. Finally, we conclude with a short discussion in 5.5.

5.3 Methodology

In this section, we briefly describe the four-fiber family model in 5.3.1, PI-GAN for learning functional priors in 5.3.2, and the sampling algorithm for estimating posteriors in 5.3.3.

5.3.1 Four-Fiber family model

We adopt the four-fiber family model as the constitutive law for describing the mechanical behavior of the aorta due to its wide acceptance and accuracy [10, 107, 62]. The model serves as a proof-of-concept for illustrating the learning ability of the functional prior framework, which, in general, can learn a wider class of constitutive models without a given function form. The four-fiber model describes the aorta as hyperelastic, anisotropic, and fiber-reinforced in four diverse directions. The strain energy density is written as:

$$W = \frac{\mu}{2}(I_1 - 3) + \sum_{i=1}^4 \frac{k_1^i}{4k_2^i} (\exp\{k_2^i(I_{4i} - 1)^2\} - 1), \quad (5.1)$$

where μ is shear modulus and $k_{1,2}^i, i = 1, \dots, 4$ are fiber modulus and exponential coefficients for the four fiber groups that are oriented along the axial ($i = 1$), circumferential ($i = 2$), and two symmetric diagonal directions ($i = 3, 4$). The above parameters are usually calibrated using nonlinear regression methods based on experimental data [62]. I_1 is the first principal invariant and I_{4i} denotes the fourth principal invariant corresponding to the i -th fiber group:

$$I_1 = \text{tr} \mathbf{C},$$

$$I_{4i} = \mathbf{C} : \mathbf{M}_i \otimes \mathbf{M}_i, \quad i = 1, \dots, 4$$

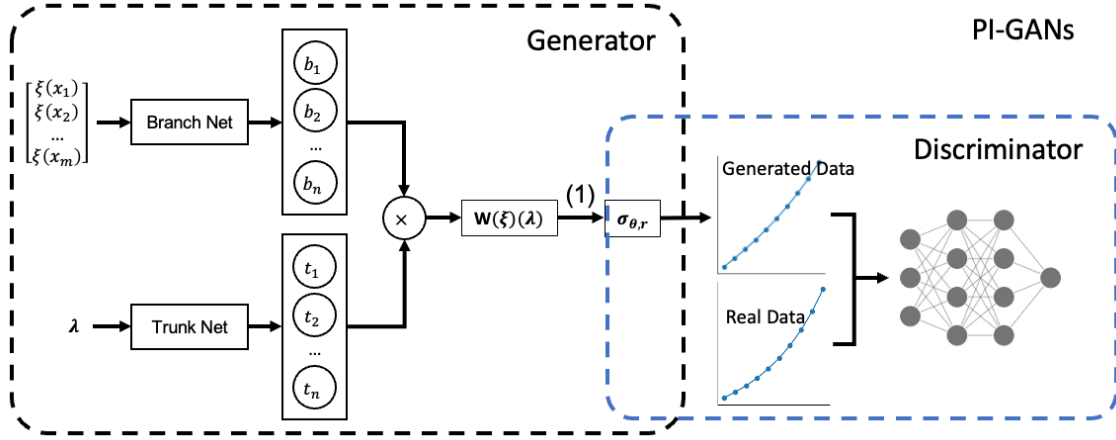


Figure 5.2: Physics-informed generative adversarial networks (PI-GANs). The generator of PI-GAN is a DeepONet with a vector of noise ξ and stretch λ as the input of the branch and trunk net. The network predicts the corresponding stress $\sigma_{\theta,r}$ which can be calculated by taking the partial derivatives (physics-informed) of the intermediate output (strain energy density W). The generated data and real data will be transmitted to the discriminator for estimating their difference.

where $\mathbf{C}$ is the right Cauchy Green tensor and $\mathbf{M}_i$ indicates the fiber family orientations:

$$\mathbf{M}_1 = [0, 0, 1],$$

$$\mathbf{M}_2 = [0, 1, 0],$$

$$\mathbf{M}_3 = [0, \cos(\alpha), \sin(\alpha)],$$

$$\mathbf{M}_4 = [0, \cos(\alpha), -\sin(\alpha)]$$

α is the angle of the diagonal families with respect to the circumferential direction.

5.3.2 Learning functional priors: PI-GAN and DeepONet

PI-GANs: We briefly introduce PI-GANs in this section. As shown in Fig. 5.2, the overall framework is divided into two parts: a generator and a discriminator. The generator $\mathcal{G}$ aims at approximating the data distribution P_D as accurate as possible

whereas the discriminator $\mathcal{D}$ is trained to distinguish the distance between real data and generated data from $\mathcal{G}$. The two parts are trained in an adversarial approach until the networks reach the equilibrium [81]. Quantitatively, the generator aims at minimizing:

$$L_{\mathcal{G}} = -\mathbb{E}_{\xi \sim P(\xi)}[\mathcal{D}(\mathcal{G}(\xi))], \quad (5.2)$$

and the discriminator is trained to minimize:

$$L_{\mathcal{D}} = \mathbb{E}_{\xi \sim P(\xi)}[\mathcal{D}(\mathcal{G}(\xi))] - \mathbb{E}_{D \sim P_D}[\mathcal{D}(D)] + \lambda_{reg} \mathbb{E}_{D \sim P_D}(\|\nabla_T \mathcal{D}(D)\|_2 - 1)^2, \quad (5.3)$$

The first two terms denotes that to discriminate the generated data $\mathcal{G}(\xi)$ and real data D . The last term serves as a gradient penalty term to regularize the training dynamics. The gradient coefficient penalty λ_{reg} is chosen as 0.1. More details can be found in [174].

DeepONet: We choose the Deep Operator Network (DeepONet) as the generator structure [159]. The network consists of a branch and a trunk net. The branch net takes a vector of random variables ξ evaluated at $x_1, x_2, \dots, x_m$ as input and yields a latent vector $\mathbf{b} = [b_1, b_2, \dots, b_n]$. The trunk net takes stretches λ as the input and outputs a vector $\mathbf{t} = [t_1, t_2, \dots, t_n]$ with the same dimensionality as $\mathbf{b}$. In the original framework of DeepONet, the network outputs takes a linear combination of $\mathbf{b}$ and $\mathbf{t}$ ($y_{out} = \sum_{i=1}^n b_i t_i$) to approximate the function of interest. In this research, we modify the network output as the energy density of material W since energy is more informative and general than stresses. Because the energy density has a tendency of exponential growth at large stretches, we adopt an exponential form as the model output after pointwise multiplication: $y = \exp\{(y_{out})\} - 1$. The modification has two advantages: first, we empirically observe that such a modification increases the expressivity of networks for the exponential family; second,

the results of linear combination y_{out} is controlled in $O(10^0)$.

The standard bi-axial loading tests can only measure stresses, not energy [77]. Hence, it is practical to utilize stresses as training data while keeping the energy density as an output. Introducing physics-informed learning is a suitable solution for solving such an issue. Physics-informed machine learning [119, 213, 30] is an emerging field where the network is trained and regularized using the governing equations or other physical prior knowledge in either FNN or DeepONet [84]. Assume the material is incompressible ($J = 1$), the Cauchy stresses (σ) can be calculated by taking partial derivatives of energy density:

$$\sigma = -p + \mathbf{F} \mathbf{S} \mathbf{F}^T = -p + \begin{bmatrix} \lambda_r & & \\ & \lambda_\theta & \\ & & \lambda_z \end{bmatrix} \frac{\partial W}{\partial \lambda} \quad (5.4)$$

where $\frac{\partial W}{\partial \lambda}$ is calculated using automatic differentiation in machine learning packages [1]. Notice that a Lagrange multiplier p is added to each direction for imposing incompressibility. Since we assume the aorta is a thin wall by neglecting the radial stress:

$$\sigma_r = -p + \frac{\partial W}{\partial \lambda_r} \lambda_r = 0, \quad (5.5)$$

where $\lambda_r = 1/\lambda_\theta \lambda_z$. As for the discriminator, we choose FNN as the network structure whose size will be introduced in Sec. 5.4.

Remark: Constitutive modeling using machine learning is a fast growing area where many insightful works has been proposed [155, 169, 66, 26, 151, 97, 150]. One of the most common network structures is FNN, which learns the deterministic relations between stretch and strain for certain types of material. However, DeepONet is more flexible and powerful on learning mappings between func-

tions [159, 304, 301, 146] and can be further adopted into a probabilistic framework: we seek to find a function (ξ) in a latent space that could implicitly represent the constitutive relation of a new sample, then DeepONet could be trained to learn the mapping between the latent space representation and its corresponding constitutive law. The function in a latent space ξ is sampled conditioned on some measurements T by a sampling algorithm stemmed from MCMC, which will be introduced in the Sec. 5.3.3.

5.3.3 Posterior inference

In this section, we introduce the methodology of posterior inference. As mentioned in the previous section, the generator is trained to learn the functional relations between a distribution of noise ξ and $\sigma(\lambda; \xi)$, from which we compute the mean and standard deviation of σ . However, the distribution of ξ is not a multivariate Gaussian as that in training: ξ is sampled from $P(\xi|D)$ via MCMC condition conditioned on new data D . Consider we have N new data measurements D , Bayes's theorem states that

$$P(\xi|D) = \frac{P(D|\xi)P(\xi)}{P(D)} \simeq P(D|\xi)P(\xi) \quad (5.6)$$

where $\simeq$ denotes that equality up to a constant since $P(D)$ is non-tractable. $P(D|\xi)$ is the likelihood (probability of D conditioned on ξ) defined as:

$$P(D|\xi) = \prod_{i=1}^N \frac{1}{\sqrt{2\pi\epsilon^2}} \exp\left(-\frac{(\sigma_{\mathcal{G}}(\lambda; \xi) - \sigma)^2}{2\epsilon^2}\right),$$

$\sigma_{\mathcal{G}}(\lambda; \xi)$ is the generator prediction at λ given noise ξ and ϵ is the standard deviation of the data. σ is the measurement data. Note the if measurements of

Case	Dimension	Training Data	Measurement Data
1	1	Synthetic	Synthetic
2	2	Synthetic	Synthetic
3	2	Synthetic	Experiments

Table 5.1: Setup for the three examples.

multiple variables are available (e.g. circumferential and axial stresses), the new likelihood is simply the joint distribution of those variables assuming the data are independent. The prior $P(\xi)$ is defined as a multivariate Gaussian distribution:

$$P(\xi) = \left(\frac{1}{\sqrt{2\pi}}\right)^{d_\xi} \exp\left\{-\frac{\|\xi\|^2}{2}\right\} \quad (5.7)$$

In this work, we adopt the No-U-Turn sampling scheme, which is a HMC with adaptive path length as the sampling scheme [94]. Once the noise ξ conditioned on D is generated, we will use the generator G to compute $u_G(x; \xi)$. We refer the sampling algorithm in this chapter as No-U-Turn for simplicity. More technical details can be found in [174].

5.4 Results

In this section, we illustrate the performance of our framework by several examples: inferring one-dimension (1D) and two-dimension (2D) constitutive modeling with synthetic data and experimental data. All of the networks are trained on a NVIDIA GeForce RTX 2060 GPU.

Name	Type	Structure	Input	Output
Branch Net (Generator)	FNN	[50, 64, 64, 64, 50]	ξ	$\mathbf{t}$
Trunk Net (Generator)	FNN	[1, 64, 64, 64, 50]	λ_θ	$\mathbf{b}$
Discriminator	FNN	[15, 64, 64, 64, 1]	y_i	y_{dis}

Table 5.2: Structure and size of the PI-GAN in Case I. FNN: fully-connected neural networks.

5.4.1 Case I: 1D Constitutive Law

We first study the feasibility of applying functional priors for learning the four-fiber family model in 1D. We generate training data at a fixing axial stretch $\lambda_z = 1.72$. The input dimension of the trunk network is set as 1 (λ_θ , circumferential direction). One empirical observation from experiments [63, 170] shows that the abdominal aorta in a mouse preserves its length over a cardiac cycle, meaning that the in-vivo axial stretch is nearly a constant. Hence, the problem setting is meaningful in this sense.

In Table 5.2, we tabulate more details of the PI-GAN model. All of the networks in case I are chosen as fully-connected neural network (FNN) with three hidden layers. Each hidden layer contains 64 neurons. We adopt DeepONet [159] as the generator whose branch net takes a vector of random variable ξ with 50 dimensions. The trunk net takes the circumferential stretch (λ_θ) as input. The output of the generator is formulated as $y_{out} = \exp\{(\mathbf{b}\mathbf{t})\} - 1$ where $\mathbf{b}$ and $\mathbf{t}$ denotes the output of the branch and trunk net.

To train the PI-GAN model, we randomly sample the four-fiber parameters from the range listed in table 5.3. The mean values are adopted from [62] with a relatively wide variation. An illustration of the four-fiber model in 1D is presented

Parameters	Mean Value	Magnitude Range
μ	8.633	[0.1, 5.0]
k_1^1	9.34	[0.1, 5.0]
k_2^1	0.137	[0.01, 2.0]
k_1^2	4.447	[0.1, 5.0]
k_2^2	0.046	[0.01, 2.0]
$k_1^{3,4}$	0.015	[0.1, 5.0]
$k_2^{3,4}$	1.193	[0.01, 2.0]
α	28	[0.1, 2.0]

Table 5.3: Four-fiber family parameters and their range. All of the training data are generated with parameters uniformly sampled within the range. The data is adopted from [62].

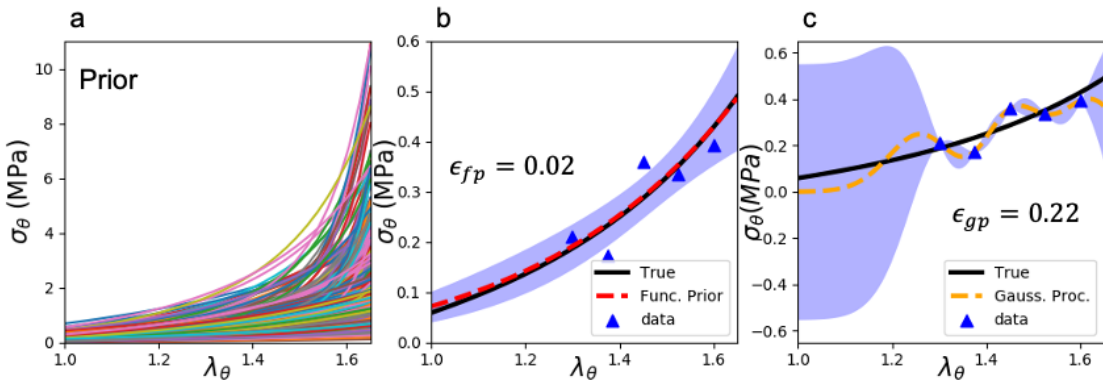


Figure 5.3: Uncertainty quantification of 1D four-fiber model. a: 1280 random samples of σ_θ . b: Mean and standard deviation band computed from Bayesian inference with functional priors. c: Mean and standard deviations predicted by Gaussian process.

in Fig. 5.3(a). The figure contains 1,280 samples in a training batch with σ_θ ranges from $O(10^{-1})$ to $O(10^1)$. The training epochs is set as 100,000 with the Adam optimizer [126]. Upon the completion of training, the objective now is to infer the posteriors of circumferential stresses based on sparse and noisy measurements. Assume that we have five noisy measurement points of σ_θ that are randomly distributed in $\lambda_\theta \in [1.3, 1.6]$. These points are sampled from the four-fiber model with parameter values set as the mean in Table 5.3. Specifically, the noise e for the sampled points is assumed to satisfy a Gaussian distribution with its mean and variance at 0 and 1. The noise scale ϵ is set as 0.1. Hence, measurements σ_θ are computed as:

$$\sigma_\theta = \bar{\sigma}_\theta + \epsilon e, e \sim N(0, 1), \quad (5.8)$$

In Fig. 5.3(b), we present the posterior inference from our framework (red dashed line), which matches well with the true data (black line). The relative error ϵ_{fp} is approximately at 0.02. The error is defined as follows:

$$err = \frac{\|\bar{\sigma}_{pred} - \bar{\sigma}_{true}\|}{\|\bar{\sigma}_{true}\|} \quad (5.9)$$

The means and standard deviations are averaged over 1,000 predictions from PI-GAN based on noise vectors sampled from No-U-Turn. The shaded area indicates two standard deviations. As a comparison, Fig. 5.3(c) presents the means and standard deviations from Gaussian process (GP) regression. The GP model does not incorporate any prior knowledge from the four-fiber family, hence it renders a prediction with a higher error ($err = 0.22$). Notice that, under current circumstance, only five measurement points are available, making the standard nonlinear regression fails since the system is under-estimated: the number of variables is larger than the number of available points.

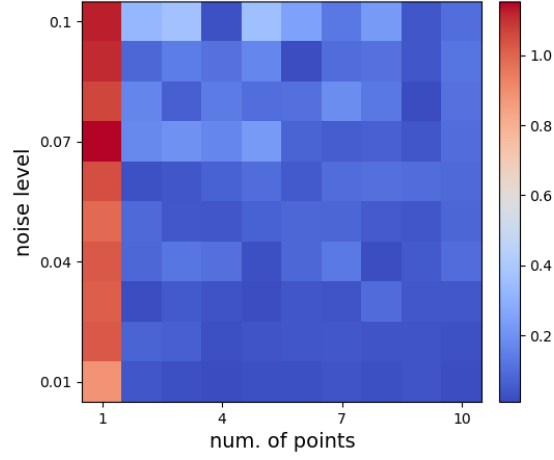


Figure 5.4: Prediction errors at different noise scales and number of points. The color represents the magnitude of error defined in Equ. 5.9.

Name	Type	Structure	Input	Output
Branch Net (Generator)	FNN	[50, 64, 64, 64, 50]	ξ	t
Trunk Net (Generator)	FNN	[2, 64, 64, 64, 50]	$[\lambda_\theta, \lambda_z]$	b
Discriminator (σ_θ)	FNN	[625, 200, 200, 200, 1]	σ_i	y_{dis}
Discriminator (σ_θ and σ_z)	FNN	[1250, 250, 250, 250, 1]	$[\sigma_i, \sigma_z]$	y_{dis}

Table 5.4: Structure and size of the PI-GAN in Case II.

Then, we investigate the impact of noise and the number of measurements on the model performance. It is clear to observe that, in Fig. 5.4, more measurement points with a lower noise level (ϵ) could lead to more accurate predictions whereas fewer points with larger noise scales would deteriorate the accuracy.

5.4.2 Case II: 2D Constitutive Law

Next, we proceed to examine the model performance in inferring 2D constitutive relationships using synthetic data.

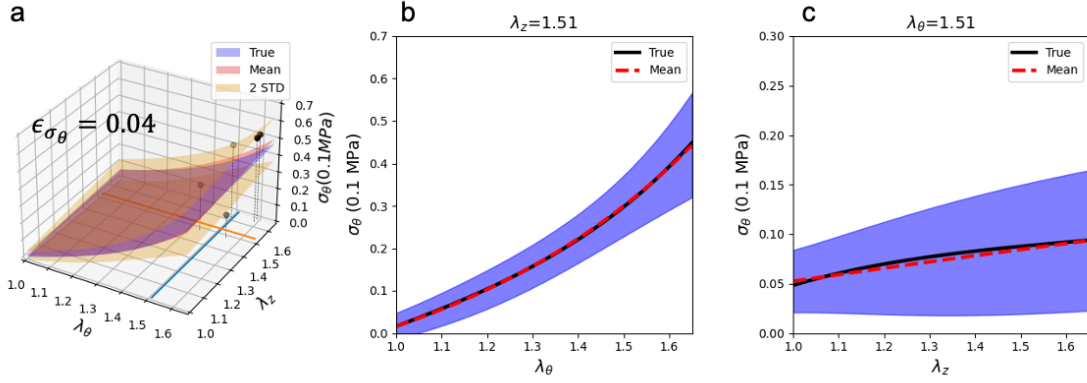


Figure 5.5: Mean and standard deviations (STD) for inferring σ_θ given five sparse measurements. (a) Mean (red) and two STD (yellow) of σ_θ versus true stress (blue). Line comparison at $\lambda_z = 1.51$ and $\lambda_\theta = 1.51$ are plotted in (b) and (c). The network is only trained to with σ_θ .

We first consider inferring only circumferential stress σ_θ . In the training stage, we inform the PI-GAN by σ_θ sampled at stretches ranging from $(\lambda_\theta, \lambda_z) \in [1.0, 1.65] \times [1.0, 1.65]$. The parameters are sampled from the range in Table 5.3. The structure of PI-GAN is shown in Table 5.4. We still consider FNN as the model structure with a larger network size than that of case I. A distribution of ξ with 50 random variables sampled from multivariate Gaussian distribution is sent to the branch net as training input. The energy W and circumferential stress σ_θ at λ_θ and λ_z are predicted as the output of the generator. In particular, σ_θ at a mesh size of 25 by 25 is estimated and transmitted to the discriminator whose outputs will be used to calculate the statistical distance between the generated data and true data. The noise level ϵ is set to 0.1. The network is trained adversarially over 100,000 epochs.

Once the network is trained, we move to infer the posterior. We first collect five noisy measurements of σ_θ at high stretches as the data for No-U-Turn. In particular, the noisy measurements are generated with the same parameter values shown in the second column of Table 5.3. Then, we generate 1,000 distributions of ξ conditioned on the five measurement points and feed the ξ into the generator.

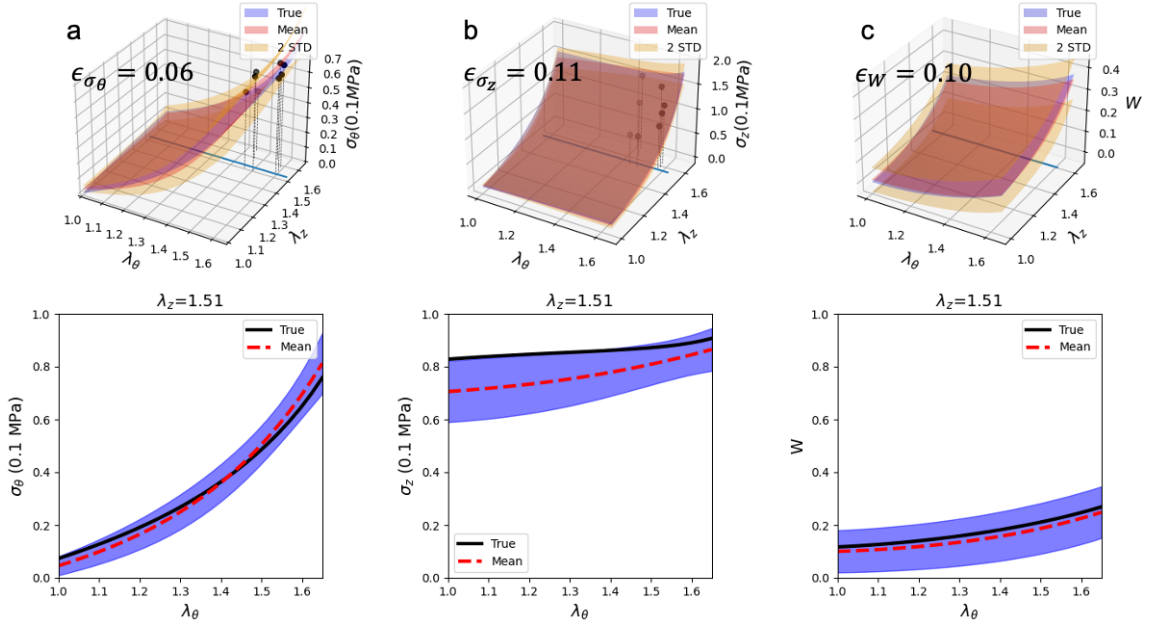


Figure 5.6: Informative inference of stresses and energy based on 8 measurements of σ_θ and σ_z . The mean (red) and 2 STD (orange) are plotted with the true data (blue) for (a) σ_θ , (b) σ_z , and (c) W . The network is trained with both σ_θ and σ_z .

At a result, 1,000 realizations of $\sigma_\theta(\lambda_z, \lambda_\theta; \xi)$ are collected to calculate the statistical results (Fig. 5.5). In Fig. 5.5(a), the mean σ_θ matches with the true solution with an error at 0.04. The prediction error of σ_θ is bounded by two standard deviations in the whole domain. Notice that the posterior inference is conducted only based on five noisy measurements whereas the nonlinear regression is underdetermined in this scenario. We further compare the predicted σ_θ at $\lambda_z = 1.51$ and $\lambda_\theta = 1.51$ in Fig. 5.5(b) and (c). It is interesting to observe that the uncertainty is larger at a higher stretch, corresponding to the region where σ_θ vary the most.

Then, we move to infer the circumferential (σ_θ) and axial (σ_z) stress simultaneously. We keep the generator structure the same as in the previous example. To inform the priors of σ_θ and σ_z to PI-GANs, we choose to concatenate the predicted σ_θ and σ_z as the input of the discriminator where the input dimension increases from 625 to 1,250. The generator in conjunction with the No-U-Turn model infer the posterior of $\sigma_{\theta,z}$ based on eight randomly distributed measurement data at

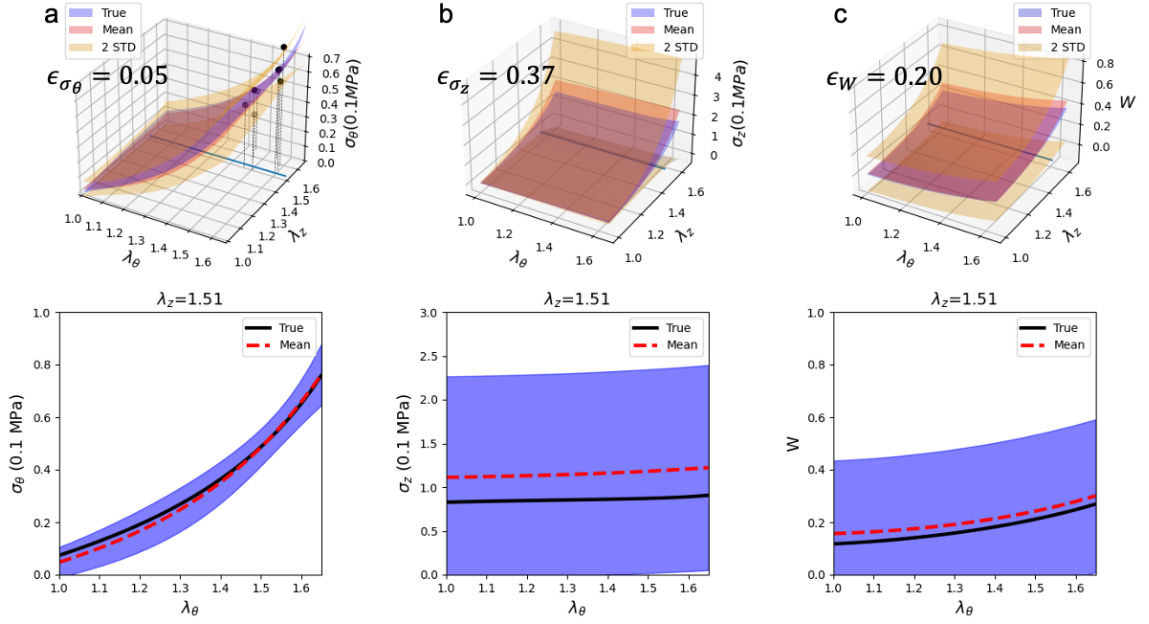


Figure 5.7: Uninformative inference of stresses and energy based on 8 measurements of σ_θ . The mean (red) and 2 STD (orange) are plotted with the true data (blue) for (a) σ_θ , (b) σ_z , and (c) W . The network is trained with both σ_θ and σ_z .

high stretches (shown in Fig. 5.6). We present the response surface of σ_θ , σ_z and energy density W in Fig. 5.6 (a), (b), and (c), respectively. The posterior mean of σ_θ and W correspond well with the true solution with errors at 0.06 and 0.10. However, there is a discrepancy between the mean σ_z and its true solution as the error falls at the edge of two standard deviations.

