UC Berkeley

UC Berkeley Electronic Theses and Dissertations

Title

Machine Learning Approaches Toward Diagnosis and Biomechanical Analysis of Cardiovascular Disease

Permalink

https://escholarship.org/uc/item/0jz631mg

Author

Madani, Ali

Publication Date

2019

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

Machine Learning Approaches Toward Diagnosis and Biomechanical Analysis of Cardiovascular Disease

by

Ali Madani

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering – Applied Science and Technology

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Mohammad Mofrad, Chair
Professor Tarek Zohdi
Professor Alexei Efros

Spring 2019

Machine Learning Approaches Toward Diagnosis and Biomechanical Analysis of Cardiovascular Disease

Copyright 2019
by
Ali Madani

Abstract

Machine Learning Approaches Toward Diagnosis and Biomechanical Analysis of Cardiovascular Disease

by

Ali Madani

Doctor of Philosophy in Engineering – Applied Science and Technology

University of California, Berkeley

Professor Mohammad Mofrad, Chair

Machine learning with deep neural networks has demonstrated high performance for high dimensionality prediction tasks across multiple domains with sufficient sample data. Cardiovascular disease is a pertinent public health issue that has the potential to be better understood and addressed via deep learning approaches. In this work, we study machine learning approaches toward diagnosing various forms of cardiovascular disease and predicting its biomechanical behavior across multiple scales.

We begin by training deep learning models for an initial classification objective in echocardiography, a ubiquitous imaging modality for cardiologists. For view classification, we are able to demonstrate physician-level performance. We then expand the work from a methods and clinical application perspective. We address the high cost of annotation in medical imaging by examining data-efficient supervised and semi-supervised algorithms. In addition, we expand our prediction tasks towards the ultimate goal of automated, accurate cardiovascular disease diagnosis by predicting left ventricular hypertrophy.

To understand the nature of cardiovascular disease and develop treatments, a close look at the underlying biomechanics is important. For atherosclerosis, a leading cause of morbidity and mortality, we bridge finite element methods and machine learning to predict arterial tissue stress. Likewise for cytoskeletal proteins, which are the structural building blocks of human biology and influence cardiovascular health, we develop graph neural network algorithms to predict force response and conformational dynamics in calponin domains.

Moreover, we hope to lay the groundwork to advance the intersection of machine learning, biomechanics, and cardiovascular disease.

In the Name of God, the most Gracious, the most Merciful
… Dedicated to all those that needed a second chance …

Contents

Contents **ii**

List of Figures **iv**

List of Tables **x**

1 Introduction **1**
1.1 Neural Network Architectures 3
1.2 Dissertation Structure 6

2 View Classification in Echocardiography using Deep Learning: Initial Case Study for Cardiovascular Imaging **8**
2.1 Introduction 8
2.2 Results 9
2.3 Discussion 15
2.4 Methods 18

3 Data-Efficient Supervised and Semi-supervised Learning Towards Automated Diagnosis **25**
3.1 Introduction 25
3.2 Results 27
3.3 Discussion 36
3.4 Methods 42

4 Bridging Finite Element Modeling and Machine Learning for Atherosclerosis **50**
4.1 Introduction 50
4.2 Methods 52
4.3 Results 62
4.4 Discussion 65

5 Bridging Molecular Dynamics Modeling and Graph Neural Networks for CH Domain Conformational Mechanics **68**

5.1 Introduction 68
5.2 Methods 69
5.3 Results and Discussion 75

6 Conclusion **78**
6.1 Synopsis 78
6.2 Future Work 79
6.3 Closure 80

Bibliography **81**

List of Figures

2.1 Convolutional neural net architecture for image classification. a The neural network algorithm used for classification included six convolutional layers and two fully-connected layers of 1028 and 512 nodes, respectively. The softmax classifier (pink circles) consisted of up to 15 nodes, depending on the classification task at hand. b Training, validation, and test data were split by study, and test data was not used for training or validating the model. The model was trained to classify images, with video classification as a majority rules vote on related image frames. Conv convolutional layer, Max Pool max pooling layer, FC fully connected layer 10