Interestingly, we then investigate if posterior inference is accurate based only on partial information, i.e., eight measurements of σ_θ . As shown in Fig. 5.7, the estimation of σ_θ is accurate and informative whereas the predicted σ_z and energy density are uninformative, represented by large uncertainty bands. This result indicates that accurate inference of stresses and energy density fields relies on stresses information on both circumferential and axial directions. Otherwise, the inference are not useful as the uncertainty becomes overwhelmingly large. This observation not only reaffirm that deformation/stress in λ_θ and λ_z are biomechanically independent, but also, from a statistically perspective, they are uncorrelated

and independent. Another interesting point is that the proposed framework is able to predict the energy density given that no prior knowledge of the energy profile is informed to the network.

5.4.3 Case III: 2D Constitutive Law with Ex-vivo Data

To study the framework's potential for inferring constitutive relations from realistic data, we present the inference results based on experimental data collected from mice with various genotypes. The three exemplary animal models collected from published studies are re-analyzed herein [201, 294, 245, 129, 109].

- Wild-type (WT) C57BL/6J control mice [109],
- *Fbn1*^{mgR/mgR} mice: the fibrillin-1 is expressed at 15-25% of its normal level [129],
- *Eln*^{+/+} (hBAC+mWT) mice: the elastin is expressed to 115% of its normal level [109].

The stress/stretch data were collected from standard ex-vivo bi-axial loading tests [62]. After data acquisition, we perform a nonlinear regression algorithm for searching the best four-fiber parameters based on all the collected data (typically more than 900 points). Despite discrepancies at certain stretches, four-fiber models with fitted parameters still render a good approximation of the overall subject-specific biomechanics, which is taken as the best approximation of the experimental data in this study.

Informing good parameter priors is essential for inference accuracy. However, it is hard to quantify the range of parameter distributions for various genotypes

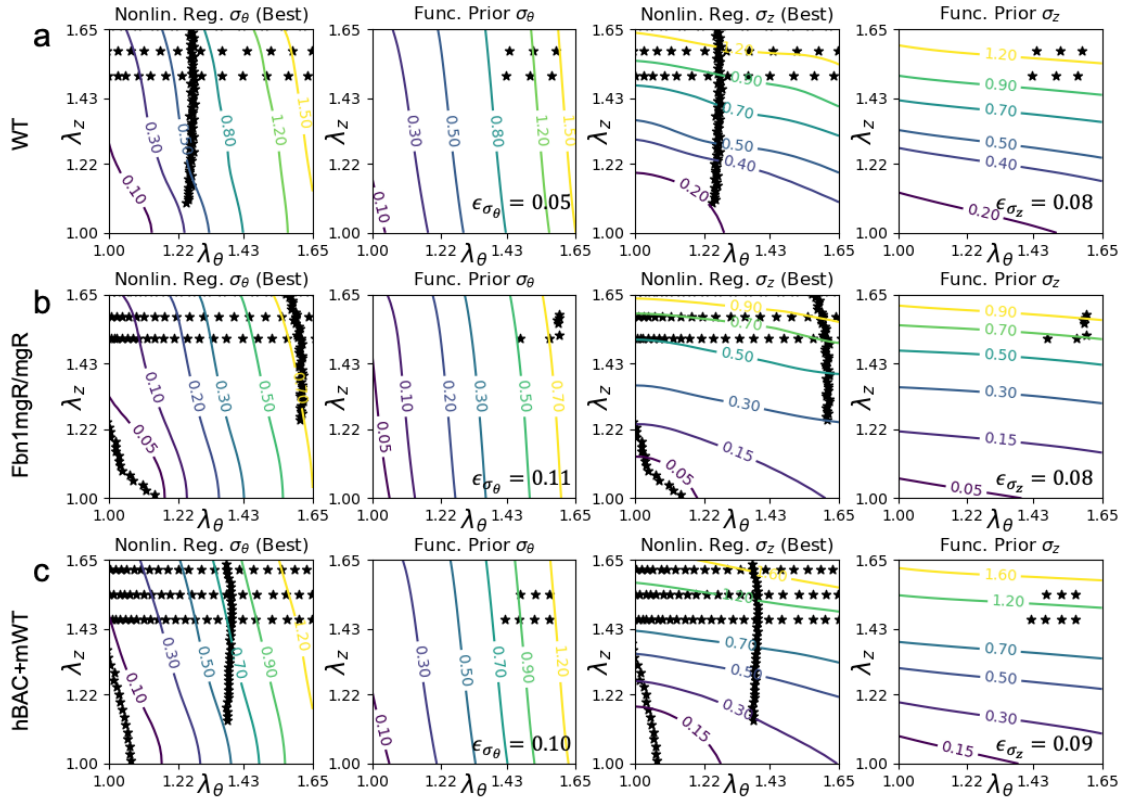


Figure 5.8: Stresses inference for three different genotypes: (a) wild-type, (b) Fbn1mgR/mgR, and (c) hBAC+mWT. The contours of nonlinear regression (Nonlin. Reg.) are calculated with all data collected from ex-vivo experiments whereas the functional prior (func. prior) estimates the mean of subject-specific constitutive relations only based on 6-7 points cluster at high stretches. Taking the nonlinear regression results as reference, the prediction errors range from 0.05 to 0.11.

since the data acquisition is expensive and time-consuming. The costly experimental procedure forbids the viability of high-throughput experiments and further restricts quantification of the physiological parameters range from existing data. As a proof-of-concept, we adopt the parameters range shown in Table 5.3 since it assumes a relatively wide range of variation based on the WT model collected in [62]. Using the same PI-GANs in Sec. 5.4.2, we present the uncertainty quantification to three exemplary model with different genotypes shown in Fig. 5.8. We randomly sample 6 or 7 points that contains $(\lambda_\theta, \lambda_z; \sigma_\theta, \sigma_z)$ in $(\lambda_\theta, \lambda_z \in [1.4, 1.6] \times [1.4, 1.6])$ from each case.

As shown Fig. 5.8(a), stresses inference for WT renders a good approximation with errors at $\epsilon_{\sigma_\theta} = 0.05$ and $\epsilon_{\sigma_z} = 0.08$, respectively. As mentioned, the generator is trained based on parameters sampled from a WT model. Hence, the model is expected to have a good performance for the inference of WT models. The posterior inference for the other two genotypes, however, ends up with higher errors, ranging from 0.08–0.11. One possible explanation is that the distributions of the other genotypes are essentially different from the WT model. *Fbn1^{mgR/mgR}* and *hBAC+mWT* have inherently different microstructure and biomechanical behavior, which are reflected by variations in parameters values and distributions for the two diseases group. Also, it is interesting to notice that the stress contour lines from the PI-GAN are rather flat whereas the contour lines from nonlinear regression exhibit a more complex and nonlinear behavior, reflected by local curvatures of the contour plots. However, the overall results indicate that our model is capable of capturing the overall trend of stresses based on sparse measurements.

5.5 Discussion

In this chapter, we adopt a generative machine learning model to infer subject-specific constitutive relations of the aorta with uncertainty quantification. The model is trained with generated data from the four-fiber family model. The model parameters are adopted from published data [62] for WT mice with relatively wide variations, which improves the model performance for different samples. The prior knowledge of the constitutive model is informed to PI-GANs, which is utilized to estimate the likelihood function in No-U-Turn in a following step. The posterior of noise vectors conditioned on sparse measurements are then sampled from No-U-Turn and transmitted to the generator of the PI-GAN for inferring the posterior of stresses in the functional space. We illustrate the performance of this framework with three examples. The first two examples utilize synthetic data as measurements. The last example demonstrates the model's potential for inferring constitutive relationships based on realistic measurements from various genotypes.

Informing prior knowledge into neural networks can enhance the accuracy and efficiency of the framework: Prediction errors and standard deviations from our framework are reduced compared to that of GP, whose results are more likely to be affected by noise and are uninformative outside the sampling range. Unlike linear or nonlinear regressions, the framework is not constrained by certain numerical rules, e.g., the number of data points must be greater than the number of unknown parameters. This feature shows strength over traditional nonlinear regression algorithms for inference with sparse measurements and for complex models with considerable parameters. Moreover, the mean and standard deviations are more informative than traditional algorithms without uncertainty quantification as they

represent the expectation and confidence of the prediction.

The advantages of adopting functional priors model lie in learning systems without explicitly function forms. Admittedly, for systems with explicit form, e.g., four-fiber models, traditional Bayesian inference with model equations could be more efficient and reliable [223] (referred as model-driven approach). However, there are certain limitations of the four-fiber model, which could restrict the applicability of the model-driven approach. One basic assumption of the four-fiber model is that fiber interactions between different alignments are neglected. In addition, as much success as the four-fiber family for healthy aortas, modeling mechanical behavior of the aorta with microstructure defects or genetic deficiencies still needs to be investigated. Nowadays, four-fiber models are still widely used as the constitutive model for mice with various genotypes. The model parameters are quantified following the same regression procedures as the healthy one. However, some of the model assumptions may not work well in this scenario, leading to nonphysical predictions of the constitutive model under some loading conditions.

Data-driven modeling is a promising approach to this problem. Training the network with realistic data could inform hidden knowledge that is not described by the four-fiber model a priori to the generator. Utilizing machine learning on modeling constitutive behavior could circumvent such problems by data-driven modeling without assuming a known form as the network learns the system behavior implicitly. On the other hand, however, traditional data-driven modeling is data-intensive, which could be further exacerbated by introducing the uncertainty of the model. In reality, it is not practical to collect as much data as needed to train a generative neural network. One possible mitigation is to incorporate both synthetic and experimental data in network training. The generative model is

trained mostly by synthetic data and is regularized by experimental data at certain stretches. Another possible improvement is to develop a multifidelity Bayesian neural network where data from four-fiber models and experiments serve as the low and high-fidelity data, respectively [172]. In addition, other improvements that may consider in the future are: (1) adopting more accurate parameter for different genotypes, (2) searching for better prior distributions, or (3) developing generative networks that are easier to train than GANs.

Acknowledgement

MY would like to thank Zongren Zou and Dr. Xuhui Meng for helpful discussion. We acknowledge the support by grant U01 HL142518 from the National Institutes of Health.

CHAPTER SIX

Interfacing Finite Elements with Deep Neural Operators for Fast Multiscale Modeling of Mechanics Problems

6.1 Abstract

Multiscale modeling is an effective approach for investigating multiphysics systems with largely disparate size features, where models with different resolutions or heterogeneous descriptions are coupled together for predicting the system's response. The solver with lower fidelity (coarse) is responsible for simulating domains with homogeneous features, whereas the expensive high-fidelity (fine) model describes microscopic features with refined discretization, often making the overall cost prohibitively high, especially for time-dependent problems. In this work, we explore the idea of multiscale modeling with machine learning and employ DeepONet, a neural operator, as an efficient surrogate of the expensive solver. DeepONet is trained offline using data acquired from the fine solver for learning the underlying and possibly unknown fine-scale dynamics. It is then coupled with standard PDE solvers for predicting the multiscale systems with new boundary/initial conditions in the coupling stage. The proposed framework significantly reduces the computational cost of multiscale simulations since the DeepONet inference cost is negligible, facilitating readily the incorporation of a plurality of interface conditions and coupling schemes. We present various benchmarks to assess accuracy and speedup, and in particular we develop a coupling algorithm for a time-dependent problem, and we also demonstrate coupling of a continuum model (finite element methods, FEM) with a neural operator representation of a particle system (Smoothed Particle Hydrodynamics, SPH) for a uniaxial tension problem with hyperelastic material. What makes this approach unique is that a well-trained over-parametrized DeepONet can generalize well and make predictions at a negligible cost.

6.2 Introduction

Predicting and monitoring complex systems, where small-scale dynamics and interactions affect global behavior are ubiquitous in science and engineering [280, 5, 54, 255, 14]. In disciplines ranging from material fracture [95] to design problems [64], models at microscale have shown their capability in representing detailed material response. However, despite their improved accuracy, the usability of microscopic models is often compromised by several computational challenges. Specifically, microscopic models require a small spatio-temporal scale to fully resolve small-scale details and capture underlying stochastic dynamics, leading to a prohibitive computational expense. Therefore, simulating material dynamics at meso- or macroscale using microscopic models is still largely beyond reach. In addition, although bottom-up approaches such as fine-grained atomistic models have provided important insights into processes at microscale, they generally do not scale up to finite-size samples [276, 319, 113]. These challenges raise the need for efficient mathematical models and algorithms, which are capable of capturing small-scale behaviors while being computationally expedient.

In the last two decades, a variety of multiscale approaches have been proposed to address these challenges. Microscopic effects often concentrate locally, whereas a continuum model can accurately describe the system in the rest of the domain. Solving such a continuum model by well-established numerical methods would reduce the computational cost. Following this domain-decomposition strategy, several works that combine continuum and microscopic models have been proposed [193, 149, 289, 188, 258, 123, 252, 122, 48] to model multiscale systems. Another approach focuses on developing a fast surrogate as the fine-scale model represented by homogenization [323, 16, 281, 58, 308, 21]. For example,

[177] considered an approximation model of a partial differential equation (PDE) that contains small-scale oscillations in its coefficients, in essence, replacing these coefficients in the model with effective properties so that the resulting solutions can adequately approximate the solutions of the original problem. However, quantifying effective properties poses a challenging task in light of the feasibility of acquiring parameter values and the level of accuracy of the homogenized PDE model compared to the original microscopic model.

Recently, deep learning algorithms have been proposed for simulating physical problems [35, 119, 318, 30, 202]. In particular, neural networks have been employed in conjunction with standard numerical models to address the aforementioned challenges in multiscale modeling [274, 8, 212, 199, 5, 222, 40]. In [8], Arbabi et al. proposed a data-driven method that trains deep neural networks to learn coarse-scale partial differential operators based on fine-scale data. In [18], Bhatia et al. presented a novel paradigm of multiscale modeling that couples models at different scales using a dynamic-important sampling approach. A machine learning model is employed to dynamically sample in the phase space, hence enabling an automatic feedback from micro to macro scale. In [169], Masi and his collaborators developed a thermodynamics-based artificial neural network (TANN) and applied it in multiscale modeling of materials with microstructure. Their results demonstrated that TANN is capable of implicitly learning the constitutive model from data and predicting the corresponding stress fields based on state variables. The authors also demonstrated that TANN is capable of solving boundary value problem in a multiscale system with complex microstructure using double-scale homogenization scheme. Other applications of machine learning in multiscale modeling include parameter inference [285, 204, 291, 196], uncertainty quantification [39, 229], data-driven modeling [39, 222, 209, 105, 308, 153], etc.

In the last few years, a new family of machine learning model, deep neural operators, have been proposed to learn the solution operator of a PDE system implicitly [139, 141, 159, 307]. Unlike another type of scientific machine learning, physics-informed neural networks (PINNs) [213], these neural operators can solve a PDE system given a new instance of interface conditions or model parameters without retraining [32, 147, 302, 146, 140, 84, 167]. Hence, such computational advantage enables neural operators to serve as an efficient surrogate model in multiscale coupling tasks, and especially for time-dependent multiscale problems, which even today have remained prohibitively expensive. Among the state-of-the-art neural operators, the Deep Operator Network (DeepONet) serves as a unique model with exceptional generalization capability and flexibility, which can learn the solution operator in irregular domains even in the presence of noise [160]. Another possibility is to use the Fourier neural operator (FNO), which is fast but is limited to complex geometries and structured data [139, 141]. Herein, we choose DeepONet as the surrogate model. For a more thorough comparison between DeepONet and FNO, we refer the reader to [160, 131].

The present chapter aims to address the aforementioned challenges by developing a new multiscale coupling framework, as demonstrated in Fig.6.1. Specifically, the new framework couples a surrogate of the microscopic system (DeepONet) with a standard finite element method (FEM) that represents the macroscopic model. The coupling framework is flexible in choosing the domain decomposition algorithms, the types of boundary condition, or the nature of multiphysics problems (static or time-dependent). In addition, the surrogate model can learn from a plurality of fine-resolution microscopic systems or from experimental data (Fig. 6.1(b)). Our approach is the first attempt to couple neural operators with standard numerical solvers using a concurrent coupling method.

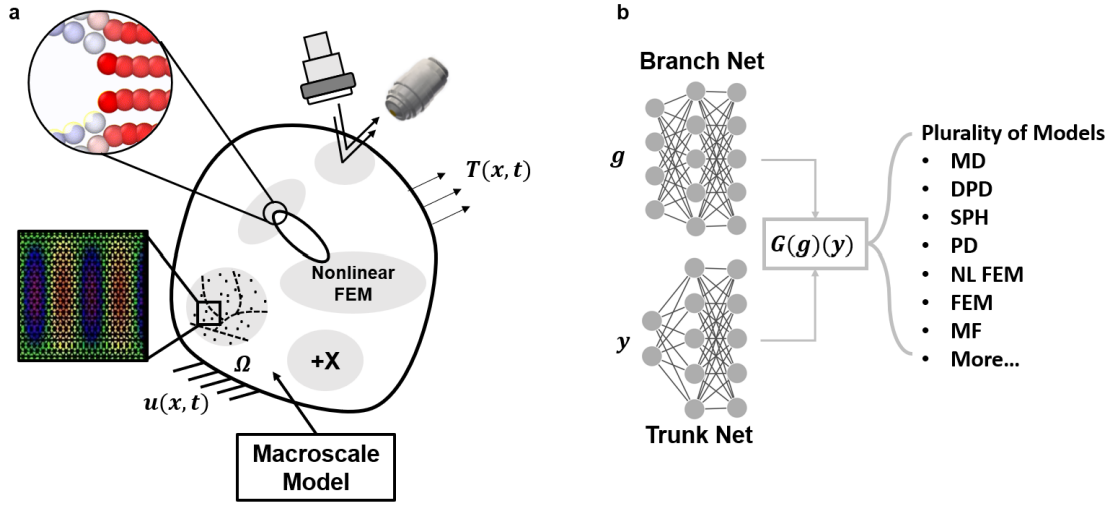


Figure 6.1: Schematic of multiscale modeling with DeepONet. (a) In a multiscale mechanics system, traction $T(x, t)$ acts on the boundary of domain Ω , which is decomposed into sub-domains described by a variety of microscopic models. A macroscopic model describes the response in the bulk region whereas nonlinear, microscopic, or data-driven models capture the detailed response in the small regions. (b) DeepONet, composed of a branch and a trunk net, is able to learn the response of the microscopic systems (from simulated or multi-modal data) and serve as a surrogate in the multiscale system. MD: molecular dynamics, DPD: dissipative particle dynamics, SPH: smoothed particle hydrodynamics, PD: peridynamics, NL FEM: nonlinear finite element, FEM: finite element, MF: multifidelity.

The chapter is organized as follows: In Sec. 6.3.1, we briefly introduce DeepONet and the adopted coupling algorithms. In Sec. 6.4, we report the coupling performance for a series of benchmarks, including a 2D Poisson equation, a 1D heat problem, and a uniaxial tension problem with elastoplastic materials and hyperelastic materials. We conclude by a brief discussion on the implications of the presented model in Sec. 6.5. In the appendices, we give an overview of the smoothed particle hydrodynamics (SPH) method and present additional results for FEM and SPH simulations. Finally, we provide further details on the network training and data generation.

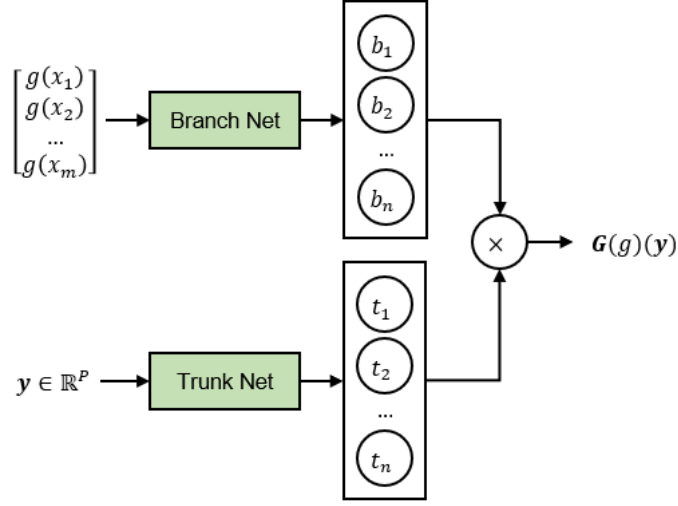


Figure 6.2: Schematic architecture of DeepONet. DeepONet learns the mapping operator G from an input function g to its corresponding output function $G(g)$. The input of branch and trunk net are g and $\mathbf{y} \in \mathbb{R}^P$, which represents a discretized function from $g(x_1)$ to $g(x_m)$ and information such as coordinates and time, respectively. Note that the output dimension of the trunk net, n , is consistent with that of the branch net. The final output $G(g)(\mathbf{y})$ is computed as the dot product of $\mathbf{b}$ and $\mathbf{t}$.

6.3 Methodology

In this section, we introduce the general architecture of DeepONet (Sec. 6.3.1) and the domain decomposition methods for the coupling framework (Sec 6.3.2).

6.3.1 Deep Operator Network (DeepONet)

We briefly summarize the general architecture of DeepONet employed in this work. Let $\Omega \subset \mathbb{R}^p$ be a bounded open set, which is the domain of our input and output functions; DeepONet can learn a general continuous operator between two Banach spaces of functions taking values in $\mathbb{R}^{d_f}$ and $\mathbb{R}^{d_u}$, respectively. We denote the input and output function spaces as $\mathcal{A} = \mathcal{A}(\Omega; \mathbb{R}^{d_f})$ and $\mathcal{U} = \mathcal{U}(\Omega; \mathbb{R}^{d_u})$. The network aims at approximating a mapping $G : \mathcal{A} \rightarrow \mathcal{U}$ between an input function

$g \in \mathcal{A}$ and its corresponding output function $G(g) \in \mathcal{U}$. Although DeepONet can learn mappings between vector-valued functions, for simplicity of exposition we focus on learning scalar-valued functions ($d_u = 1$) in the following. For any $\mathbf{y} \in \Omega \subset \mathbb{R}^p$, $G(g)(\mathbf{y}) \in \mathbb{R}$ is the evaluation of function $G(g)$ at $\mathbf{y}$. As shown in Fig. 6.2, the branch network takes a function g in its discrete representation $[g(\mathbf{x}_1), g(\mathbf{x}_2), \dots, g(\mathbf{x}_m)]$ at locations $\{\mathbf{x}_j\}_{j=1}^m$ as input and yields an array of operator features, $\{b_i\}_{i=1}^n$, as its output. The trunk network takes $\mathbf{y}$ as input and yields another array of operator features, $\{t_i\}_{i=1}^n$. Finally, $G(g)(\mathbf{y})$ can be approximated by [159, 41]:

$$G(g)(\mathbf{y}) \approx \sum_{i=1}^n b_i t_i. \quad (6.1)$$

In this work, we adopt fully-connected neural networks as the architecture of both sub-networks. We refer the readers to [159, 135] for theoretical analysis and error estimations of DeepONet.

Regarding the loss function, we adopt the mean squared error (MSE) which measures the square of L_2 norm between model predictions and training data. Given M sample input functions $g^{(i)}$, $i = 1, \dots, M$, and the ground-truth of their corresponding output functions, $G_{\text{data}}(g^{(i)})(\cdot)$ on a set of N points $\{\mathbf{y}_j\}_{j=1}^N$, the loss is expressed as:

$$\mathcal{L} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (G_{\text{model}}(g^{(i)})(\mathbf{y}_j) - G_{\text{data}}(g^{(i)})(\mathbf{y}_j))^2 + \hat{R}, \quad (6.2)$$

where G_{model} is the learned operator of DeepONet for approximation and $G_{\text{data}}(g^{(i)})(\mathbf{y}_j)$ denotes the data measurements for the i -th output function at j -th evaluation point. $\hat{R}$ is an additional loss term for the purpose of regularization (see Sec. 6.4.1 and Eq. 6.11).

6.3.2 Coupling Methods

Here we introduce the procedure of coupling DeepONet with a numerical model. Consider a system defined on a computational domain Ω ; we decompose the domain into Ω_I and Ω_{II} according to the features in each domain. These two domains are described by a macroscopic model (model I) and a microscopic model (model II), respectively. In this chapter, we choose model I as FEM. Nonetheless, the proposed procedure can be generalized to couple other mesh-based models [198, 120, 100, 269] or particle (meshfree) models [181, 60, 85, 218] as shown in Fig. 6.1(b). Model II represents a DeepONet, which serves as a surrogate for the microscopic, fine-scale model.

Notably, the domain decomposition can be either overlapping or non-overlapping. Fig. 6.3(a) shows an one-dimensional illustration of the domain decomposition method. For the overlapping setting, $\Gamma_1 (= \{x_1\})$ for this 1D case) represents the internal boundary of model I and $\Gamma_2 (= \{x_2\})$ is the boundary of model II. The overlapping region is $\overline{\Omega_I} \cap \overline{\Omega_{II}} = [x_2, x_1]$. The two models communicate with each other by exchanging interface conditions on Γ_1 and Γ_2 . For the non-overlapping setting, we have $\Gamma_1 = \Gamma_2 := \Gamma$ ($x_1 = x_2$ for this 1D case) and the two model exchanges interface information on Γ . Fig. 6.3(b) and Algorithm 1 summarize the iterative procedure of the coupling framework with a Robin-type boundary condition [311]. For the n -th iteration, we utilize quantities $(\cdot)^n$ to calculate $(\cdot)^{n+1}$. To initiate the coupling procedure ($n = 0$), we start with an initial guess of the interface solution value $u^0(x_1)$ and derivative-related information $T^0(x_1)$ on Γ_1 . Then, we take a linear combination of these quantities and forms a Robin-type boundary condition on Γ_1 , namely, $\tilde{h}^0(x_1) = R_1 u^n(x_1) + R_2 T^0(x_1)$, which is applied on Model I. The method proceeds by solving for $u_I^{n+1}(x)$, $x \in \Omega_I$, from Model I, and interpolat-

ing the computed solution or its derivatives at Γ_2 . The interpolated information will then be transmitted to Model II as the boundary condition. Then, we solve for $u_{II}^{n+1}(x)$, $x \in \Omega_{II}$, from Model II with the transmitted interface condition on Γ_2 and interpolate its solution on Γ_1 . If the stopping criterion

$$\|u_I^{n+1} - u_I^n\|_{L^2(\Omega_I)}^2 + \|u_{II}^{n+1} - u_{II}^n\|_{L^2(\Omega_{II})}^2 < \epsilon, \quad (6.3)$$

is satisfied, the coupling result is considered to be converged. The solution for the two domains is:

$$u_I(x) := u_I^{n+1}(x), \quad x \in \Omega_I, \quad (6.4)$$

$$u_{II}(x) := u_{II}^{n+1}(x), \quad x \in \Omega_{II}. \quad (6.5)$$

If the solution is not converged, we proceed to update the interface information on Γ_1 with a relaxation formulation: $\tilde{h}^{n+1}(x_1) = (1 - \theta)h_I^{n+1}(x_1) + \theta h_{II}^{n+1}(x_1)$ where $h_{I,II}^{n+1} = R_1 u_{I,II}^n(x_1) + R_2 T_{I,II}^0(x_1)$ are the Robin boundary condition from Model I and II. Here, the relaxation parameter $\theta \in [0, 1]$ can be either fixed or updated according to the Aitken's rule [179, 311]. Then, we proceed to a new iteration by transmitting the updated boundary condition $\tilde{h}^{n+1}(x_1)$ to Model I and repeating the procedure stated above with $n \leftarrow n+1$. This procedure is repeated until the stopping criterion is satisfied. These coupling algorithms have their origin to the classical Schwarz coupling methods [152, 182, 67] and an iterative patching algorithms [67].

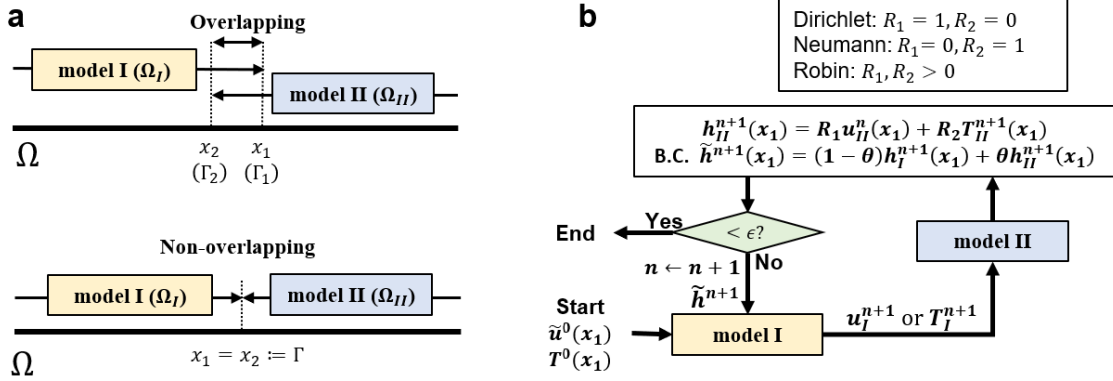


Figure 6.3: A flexible framework in domain decomposition. The proposed coupling framework is able to adopt either (a) overlapping or non-overlapping domain decomposition coupled with Dirichlet, Neumann, or Robin boundary conditions at the interface. As an illustration, (b) presents a Robin-type boundary condition imposed on model I with interfacial solution $\tilde{u}(x_1)$ updated by a relaxation scheme. Dirichlet and Neumann boundary can be imposed by adjusting the value of R_1 and R_2 . The relaxation parameter, θ , is either fixed at a value or updated dynamically. $T_{II}^n(x_1)$ represents the Neumann-related information at x_1 from Model II, e.g., traction or derivatives.

Algorithm 1 Coupling Method. $\Gamma_1 := \partial\Omega_I$, $\Gamma_2 := \partial\Omega_{II}$. If non-overlapping, $\Gamma_1 = \Gamma_2$, else $\Gamma_1 \neq \Gamma_2$.

Initialization: Set model I with $u(x_1) = 0$ and model II with $u(x_1) = 0$

Main Loop:

for $n = 0 : n_{max} - 1$ **do**

Model I (FEM):

- Receive the interface information $\tilde{h}^n(x_1)$ from Model II ($x_1 \in \Gamma_1$).
- Solve for u_I^{n+1} from Model I.
- Calculate $u_I^{n+1}(x_2)$ or $\frac{\partial u_I^{n+1}}{\partial x}|_{x_2}$ and pass it to Model II ($x_2 \in \Gamma_2$).

Model II (NN):

- Receive the interface information $u_I^{n+1}(x_2)$ or $\frac{\partial u_I^{n+1}}{\partial x}|_{x_2}$ from Model I ($x_2 \in \Gamma_2$).
- Solve for $u_{II}^{n+1}/T_{II}^{n+1}$ from Model II.
- Calculate $h_{II}^{n+1}(x_1) = R_1 u_I^{n+1}(x_1) + R_2 T_{II}^{n+1}(x_1)$.
- Calculate $\tilde{h}^{n+1}(x_1) = (1 - \theta) h_I^{n+1}(x_1) + \theta h_{II}^{n+1}(x_1)$ and pass it to Model I ($x_1 \in \Gamma_1$).

If converged, stop;
end for

Problem	Model I	Model II/Training Data	Interface Condition	Overlap
2D Poisson	FEM	DeepONet/FEM	D-D, R-D	Yes
1D Heat	FEM	DeepONet/FEM	N-D	No
2D Elastoplasticity	Linear Elastic FEM	DeepONet/Elastoplastic FEM	N-D	No
2D Hyperelasticity	Hyperelastic FEM	DeepONet/Hyperelastic SPH	N-D, R-D/R-N	No

Table 6.1: Setup for the four examples. D-D: Dirichlet-Dirichlet, R-D: Robin-Dirichlet, N-D: Neumann-Dirichlet, R-N: Robin-Neumann

6.4 Results

In this section, we present the simulation results of four benchmark problems: 2D Poisson equation, 1D heat equation, 2D elastoplasticity, and 2D hyperelasticity, to demonstrate the applicability of our method. The detailed settings of these four examples, including the choices of model I and model II, interface conditions, and whether domains overlap, are provided in Table 6.1. For interface conditions, we studied four types of conditions, namely, Dirichlet-Dirichlet (D-D), Robin-Dirichlet (R-D), Neumann-Dirichlet (N-D), and Robin-Neumann (R-N). The first letter represents the boundary condition for Model I and the second letter for Model II. For each problem, our framework consists of two stages. In the first stage, we train a DeepONet offline to obtain a surrogate of model II. Then, we couple Model I with DeepONet in the online stage using our proposed coupling method. Training data of elastoplastic FEM in Sec. 6.4.3 and hyperelasticity in Sec. 6.4.4 are generated by Abaqus [2] and an SPH solver [71]. All the other usages of FEM (including FEM solver for model I and generation of training data for model II) are based on the FEniCS package [6] using second-order Lagrange polynomials. For all experiments, the networks are trained on a NVIDIA GeForce RTX 2060 GPU. The FEM simulations are performed on two 6-core Intel Core i7-8700 CPUs.

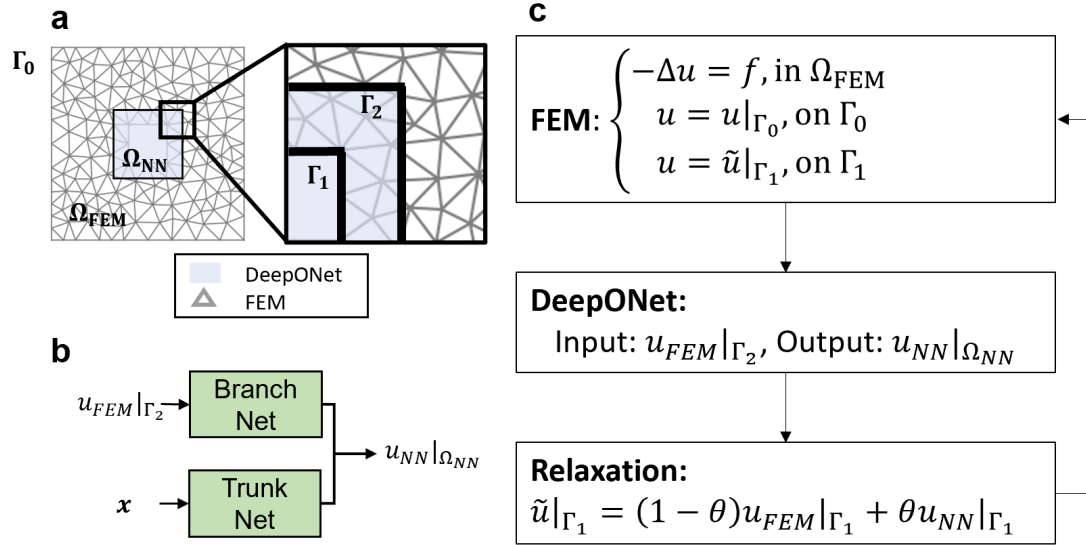


Figure 6.4: Setup of the Poisson equation. (a) A 1×1 unit square is decomposed into two overlapping regions, Ω_{NN} (light blue) and Ω_{FEM} (meshed). Ω_{FEM} is bounded by Γ_1 and Γ_0 , whereas Ω_{NN} is a 0.3×0.3 square in the center with boundary Γ_2 . (b) DeepONet takes $u_{FEM}|_{\Gamma_2}$ as input and yields the solution in Ω_{NN} , denoted as $u_{NN}|_{\Omega_{NN}}$. (c) Formulation of an overlapping D-D method. Given the boundary condition on Γ_0 , the coupling framework iteratively updates the corresponding boundary condition on Γ_1 . The relaxation parameter θ is either fixed or updated based on the Aitken's rule [311].

6.4.1 Poisson equation

We first study the feasibility of the proposed framework for solving a static problem. Let us consider a Poisson equation described by the PDE system in $\Omega := \Omega_{FEM} \cup \Omega_{NN}$:

$$-\Delta u(\mathbf{x}) = f(\mathbf{x}), \text{ in } \Omega = [0, 1]^2, \quad (6.6)$$

$$u(\mathbf{x}) = u|_{\Gamma_0}(\mathbf{x}), \text{ on } \Gamma_0, \quad (6.7)$$

where we set $f(\mathbf{x}) = 6$ for all $\mathbf{x} \in \Omega$. As shown in Fig. 6.4(a), we decompose one domain into two overlapping subdomains, namely $\Omega_{FEM} = [0, 1]^2 / [0.4, 0.6]^2$ (with an internal boundary Γ_1) and $\Omega_{NN} = [0.3, 0.7]^2$ (with a boundary Γ_2). In Ω_{FEM} , the

system is governed by Eqs. (6.6) and (6.7) with an boundary condition

$$u(\mathbf{x}) = \tilde{u}|_{\Gamma_1}(\mathbf{x}), \text{ on } \Gamma_1. \quad (6.8)$$

In this example, we first verify the efficacy of the coupling framework with a D-D interface condition. For this setup, in Ω_{NN} , the system is governed by Eqs. (6.6) and (6.7) with input

$$u(\mathbf{x}) = u_{FEM}|_{\Gamma_2}, \text{ on } \Gamma_2. \quad (6.9)$$

After presenting the results of the D-D case, we display results of parametric studies for better demonstrating the influence of other factors that may influence the convergence rate.

In the offline stage, we train a DeepONet as a surrogate of FEM for predicting solution in Ω_{NN} . First, we sample a set of boundary conditions $u|_{\Gamma_2}$ from a random field with $\alpha = 5$, a parameter in the correlation function in Eq. (6.53). Qualitatively, a smaller α results in a less smooth function. More details of the random field generation are provided in 6.D. Then, we solve the Poisson equation with the randomly sampled boundary conditions using FEM, whose computational results in Ω_{NN} are collected as the training cases. We generate 1,000 cases as the training dataset of DeepONet following the aforementioned process. As shown in Fig. 6.4(b), the branch network input $u_{FEM}|_{\Gamma_2}$ comes from an interpolation of the FEM solution on Γ_2 , while the trunk network takes the coordinate $\mathbf{x}$ ($\in \Omega_{NN}$) as its input. The network output is $u_{NN}(\mathbf{x})$. We present more details related to the network training in 6.C.

After the completion of the offline training, we proceed to the coupling stage

as shown in Fig. 6.4(c). Given a fixed boundary condition on Γ_0 , we initiate the coupling framework with an initial guess of u on Γ_1 , which is typically zero. Then, we solve the governing equation in Ω_{FEM} with FEM and interpolate the solution on Γ_2 (denoted as $u_{FEM}|_{\Gamma_2}$), which will be used as the input of the branch network. With this input, the DeepONet then predicts $u_{NN}|_{\Omega_{NN}}$, the solution of the Poisson equation in Ω_{NN} . Following that, a relaxation scheme is adopted to update the interface condition on Γ_1 as: $\tilde{u}|_{\Gamma_1} = (1 - \theta)u_{FEM}|_{\Gamma_1} + \theta u_{NN}|_{\Gamma_1}$, which will later be used as the boundary condition of FEM in the next iteration. The coupling iteration continues until the solution u converges.

We show the performance of our coupling framework for the Poisson equation in Fig. 6.5. We fix the relaxation parameter as $\theta = 0.5$ and impose a boundary condition on Γ_0 from the testing dataset (unseen to the DeepONet in the training stage). Figs. 6.5(a-b) show a comparison between the FEM ground truth (first column; computed directly in Ω) and the predictions from the coupling framework (second column; first row in Ω_{FEM} from FEM, second row in Ω_{NN} from DeepONet) together with their difference (third column). We observe an agreement between the coupling predictions and the ground truth. The maximum absolute error presented in the third column is less than 1%. We also note that the error is larger in Ω_{FEM} , especially in the region close to its external boundary Γ_0 . The observed error is mostly dominated by the interpolation accuracy in the FEM, not the error of the coupling framework. We further show a quantitative comparison between models prediction and ground-truth on Γ_1 and Γ_2 in Fig. 6.5(c): the predictions from coupled FEM/NN (blue dashed lines and green triangles, respectively) accurately reproduce the true solution (red lines) with a relative error at around 0.07%.

In addition to the results shown in Fig. 6.5, we also conducted parametric studies on diverse factors that influence the performance of our framework, including:

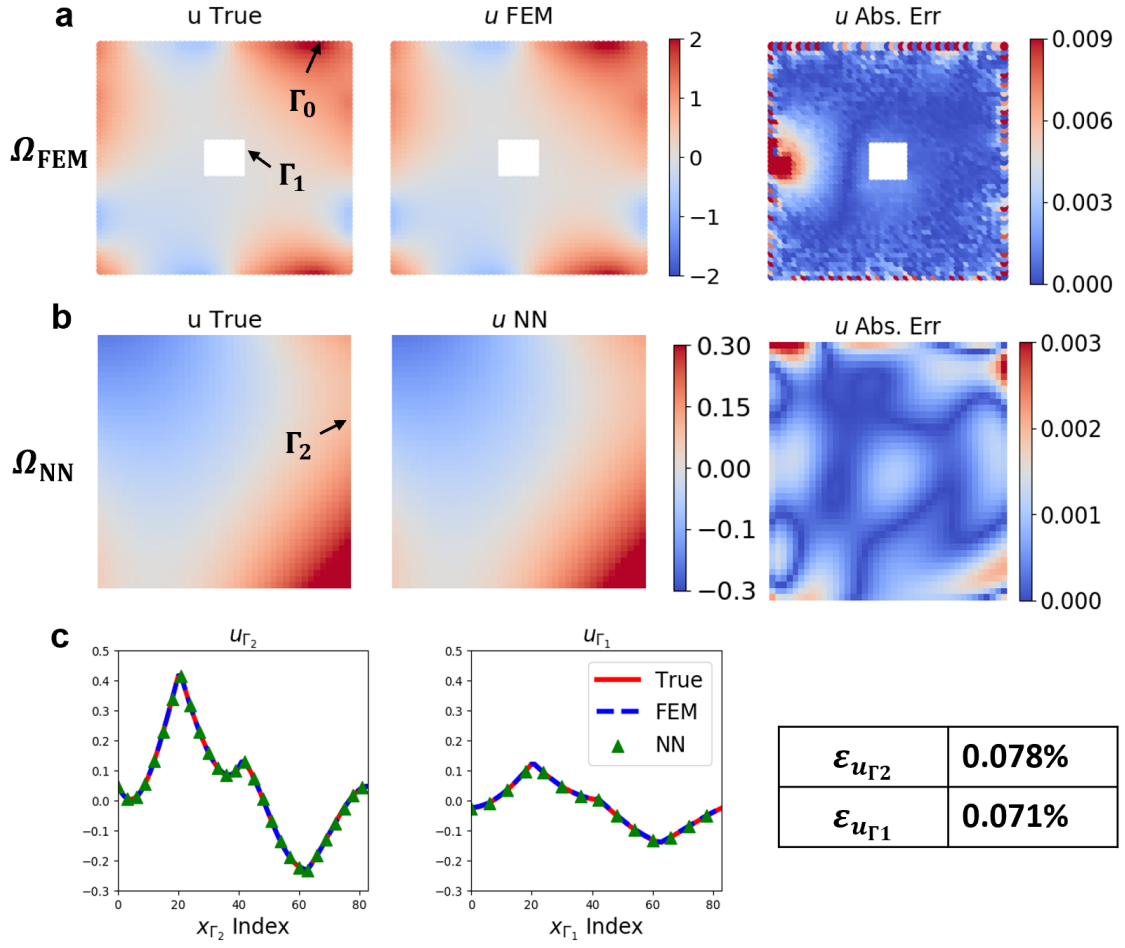


Figure 6.5: Results of coupling FEM and DeepONet for the Poisson equation. (a-b) Model predictions from the FEM and DeepONet with the corresponding absolute errors in Ω_{FEM} and Ω_{NN} . (c) Model predictions at the interfaces ($u|_{\Gamma_1}$ and $u|_{\Gamma_2}$) and the true solution (red line). The relative errors of model predictions at the interfaces are far less than 1%. Please see Fig. 6.4 for Ω_{NN} and Ω_{FEM} .

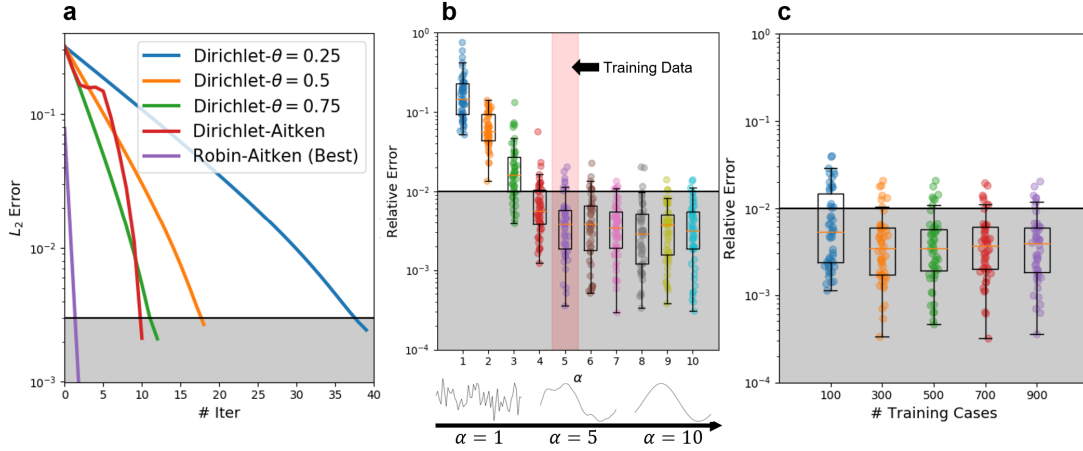


Figure 6.6: Parametric studies of the Poisson equation. (a) Given a boundary condition on Γ_0 , the convergence history of the coupling model varies with the coupling method (Robin or Dirichlet) and parameter θ . (b) The model is trained with data generated from a random field with $\alpha = 5$. We test the generalization ability represented by the relative errors of the coupling model for cases corresponding to $\alpha = 1$ to 10. (c) Box plots of relative errors for DeepONets trained with 100 to 900 cases. Each column shows the coupling results by testing with 50 different cases. The shaded area indicates that the relative error on Γ_1 is less than 1%. Notice that we present the error with respect to the ground truth to show the convergence and accuracy of our framework.

convergence of the framework with different coupling methods (Fig. 6.6(a)); the coupling accuracy influenced by the extrapolation capability of the DeepONet (Fig. 6.6(b)) and the number of training cases (Fig. 6.6(c)). In Fig. 6.6(a), convergence of errors with different boundary conditions is plotted against iterations. The shaded area indicates that the relative error is less than 1%. Specifically, given various values of the relaxation parameter θ ($\theta = 0.25, 0.5$ and 0.75), the displacement errors with Dirichlet boundary conditions satisfy the stopping criterion (L_2 error less than 2×10^{-3} , or equivalently, relative error less than 1%) at iteration 37, 18, and 12, respectively. With the Aitken’s relaxation strategy for θ (see, e.g., [311]), the coupling error (denoted as “Dirichlet-Aitken”) converges a bit faster than that of a fixed relaxation parameter $\theta = 0.75$. We also adopt a Robin-type boundary (Robin-Aitken) with dynamic update in θ (purple line). In this case of Robin boundary condition, the system in domain Ω_{FEM} is governed by Eqs. (6.6) and

(6.7) with a Robin boundary:

$$R_1 u + R_2 \frac{\partial u}{\partial n} = g, \text{ on } \Gamma_1, \quad (6.10)$$

where we set $R_1 = 1$ and $R_2 = 1$. Since the FEM needs both the information of the solution and its derivative in the normal direction from DeepONet to update the Robin boundary condition, we adjust the training loss of the network following the strategy. The regularization term $\hat{R}$ in Eq. (6.2) is set as:

$$\hat{R} = \left(\frac{\partial u}{\partial n} \Big|_{NN} - \frac{\partial u}{\partial n} \Big|_{true} \right)^2, \mathbf{x} \in \Gamma_1 \quad (6.11)$$

to regularize the partial derivative with respect to the normal direction of Γ_1 . In Eq. (6.2), the partial derivative term is computed by taking the automatic differentiation of the network output with respect to the trunk net input $\mathbf{x}$ [84, 275, 305, 317, 315]. We assign the weight of the regularization term as 1. With the employment of the Robin boundary condition, we observe that the method only takes two iterations to reach a relatively small error.

In Fig. 6.6(b), we show the generalization ability of DeepONet and its impact on the accuracy of the coupling framework by testing with boundary conditions outside the training region. The minimal relative errors of $u|_{\Gamma_2}$ are plotted against the correlation length α of the random field that was used to generate training data. When the correlation length α increases, the sampled curves become smoother and vice versa. Notice that we train the network on training samples with $\alpha = 5$ and test its performance on α ranging from 1 to 10, each with 50 testing cases. The relative errors drop from around 10% to less than 1% with α increasing from 1 to 5. For $\alpha > 5$, the errors are statistically stable.

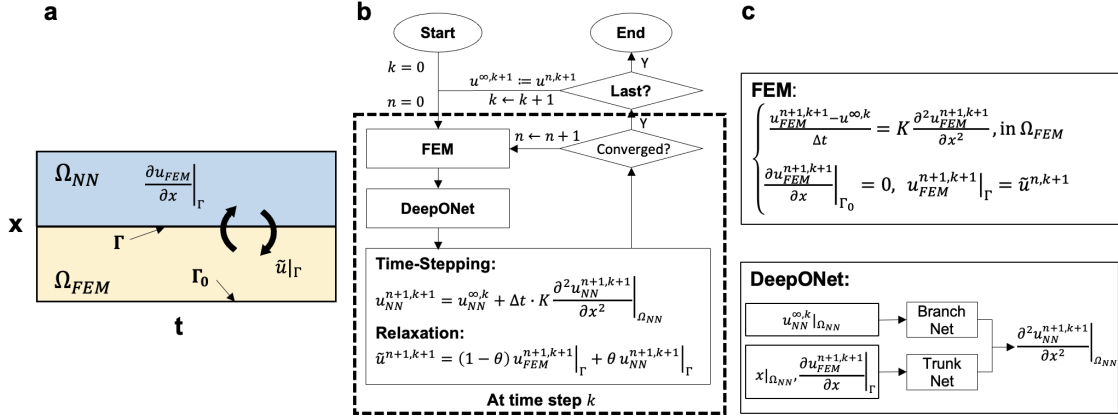


Figure 6.7: Setup of the time-dependent problem (heat equation). (a) The spatio-temporal domain is decomposed into two sub-domains with Γ as the interface between FEM and NN domain. (b-c) Formulation of the coupling framework for the heat equation. The framework first solves for $u_{FEM}^{n+1,k+1}$ from FEM with updated information at the interface $\tilde{u}$ at k -th step. Then, the computed derivative at Γ is transmitted into DeepONet, which predicts the spatial derivative in Ω_{NN} (c). The solution in Ω_{NN} is computed based on a time-stepping scheme, followed by a relaxation update (b). The initialization at $k=0$ and $n=0$ is described in the main text.

Fig. 6.6(c) exhibits the accuracy of the coupling framework as a function of the generalization ability of DeepONet, which is reflected by the number of training cases for DeepONet. The median of the relative errors of $u|_{\Gamma_1}$ decreases slightly when the number of training cases increases from 100 to 300 and then stays statistically stable even with further increase. The computational results implicitly suggest that after 300 training cases, the errors are most contributed by the accuracy of boundary interpolation, not the generalization of DeepONet. The shaded area in (b-c) denotes relative error at 1%.

6.4.2 Heat equation

Next, we investigate the performance of the coupling framework for a dynamic problem, namely, 1D heat equation, with a N-D method. Consider a PDE system

in $\Omega := \Omega_{FEM} \cup \Omega_{NN}$:

$$\frac{\partial u}{\partial t} = K \frac{\partial^2 u}{\partial x^2}, \quad \text{for } (x, t) \in \Omega := [0, 1.0] \times [0, 10.0], \quad (6.12)$$

$$u(x, 0) = u_0(x), \quad \text{for } x \in [0, 1.0], \quad (6.13)$$

$$u_x(x, t) = 0, \quad \text{for } (x, t) \in \Gamma_0 := \{0, 1.0\} \times [0, 10.0], \quad (6.14)$$

where the thermal diffusivity K is set as 0.1. As shown in Fig. 6.7(a), we decompose the computational domain Ω into two non-overlapping subdomains: $\Omega_{FEM} := (x, t) \in [0, 0.5] \times [0, 10.0]$ for FEM and $\Omega_{NN} := (x, t) \in [0.5, 1] \times [0, 10.0]$ for DeepONet. The interface is $\Gamma := \{0.5\} \times [0, 10]$. For the FEM subdomain Ω_{FEM} , we set a Dirichlet boundary condition at the interface Γ :

$$u(x, t) = \tilde{u}(x, t) \quad \text{for } (x, t) \in \Gamma, \quad (6.15)$$

where $\tilde{u}$ is the updated and relaxed solution at the interface. For the DeepONet subdomain Ω_{NN} , we impose the interfacial flux from FEM $K \frac{\partial u_{FEM}}{\partial x}$ as input.

The coupling method is illustrated in Fig. 6.7(b-c). The solution process is a nested loop with indices n and k , where k represents the time step and n is the current iteration step. To initiate the framework, we set both n and k as zero with an initial guess of interfacial flux for the FEM. The FEM solves for the solution $(u_{FEM}^{n+1,k+1}|_{\Omega_{FEM}})$ in Ω_{FEM} at time step $k+1$ given the solution at the previous time step k , denoted as $u_{FEM}^{\infty,k}|_{\Omega_{FEM}}$. Then, we transmit the flux at Γ , $K \frac{\partial u_{FEM}^{n+1,k+1}}{\partial x}|_{\Gamma}$, to DeepONet as a part of the trunk net input (Fig. 6.7(c)). The branch network takes the system solution in Ω_{NN} at k -th time step as input. The output of DeepONet estimates an approximation of the diffusion term $K \frac{\partial^2 u_{NN}^{n+1,k+1}}{\partial x^2}$. The solution in Ω_{NN} is then calculated by a semi-discretized heat equation with the backward Euler method: $u_{NN}^{n+1,k+1}(x) = u_{NN}^{n,k}(x) + \Delta t \cdot K \frac{\partial^2 u_{NN}^{n+1,k+1}}{\partial x^2}$. Next, the interfacial flux $\tilde{u}^{n+1,k+1}$ is updated by

the relaxation scheme:

$$\tilde{u}^{n+1,k+1} = (1 - \theta)u_{FEM}^{n+1,k+1}|_{\Gamma_1} + \theta u_{NN}^{n+1,k+1}|_{\Gamma_1} \quad (6.16)$$

In this example, we fix the relaxation parameter $\theta = 0.5$. If the updated interfacial flux is not yet converged, we continue to the next iteration. Otherwise, if the flux is converged, then we proceed to the next time step in the outer loop and restart the inner loop with the reset iteration step $n = 0$.

Herein, we summarize the training procedure of DeepONet. We generate 1,000 initial conditions in $x \in [0.5, 1.0]$ from a Gaussian random field with constant mean 0 and correlation length 0.3. For each case, we randomly sample the flux at $u|_{\Gamma_1}$ 20 times based on a uniform distribution from -3 to 3 as the boundary conditions. The training data of DeepONet is then generated by running FEM simulations on Ω_{NN} with the sampled initial/boundary conditions. Note that training a DeepONet in a spatio-temporal domain with disparate boundary/initial condition is a data-demanding task: one needs to sample in the spatio-temporal domain. Also, performance of the framework may be deteriorated when the network extrapolates the solution outside the training domain. Hence, we alleviate the challenges by training the network to learn the implicit spatial derivative operator $K \frac{\partial^2 u}{\partial x^2}$ with an explicit method to advance in time. In practice, we acquire the spatial derivative $K \frac{\partial^2 u}{\partial x^2}$ in Ω_{NN} from FEM simulations running from $t = 0$ to 1.0 with $dt = 0.1$. The simulated spatial derivative is then collected and utilized for DeepONet training. For more details of the network training, we refer the reader to Sec. 6.C.

The coupling results are shown in Fig. 6.8. Fig. 6.8(a) shows the prediction from both models in the spatio-temporal domain Ω . In Fig. 6.8(b), the solution of DeepONet at $x = 0.75$ is plotted against time. The prediction from DeepONet

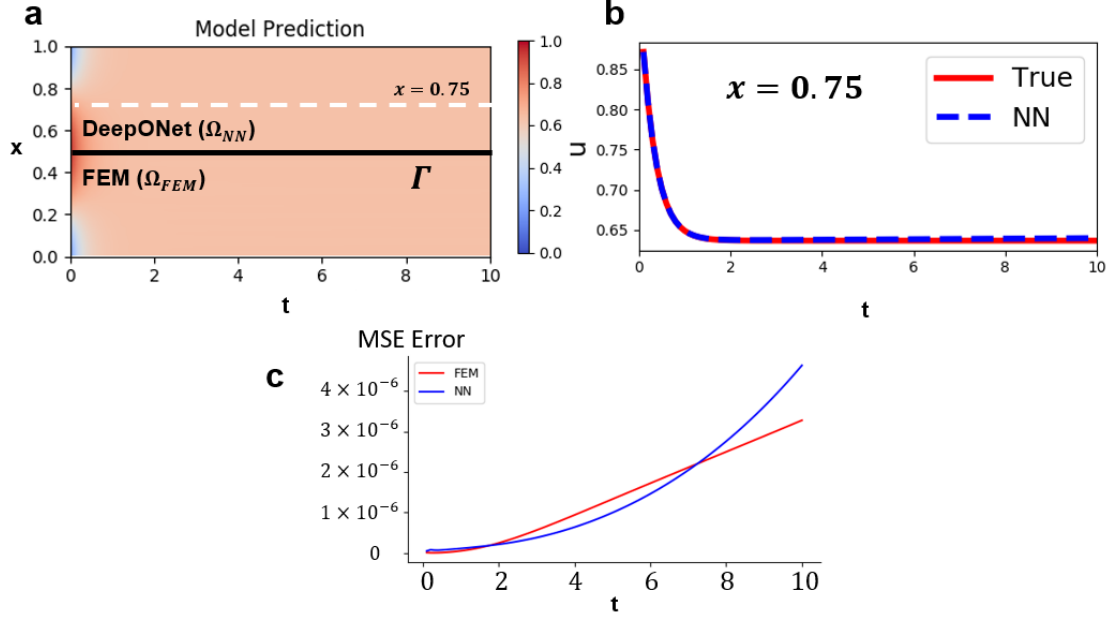


Figure 6.8: Results of coupling FEM and DeepONet for the heat equation. (a) Model predictions from DeepONet and FEM. DeepONet predicts the solution in $x \in \Omega_{NN}$ while FEM predicts the solution for $x \in \Omega_{FEM}$. The black line at $x = 0.5$ denotes the interface of the two domains. The solution at $x = 0.75$ (indicated by the white dashed line) is presented in (b). (c) The mean square errors of FEM (red) and DeepONet (blue) vs time (t).

shows a good agreement with the ground truth solution even for the temporal region outside the training dataset ($t > 1$), indicating that the proposed framework works well for extrapolation. The mean-squared errors (MSE) of both models in the coupling framework are plotted against time in Fig. 6.8(c): the coupling framework shows stability and accuracy over long-time integration. Although the prediction errors grow with the increase of t , we note that the errors accumulate at a relatively low rate, demonstrating that the proposed coupling framework is capable of solving a time-dependent system. It may be misleading that, from $t \in [2, 7]$, DeepONet outperforms FEM since the mean squared error of DeepONet is lower than that of FEM. In our concurrent coupling framework, the errors in FEM is induced by the oscillatory errors from DeepONet. Therefore, one can observe that the errors from FEM and DeepONet are both in a similar scale. In general, one might observe a slightly better accuracy from DeepONet, but these

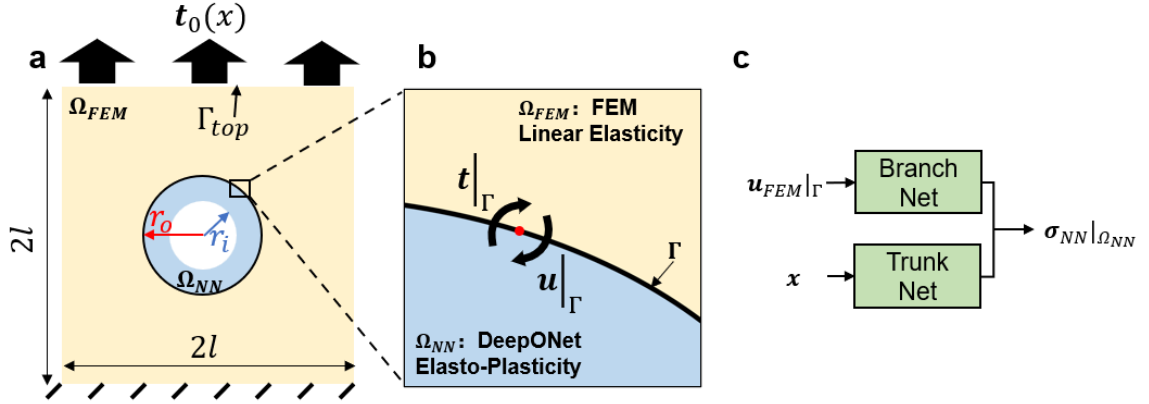


Figure 6.9: Setup of the elastoplasticity problem. (a) A $2l \times 2l$ solid plate is clamped on the bottom edge with a traction t_0 distributed on the top edge. (b) The yellow region (Ω_{FEM}) is modeled with linear elasticity, whereas the blue region (Ω_{NN}) is modeled with elastoplasticity. Interfacial displacement (Dirichlet boundary condition) $u|_\Gamma$ computed from FEM is transmitted to DeepONet as input. (c) The DeepONet estimates the stress components in Ω_{NN} , $\sigma_{NN}|_{\Omega_{NN}}$, based on which the interfacial traction $t_{NN}|_\Gamma$ can be calculated accordingly. The predicted traction is updated and provided to FEM as a Neumann-type boundary condition. In this example, we set the interior circle radius $r_i = 0.1$, the interface circle radius $r_o = 0.3$, and the plate size $l = 1$.

observations are generally problem-dependent and do not indicate that the data-driven DeepONet has a better generalization ability than FEM.

6.4.3 Elastoplasticity

In this section, we test the proposed framework for predicting the elastoplastic behavior of a solid material. As shown in Fig. 6.9(a), we consider a plane strain problem for a square-shaped solid of size $2l \times 2l$ with a circular void of radius r_i , where vertical tension is applied on its top edge. In this example, we take $l = 1$, $r_i = 0.1$. For linear elastic materials, the kinematics, constitutive relation, and

equilibrium equations are as follows:

$$\boldsymbol{\varepsilon} = \frac{1}{2}(\boldsymbol{\nabla} \mathbf{u} + \boldsymbol{\nabla} \mathbf{u}^T), \quad (6.17)$$

$$\boldsymbol{\sigma} = \lambda \text{tr}(\boldsymbol{\varepsilon}) \mathbf{I} + 2\mu \boldsymbol{\varepsilon}, \quad (6.18)$$

$$\mathbf{0} = \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} + \mathbf{b}, \quad (6.19)$$

where $\boldsymbol{\varepsilon}$, $\mathbf{u}$, and $\boldsymbol{\sigma}$ are the strain, displacement, and (Cauchy) stress; λ and μ are Lamé moduli, which we take as $\lambda = 0.5769$ and $\mu = 0.3846$, and $\mathbf{I}$ is the identity tensor. In this example, we consider a solid subject to no body load, and therefore set the body force term $\mathbf{b}$ as zero.

To model the plastic behavior of the material, we consider small-deformation, rate-independent elastoplasticity with isotropic hardening. The additive decomposition of the strain tensor writes $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^e + \boldsymbol{\varepsilon}^p$, where $\boldsymbol{\varepsilon}^e$ and $\boldsymbol{\varepsilon}^p$ are the elastic and plastic strains, respectively. The elastic strain $\boldsymbol{\varepsilon}^e$ is related to the stress by

$$\boldsymbol{\sigma} = \lambda \text{tr}(\boldsymbol{\varepsilon}^e) \mathbf{I} + 2\mu \boldsymbol{\varepsilon}^e. \quad (6.20)$$

The plastic strain $\boldsymbol{\varepsilon}^p$ is purely deviatoric (i.e., $\text{tr}(\boldsymbol{\varepsilon}^p) = 0$). We define the deviatoric stress $\mathbf{s}$, the increment of the equivalent plastic strain $d\bar{\varepsilon}^p$, and the equivalent tensile stress (Mises stress) $\bar{\sigma}$ as

$$\mathbf{s} = \boldsymbol{\sigma} - \frac{1}{3} \text{tr}(\boldsymbol{\sigma}) \mathbf{I}, \quad (6.21)$$

$$d\bar{\varepsilon}^p = \sqrt{\frac{2}{3}} d\boldsymbol{\varepsilon}^p : d\boldsymbol{\varepsilon}^p, \quad (6.22)$$

$$\bar{\sigma} = \sqrt{\frac{3}{2}} \mathbf{s} : \mathbf{s}, \quad (6.23)$$

respectively. The flow direction N^p and the increment of the plastic strain $d\boldsymbol{\varepsilon}^p$ are

given by

$$\mathbf{N}^p = \sqrt{\frac{3}{2}} \frac{\mathbf{s}}{\bar{\sigma}}, \quad (6.24)$$

$$d\epsilon^p = \sqrt{\frac{3}{2}} d\bar{\epsilon}^p \mathbf{N}^p. \quad (6.25)$$

Then, the yield function f can be defined as

$$f = \bar{\sigma} - Y(\bar{\epsilon}^p), \quad (6.26)$$

where the linear strain-hardening function Y is taken as

$$Y(\bar{\epsilon}^p) = Y_0 + H_0 \bar{\epsilon}^p, \quad (6.27)$$

The initial strength $Y_0 = 0.1$ and the hardening modulus $H_0 = 0.3$ are taken as two constant material parameters. The aforementioned mechanical quantities are subject to the Kuhn-Tucker complementary conditions:

$$f \leq 0, \quad d\bar{\epsilon}^p \geq 0, \quad (d\bar{\epsilon}^p)f = 0. \quad (6.28)$$

In addition, when $f = 0$, the consistency condition $d\bar{\epsilon}^p df = 0$ also needs to be satisfied.

Due to the setup of our boundary value problem, the plastic deformation concentrates around the void whereas the material is dominated by elastic behaviors in regions away from the void. Hence, we decompose the square domain into two non-overlapping subdomains (see Fig. 6.9(a-b)): the internal region Ω_{NN} , which is an annulus with internal radius $r_i = 0.1$ and external radius $r_o = 0.3$ (on Γ), modeled by DeepONet as a surrogate for the solid's elastoplastic response; the external

region Ω_{FEM} , modeled by FEM for linear elasticity. The two regions share a common interface on Γ . In Ω_{NN} , we train several DeepONets as surrogates of each stress components to capture the plastic behavior. First, we sample 1,000 displacement boundary conditions on the top edge using the sampling method described in 6.D.2, and employ the sampled data as boundary conditions. Then, we solve for the displacement and stress fields in the entire domain with a FEM solver based on the elastoplasticity model described above. Next, we collect the simulation results in Ω_{NN} , which will be employed as the training data of DeepONets. As depicted in Fig. 6.9(c), DeepONet can predict the corresponding Cauchy stresses in Ω_{NN} with input as the displacement at the interface. In 2D problems, the Cauchy stress $\sigma(\mathbf{x})$ for each material point is a 2×2 symmetric matrix. Therefore, to model the stress we only need to predict its three components, namely, σ_{11} , σ_{12} , and σ_{22} . For each of these components, we train an independent DeepONet separately as a surrogate of these quantities. The traction at the interface $\mathbf{t}|_{\Gamma}$ is calculated accordingly based on the predicted stress from the DeepONets. More details regarding the training of this network are presented in 6.C.

Then, we employ the trained DeepONets in the coupling framework. As depicted in Fig. 6.9, DeepONets and FEM communicate at the interface Γ by transmitting the information of displacement and traction (Fig. 6.9). The interfacial displacement, $\mathbf{u}_{FEM}|_{\Gamma}$, is computed in FEM and transmitted to the DeepONets as the input of the branch network. Then, the network solves for the corresponding stress in Ω_{NN} and calculates the traction at the interface ($\mathbf{t}_{NN}|_{\Gamma}$). In the next iteration, the computed traction $\mathbf{t}_{NN}|_{\Gamma}$ will be imposed as the boundary condition of the FEM model. In this example, we employ a relaxation scheme for the traction from the DeepONets. The relaxation parameter θ is fixed at 0.5.

We present the coupling results with a N-D interface condition in Fig. 6.10. In

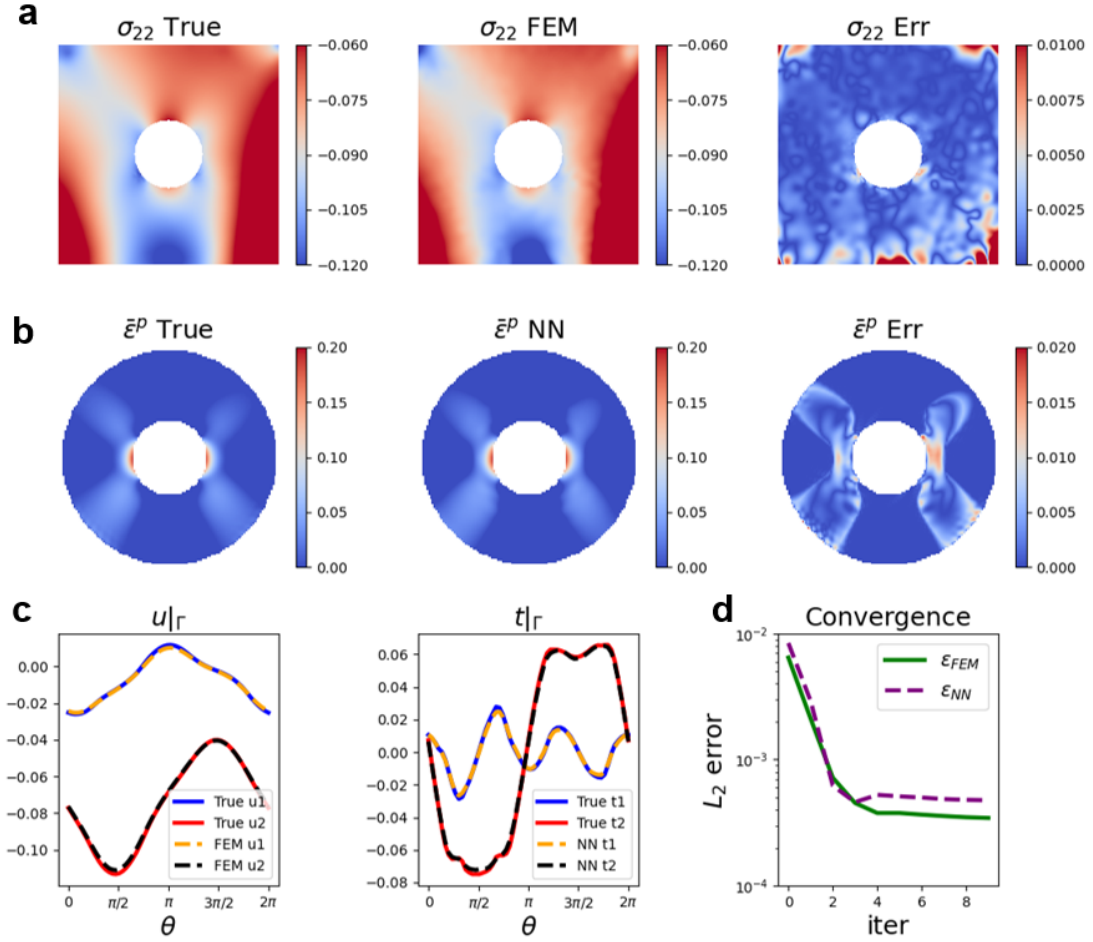


Figure 6.10: Results of coupling FEM and DeepONet in elastoplasticity. (a-b) Results of the normal stress in y direction (σ_{22}) and equivalent plastic strain ($\bar{\epsilon}^p$). From left to right: True value, FEM/DeepONet predicted value from coupling, and their absolute errors. (c) Displacement and traction at the interface Γ . True values and predicted values from coupling are presented. Solid lines: displacement and traction of true data in x (blue) and y (red) directions. Dashed lines: displacement and traction of the model predictions in x (yellow) and y (black) directions. (d) The history of the relative errors of the coupling model at the interface. ϵ_{FEM} and ϵ_{NN} refer to the L_2 error of displacement and traction from FEM and DeepONet, respectively. Notice that we present the error with respect to the ground truth to show the convergence and accuracy of our framework.

Figs. 6.10(a-b), we plot the ground truth solution from FEM in the first column, the predictions from our coupling framework in the second column, together with their differences in the third column. In plot (a), we show the results of the normal stress (σ_{22}) in the vertical direction of Ω_{FEM} . The results of the equivalent plastic strain, $\bar{\epsilon}^p$ in Ω_{NN} are provided in plot (b). Although the coupling framework has generally well reproduced the stress component σ_{22} in the bulk region of Ω_{FEM} , there exist relatively large errors at the top edge and the bottom corners. These errors either originate from numerical interpolation in the FEM solver or are caused by a reduced solution regularity. Apart from these regions, the errors are controlled at a low value with the maximum relative error lower than 10%. In Ω_{NN} , the profile of $\bar{\epsilon}^p$ is well captured by the coupling scheme with the maximum relative error of $\bar{\epsilon}^p$ less than 10%. The results demonstrate that plasticity in the region of interest is well predicted by the surrogate model. To provide a further quantitative verification of our coupling framework, we plot the ground-truth, the predictions of displacement, and traction components on Γ in Fig. 6.10(c) as functions of θ , the angle in polar coordinate. The solid lines denote the ground-truth results from the nonlinear FEM. The dashed lines denote the predicted values of the framework. The model predictions match the ground truth well, with relative errors smaller than 2%. Fig. 6.10(d) shows the efficiency of the coupling framework: the L_2 errors between model predictions and the ground truth reach a plateau at the fourth iteration.

6.4.4 Hyperelasticity

In the previous examples, DeepONet was trained based on data generated from FEM and was coupled with another FEM in the online stage. These examples

Parameter	μ	K	k_1	k_2	α
Value	0.3846	0.8333	0.1	1.5	$\pi/2$

Table 6.2: Parameters value of the HGO model. We set the value of k_1, k_2 , and α the same for $i = 1$ and 2.

illustrated the capability of our coupling framework in solving both static and dynamic problems. In this section, we demonstrate the capability of our framework in concurrently coupling a continuum model (FEM) with a surrogate from smooth particle dynamics (SPH) for describing a microscopic particle system. We consider using the coupled framework to predict the mechanics of a hyperelastic material. We first derive the energy minimization formulation of the continuum model. We denote by ψ the strain energy density of the hyperelastic model and seek to find a displacement field $\mathbf{u} : \Omega \rightarrow \mathbb{R}^2$ that minimizes the total potential energy Ψ :

$$\Psi = \int_{\Omega} \psi(\mathbf{u}) d\mathbf{x} - \int_{\Omega} \mathbf{b} \cdot \mathbf{u} d\mathbf{x} - \int_{\Gamma_N} \mathbf{T}_0 \cdot \mathbf{u} ds. \quad (6.29)$$

Here, $\mathbf{b}$ denotes the body force in Ω , $\mathbf{T}$ is the traction load applied on the Neumann boundary Γ_N . Hence, the total potential energy Ψ is the integration of strain energy density, ψ , over the entire domain Ω , deduced by the energy contributions from the body force $\mathbf{b}$, the traction $\mathbf{T}$.

We consider the Holzapfel-Gasser-Odgen (HGO) model [96] to describe the constitutive behavior of the material in this example. Essentially, the material is hyperelastic, anisotropic, fiber-reinforced in diverse directions. Its strain energy density is:

$$\psi = \frac{\mu}{2}(I_1 - 3) - \mu \ln(J) + \frac{k_1}{2k_2} \sum_{i=1}^2 (\exp\{(k_2 \langle E_i \rangle^2)\} - 1) + \frac{K}{2} \left(\frac{J^2 - 1}{2} - \ln J \right), \quad (6.30)$$

where $\langle \cdot \rangle$ denotes the Macaulay bracket. In this model, the fiber strain of the two fiber groups is expressed as:

$$E_i = \kappa(I_1 - 3) + (1 - 3\kappa)(I_{4i} - 1), \quad i = 1, 2, \quad (6.31)$$

where k_1 and k_2 are fiber modulus and the exponential coefficient, respectively, I_1 is the first principal invariant, and I_{4i} is the fourth principal invariants corresponding to the i -th fiber group. Mathematically, for the i -th fiber group with angle direction α_i from the reference direction, I_{4i} is calculated by $\mathbf{n}_i^T \mathbf{C} \mathbf{n}_i$, where $\mathbf{C}$ is the right Cauchy-Green tensor and $\mathbf{n}_i = [\cos\alpha_i, \sin\alpha_i]^T$. In our simulations, we consider a material with fiber reinforcement in the vertical direction (see Fig. 6.11 right for illustration). Therefore, for both fiber groups we set $\alpha_i = \pi/2$. In Eq. (6.31), fiber dispersion is denoted as κ , whose value ranges from 0 to $\frac{1}{3}$. Intuitively, $\kappa = 0$ means no fiber dispersion whereas $\kappa = \frac{1}{3}$ represents an isotropic fiber dispersion. In this example, we consider the fiber oriented vertically with no dispersion ($\kappa = 0$). All parameter values in this example are summarized in Table 6.2.

We now present the problem set up of this example. As depicted in Fig. 6.11(a), we consider a 2D unit square plate with a centered circular void (radius as 0.1). The plate deforms under a uniaxial tension, $\mathbf{T}_0(\mathbf{x})$, applied on its top edge. The bottom edge is clamped. We model the material response of the entire domain using an SPH model, whose solution is taken as the ground-truth solution. More details of the SPH model are provided in 6.A. To develop a coupling model, we consider a similar setting as in the previous example and decompose the entire domain Ω into two non-overlapping subdomains. Then, we train DeepONets using SPH data to obtain a surrogate for the internal domain while the external region is described by a continuum FEM model. Models in these two domains communicate by exchanging proper interface conditions on their common inter-

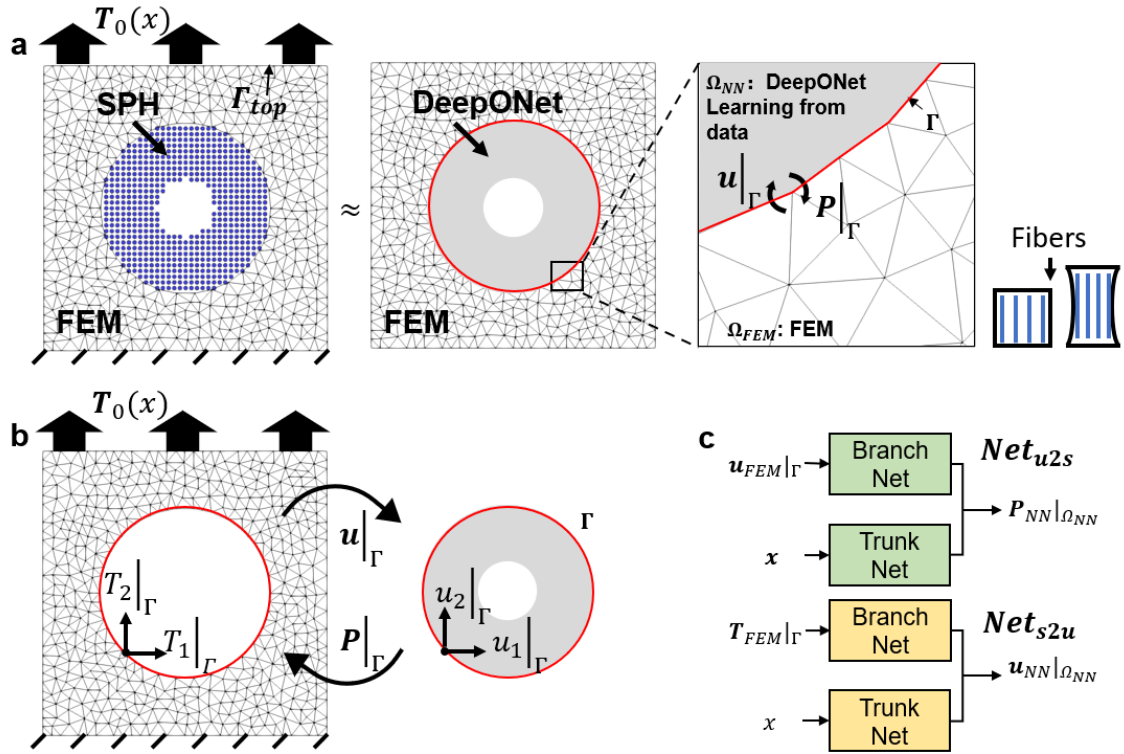


Figure 6.11: Setup of the hyperelasticity problem. (a) A unit square is decomposed into Ω_{NN} and Ω_{FEM} . We impose a non-uniform traction boundary condition on the top edge and fix the displacement at the bottom and train multiple DeepONets to represent the mechanics of an SPH model. The material is reinforced by fibers in the vertical direction. Information at the interface (displacement u and first Piola-Kirchhoff stress P) is transmitted between DeepONet and FEM. (b) Traction and displacement in different directions are exchanged at the interface Γ . FEM predicts the external domain while the DeepONet is trained based on SPH data. (c) Two types of DeepONet are proposed to predict the mechanics of the system: predicting stresses based on displacement information (green boxes) and vice versa (yellow boxes).

face, Γ . In Fig. 6.11(b) we present a schematic of information transmission of the coupling framework with a N-D method. At the n -th iteration, the FEM solver receives an updated distributed traction on Γ from DeepONets and solves the updated displacement field, $\mathbf{u}_{FEM}^n$, with the given information. Then, the FEM transmits the updated displacement information $\mathbf{u}_{FEM}^n|_{\Gamma}$ to DeepONets. With the displacement on Γ as input, DeepONets estimate the first Piola-Kirchhoff (PK1) stresses in Ω_{NN} . Based on the predicted PK1 stresses, we then calculate the surface traction on Γ and other associated quantities, such as the equivalent plastic strain $\bar{\epsilon}^P$ and von Mises stress $\bar{\sigma}$, accordingly. At the n -th iteration, the system solution at the interface Γ is updated as $\tilde{\mathbf{T}}(\mathbf{x}) = (1 - \theta)\mathbf{T}_I^n(\mathbf{x}) + \theta\mathbf{T}_{II}^n(\mathbf{x})$.

Next, we briefly describe the training process of DeepONet. To generate the training/testing dataset, we sample 1,000 different traction loading $\mathbf{T}_0(\mathbf{x})$ on the top edge from a random field (see Algorithm in Sec. 6.D). Then, for each sampled traction loading, we perform an SPH simulation to obtain the solutions in the entire domain and collect the corresponding solutions of displacement and PK1 stress fields of in Ω_{NN} . Among these 1000 samples, 900 cases are employed as the training data while the rest is kept as testing data. As depicted in Fig. 6.11(c), we consider two approaches for network training. In the first approach, the network (Net_{u2s}) takes the displacement at the interface ($\mathbf{u}|_{\Gamma}$) as input and predicts PK1 stress as the output. In the second approach, the interfacial traction $T|_{\Gamma}$ is employed as the input of the network (Net_{s2u}), yielding the displacement field as output. These approaches provide a flexibility of imposing different interface conditions. We adopt the first approach in the N-D method and combine the information of the two approaches ($\hat{\mathbf{u}}$ and $\hat{\mathbf{T}}$) in the R-D/R-N method.

Fig. 6.12 shows the coupling results of a typical testing case with a N-D method. In Fig. 6.12(a-b), we compare the vertical displacement u_2 in Ω_{FEM} (first row) and

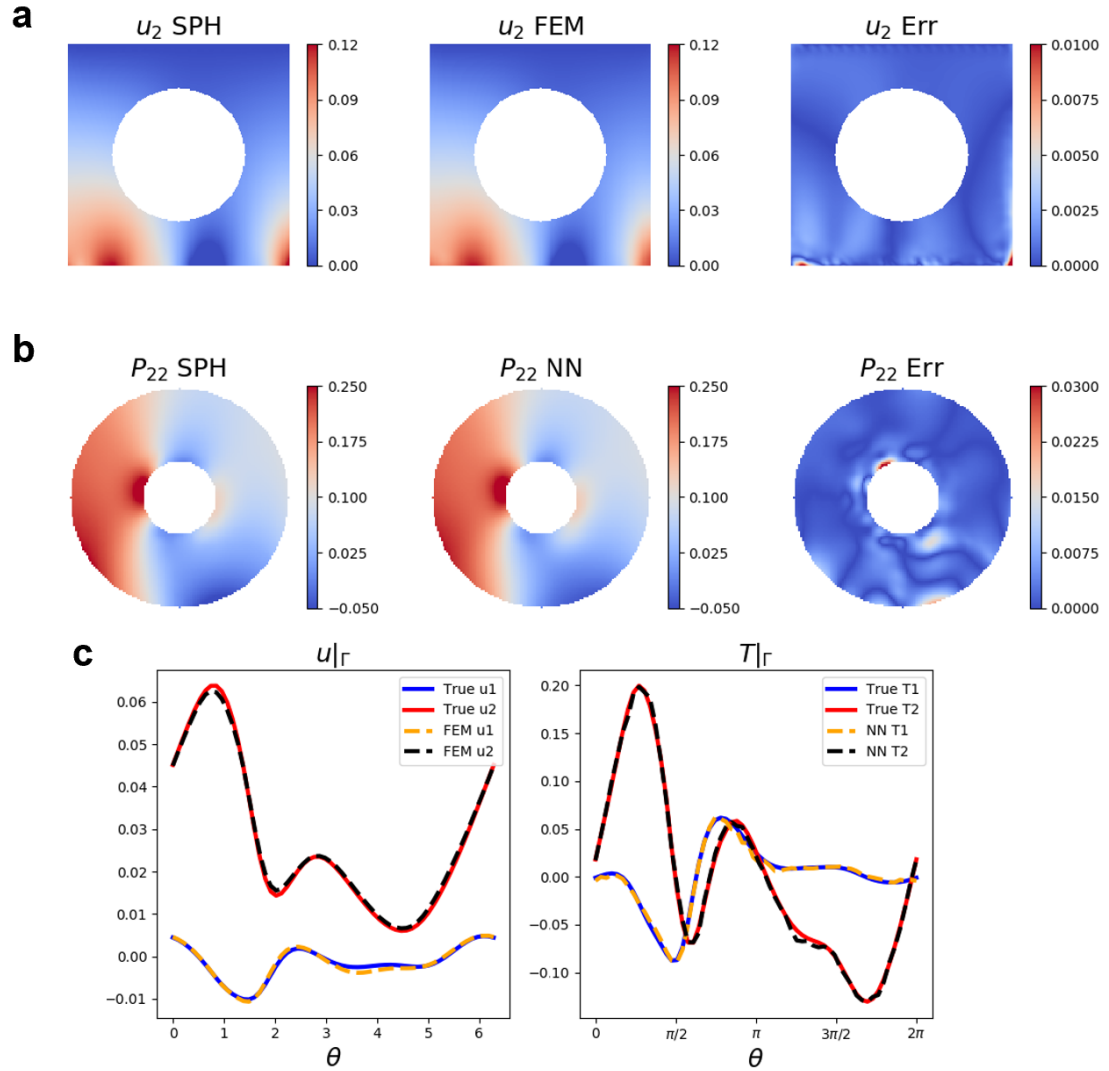


Figure 6.12: Results of coupling FEM and DeepONet in the hyperelasticity problem. Results of (a) displacement in Ω_{FEM} and (b) P_{22} in Ω_{NN} . From left to right: prediction from SPH, prediction from FEM, and their absolute differences. (c) Predicted displacement and traction at the interface. Predictions from both FEM and DeepONets are compared with the true data. Solid lines: displacement and traction of true data in x (blue) and y (red) directions. Dashed lines: displacement and traction of the model predictions in x (yellow) and y (black) directions.

Model	Wall Time
FEM-SPH	$\sim 4h$
FEM-DeepONet	$< 1s$

Table 6.3: Wall time comparison for SPH and DeepONet per iteration in the hyperelastic problem. The excessive cost of SPH is exacerbated because we use a time-dependent SPH solver to simulate a static problem.

the PK1 stress component P_{22} in Ω_{NN} (second row) between the SPH ground truth (first column) and the FEM/NN prediction (second column). The prediction errors are displayed in the third column. We observe that the coupling results match well with the ground truth solution with the largest prediction errors distributed near the bottom corners. The errors are partially induced by the numerical interpolations in the FEM and reduced regularity in that region. To further examine the prediction accuracy, in Fig. 6.12(c) we plot a quantitative comparison of displacement and traction on the interface Γ as functions of the polar coordinate angle θ . Despite some numerical discrepancies between SPH and FEM (see 6.B), the FEM/DeepONet predictions shown in dashed lines successfully reproduce the SPH simulation results depicted in solid lines. Therefore, our proposed method is capable of capturing the mechanics from SPH with a substantially improved efficiency: the time cost of performing an SPH simulation is approximately 4 hours whereas running its surrogate just takes a fraction of a second (Table 6.3). Admittedly, the excessive cost of SPH is exacerbated because we use a time-dependent SPH solver to simulate a static problem. Nonetheless, we can see that replacing a particle model with its surrogate in a multiscale coupling framework poses unique advantages in both efficiency and programming easiness.

To further illustrate the flexibility and investigate the convergence rate of our coupling framework, we employ a R-D/R-N coupling method to our framework with a variety of values of the Robin coefficient R . To clarify, R-D/R-N means that

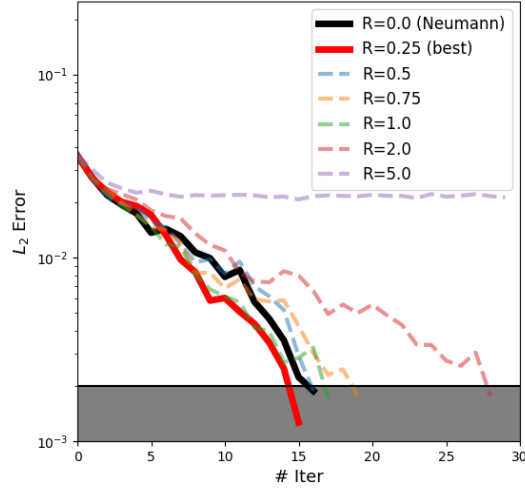


Figure 6.13: L_2 errors of the interfacial displacement in the hyperelastic multiscale model with a Robin boundary. The red line with Robin coefficient $R=0.25$ converges fastest among the testing cases. The black solid line represents the relative error history of a Neumann boundary condition ($R=0$). The shaded area indicates that the relative error is less than 2%. Notice that we present the error with respect to the ground truth to show the convergence and accuracy of our framework.

the FEM, imposed with a Robin-type boundary condition, separately transmits the information of displacement and traction to Net_{u2s} and Net_{s2u} in order to update the Robin information in the next step. The continuum model with a Robin boundary condition is modified as:

$$\Psi = \int_{\Omega} \psi(\mathbf{u}) d\mathbf{x} - \int_{\Omega} \mathbf{b} \cdot \mathbf{u} d\mathbf{x} - \int_{\Gamma_N} \mathbf{T}_0 \cdot \mathbf{u} ds - \int_{\Gamma_R} \left(\mathbf{r} - \frac{1}{2} R \mathbf{u} \right) \cdot \mathbf{u} ds. \quad (6.32)$$

$\mathbf{r}$ is the Robin boundary condition applied on the interface Γ_R . We define $\mathbf{r}$ as:

$$\mathbf{r} = \hat{\mathbf{T}} + R \cdot \hat{\mathbf{u}}, \text{ for } \mathbf{x} \in \Gamma_R, \quad (6.33)$$

where R is the (positive) Robin coefficient, $\hat{\mathbf{T}}$ and $\hat{\mathbf{u}}$ are the known traction and displacement.

The coupling procedure is changed as well. At the n -th iteration, we first solve

the FEM model with a Robin boundary condition $\tilde{\mathbf{r}}^n(\mathbf{x})$. Then, we transmit the interfacial traction and displacement of FEM to Net_{u2s} and Net_{s2u} , respectively. The corresponding displacement $\hat{\mathbf{u}}^{n+1}$ and tractions $\hat{\mathbf{T}}^{n+1}$ are estimated by DeepONets and used to update the Robin boundary condition: $\tilde{\mathbf{r}}^{n+1}(\mathbf{x}) = (1-\theta)\mathbf{r}_{FEM}^{n+1}(\mathbf{x}) + \theta\mathbf{r}_{NN}^{n+1}(\mathbf{x})$. Here, $\theta = 0.5$ is taken as a fixed relaxation parameter.

We further study the convergence of interfacial displacement errors with different values of R . In Fig. 6.13, we highlight the results from the N-D coupling method with the black line and the results of R-D/R-N with $R = 0.25$ (best result) in red. The shaded area indicates 2% equivalent relative error of the test case. Due to the nonlinearity of the problem, it generally takes more iterations for the coupling framework to reach the stopping criterion. When taking a R-N coupling method with a sub-optimal Robin coefficient (such as $R = 5.0$ as shown in the purple line of Fig. 6.13), the coupling framework fails to converge. This fact again demonstrates the importance of choosing an appropriate coupling method.

6.5 Discussion

In this chapter, we propose an efficient concurrent coupling framework for multiscale modeling of mechanics problems. In lieu of coupling an expensive microscopic model, we propose to employ a deep neural operator, DeepONet, as a surrogate to approximate the microscopic solution in the domain with fine-scale features. The response in the coarse-scale domain is simulated by a standard numerical model, such as the finite element method. The two models are coupled concurrently by exchanging information at the interface until convergence. To verify the performance of this framework, we study four benchmarks including

solving static and dynamic problems for different materials. We have also demonstrated that the framework is readily applicable for various interface conditions. The results show that the cases with a Robin boundary condition tend to converge faster than a Neumann/Dirichlet boundary condition. Moreover, the predictions of the coupled model agree well with the true solution acquired from a numerical method, indicating the generalization ability of DeepONet and the accuracy of the proposed framework.

In addition, using a neural operator as surrogate enables the model to be trained directly from data. Such a model is particularly promising for learning dynamics of complex materials without explicit constitutive models. Moreover, coupling DeepONet with FEM substantially improves the computational efficiency in multiscale modeling: DeepONet predicts the expensive microscopic behavior only at a fraction of second. Hence, the overall computational cost of the proposed framework could be shortened by orders of magnitude than the existing multiscale coupling methods. In addition, we can train the surrogate model to learn based on partial information: it can predict the information only at the interface and neglect dynamics inside the microscopic region. Thus, the overall framework behaves as an artificial-intelligence type boundary condition, which would potentially improve the efficiency especially in the scenarios where only the macroscopic dynamics is of interest.

As a further note, we would like to point out the keys to build a successful neural operator based coupling framework with guaranteed convergence and to avoid possible pitfalls. First, the generalization ability of DeepONet determines if the result converges and its convergence rate. Utilizing an accurate neural operator model takes fewer iterations to converge. However, a poorly trained model could lead to slow convergence or even divergence. Second, normalizing data is another

key to convergence. In our experiments, the training data are at disparate scales, ranging from 10^{-3} – 10^1 . Properly normalization of the training data would be essential to the convergence of numerical iterations. In addition, choosing proper coupling strategies, such as the right Robin coefficient, also plays a critical role in guaranteeing fast numerical convergence as it affects the condition number of the stiffness matrix in FEM and the coupling system [52]. A large condition number may lead to a slow convergence or even divergent result in the coupling framework (see Sec. 6.4.4 and [87, 55]).

Certainly, more improvements and directions can be considered in the future. Learning directly from data facilitates a unique feature that DeepONet can learn data that comes from different scales, which has been demonstrated in Sec. 6.4.4. In the future, it would be interesting to develop a data-driven model from noisy data, such as molecular dynamics and dissipative particle dynamics, as presented in Fig. 6.1. In addition, overcoming the multiscale characteristic length and time scale would be another challenge in multiscale modeling with machine learning, that is, the time and length of a microscopic model are usually orders of magnitude smaller than the continuum model, causing challenges in network training and long-term predictions. The coupling results may drift away from the true solution or even diverge due to inaccurate predictions from surrogate models. Another improvement would be considering microstructures and geometric variations (see [169]). Developing an operator-learning neural network that can predict dynamics with different geometric variations would be a great improvement for broadening the application of the proposed method. Also, simulating fracture progression [110] is another natural and promising application of the multiscale coupling framework. As fracture progresses, the field of interest that includes the damage region may also move along with the tip of a crack.

Such modification is particularly useful for problems with discontinuity and/or multiple phases, such as fracture, phase transition, shock wave capturing, and etc. Our framework can be further extended to employ other coupling methods such as quasicontinuum [248], multigrid [25], or serves as a surrogate model for heterogeneous multiscale method (HMM) [282], etc.

Acknowledgment

MY, EZ, and GEK acknowledge the support by grant U01 HL142518 from the National Institutes of Health. Y. Yu would like to acknowledge support by the National Science Foundation under award DMS 1753031.

6.A Smoothed Particle Hydrodynamics

In this section, we briefly introduce the basic formulation of the Total Lagrangian Smoothed Particle Hydrodynamics (TLSPH). We refer the reader to [221, 71] for more details. In the SPH framework, physical quantities are approximated with the neighboring information in a kernel. Consider a function $f(X)$ at X_i can be approximated by $\hat{g}(X_i)$ with the integration

$$\hat{g}(X_i) = \int f(X)W(X - X_i)dX, \quad (6.34)$$

where $W(X)$ is a weighting kernel which is chosen as a third-order polynomial

$$W(R_j, h) = A \begin{cases} (h - R_j)^3, & R_j < h, \\ 0, & R_j \leq h, \end{cases} \quad (6.35)$$

h denotes the radius of the integration kernel; R_j is defined as the distance between X_j and the reference point in the kernel X_i with $A = 10/\pi h^5$ in two dimensions or $A = 10/\pi h^6$ in three dimensions. SPH numerically approximates the integration as

$$g(X_i) = \sum_{j \in S} f_j V_j W(R_j, h), \quad (6.36)$$

where V_j is the volume of particle j . The gradient of $g(X_i)$ with respect to its reference coordinate is

$$\nabla_X g(X_i) = \sum_{j \in S} f_j V_j \nabla_X W(R_j, h) \quad (6.37)$$

with

$$\nabla_X W(R_j, h) = \left(\frac{\partial W(R_j, h)}{\partial R_j} \right) \frac{R_j}{R_j} \quad (6.38)$$

For a vector $f(X)$, its gradient to the referential coordinate is approximated as the following using the integration rule:

$$\nabla_X g(X_i) = \sum_{j \in S} f_j \otimes V_j \nabla_X W(R_j, h) \quad (6.39)$$

We then introduce two *ad-hoc* corrections for keeping symmetrization and first-order completeness [180].

$$\nabla_X g(X_i) = \sum_{j \in S} (f_j - f_i) V_j \nabla_X W(R_j, h). \quad (6.40)$$

Another correction guarantees the first-order completeness [217], the corrected gradient tensor is defined as

$$\tilde{\nabla}_X W(R_j, h) = A^{-1} \nabla_X W(R_j, h) \quad (6.41)$$

where the shape tensor A is

$$A_i = \sum_{j \in S} V_j \nabla_X W(R_j, h) \otimes R_j \quad (6.42)$$

We introduce the SPH integration to the governing equation of solid at the continuum level. The equilibrium equation is:

$$\nabla \cdot P + \rho_0 \mathbf{b} = \rho_0 \ddot{\mathbf{x}} \quad (6.43)$$

where P is the first Piola-Kirchhoff stress tensor, $\mathbf{b}$ the body force vector, ρ_0 the referential mass density, and $\ddot{\mathbf{x}}$ the acceleration. We define the deformation gradient as

$$F_i = \frac{\partial x_i}{\partial X_i} = \sum_{j \in S} r_j \otimes V_j \tilde{\nabla}_X W(R_j, h) \quad (6.44)$$

According to basic continuum mechanics law,

$$P_i = F_i S_i = 2F_i \frac{\partial \mathcal{W}}{\partial C} \quad (6.45)$$

As proved in [22], the internal forces emerge from the divergence of the first Piola-Kirchhoff stress P :

$$m_i \ddot{x}_i = f^{int} + f^{ext}, \quad (6.46)$$

where the internal forces is expressed as

$$f_i^{int} = \sum_{j \in S} V_i V_j (P_i \tilde{\nabla}_X W(R_i, h) - P_j \tilde{\nabla}_X W(R_i, h)). \quad (6.47)$$

We adopt another correction for suppressing spurious hourglassing mode [71]

$$f_i^{hg} = \sum_{j \in S} -\alpha \frac{EV_i V_j W(R_j, h)}{2R_j^2} (\delta_i + \delta_j) \frac{r_j}{r_j} \quad (6.48)$$

The deformation derivative in time is approximated as

$$\dot{F} = \frac{1}{\Delta t} (F^{t+1} - F^t) \quad (6.49)$$

We adopt the strain energy function $\mathcal{W}(\mathbf{C}, \mathbf{M})$ for a fiber-enhanced tissue proposed in [96]

$$\mathcal{W}(C, M) = \frac{c}{2} (I_1 - 3) - c \ln(J) + \frac{k_1}{2k_2} \sum_{i=1}^2 (\exp\{(k_2 \langle E_i \rangle^2)\} - 1) + \frac{K_0}{2} \left(\frac{J^2 - 1}{2} - \ln J \right), \quad (6.50)$$

where principal invariants, I_1 and I_4 are defined as,

$$I_1 = C : I, J = \det F, I_4 = C : M \otimes M \quad (6.51)$$

The parameters are set the same as the FEM model in Sec. 6.4.4. We refer the reader to [221, 4] for more details.

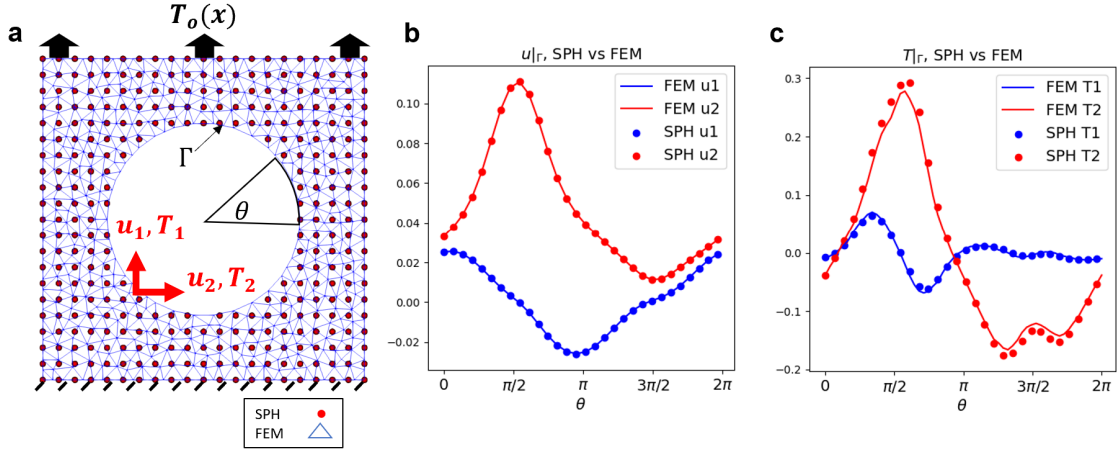


Figure 6.14: Comparison of SPH and FEM. (a) We impose the same Dirichlet boundary condition on the top and internal circle for a FEM and SPH model. (b-c) Interfacial displacement and traction of both models.

6.B Accuracy of FEM and SPH

To compare the numerical solution of the FEM and SPH model, We present their computational results for a same benchmark problem. The problem is set up as shown in Fig. 6.14(a). We impose a distributed traction boundary condition on the top edge and impose a displacement boundary condition (Dirichlet-type) at the interface for both models shown in Fig. 6.14(b): the displacement in x and y at the interface are plotted against the angle θ . The corresponding tractions at the interface are presented in Fig. 6.14(c). The solid lines are the FEM tractions in x and y directions with the dashed lines denoting the SPH results. The results correspond well in general with some deviation around $\theta = \pi/2$ and $\theta = 3\pi/2$ due to the difference in numerical integration methods. This discrepancy partially contributes to the overall errors observed in Sec. 6.4.4.

Problem	Branch Size	Trunk Size	Train Epochs
Poisson	[84, 100, 100, 100]	[2, 100, 100, 100]	100,000
Heat	[51, 100, 100, 100]	[2, 100, 100, 100]	100,000
Elastoplastic	[120, 150, 150, 150]	[2, 150, 150, 150]	1,000,000
Hyperelastic	[120, 150, 150, 150]	[2, 150, 150, 150]	1,000,000

Figure 6.15: Network size for various problems. We tabulate the branch/trunk network size for the four examples in Sec. 6.4 with the number of training epochs.

6.C Network Training Details

We explain the details of network training in this section. Network architecture for each problem is presented in Fig. 6.15. We adopt a fully-connected neural network (FNN) with four layers for all the sub-networks. The input layer dimension of the branch net depends on the number of points at the interface, which ranges from 51 to 120 in our examples. Three hidden layers are concatenated with the input layer where we adopt the hyperbolic tangent function as the activate function. Each hidden layer has 100 or 150 neurons shown in Fig. 6.15. To minimize the loss function for each case, we set the training epoch as 100,000 steps for the heat and Poisson example and 1,000,000 for the other two examples. Moreover, we illustrate the error history of a few cases in Fig. 6.16 where each cases shows a relatively small difference between the training and testing errors. The low testing errors indicating a good generalization capability of the network. More specifically, we tabulate the input/output of each network in Fig. 6.17 with MSEs of the training and testing data for each case.

We define the following metrics proposed in [50] to quantitatively characterize the efficiency of our framework.

$$R_t = \frac{C_t}{N_s C_s}, R_e = \frac{C_e}{C_s}, N_e^* = N_s + \frac{C_t}{C_s}. \quad (6.52)$$

Here, R_t is the training ratio where the time cost for training the DeepONet (C_t) is divided by the number of simulations (N_s) and the cost in time (C_s) for each simulation in the training dataset. R_e represents the ratio of time on one forward prediction in DeepONet over performing a standard numerical solver. The break even time N_e^* indicates the number of evaluations in the numerical solver where training a DeepONet costs as much as using numerical solvers. Beyond this point, surrogate modeling will be more efficient than a classical PDE solver. Taking the elastoplasticity case as an example, each elastoplastic FEM takes approximately 30s ($C_s = 30s$) on a single CPU where we generated 1,000 ($N_s = 1000$) samples as the training data. The DeepONet training takes 62 min ($C_t = 3720s$) on a single GPU. The time cost of each forward evaluation using DeepONet (C_e) is 0.0014s. Thus, using this information, we obtained $R_t = 0.124$, $R_e = 4.6 \times 10^{-5}$, and $N_e^* = 1124$. That means, as far as the resultant surrogate model is used for more than 1124 times in prediction tasks, DeepONet will be a more efficient choice.

6.D Data Generation

6.D.1 Random field generation

We provide an overview of the algorithms of random fields generation using Fast Fourier Transformation (FFT). Let $W(\mathbf{x})$ be a Gaussian white noise random field on $\mathbb{R}^d$. A random field $\phi(\mathbf{x})$ can be sampled by

$$\phi(\mathbf{x}) = \mathcal{F}^{-1}(\gamma^{1/2}\mathcal{F}(W))(\mathbf{x}) \quad (6.53)$$

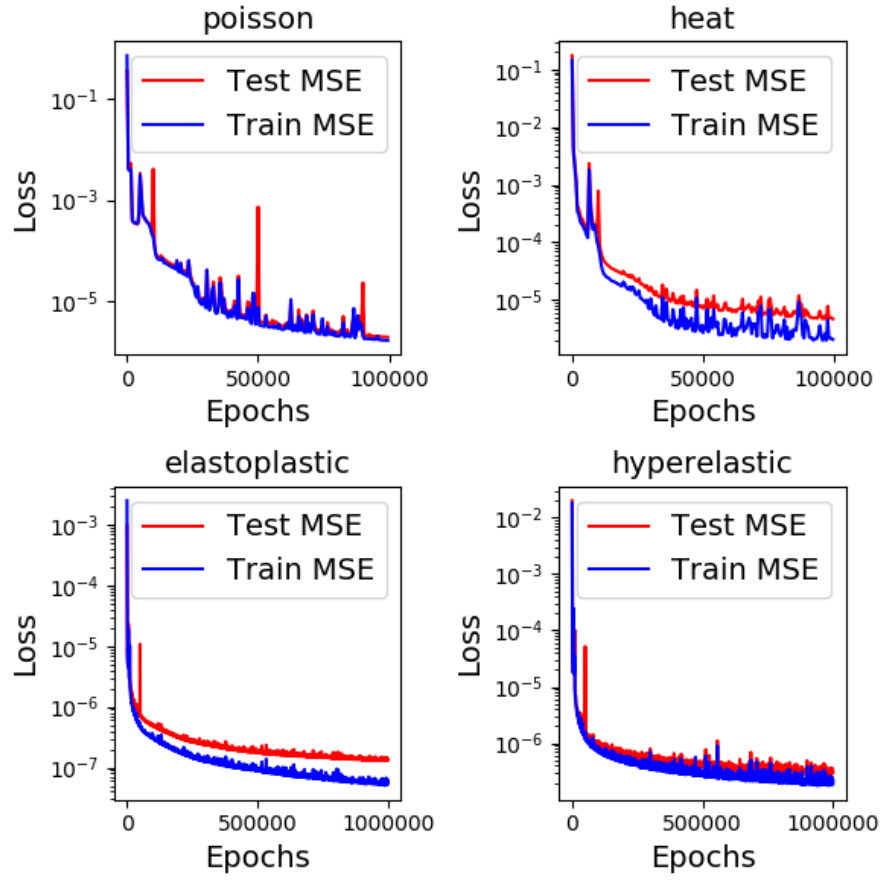


Figure 6.16: History of Training/testing errors of DeepONet for different problems.

Problem	Branch Input	Trunk Input	Output	MSE_{train}	MSE_{test}
Poisson	$u _{\Gamma}$	$[x, y]$	u	1.675×10^{-6}	1.946×10^{-6}
Heat	u^n	$[x, \frac{\partial u}{\partial x} _{\Gamma}]$	$\frac{\partial^2 u^{n+1}}{\partial x^2}$	1.877×10^{-6}	4.471×10^{-6}
Elastoplastic	$u _{\Gamma}$	$[x, y]$	ε_{11}	1.028×10^{-8}	4.523×10^{-8}
			ε_{12}	5.578×10^{-9}	1.183×10^{-8}
			ε_{22}	4.854×10^{-8}	1.262×10^{-7}
Hyperelastic	$u _{\Gamma}$	$[x, y]$	P_{11}	1.355×10^{-8}	2.347×10^{-8}
			P_{12}	6.393×10^{-9}	1.278×10^{-8}
			P_{21}	7.540×10^{-9}	1.798×10^{-8}
			P_{22}	1.627×10^{-7}	2.701×10^{-7}
	$T _{\Gamma}$	$[x, y]$	u_1	3.396×10^{-10}	2.430×10^{-9}
			u_2	1.716×10^{-9}	3.841×10^{-9}

Figure 6.17: Details of network setup (inputs/outputs and MSE errors of training/testing) for each example in Sec. 6.4.

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote Fourier transformation and its inverse. γ represents a correlation function $\|k^{-\alpha}\|$ where $\|k\|$ is the L_2 norm of the wave number $k \in \mathbb{R}^d$. In Sec. 6.4.1, 6.4.2, and 6.4.4, we adopt this method for generating the training data with $\alpha = 5$. We refer the reader to find more theoretical details in [134].

6.D.2 Sampling in elastoplasticity

In the case of elastoplasticity, the non-uniform tension $t_0(x)$ (see Fig. 6.9) is generated by

$$t_0(x) = \sum_{i=1}^3 \frac{A_i}{i} \cos(ix) + \sum_{i=1}^3 \frac{B_i}{i} \sin(ix), \quad (6.54)$$

where $A_i, B_i \sim N(0, 0.05^2)$ ($i \in \{1, 2, 3\}$) are independent normal random variables. We choose the standard deviation as 0.05 to make sure that the dataset contains similar numbers of cases with and without plastic deformation.

CHAPTER SEVEN

Conclusion

7.1 Summary

Aiming to address the need for quantitative understanding of GAG-initiated aortic dissection and its relevant biomechanical systems, I develop a series of computational and machine learning models for both forward predictions of aortic mechanics and dissection and inverse characterizations of vascular biomechanical properties. First, I employ an operator-regression network for predicting injection-caused dissection progression for diverse histological microstructures. We implicitly demonstrate that our model can effectively extract the underlying functional relationship between distinctive microstructures and their mechanical properties. Yet, 3D realistic microstructures are simplified into 2D images in the model where linkings between elastic lamellae are neglected. Hence, I developed a computational model for investigating dissection progression based on ex-vivo mouse aortas to incorporate the missing information. Our results recover the measured mechanical data with detailed descriptions of internal stress conditions. The proposed computational model provides further evidence that microstructures play an important role in dissection progression that eventually leads to vessel rupture. In the next two chapters, I explore the application of PINNs/DeepONets in inferring thrombus and vascular properties based on measurable data. The material properties, including permeability, viscoelasticity, or constitutive relationships, are accurately recovered from data in a deterministic and probabilistic approach. Lastly, I develop a novel framework by combining machine learning with finite elements which can be used for facilitating fast multiscale modeling in biomechanical systems.

7.2 Open Questions and future work

As discussed by Karnik [121], the delicate yet complex interactions between the internal microstructure and a myriad of proteins, proteoglycans, and growth factors of the vascular matrix lead to difficulties in delineating, analyzing, and measuring the properties and functions of various constituents. Quantitative investigations of these constituents and their interplay can greatly facilitate the understanding of vascular physiology/pathophysiology and diseases, including the mechanism of the initiation and propagation of aortic dissection. The thesis serves as an endeavor to fill these gaps by synergistically integrating computational and machine learning models with experimental measurements for aortic dissection. Although some insights have been gleaned by the computational research in this thesis, a series of open questions still exists as the motivations of my research. Possible mitigations and future directions are briefly discussed after each question.

- **How do GAGs affect dissection progression?** Although we have developed a computational model based on two-photon microscope in Chap 3, where information of the main components is incorporated in the reconstructed geometry. Yet, we need perform more simulations to investigate local stresses with presence of GAGs. In addition, because two-photon imaging cannot reflect distributions of GAGs, simulating the aortic biomechanics with GAGs need to artificially incorporate different GAG distributions, which may not faithful predict the correct progression path if delamination is initiated. Hence, we will address this by sampling different GAGs distribution and summarize the statistics. Possible significant factors, such as mass fraction, the concentration of ions, distribution of GAGs (localized or distributed) as well as their interactions with different mouse models and impacts on the

directions of damage progression will be studied completely and thoroughly.

- **How to combine knowledge from experimental with synthetic data into a network for inferring a class of constitutive modeling?** We employ a network for accurate learning functional priors of constitutive relationships and efficient inference based on sparse measurements (Chap 5). However, the prior knowledge for the PI-GANs was provided by the four-fiber model, which is a widely-accepted yet simplified model since important information, such as fiber interactions, is missing. The model assumes that fibers in each direction are independent and do not interact with each other whereas developing a more accurate constitutive modeling requires tremendous insights and is time-consuming. Leveraging abundant data generated from four-fiber models with accurate yet expensive ex-vivo data, or multifidelity modeling, is a possible way of developing more accurate constitutive modeling and efficient constitutive inference based on sparse measurements [172, 61, 186, 173, 161, 99].
- **How to incorporate data from microstructures, genetics, and molecular level information to constitutive modeling?** Macroscopic stretch-stress relationships are a resultant effect that is affected by collective factors at micro-scales, such as genetics, underlying microstructure, molecular expression, and others. There exist hundreds, if not thousands, of proteins, proteoglycans, genotypes, and other factors that control the biomechanical phenotypes of a vascular system. Modeling such a complex process with explicit expression is expensive, even unrealistic. Hence, more recent works address this question by incorporating information from ex-vivo imaging [97, 150] and genetics [316] using data-driven models.

Bibliography

- [1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.
- [2] Abaqus. *Abaqus 2020 documentation*. Dassault Systèmes, 2020.
- [3] H. Ahmadzadeh, M. K. Rausch, and J. D. Humphrey. Particle-based computational modelling of arterial disease. *Journal of the Royal Society Interface*, 15(149):20180616, 2018.
- [4] H. Ahmadzadeh, M. K. Rausch, and J. D. Humphrey. Modeling lamellar disruption within the aortic wall using a particle-based approach. *Scientific Reports*, 9(1):1–17, 2019.
- [5] M. Alber, A. Buganza Tepole, W. R. Cannon, S. De, S. Dura-Bernal, K. Garikipati, G. E. Karniadakis, W. W. Lytton, P. Perdikaris, L. Petzold, et al. Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences. *NPJ Digital Medicine*, 2(1):1–11, 2019.

- [6] M. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells. The fenics project version 1.5. *Archive of Numerical Software*, 3(100), 2015.
- [7] L. Ambrosio and V. M. Tortorelli. Approximation of functional depending on jumps by elliptic functional via t-convergence. *Communications on Pure and Applied Mathematics*, 43(8):999–1036, 1990.
- [8] H. Arbabi, J. E. Bunder, G. Samaey, A. J. Roberts, and I. G. Kevrekidis. Linking machine learning with multiscale numerics: data-driven discovery of homogenized equations. *JOM*, 72(12):4444–4457, 2020.
- [9] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.
- [10] S. Baek, R. L. Gleason, K. Rajagopal, and J. Humphrey. Theory of small on large: potential utility in computations of fluid–solid interactions in arteries. *Computer methods in applied mechanics and engineering*, 196(31-32):3070–3078, 2007.
- [11] E. Ban, C. Cavinato, and J. D. Humphrey. Differential propensity of dissection along the aorta. *Biomechanics and Modeling in Mechanobiology*, pages 1–13, 2021.
- [12] E. Ban, C. Cavinato, and J. D. Humphrey. Critical pressure of intramural delamination in aortic dissection. *Annals of biomedical engineering*, pages 1–12, 2022.
- [13] K.-J. Bathe and S. Bolourchi. Large displacement analysis of three-dimensional beam structures. *International journal for numerical methods in engineering*, 14(7):961–986, 1979.

- [14] Y. Bazilevs, V. Calo, J. Cottrell, T. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 197(1-4):173–201, 2007.
- [15] C. Bellini, J. Ferruzzi, S. Roccabianca, E. Di Martino, and J. Humphrey. A microstructurally motivated model of arterial wall mechanics with mechanobiological implications. *Annals of biomedical engineering*, 42(3):488–502, 2014.
- [16] A. Bensoussan, J.-L. Lions, and G. Papanicolaou. *Asymptotic analysis for periodic structures*, volume 374. American Mathematical Soc., 2011.
- [17] M. R. Bersi, V. A. A. Santamaría, K. Marback, P. Di Achille, E. H. Phillips, C. J. Goergen, J. D. Humphrey, and S. Avril. Multimodality imaging-based characterization of regional material properties in a murine model of aortic dissection. *Scientific Reports*, 10(1):1–23, 2020.
- [18] H. Bhatia, T. S. Carpenter, H. I. Ingólfsson, G. Dharuman, P. Karande, S. Liu, T. Oppelstrup, C. Neale, F. C. Lightstone, B. Van Essen, et al. Machine-learning-based dynamic-importance sampling for adaptive multiscale simulations. *Nature Machine Intelligence*, 3(5):401–409, 2021.
- [19] J. Biehler, M. W. Gee, and W. A. Wall. Towards efficient uncertainty quantification in complex and large-scale biomechanical problems based on a bayesian multi-fidelity scheme. *Biomechanics and modeling in mechanobiology*, 14(3):489–513, 2015.
- [20] J. P. Bishwal. *Parameter estimation in stochastic differential equations*. Springer, 2007.
- [21] A. Blumens, M. Yin, H. Nakajima, Y. Hasegawa, Z. Li, and G. E. Karniadakis. Multiscale parareal algorithm for long-time mesoscopic simulations of mi-

- crovascular blood flow in zebrafish. *Computational Mechanics*, 68:1131–1152, 2021.
- [22] J. Bonet and T.-S. Lok. Variational and momentum preservation aspects of smooth particle hydrodynamic formulations. *Computer Methods in Applied Mechanics and Engineering*, 180(1-2):97–115, 1999.
- [23] Z. I. Botev, J. F. Grotowski, and D. P. Kroese. Kernel density estimation via diffusion. *The Annals of Statistics*, 38(5):2916–2957, 2010.
- [24] B. Bourdin, G. A. Francfort, and J.-J. Marigo. The variational approach to fracture. *Journal of Elasticity*, 91(1):5–148, 2008.
- [25] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390, 1977.
- [26] P. T. Brewick and K. Teferra. Uncertainty quantification for constitutive model calibration of brain tissue. *Journal of the mechanical behavior of biomedical materials*, 85:237–255, 2018.
- [27] S. L. Brunton, B. R. Noack, and P. Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52:477–508, 2020.
- [28] 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*, 113(15):3932–3937, 2016.
- [29] S. Cai, Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis. Physics-informed neural networks (PINNs) for fluid mechanics: A review. *arXiv preprint arXiv:2105.09506*, 2021.

- [30] S. Cai, Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis. Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mechanica Sinica*, pages 1–12, 2022.
- [31] S. Cai, Z. Wang, L. Lu, T. A. Zaki, and G. E. Karniadakis. DeepM&MNet: Inferring the electroconvection multiphysics fields based on operator approximation by neural networks. *arXiv preprint arXiv:2009.12935*, 2020.
- [32] S. Cai, Z. Wang, L. Lu, T. A. Zaki, and G. E. Karniadakis. DeepM&Mnet: Inferring the electroconvection multiphysics fields based on operator approximation by neural networks. *Journal of Computational Physics*, 436:110296, 2021.
- [33] B. Calderhead, M. Girolami, and N. D. Lawrence. Accelerating bayesian inference over nonlinear differential equations with gaussian processes. In *Advances in neural information processing systems*, pages 217–224, 2009.
- [34] L. Cardamone, A. Valentin, J. Eberth, and J. Humphrey. Origin of axial pre-stretch and residual stress in arteries. *Biomechanics and modeling in mechanobiology*, 8(6):431–446, 2009.
- [35] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová. Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4):045002, 2019.
- [36] M. W. Carson and M. R. Roach. The strength of the aortic media and its role in the propagation of aortic dissection. *Journal of Biomechanics*, 23(6):579–588, 1990.
- [37] C. Cavinato, P. Badel, W. Krasny, S. Avril, and C. Morin. Experimental characterization of adventitial collagen fiber kinematics using second-harmonic

- generation imaging microscopy: similarities and differences across arteries, species and testing conditions. In *Multi-scale Extracellular Matrix Mechanics and Mechanobiology*, pages 123–164. Springer, 2020.
- [38] C. Cavinato, S.-I. Murtada, A. Rojas, and J. D. Humphrey. Evolving structure-function relations during aortic maturation and aging revealed by multiphoton microscopy. *Mechanisms of Ageing and Development*, 196:111471, 2021.
- [39] S. Chan and A. H. Elsheikh. A machine learning approach for efficient uncertainty quantification using multiscale methods. *Journal of Computational Physics*, 354:493–511, 2018.
- [40] A. Chattopadhyay, P. Hassanzadeh, and D. Subramanian. Data-driven predictions of a multiscale lorenz 96 chaotic system using machine-learning methods: reservoir computing, artificial neural network, and long short-term memory network. *Nonlinear Processes in Geophysics*, 27(3):373–389, 2020.
- [41] T. Chen and H. Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks*, 6(4):911–917, 1995.
- [42] Y. Chen, L. Lu, G. E. Karniadakis, and L. Dal Negro. Physics-informed neural networks for inverse problems in nano-optics and metamaterials. *Optics Express*, 28(8):11618–11633, 2020.
- [43] O. A. Chkrebtii, D. A. Campbell, B. Calderhead, M. A. Girolami, et al. Bayesian solution uncertainty quantification for differential equations. *Bayesian Analysis*, 11(4):1239–1267, 2016.
- [44] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1251–1258, 2017.

- [45] F. S. Cikach, C. D. Koch, T. J. Mead, J. Galatioto, B. B. Willard, K. B. Emerton, M. J. Eagleton, E. H. Blackstone, F. Ramirez, E. E. Roselli, et al. Massive aggrecan and versican accumulation in thoracic aortic aneurysm and dissection. *JCI insight*, 3(5), 2018.
- [46] J. L. Crassidis and J. L. Junkins. *Optimal estimation of dynamic systems*. CRC press, 2011.
- [47] M. Cury, F. Zeidan, and A. C. Lobato. Aortic disease in the young: genetic aneurysm syndromes, connective tissue disorders, and familial aortic aneurysms and dissections. *International journal of vascular medicine*, 2013, 2013.
- [48] M. D’Elia, D. Littlewood, J. Trageser, M. Perego, and P. Bochev. An optimization-based strategy for peridynamic-fem coupling and for the prescription of nonlocal boundary conditions. *arXiv preprint arXiv:2110.04420*, 2021.
- [49] J. Dhamala, H. J. Arevalo, J. Sapp, B. M. Horáček, K. C. Wu, N. A. Trayanova, and L. Wang. Quantifying the uncertainty in model parameters using gaussian process-based markov chain monte carlo in cardiac electrophysiology. *Medical image analysis*, 48:43–57, 2018.
- [50] P. C. Di Leoni, L. Lu, C. Meneveau, G. Karniadakis, and T. A. Zaki. Deeponet prediction of linear instability waves in high-speed boundary layers. *arXiv preprint arXiv:2105.08697*, 2021.
- [51] M. Di Nisio, N. van Es, and H. R. Büller. Deep vein thrombosis and pulmonary embolism. *The Lancet*, 388(10063):3060–3073, 2016.
- [52] W. Dijkstra and R. Mattheij. The condition number of the bem-matrix arising from laplace’s equation. *Electronic Journal of Boundary Elements*, 4(2), 2006.

- [53] K. P. Dingemans, P. Teeling, J. H. Lagendijk, and A. E. Becker. Extracellular matrix of the human aortic media: an ultrastructural histochemical and immunohistochemical study of the adult aortic media. *The Anatomical Record: An Official Publication of the American Association of Anatomists*, 258(1):1–14, 2000.
- [54] M. Dobson, M. Luskin, and C. Ortner. Stability, instability, and error of the force-based quasicontinuum approximation. *Archive for Rational Mechanics and Analysis*, 197(1):179–202, 2010.
- [55] J. Douglas and C.-S. Huang. An accelerated domain decomposition procedure based on robin transmission conditions. *BIT Numerical Mathematics*, 37(3):678–686, 1997.
- [56] E. N. Dvorkin and K.-J. Bathe. A continuum mechanics based four-node shell element for general non-linear analysis. *Engineering computations*, 1984.
- [57] V. G. Eck, W. P. Donders, J. Sturdy, J. Feinberg, T. Delhaas, L. R. Hellevik, and W. Huberts. A guide to uncertainty quantification and sensitivity analysis for cardiovascular applications. *International journal for numerical methods in biomedical engineering*, 32(8):e02755, 2016.
- [58] Y. Efendiev, J. Galvis, and T. Y. Hou. Generalized multiscale finite element methods (GMsFEM). *Journal of Computational Physics*, 251:116–135, 2013.
- [59] S. Endo, N. Kuwayama, Y. Hirashima, T. Akai, M. Nishijima, and A. Takaku. Results of urgent thrombolysis in patients with major stroke and atherothrombotic occlusion of the cervical internal carotid artery. *American Journal of Neuroradiology*, 19(6):1169–1175, 1998.
- [60] P. Espanol and P. Warren. Statistical mechanics of dissipative particle dynamics. *EPL (Europhysics Letters)*, 30(4):191, 1995.

- [61] M. G. Fernández-Godino, C. Park, N.-H. Kim, and R. T. Haftka. Review of multi-fidelity models. *arXiv preprint arXiv:1609.07196*, 2016.
- [62] J. Ferruzzi, M. Bersi, and J. Humphrey. Biomechanical phenotyping of central arteries in health and disease: advantages of and methods for murine models. *Annals of biomedical engineering*, 41(7):1311–1330, 2013.
- [63] J. Ferruzzi, P. Di Achille, G. Tellides, and J. D. Humphrey. Combining in vivo and in vitro biomechanical data reveals key roles of perivascular tethering in central artery function. *PloS one*, 13(9):e0201379, 2018.
- [64] J. Fish, G. J. Wagner, and S. Keten. Mesoscopic and multiscale modelling in materials. *Nature Materials*, 20(6):774–786, 2021.
- [65] B. FitzGibbon, B. Fereidoonhezad, J. Concannon, N. Hynes, S. Sultan, K. M. Moerman, and P. McGarry. A numerical investigation of the initiation of aortic dissection. *engrXiv*, 2020.
- [66] M. Flaschel, S. Kumar, and L. De Lorenzis. Unsupervised discovery of interpretable hyperelastic constitutive laws. *Computer Methods in Applied Mechanics and Engineering*, 381:113852, 2021.
- [67] D. Funaro, A. Quarteroni, and P. Zanolli. An iterative procedure with interface relaxation for domain decomposition methods. *SIAM Journal on Numerical Analysis*, 25(6):1213–1236, 1988.
- [68] Y. Fung. Elasticity of soft tissues in simple elongation. *American Journal of Physiology-Legacy Content*, 213(6):1532–1544, 1967.
- [69] Y. Fung, K. Fronek, and P. Patitucci. Pseudoelasticity of arteries and the choice of its mathematical expression. *American Journal of Physiology-Heart and Circulatory Physiology*, 237(5):H620–H631, 1979.

- [70] Y. C. Fung. *Biomechanics: mechanical properties of living tissues*. Springer Science & Business Media, 2013.
- [71] G. C. Ganzenmüller. An hourglass control algorithm for lagrangian smooth particle hydrodynamics. *Computer Methods in Applied Mechanics and Engineering*, 286:87–106, 2015.
- [72] G. C. Ganzenmüller, M. O. Steinhauser, P. Van Liedekerke, and K. U. Leuven. The implementation of smooth particle hydrodynamics in lammmps. *Paul Van Liedekerke Katholieke Universiteit Leuven*, 1:1–26, 2011.
- [73] F. Gao, Z. Guo, M. Sakamoto, and T. Matsuzawa. Fluid-structure interaction within a layered aortic arch model. *Journal of Biological Physics*, 32(5):435–454, 2006.
- [74] T. C. Gasser, R. W. Ogden, and G. A. Holzapfel. Hyperelastic modelling of arterial layers with distributed collagen fibre orientations. *Journal of The Royal Society Interface*, 3(6):15–35, 2006.
- [75] H. P. Gavin. The levenberg-marquardt algorithm for nonlinear least squares curve-fitting problems. *Department of Civil and Environmental Engineering, Duke University*, 19, 2019.
- [76] J. P. V. Geest, M. S. Sacks, and D. A. Vorp. Age dependency of the biaxial biomechanical behavior of human abdominal aorta. *Journal of Biomechanical Engineering*, 126(6):815–822, 2004.
- [77] K. Genovese, Y. Lee, A. Lee, and J. Humphrey. An improved panoramic digital image correlation method for vascular strain analysis and material characterization. *Journal of the mechanical behavior of biomedical materials*, 27:132–142, 2013.

- [78] A. Gizzi, M. L. De Bellis, M. Vasta, and A. Pandolfi. Diffusion-based de-generation of the collagen reinforcement in the pathologic human cornea. *Journal of Engineering Mathematics*, 127(1):1–10, 2021.
- [79] J. Golledge and K. A. Eagle. Acute aortic dissection. *The Lancet*, 372(9632):55–66, 2008.
- [80] K. Gong, H. Liu, and B.-l. Wang. Water entry of a wedge based on sph model with an improved boundary treatment. *Journal of Hydrodynamics*, 21(6):750–757, 2009.
- [81] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- [82] S. Goswami, C. Anitescu, S. Chakraborty, and T. Rabczuk. Transfer learning enhanced physics informed neural network for phase-field modeling of fracture. *Theoretical and Applied Fracture Mechanics*, 106:102447, 2020.
- [83] S. Goswami, M. Yin, Y. Yu, and G. E. Karniadakis. A physics-informed variational deeponet for predicting the crack path in brittle materials. *arXiv preprint arXiv:2108.06905*, 2021.
- [84] S. Goswami, M. Yin, Y. Yu, and G. E. Karniadakis. A physics-informed variational deeponet for predicting crack path in quasi-brittle materials. *Computer Methods in Applied Mechanics and Engineering*, 391:114587, 2022.
- [85] R. D. Groot and P. B. Warren. Dissipative particle dynamics: Bridging the gap between atomistic and mesoscopic simulation. *The Journal of Chemical Physics*, 107(11):4423–4435, 1997.

- [86] O. Gültekin, S. P. Hager, H. Dal, and G. A. Holzapfel. Computational modeling of progressive damage and rupture in fibrous biological tissues: application to aortic dissection. *Biomechanics and Modeling in Mechanobiology*, 18(6):1607–1628, 2019.
- [87] K. Gustafson. Domain decomposition, operator trigonometry, robin condition. *Contemporary Mathematics*, 218:432–437, 1998.
- [88] P. G. Hagan, C. A. Nienaber, E. M. Isselbacher, D. Bruckman, D. J. Karavite, P. L. Russman, A. Evangelista, R. Fattori, T. Suzuki, J. K. Oh, et al. The international registry of acute aortic dissection (irad): new insights into an old disease. *Jama*, 283(7):897–903, 2000.
- [89] S. J. Hamilton and A. Hauptmann. Deep d-bar: Real-time electrical impedance tomography imaging with deep neural networks. *IEEE transactions on medical imaging*, 37(10):2367–2377, 2018.
- [90] D. Haskett, E. Speicher, M. Fouts, D. Larson, M. Azhar, U. Utzinger, and J. V. Geest. The effects of angiotensin ii on the coupled microstructural and biomechanical response of c57bl/6 mouse aorta. *Journal of biomechanics*, 45(5):772–779, 2012.
- [91] C. M. He and M. R. Roach. The composition and mechanical properties of abdominal aortic aneurysms. *Journal of Vascular Surgery*, 20(1):6–13, 1994.
- [92] F. Helmchen and W. Denk. Deep tissue two-photon microscopy. *Nature methods*, 2(12):932–940, 2005.
- [93] P. Hendrickx, P. Rieder, M. Prokop, H. Milbradt, M. Karck, and J. Laas. Intravenous dsa and dynamic computed tomography for postoperative follow-up of type a aortic dissections. *European Journal of Radiology*, 9(3):158–162, 1989.

- [94] M. D. Hoffman, A. Gelman, et al. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014.
- [95] B. L. Holian and R. Ravelo. Fracture simulations using large-scale molecular dynamics. *Physical Review B*, 51(17):11275, 1995.
- [96] G. A. Holzapfel, T. C. Gasser, and R. W. Ogden. A new constitutive framework for arterial wall mechanics and a comparative study of material models. *Journal of elasticity and the physical science of solids*, 61(1):1–48, 2000.
- [97] G. A. Holzapfel, K. Linka, S. Sherifova, and C. J. Cyron. Predictive constitutive modelling of arteries by deep learning. *Journal of the Royal Society Interface*, 18(182):20210411, 2021.
- [98] G. A. Holzapfel, G. Sommer, C. T. Gasser, and P. Regitnig. Determination of layer-specific mechanical properties of human coronary arteries with nonatherosclerotic intimal thickening and related constitutive modeling. *American Journal of Physiology-Heart and Circulatory Physiology*, 289(5):H2048–H2058, 2005.
- [99] A. A. Howard, M. Perego, G. E. Karniadakis, and P. Stinis. Multifidelity deep operator networks. *arXiv preprint arXiv:2204.09157*, 2022.
- [100] T. J. Hughes. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2012.
- [101] J. D. Humphrey. Mechanics of the arterial wall: review and directions. *Critical Reviews™ in Biomedical Engineering*, 23(1-2), 1995.

- [102] J. D. Humphrey. Possible mechanical roles of glycosaminoglycans in thoracic aortic dissection and associations with dysregulated transforming growth factor- β . *Journal of Vascular Research*, 50(1):1–10, 2013.
- [103] J. D. Humphrey and G. A. Holzapfel. Mechanics, mechanobiology, and modeling of human abdominal aorta and aneurysms. *Journal of biomechanics*, 45(5):805–814, 2012.
- [104] J. D. Humphrey, M. A. Schwartz, G. Tellides, and D. M. Milewicz. Role of mechanotransduction in vascular biology: focus on thoracic aortic aneurysms and dissections. *Circulation research*, 116(8):1448–1461, 2015.
- [105] H. I. Ingólfsson, C. Neale, T. S. Carpenter, R. Shrestha, C. A. López, T. H. Tran, T. Oppelstrup, H. Bhatia, L. G. Stanton, X. Zhang, et al. Machine learning–driven multiscale modeling reveals lipid-dependent dynamics of ras signaling proteins. *Proceedings of the National Academy of Sciences*, 119(1), 2022.
- [106] F. Isensee, P. F. Jaeger, S. A. Kohl, J. Petersen, and K. H. Maier-Hein. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18(2):203–211, 2021.
- [107] M. Jadidi, S. Sherifova, G. Sommer, A. Kamenskiy, and G. A. Holzapfel. Constitutive modeling using structural information on collagen fiber direction and dispersion in human superficial femoral artery specimens of different ages. *Acta biomaterialia*, 121:461–474, 2021.
- [108] A. D. Jagtap, E. Kharazmi, and G. E. Karniadakis. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 365:113028, 2020.

- [109] Y. Jiao, G. Li, A. Korneva, A. W. Caulk, L. Qin, M. R. Bersi, Q. Li, W. Li, R. P. Mecham, J. D. Humphrey, et al. Deficient circumferential growth is the primary determinant of aortic obstruction attributable to partial elastin deficiency. *Arteriosclerosis, thrombosis, and vascular biology*, 37(5):930–941, 2017.
- [110] H. Jin, T. Jiao, R. J. Clifton, and K.-S. Kim. Dynamic fracture of a bicontinuously nanostructured copolymer: A deep-learning analysis of big-data-generating experiment. *Journal of the Mechanics and Physics of Solids*, page 104898, 2022.
- [111] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, 2017.
- [112] X. Jin, S. Cai, H. Li, and G. E. Karniadakis. NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible navier-stokes equations. *Journal of Computational Physics*, 426:109951, 2021.
- [113] N. Jing, Q. Xue, C. Ling, M. Shan, T. Zhang, X. Zhou, and Z. Jiao. Effect of defects on young’s modulus of graphene sheets: a molecular dynamics simulation. *RSC Advances*, 2(24):9124–9129, 2012.
- [114] H. D. Kabir, A. Khosravi, M. A. Hosen, and S. Nahavandi. Neural network-based uncertainty quantification: A survey of methodologies and applications. *IEEE access*, 6:36218–36234, 2018.
- [115] C. Karmonik, M. Müller-Eschner, S. Partovi, P. Geisbüsch, M.-K. Ganten, J. Bismuth, M. G. Davies, D. Böckler, M. Loebe, A. B. Lumsden, and H. von Tengg-Kobligk. Computational fluid dynamics investigation of chronic aor-

- tic dissection hemodynamics versus normal aorta. *Vascular and Endovascular Surgery*, 47(8):625–631, 2013.
- [116] C. Karmonik, S. Partovi, M. Müller-Eschner, J. Bismuth, M. G. Davies, D. J. Shah, M. Loebe, D. Böckler, A. B. Lumsden, and H. von Tengg-Kobligk. Longitudinal computational fluid dynamics study of aneurysmal dilatation in a chronic debakey type iii aortic dissection. *Journal of Vascular Surgery*, 56(1):260–263, 2012.
- [117] G. Karniadakis and S. Sherwin. *Spectral/hp element methods for computational fluid dynamics*. Oxford University Press, 2013.
- [118] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, W. Sifan, and L. Yang. Physics-informed machine learning. *Nature Reviews Physics*, 2021.
- [119] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.
- [120] G. E. Karniadakis and S. Sherwin. *Spectral/hp element methods for computational fluid dynamics*. Oxford University Press, 2005.
- [121] S. K. Karnik, B. S. Brooke, A. Bayes-Genis, L. Sorensen, J. D. Wythe, R. S. Schwartz, M. T. Keating, and D. Y. Li. A critical role for elastin signaling in vascular morphogenesis and disease. *Development*, 2003.
- [122] I. G. Kevrekidis, C. W. Gear, J. M. Hyman, P. G. Kevrekidis, O. Runborg, C. Theodoropoulos, et al. Equation-free, coarse-grained multiscale computation: enabling microscopic simulators to perform system-level analysis. *Commun. Math. Sci*, 1(4):715–762, 2003.

- [123] I. G. Kevrekidis and G. Samaey. Equation-free multiscale computation: Algorithms and applications. *Annual Review of Physical Chemistry*, 60:321–344, 2009.
- [124] E. Kharazmi, Z. Zhang, and G. E. Karniadakis. Variational physics-informed neural networks for solving partial differential equations. *arXiv preprint arXiv:1912.00873*, 2019.
- [125] E. Kharazmi, Z. Zhang, and G. E. Karniadakis. hp-vpinns: Variational physics-informed neural networks with domain decomposition. *arXiv preprint arXiv:2003.05385*, 2020.
- [126] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [127] M. G. Kinsella, S. L. Bressler, and T. N. Wight. The regulated synthesis of versican, decorin, and biglycan: extracellular matrix proteoglycans that influence cellular phenotype. *Critical Reviews™ in Eukaryotic Gene Expression*, 14(3), 2004.
- [128] G. Kissas, Y. Yang, E. Hwuang, W. R. Witschey, J. A. Detre, and P. Perdikaris. Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive 4d flow mri data using physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 358:112623, 2020.
- [129] A. Korneva, L. Zilberberg, D. Rifkin, J. Humphrey, and C. Bellini. Absence of ltbp-3 attenuates the aneurysmal phenotype but not spinal effects on the aorta in marfan syndrome. *Biomechanics and modeling in mechanobiology*, 18(1):261–273, 2019.

- [130] N. T. Kouchoukos and D. Dougenis. Surgery of the thoracic aorta. *New England Journal of Medicine*, 336(26):1876–1889, 1997.
- [131] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar. Neural operator: Learning maps between function spaces. *arXiv preprint arXiv:2108.08481*, 2021.
- [132] P. A. Kyrle and S. Eichinger. Deep vein thrombosis. *The Lancet*, 365(9465):1163–1174, 2005.
- [133] I. E. Lagaris, A. Likas, and D. I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9(5):987–1000, 1998.
- [134] A. Lang and J. Potthoff. Fast simulation of gaussian random fields. *Monte Carlo Methods and Applications*, 17(3):195–214, 2011.
- [135] S. Lanthaler, S. Mishra, and G. E. Karniadakis. Error estimates for deep-onets: A deep learning framework in infinite dimensions. *arXiv preprint arXiv:2102.09618*, 2021.
- [136] K. Lee and K. T. Carlberg. Model reduction of dynamical systems on non-linear manifolds using deep convolutional autoencoders. *Journal of Computational Physics*, 404:108973, 2020.
- [137] A. W. Lensing, P. Prandoni, M. H. Prins, and H. Büller. Deep-vein thrombosis. *The Lancet*, 353(9151):479–485, 1999.
- [138] B. Li and N. Bouklas. A variational phase-field model for brittle fracture in polydisperse elastomer networks. *International Journal of Solids and Structures*, 182:193–204, 2020.

- [139] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.
- [140] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Multipole graph neural operator for parametric partial differential equations. *arXiv preprint arXiv:2006.09535*, 2020.
- [141] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020.
- [142] H. Liang and H. Wu. Parameter estimation for differential equation models using a framework of measurement error in regression models. *Journal of the American Statistical Association*, 103(484):1570–1583, 2008.
- [143] L. Liang, M. Liu, C. Martin, J. A. Elefteriades, and W. Sun. A machine learning approach to investigate the relationship between shape features and numerically predicted risk of ascending aortic aneurysm. *Biomechanics and Modeling in Mechanobiology*, 16(5):1519–1533, 2017.
- [144] C. Lieberman and K. Willcox. Goal-oriented inference: Approach, linear theory, and application to advection diffusion. *SIAM Review*, 55(3):493–519, 2013.
- [145] C. Lin, Z. Li, L. Lu, S. Cai, M. Maxey, and G. E. Karniadakis. Operator learning for predicting multiscale bubble growth dynamics. *arXiv preprint arXiv:2012.12816*, 2020.
- [146] C. Lin, Z. Li, L. Lu, S. Cai, M. Maxey, and G. E. Karniadakis. Operator learning for predicting multiscale bubble growth dynamics. *The Journal of Chemical Physics*, 154(10):104118, 2021.

- [147] C. Lin, M. Maxey, Z. Li, and G. E. Karniadakis. A seamless multiscale operator neural network for inferring bubble dynamics. *Journal of Fluid Mechanics*, 929, 2021.
- [148] F.-H. Lin, C. Liu, and P. Zhang. On hydrodynamics of viscoelastic fluids. *Communications on Pure and Applied Mathematics*, 58(11):1437–1471, 2005.
- [149] P. Lin. Theoretical and numerical analysis for the quasi-continuum approximation of a material particle model. *Mathematics of Computation*, 72(242):657–675, 2003.
- [150] K. Linka, C. Cavinato, J. D. Humphrey, and C. J. Cyron. Predicting and understanding arterial elasticity from key microstructural features by bidirectional deep learning. *Acta Biomaterialia*, 2022.
- [151] K. Linka, M. Hillgärtner, K. P. Abdolazizi, R. C. Aydin, M. Itskov, and C. J. Cyron. Constitutive artificial neural networks: a fast and general approach to predictive data-driven constitutive modeling by deep learning. *Journal of Computational Physics*, 429:110010, 2021.
- [152] P.-L. Lions et al. On the schwarz alternating method. i. In *First international symposium on domain decomposition methods for partial differential equations*, volume 1, page 42. Paris, France, 1988.
- [153] B. Liu, N. Kovachki, Z. Li, K. Azizzadenesheli, A. Anandkumar, A. M. Stuart, and K. Bhattacharya. A learning-based multiscale method and its application to inelastic impact problems. *Journal of the Mechanics and Physics of Solids*, 158:104668, 2022.
- [154] M. Liu, L. Liang, and W. Sun. Estimation of in vivo constitutive parameters of the aortic wall using a machine learning approach. *Computer Methods in Applied Mechanics and Engineering*, 347:201–217, 2019.

- [155] M. Liu, L. Liang, and W. Sun. A generic physics-informed neural network-based constitutive model for soft biological tissues. *Computer methods in applied mechanics and engineering*, 372:113402, 2020.
- [156] M. Liu, L. Liang, Q. Zou, Y. Ismail, X. Lou, G. Iannucci, E. P. Chen, B. G. Leshnower, J. A. Elefteriades, and W. Sun. A probabilistic and anisotropic failure metric for ascending thoracic aortic aneurysm risk assessment. *Journal of the Mechanics and Physics of Solids*, page 104539, 2021.
- [157] M. Liu and G. Liu. Smoothed particle hydrodynamics (sph): an overview and recent developments. *Archives of computational methods in engineering*, 17(1):25–76, 2010.
- [158] L. Lu, M. Dao, P. Kumar, U. Ramamurty, G. E. Karniadakis, and S. Suresh. Extraction of mechanical properties of materials through deep learning from instrumented indentation. *Proceedings of the National Academy of Sciences*, 117(13):7052–7062, 2020.
- [159] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis. Learning nonlinear operators via deepnet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021.
- [160] L. Lu, X. Meng, S. Cai, Z. Mao, S. Goswami, Z. Zhang, and G. E. Karniadakis. A comprehensive and fair comparison of two neural operators (with practical extensions) based on fair data. *arXiv preprint arXiv:2111.05512*, 2021.
- [161] L. Lu, R. Pestourie, S. G. Johnson, and G. Romano. Multifidelity deep neural operators for efficient learning of partial differential equations with application to fast inverse design of nanoscale heat transport. *arXiv preprint arXiv:2204.06684*, 2022.

- [162] A. Lucas, M. Iliadis, R. Molina, and A. K. Katsaggelos. Using deep neural networks for inverse problems in imaging: beyond analytical methods. *IEEE Signal Processing Magazine*, 35(1):20–36, 2018.
- [163] N. F. MacLean, N. L. Dudek, and M. R. Roach. The role of radial elastic properties in the development of aortic dissections. *Journal of Vascular Surgery*, 29(4):703–710, 1999.
- [164] E. F. Mammen. Pathogenesis of venous thrombosis. *Chest*, 102(6):640S–644S, 1992.
- [165] N. M. Mangan, S. L. Brunton, J. L. Proctor, and J. N. Kutz. Inferring biological networks by sparse identification of nonlinear dynamics. *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, 2(1):52–63, 2016.
- [166] Z. Mao, A. D. Jagtap, and G. E. Karniadakis. Physics-informed neural networks for high-speed flows. *Computer Methods in Applied Mechanics and Engineering*, 360:112789, 2020.
- [167] Z. Mao, L. Lu, O. Marxen, T. A. Zaki, and G. E. Karniadakis. Deepm&mnet for hypersonics: Predicting the coupled flow and finite-rate chemistry behind a normal shock using neural-network approximation of operators. *Journal of Computational Physics*, 447:110698, 2021.
- [168] C. Martin, W. Sun, and J. Elefteriades. Patient-specific finite element analysis of ascending aorta aneurysms. *American Journal of Physiology-Heart and Circulatory Physiology*, 308(10):H1306–H1316, 2015.
- [169] F. Masi and I. Stefanou. Thermodynamics-based artificial neural networks (tann) for multiscale modeling of materials with inelastic microstructure. *arXiv preprint arXiv:2108.13137*, 2021.

- [170] I. Masson, P. Boutouyrie, S. Laurent, J. D. Humphrey, and M. Zidi. Characterization of arterial wall mechanical behavior and stresses from human clinical data. *Journal of biomechanics*, 41(12):2618–2627, 2008.
- [171] K. Maute, M. Nikbay, and C. Farhat. Sensitivity analysis and design optimization of three-dimensional non-linear aeroelastic systems by the adjoint method. *International Journal for Numerical Methods in Engineering*, 56(6):911–933, 2003.
- [172] X. Meng, H. Babaei, and G. E. Karniadakis. Multi-fidelity bayesian neural networks: Algorithms and applications. *Journal of Computational Physics*, 438:110361, 2021.
- [173] X. Meng and G. E. Karniadakis. A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse pde problems. *Journal of Computational Physics*, 401:109020, 2020.
- [174] X. Meng, L. Yang, Z. Mao, J. del Águila Ferrandis, and G. E. Karniadakis. Learning functional priors and posteriors from data and physics. *Journal of Computational Physics*, 457:111073, 2022.
- [175] W. Menke. *Geophysical data analysis: Discrete inverse theory*. Academic press, 2018.
- [176] C. Michoski, M. Milosavljevic, T. Oliver, and D. Hatch. Solving irregular and data-enriched differential equations using deep neural networks. *arXiv preprint arXiv:1905.04351*, 2019.
- [177] G. W. Milton. *The Theory of Composites*. Cambridge University Press, 2002.

- [178] N. Moës, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *International journal for numerical methods in engineering*, 46(1):131–150, 1999.
- [179] D. Mok, W. Wall, and E. Ramm. Accelerated iterative substructuring schemes for instationary fluid-structure interaction. *Computational Fluid and Solid Mechanics*, 2:1325–1328, 2001.
- [180] J. J. Monaghan. An introduction to SPH. *Computer Physics Communications*, 48(1):89–96, 1988.
- [181] J. J. Monaghan. Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30(1):543–574, 1992.
- [182] A. Mota, I. Tezaur, and C. Alleman. The schwarz alternating method in solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 319:19–51, 2017.
- [183] B. E. Muhs, K. Vincken, J. Van Prehn, M. Stone, L. Bartels, M. Prokop, F. Moll, and H. Verhagen. Dynamic cine-ct angiography for the evaluation of the thoracic aorta; insight in dynamic changes with implications for thoracic endograft treatment. *European Journal of Vascular and Endovascular Surgery*, 32(5):532–536, 2006.
- [184] S.-I. Murtada, J. Ferruzzi, H. Yanagisawa, and J. Humphrey. Reduced biaxial contractility in the descending thoracic aorta of fibulin-5 deficient mice. *Journal of biomechanical engineering*, 138(5), 2016.
- [185] D. P. Nathan, C. Xu, J. H. Gorman III, R. M. Fairman, J. E. Bavaria, R. C. Gorman, K. B. Chandran, and B. M. Jackson. Pathogenesis of acute aortic dissection: a finite element stress analysis. *The Annals of Thoracic Surgery*, 91(2):458–463, 2011.

- [186] L. W. Ng and K. E. Willcox. Multifidelity approaches for optimization under uncertainty. *International Journal for numerical methods in Engineering*, 100(10):746–772, 2014.
- [187] V. T. Nguyen, D. Georges, and G. Besançon. State and parameter estimation in 1-d hyperbolic pdes based on an adjoint method. *Automatica*, 67:185–191, 2016.
- [188] X. Nie, S. Chen, M. Robbins, et al. A continuum and molecular dynamics hybrid method for micro-and nano-fluid flow. *Journal of Fluid Mechanics*, 500:55–64, 2004.
- [189] C. A. Nienaber, R. E. Clough, N. Sakalihasan, T. Suzuki, R. Gibbs, F. Mussa, M. P. Jenkins, M. M. Thompson, A. Evangelista, J. S. Yeh, et al. Aortic dissection. *Nature Reviews Disease Primers*, 2(1):1–18, 2016.
- [190] G. Ninos, V. Bartzis, N. Merlemis, and I. E. Sarris. Uncertainty quantification implementations in human hemodynamic flows. *Computer Methods and Programs in Biomedicine*, 203:106021, 2021.
- [191] M. K. O’Connell, S. Murthy, S. Phan, C. Xu, J. Buchanan, R. Spilker, R. L. Dalman, C. K. Zarins, W. Denk, and C. A. Taylor. The three-dimensional micro-and nanostructure of the aortic medial lamellar unit measured using 3d confocal and electron microscopy imaging. *Matrix Biology*, 27(3):171–181, 2008.
- [192] C. Olsson, S. Thelin, E. Stahle, A. Ekbom, and F. Granath. Thoracic aortic aneurysm and dissection. *Circulation*, 114(24):2611–2618, 2006.
- [193] M. Ortiz. A method of homogenization of elastic media. *International Journal of Engineering Science*, 25(7):923–934, 1987.

- [194] P. T. O’Gara and R. W. DeSanctis. Acute aortic dissection and its variants: toward a common diagnostic and therapeutic approach. *Circulation*, 92(6):1376–1378, 1995.
- [195] G. Pang, L. Lu, and G. E. Karniadakis. fpinns: Fractional physics-informed neural networks. *SIAM Journal on Scientific Computing*, 41(4):A2603–A2626, 2019.
- [196] J. Park and X. Zhu. Physics-informed neural networks for learning the homogenized coefficients of multiscale elliptic equations. *arXiv preprint arXiv:2202.09712v1*, 2022.
- [197] S. Pasta, J. A. Phillippi, T. G. Gleason, and D. A. Vorp. Effect of aneurysm on the mechanical dissection properties of the human ascending thoracic aorta. *The Journal of thoracic and cardiovascular surgery*, 143(2):460–467, 2012.
- [198] A. T. Patera. A spectral element method for fluid dynamics: laminar flow in a channel expansion. *Journal of Computational Physics*, 54(3):468–488, 1984.
- [199] G. C. Peng, M. Alber, A. B. Tepole, W. R. Cannon, S. De, S. Dura-Bernal, K. Garikipati, G. E. Karniadakis, W. W. Lytton, P. Perdikaris, et al. Multi-scale modeling meets machine learning: What can we learn? *Archives of Computational Methods in Engineering*, 28(3):1017–1037, 2021.
- [200] L. Peng, Y. Qiu, Z. Yang, D. Yuan, C. Dai, D. Li, Y. Jiang, and T. Zheng. Patient-specific computational hemodynamic analysis for interrupted aortic arch in an adult: implications for aortic dissection initiation. *Scientific Reports*, 9(1):1–8, 2019.
- [201] L. Pereira, S. Y. Lee, B. Gayraud, K. Andrikopoulos, S. D. Shapiro, T. Bunton, N. J. Biery, H. C. Dietz, L. Y. Sakai, and F. Ramirez. Pathogenetic sequence

- for aneurysm revealed in mice underexpressing fibrillin-1. *Proceedings of the National Academy of Sciences*, 96(7):3819–3823, 1999.
- [202] D. Pfau, J. S. Spencer, A. G. Matthews, and W. M. C. Foulkes. Ab initio solution of the many-electron schrödinger equation with deep neural networks. *Physical Review Research*, 2(3):033429, 2020.
- [203] M. Piasecki and N. D. Katopodes. Identification of stream dispersion coefficients by adjoint sensitivity method. *Journal of Hydraulic Engineering*, 125(7):714–724, 1999.
- [204] F. Pled, C. Desceliers, and T. Zhang. A robust solution of a statistical inverse problem in multiscale computational mechanics using an artificial neural network. *Computer Methods in Applied Mechanics and Engineering*, 373:113540, 2021.
- [205] S. Plimpton. Fast parallel algorithms for short-range molecular dynamics. *Journal of computational physics*, 117(1):1–19, 1995.
- [206] A. Polanczyk, A. Piechota-Polanczyk, C. Domenig, J. Nanobachvili, I. Huk, and C. Neumayer. Computational fluid dynamic accuracy in mimicking changes in blood hemodynamics in patients with acute type iiib aortic dissection treated with tevar. *Applied Sciences*, 8(8):1309, 2018.
- [207] R. Prêtre and L. K. Von Segesser. Aortic dissection. *The Lancet*, 349(9063):1461–1464, 1997.
- [208] A. F. Psaros, X. Meng, Z. Zou, L. Guo, and G. E. Karniadakis. Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons. *arXiv preprint arXiv:2201.07766*, 2022.

- [209] S. Pyrialakos, I. Kalogeris, G. Sotiropoulos, and V. Papadopoulos. A neural network-aided bayesian identification framework for multiscale modeling of nanocomposites. *Computer Methods in Applied Mechanics and Engineering*, 384:113937, 2021.
- [210] E. Qian, B. Kramer, B. Peherstorfer, and K. Willcox. Lift & learn: Physics-informed machine learning for large-scale nonlinear dynamical systems. *Physica D: Nonlinear Phenomena*, 406:132401, 2020.
- [211] A. Rachev and K. Hayashi. Theoretical study of the effects of vascular smooth muscle contraction on strain and stress distributions in arteries. *Annals of biomedical engineering*, 27(4):459–468, 1999.
- [212] A. S. Rahman, T. Hosono, J. M. Quilty, J. Das, and A. Basak. Multiscale groundwater level forecasting: coupling new machine learning approaches with wavelet transforms. *Advances in Water Resources*, 141:103595, 2020.
- [213] 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*, 378:686–707, 2019.
- [214] M. Raissi, A. Yazdani, and G. E. Karniadakis. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481):1026–1030, 2020.
- [215] K. Rajagopal, C. Bridges, and K. Rajagopal. Towards an understanding of the mechanics underlying aortic dissection. *Biomechanics and Modeling in Mechanobiology*, 6(5):345–359, 2007.

- [216] J. O. Ramsay, G. Hooker, D. Campbell, and J. Cao. Parameter estimation for differential equations: a generalized smoothing approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(5):741–796, 2007.
- [217] P. Randles and L. D. Libersky. Smoothed particle hydrodynamics: some recent improvements and applications. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):375–408, 1996.
- [218] D. C. Rapaport and D. C. R. Rapaport. *The art of molecular dynamics simulation*. Cambridge university press, 2004.
- [219] M. K. Rausch and J. D. Humphrey. A microstructurally inspired damage model for early venous thrombus. *Journal of the mechanical behavior of biomedical materials*, 55:12–20, 2016.
- [220] M. K. Rausch and J. D. Humphrey. A computational model of the biochemomechanics of an evolving occlusive thrombus. *Journal of Elasticity*, 129(1-2):125–144, 2017.
- [221] M. K. Rausch, G. E. Karniadakis, and J. D. Humphrey. Modeling soft tissue damage and failure using a combined particle/continuum approach. *Biomechanics and Modeling in Mechanobiology*, 16(1):249–261, 2017.
- [222] F. Regazzoni, L. Dedè, and A. Quarteroni. Machine learning of multiscale active force generation models for the efficient simulation of cardiac electromechanics. *Computer Methods in Applied Mechanics and Engineering*, 370:113268, 2020.
- [223] B. V. Rego, D. Weiss, M. R. Bersi, and J. D. Humphrey. Uncertainty quantification in subject-specific estimation of local vessel mechanical properties. *International Journal for Numerical Methods in Biomedical Engineering*, page e3535, 2021.

- [224] B. Reyes, A. A. Howard, P. Perdikaris, and A. M. Tartakovsky. Learning unknown physics of non-newtonian fluids. *Physical Review Fluids*, 6(7):073301, 2021.
- [225] M. R. Roach and S. Song. Variations in strength of the porcine aorta as a function of location. *Clinical and Investigative Medicine*, 17(4):308, 1994.
- [226] S. Roccabianca, G. A. Ateshian, and J. D. Humphrey. Biomechanical roles of medial pooling of glycosaminoglycans in thoracic aortic dissection. *Biomechanics and Modeling in Mechanobiology*, 13(1):13–25, 2014.
- [227] S. Roccabianca, C. Bellini, and J. D. Humphrey. Computational modelling suggests good, bad and ugly roles of glycosaminoglycans in arterial wall mechanics and mechanobiology. *Journal of The Royal Society Interface*, 11(97):20140397, 2014.
- [228] S. Roccabianca, C. Figueroa, G. Tellides, and J. D. Humphrey. Quantification of regional differences in aortic stiffness in the aging human. *Journal of the Mechanical Behavior of Biomedical Materials*, 29:618–634, 2014.
- [229] I. Rocha, P. Kerfriden, and F. van der Meer. On-the-fly construction of surrogate constitutive models for concurrent multiscale mechanical analysis through probabilistic machine learning. *Journal of Computational Physics: X*, 9:100083, 2021.
- [230] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017.
- [231] Y. Rudy and H. Oster. The electrocardiographic inverse problem. *Critical reviews in biomedical engineering*, 20(1-2):25–45, 1992.

- [232] F. Sahli Costabal, Y. Yang, P. Perdikaris, D. E. Hurtado, and E. Kuhl. Physics-informed neural networks for cardiac activation mapping. *Frontiers in Physics*, 8:42, 2020.
- [233] M. Sambridge. Geophysical inversion with a neighbourhood algorithm-i. searching a parameter space. *Geophysical journal international*, 138(2):479–494, 1999.
- [234] S. Sankaran, H. J. Kim, G. Choi, and C. A. Taylor. Uncertainty quantification in coronary blood flow simulations: impact of geometry, boundary conditions and blood viscosity. *Journal of biomechanics*, 49(12):2540–2547, 2016.
- [235] S. Sankaran and A. L. Marsden. A stochastic collocation method for uncertainty quantification and propagation in cardiovascular simulations. *Journal of biomechanical engineering*, 133(3), 2011.
- [236] A. Sarrami-Foroushani, T. Lassila, A. Gooya, A. J. Geers, and A. F. Frangi. Uncertainty quantification of wall shear stress in intracranial aneurysms using a data-driven statistical model of systemic blood flow variability. *Journal of biomechanics*, 49(16):3815–3823, 2016.
- [237] S. Sevitt and N. Gallagher. Venous thrombosis and pulmonary embolism. a clinico-pathological study in injured and burned patients. *British Journal of Surgery*, 48(211):475–489, 1961.
- [238] D. E. Shea, S. L. Brunton, and N. Kutz. Sindy-bvp: Sparse identification of nonlinear dynamics for boundary value problems. *arXiv preprint arXiv:2005.10756*, 2020.

- [239] H. Shin, J. Choi, and M. Cho. An efficient multiscale homogenization modeling approach to describe hyperelastic behavior of polymer nanocomposites. *Composites Science and Technology*, 175:128–134, 2019.
- [240] S. A. Silling and E. Askari. A meshfree method based on the peridynamic model of solid mechanics. *Computers & structures*, 83(17-18):1526–1535, 2005.
- [241] B. R. Simon, A. S. KOBAYASHI, D. E. STRANDNESS, and C. A. WIEDERHIELM. Reevaluation of arterial constitutive relations: A finite-deformation approach. *Circulation Research*, 30(4):491–500, 1972.
- [242] R. L. Smith, E. F. Blick, J. Coalson, and P. D. Stein. Thrombus production by turbulence. *Journal of applied physiology*, 32(2):261–264, 1972.
- [243] G. Sommer, T. C. Gasser, P. Regitnig, M. Auer, and G. A. Holzapfel. Dissection properties of the human aortic media: an experimental study. *Journal of Biomechanical engineering*, 130(2), 2008.
- [244] G. Sommer, S. Sherifova, P. J. Oberwalder, O. E. Dapunt, P. A. Ursomanno, A. DeAnda, B. E. Griffith, and G. A. Holzapfel. Mechanical strength of aneurysmatic and dissected human thoracic aortas at different shear loading modes. *Journal of biomechanics*, 49(12):2374–2382, 2016.
- [245] B. Spronck, M. Latorre, M. Wang, S. Mehta, A. W. Caulk, P. Ren, A. B. Ramachandra, S.-I. Murtada, A. Rojas, C.-S. He, et al. Excessive adventitial stress drives inflammation-mediated fibrosis in hypertensive aortic remodelling in mice. *Journal of the Royal Society Interface*, 18(180):20210336, 2021.
- [246] B. Staber and J. Guilleminot. A random field model for anisotropic strain energy functions and its application for uncertainty quantification in vascular mechanics. *Computer Methods in Applied Mechanics and Engineering*, 333:94–113, 2018.

- [247] Z. Sun and T. Chaichana. A systematic review of computational fluid dynamics in type b aortic dissection. *International journal of cardiology*, 210:28–31, 2016.
- [248] E. B. Tadmor, M. Ortiz, and R. Phillips. Quasicontinuum analysis of defects in solids. *Philosophical magazine A*, 73(6):1529–1563, 1996.
- [249] P. Tahmasebi, F. Javadpour, and M. Sahimi. Stochastic shale permeability matching: Three-dimensional characterization and modeling. *International Journal of Coal Geology*, 165:231–242, 2016.
- [250] A. S. Tam, M. C. Sapp, and M. R. Roach. The effect of tear depth on the propagation of aortic dissections in isolated porcine thoracic aorta. *Journal of Biomechanics*, 31(7):673–676, 1998.
- [251] A. M. Tartakovsky, C. O. Marrero, P. Perdikaris, G. D. Tartakovsky, and D. Barajas-Solano. Learning parameters and constitutive relationships with physics informed deep neural networks. *arXiv preprint arXiv:1808.03398*, 2018.
- [252] C. Theodoropoulos, Y.-H. Qian, and I. G. Kevrekidis. “coarse” stability and bifurcation analysis using time-steppers: A reaction-diffusion example. *Proceedings of the National Academy of Sciences*, 97(18):9840–9843, 2000.
- [253] M. Thubrikar, P. Agali, and F. Robicsek. Wall stress as a possible mechanism for the development of transverse intimal tears in aortic dissections. *Journal of medical engineering & technology*, 23(4):127–134, 1999.
- [254] G. Tierra, J. P. Pavissich, R. Nerenberg, Z. Xu, and M. S. Alber. Multicomponent model of deformation and detachment of a biofilm under fluid flow. *Journal of The Royal Society Interface*, 12(106):20150045, 2015.

- [255] J. Tinsley Oden, S. Prudhomme, A. Romkes, and P. T. Bauman. Multiscale modeling of physical phenomena: Adaptive control of models. *SIAM Journal on Scientific Computing*, 28(6):2359–2389, 2006.
- [256] J. Tong, Y. Cheng, and G. A. Holzapfel. Mechanical assessment of arterial dissection in health and disease: advancements and challenges. *Journal of Biomechanics*, 49(12):2366–2373, 2016.
- [257] J. S. Tran, D. E. Schiavazzi, A. M. Kahn, and A. L. Marsden. Uncertainty quantification of simulated biomechanical stimuli in coronary artery bypass grafts. *Computer methods in applied mechanics and engineering*, 345:402–428, 2019.
- [258] J. S. Tran, D. E. Schiavazzi, A. B. Ramachandra, A. M. Kahn, and A. L. Marsden. Automated tuning for parameter identification and uncertainty quantification in multi-scale coronary simulations. *Computers & Fluids*, 142:128–138, 2017.
- [259] S. Treitel and L. Lines. Past, present, and future of geophysical inversion—a new millennium analysis. *Geophysics*, (66)(1):21–24, 2001.
- [260] R. K. Tripathy and I. Billionis. Deep uq: Learning deep neural network surrogate models for high dimensional uncertainty quantification. *Journal of computational physics*, 375:565–588, 2018.
- [261] T. T. Tsai, M. S. Schlicht, K. Khanafer, J. L. Bull, D. T. Valassis, D. M. Williams, R. Berguer, and K. A. Eagle. Tear size and location impacts false lumen pressure in an ex vivo model of chronic type b aortic dissection. *Journal of vascular surgery*, 47(4):844–851, 2008.
- [262] A. Tsamis, J. A. Phillippi, R. G. Koch, S. Pasta, A. D’Amore, S. C. Watkins, W. R. Wagner, T. G. Gleason, and D. A. Vorp. Fiber micro-architecture in the

- longitudinal-radial and circumferential-radial planes of ascending thoracic aortic aneurysm media. *Journal of biomechanics*, 46(16):2787–2794, 2013.
- [263] M. S. Tweet, S. N. Hayes, S. R. Pitta, R. D. Simari, A. Lerman, R. J. Lennon, B. J. Gersh, S. Khambatta, P. J. Best, C. S. Rihal, et al. Clinical features, management, and prognosis of spontaneous coronary artery dissection. *Circulation*, 126(5):579–588, 2012.
- [264] V. D. Tzilalis, D. Kamvysis, P. Panagou, I. Kaskarelis, M. K. Lazarides, T. Perdikides, P. Prassopoulos, and H. Boudoulas. Increased pulse wave velocity and arterial hypertension in young patients with thoracic aortic endografts. *Annals of vascular surgery*, 26(4):462–467, 2012.
- [265] R. N. Vaishnav, J. T. Young, J. S. Janicki, and D. J. Patel. Nonlinear anisotropic elastic properties of the canine aorta. *Biophysical Journal*, 12(8):1008–1027, 1972.
- [266] C. van Baardwijk and M. R. Roach. Factors in the propagation of aortic dissections in canine thoracic aortas. *Journal of Biomechanics*, 20(1):67–73, 1987.
- [267] J. van Prehn, K. L. Vincken, B. E. Muhs, G. K. Barwegen, L. W. Bartels, M. Prokop, F. L. Moll, and H. J. Verhagen. Toward endografting of the ascending aorta: insight into dynamics using dynamic cine-cta. *Journal of Endovascular Therapy*, 14(4):551–560, 2007.
- [268] M. Vasta, A. Gizzi, and A. Pandolfi. A spectral decomposition approach for the mechanical statistical characterization of distributed fiber-reinforced tissues. *International Journal of Non-Linear Mechanics*, 106:258–265, 2018.
- [269] H. K. Versteeg and W. Malalasekera. *An introduction to computational fluid dynamics: the finite volume method*. Pearson education, 2007.

- [270] K. von der Mark. Localization of collagen types in tissues. *International review of connective tissue research*, 9:265–324, 1981.
- [271] Y. von Kodolitsch, S. K. Csoz, D. H. Koschyk, I. Schalwat, R. Loose, M. Karck, C. Dieckmann, R. Fattori, A. Haverich, J. Berger, et al. Intramural hematoma of the aorta: predictors of progression to dissection and rupture. *Circulation*, 107(8):1158–1163, 2003.
- [272] J. E. Wagenseil. Bio-chemo-mechanics of thoracic aortic aneurysms. *Current opinion in biomedical engineering*, 5:50–57, 2018.
- [273] J. E. Wagenseil and R. P. Mecham. Vascular extracellular matrix and arterial mechanics. *Physiological reviews*, 89(3):957–989, 2009.
- [274] 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*, 334:337–380, 2018.
- [275] S. Wang, H. Wang, and P. Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed deepnets. *Science Advances*, 7(40):eabi8605, 2021.
- [276] Y. Wang, Z. Li, J. Xu, C. Yang, and G. E. Karniadakis. Concurrent coupling of atomistic simulation and mesoscopic hydrodynamics for flows over soft multi-functional surfaces. *Soft Matter*, 15(8):1747–1757, 2019.
- [277] Z. Wang, X. Huan, and K. Garikipati. Variational system identification of the partial differential equations governing the physics of pattern-formation: inference under varying fidelity and noise. *Computer Methods in Applied Mechanics and Engineering*, 356:44–74, 2019.

- [278] Z. Wang, X. Huan, and K. Garikipati. Identification of the partial differential equations governing microstructure evolution in materials: Inference over incomplete, sparse and spatially non-overlapping data. *arXiv preprint arXiv:2001.04816*, 2020.
- [279] Z.-B. Wang, R. Chen, H. Wang, Q. Liao, X. Zhu, and S.-Z. Li. An overview of smoothed particle hydrodynamics for simulating multiphase flow. *Applied Mathematical Modelling*, 40(23-24):9625–9655, 2016.
- [280] E. Weinan. *Principles of multiscale modeling*. Cambridge University Press, 2011.
- [281] E. Weinan and B. Engquist. Multiscale modeling and computation. *Notices of the AMS*, 50(9):1062–1070, 2003.
- [282] E. Weinan, B. Engquist, et al. The heterogenous multiscale methods. *Communications in Mathematical Sciences*, 1(1):87–132, 2003.
- [283] E. E. Weinmann and E. W. Salzman. Deep-vein thrombosis. *New England Journal of Medicine*, 331(24):1630–1641, 1994.
- [284] A. Witthoft, A. Yazdani, Z. Peng, C. Bellini, J. D. Humphrey, and G. E. Karniadakis. A discrete mesoscopic particle model of the mechanics of a multi-constituent arterial wall. *Journal of The Royal Society Interface*, 13(114):20150964, 2016.
- [285] L. Wu, K. Zulueta, Z. Major, A. Arriaga, and L. Noels. Bayesian inference of non-linear multiscale model parameters accelerated by a deep neural network. *Computer Methods in Applied Mechanics and Engineering*, 360:112693, 2020.

- [286] T. Wu, B. Rosić, L. De Lorenzis, and H. G. Matthies. Parameter identification for phase-field modeling of fracture: a bayesian approach with sampling-free update. *Computational Mechanics*, 67(2):435–453, 2021.
- [287] A. Wufsus, N. Macera, and K. Neeves. The hydraulic permeability of blood clots as a function of fibrin and platelet density. *Biophysical journal*, 104(8):1812–1823, 2013.
- [288] C. Wunsch. *The ocean circulation inverse problem*. Cambridge University Press, 1996.
- [289] S. Xiao and T. Belytschko. A bridging domain method for coupling continua with molecular dynamics. *Computer Methods in Applied Mechanics and Engineering*, 193(17-20):1645–1669, 2004.
- [290] S. Xu, Z. Xu, O. V. Kim, R. I. Litvinov, J. W. Weisel, and M. Alber. Model predictions of deformation, embolization and permeability of partially obstructive blood clots under variable shear flow. *Journal of the Royal Society Interface*, 14(136):20170441, 2017.
- [291] X. Xu, M. D’Elia, C. Glusa, and J. T. Foster. Machine-learning of nonlocal kernels for anomalous subsurface transport from breakthrough curves. *arXiv preprint arXiv:2201.11146*, 2022.
- [292] Z. Xu, N. Chen, M. M. Kamocka, E. D. Rosen, and M. Alber. A multi-scale model of thrombus development. *Journal of the Royal Society Interface*, 5(24):705–722, 2008.
- [293] X. Xun, J. Cao, B. Mallick, A. Maity, and R. J. Carroll. Parameter estimation of partial differential equation models. *Journal of the American Statistical Association*, 108(503):1009–1020, 2013.

- [294] H. Yanagisawa, E. C. Davis, B. C. Starcher, T. Ouchi, M. Yanagisawa, J. A. Richardson, and E. N. Olson. Fibulin-5 is an elastin-binding protein essential for elastic fibre development in vivo. *Nature*, 415(6868):168–171, 2002.
- [295] L. Yang, X. Meng, and G. E. Karniadakis. B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data. *Journal of Computational Physics*, 425:109913, 2021.
- [296] L. Yang, D. Zhang, and G. E. Karniadakis. Physics-informed generative adversarial networks for stochastic differential equations. *SIAM Journal on Scientific Computing*, 42(1):A292–A317, 2020.
- [297] Y. Yang, B. Engquist, J. Sun, and B. F. Hamfeldt. Application of optimal transport and the quadratic wasserstein metric to full-waveform inversion. *Geophysics*, 83(1):R43–R62, 2018.
- [298] Y. Yang and P. Perdikaris. Adversarial uncertainty quantification in physics-informed neural networks. *Journal of Computational Physics*, 394:136–152, 2019.
- [299] M. Yasaka, T. Yamaguchi, and M. Shichiri. Distribution of atherosclerosis and risk factors in atherothrombotic occlusion. *Stroke*, 24(2):206–211, 1993.
- [300] A. Yazdani, H. Li, J. D. Humphrey, and G. E. Karniadakis. A general shear-dependent model for thrombus formation. *PLoS computational biology*, 13(1):e1005291, 2017.
- [301] M. Yin, E. Ban, B. V. Rego, E. Zhang, C. Cavinato, J. D. Humphrey, and G. E. Karniadakis. Simulating progressive intramural damage leading to aortic dissection using deepoNet: an operator–regression neural network. *Journal of the Royal Society Interface*, 19(187):20210670, 2022.

- [302] M. Yin, E. Ban, B. V. Rego, E. Zhang, C. Cavinato, J. D. Humphrey, and G. E. Karniadakis. Simulating progressive intramural damage leading to aortic dissection using deeponet: an operator–regression neural network. *Journal of Royal Society Interface*, 19:20140397, 2021.
- [303] M. Yin, A. Yazdani, and G. E. Karniadakis. One-dimensional modeling of fractional flow reserve in coronary artery disease: Uncertainty quantification and bayesian optimization. *Computer Methods in Applied Mechanics and Engineering*, 353:66–85, 2019.
- [304] M. Yin, E. Zhang, Y. Yu, and G. E. Karniadakis. Interfacing finite elements with deep neural operators for fast multiscale modeling of mechanics problems. *Computer Methods in Applied Mechanics and Engineering*, page 115027, 2022.
- [305] M. Yin, X. Zheng, J. D. Humphrey, and G. E. Karniadakis. Non-invasive inference of thrombus material properties with physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 375:113603, 2021.
- [306] X. Yin, S. Wanga, A. L. Fellows, J. Barallobre-Barreiro, R. Lu, H. Davaapil, R. Franken, M. Fava, F. Baig, P. Skroblin, et al. Glycoproteomic analysis of the aortic extracellular matrix in marfan patients. *Arteriosclerosis, thrombosis, and vascular biology*, 39(9):1859–1873, 2019.
- [307] H. You, Y. Yu, M. D’Elia, T. Gao, and S. Silling. Nonlocal kernel network (nkn): a stable and resolution-independent deep neural network. *arXiv preprint arXiv:2201.02217*, 2022.

- [308] H. You, Y. Yu, S. Silling, and M. D’Elia. A data-driven peridynamic continuum model for upscaling molecular dynamics. *Computer Methods in Applied Mechanics and Engineering*, 389:114400, 2022.
- [309] X. Yu, B. Suki, and Y. Zhang. Avalanches and power law behavior in aortic dissection propagation. *Science advances*, 6(21):eaaz1173, 2020.
- [310] X. Yu, R. Turcotte, F. Seta, and Y. Zhang. Micromechanics of elastic lamellae: unravelling the role of structural inhomogeneity in multi-scale arterial mechanics. *Journal of The Royal Society Interface*, 15(147):20180492, 2018.
- [311] Y. Yu, F. F. Bargas, H. You, M. L. Parks, M. L. Bittencourt, and G. E. Karniadakis. A partitioned coupling framework for peridynamics and classical theory: analysis and simulations. *Computer Methods in Applied Mechanics and Engineering*, 340:905–931, 2018.
- [312] Y. Yu, H. You, and N. Trask. An asymptotically compatible treatment of traction loading in linearly elastic peridynamic fracture. *Computer Methods in Applied Mechanics and Engineering*, 377:113691, 2021.
- [313] X. Zeng and S. Li. A multiscale cohesive zone model and simulations of fractures. *Computer Methods in Applied Mechanics and Engineering*, 199(9-12):547–556, 2010.
- [314] D. Zhang, L. Lu, L. Guo, and G. E. Karniadakis. Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems. *Journal of Computational Physics*, 397:108850, 2019.
- [315] E. Zhang, M. Dao, G. E. Karniadakis, and S. Suresh. Analyses of internal structures and defects in materials using physics-informed neural networks. *Science Advances*, 8(7):eabk0644, 2022.

- [316] E. Zhang, B. Spronck, J. D. Humphrey, and G. E. Karniadakis. G2 ϕ net: Inferring tissue genotype-to-phenotype relations with deep learning. *arXiv preprint*, 2022.
- [317] E. Zhang, M. Yin, and G. E. Karniadakis. Physics-informed neural networks for nonhomogeneous material identification in elasticity imaging. *arXiv preprint arXiv:2009.04525*, 2020.
- [318] L. Zhang, J. Han, H. Wang, R. Car, and E. Weinan. Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics. *Physical Review Letters*, 120(14):143001, 2018.
- [319] T. Zhang, X. Li, and H. Gao. Fracture of graphene: a review. *International Journal of Fracture*, 196(1-2):1–31, 2015.
- [320] H. Zhao, B. D. Storey, R. D. Braatz, and M. Z. Bazant. Learning the physics of pattern formation from images. *Physical Review Letters*, 124(6):060201, 2020.
- [321] X. Zheng, A. Yazdani, H. Li, J. D. Humphrey, and G. E. Karniadakis. A three-dimensional phase-field model for multiscale modeling of thrombus biomechanics in blood vessels. *PLOS Computational Biology*, 16(4):e1007709, 2020.
- [322] G. Zi and T. Belytschko. New crack-tip elements for xfem and applications to cohesive cracks. *International Journal for Numerical Methods in Engineering*, 57(15):2221–2240, 2003.
- [323] T. I. Zohdi. Homogenization methods and multiscale modeling. *Encyclopedia of Computational Mechanics Second Edition*, pages 1–24, 2017.
- [324] B. Zoph and Q. V. Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.