2.2 Sample input images. Views classified included parasternal long axis (psla), right ventricular inflow (rv inflow), basal short axis (sax basal), short axis at mid or mitral level (sax mid), apical four-chamber (a4c), apical five chamber (a5c), apical two chamber (a2c), apical three chamber/apical long axis (a3c), subcostal four-chamber (sub4c), subcostal inferior vena cava (ivc), subcostal/abdominal aorta (subao), suprasternal aorta/aortic arch (supao), pulsed-wave Doppler (PW), continuous-wave Doppler (CW), and m-mode (mmode). Note that these images are the actual resolution of input data to the deep learning algorithm 11

2.3 Natural variations in input data. In addition to applying data augmentation algorithms, we included in each category a range of images representing the natural variation seen in real-life echocardiography. The parasternal long-axis view is shown here for example. Variations include a range of timepoints spanning diastole and systole, differences in gain or chroma map, use of dual-mode acquisition, differences in depth and zoom, technically challenging images, use of 3D acquisition, a range of pathologies (seen here, concentric left ventricular hypertrophy and pericardial effusion), and use of color Doppler, as well as differences in angulation, sector width, and use of LV contrast. Note that these images are the actual resolution of input data to the deep learning algorithm 12

2.4 Deep learning model simultaneously distinguishes among 15 standard echocardiographic views. We developed a deep-learning method to classify among standard echocardiographic views, represented here by t-SNE clustering analysis of image classification. On the left, t-SNE clustering of input echocardiogram images. Each image is plotted in 4800-dimensional space according to the number of pixels, and projected to two-dimensional space for visualization purposes. Different colored dots represent different view classes (see legend in figure). Prior to neural network analysis, input data does not cluster into clear groups. On the right, data as processed through the last fully connected layer of the neural network are again represented in two-dimensional space, showing organization into clusters according to view category. Abbreviations: a4c apical 4 chamber, psla parasternal long axis, saxbasal short axis basal, a2c apical 2 chamber, saxmid short axis mid/mitral, a3c apical 3 chamber, sub4c subcostal 4 chamber, a5c apical 5 chamber, ivc subcostal ivc, rvinflow right ventricular inflow, supao suprasternal aorta/aortic arch, subao subcostal/abdominal aorta, cw continuous-wave Doppler, pw pulsed-wave Doppler, mmode m-mode recording 13

2.5 Echocardiogram view classification by deep-learning model. Confusional matrices showing actual view labels on y-axis, and neural network-predicted view labels on the x-axis by view category for video classification (a) and still-image classification (b) compared with a representative board-certified echocardiographer (c). Reading across true-label rows, the numbers in the boxes represent the percentage of labels predicted for each category. Color intensity corresponds to percentage, see heatmap on far right; the white background indicates zero percent. Categories are clustered according to areas of the most confusion. Rows may not add up to 100 percent due to rounding. d Comparison of accuracy by view category for deep-learning-assisted video classification, still-image classification, and still-image classification by a representative echocardiographer. e A comparison of percent of images correctly predicted by view category, when considering the models highest-probability top hit (white boxes) vs. its top two hits (blue boxes). f Receiver operating characteristic curves for view categories were very similar, with AUCs ranging from 0.985 to 1.00 (mean 0.996). Abbreviations: saxmid short axis mid/mitral, ivc subcostal ivc, subao subcostal/abdominal aorta, supao suprasternal aorta/aortic arch, saxbasal short axis basal, rvinflow right ventricular inflow, a2c apical 2 chamber, a3c apical 3 chamber, a4c apical 4 chamber, a5c apical 5 chamber, psla parasternal long axis, sub4c subcostal 4 chamber 14

2.6 Visualization of decision-making by neural network. a Occlusion experiments. All test images (a short axis basal sample image is shown here) were modified with grey masking of different shapes and sizes as shown, and test accuracy predicted for the test set based on each different modification. Masking that covered cardiac structures resulted in the poorest predictions. b Saliency maps. The input pixels weighted most heavily in the neural networks classification decision for two example images (left; suprasternal aorta/aortic arch and short axis mid/mitral input examples shown) were calculated and plotted. The most important pixels (right) make an outline of structures clinically relevant to the view shown 16

2.7 Confidence for first and second guesses on image classification. Box plot summarizing probabilities assigned to correct and incorrect images in the test set using the single highest probability to classify the test image. Confidence for correct answers was higher than for incorrect answers. Median, interquartile range for correct (0.999, 0.970-1.00) and incorrect (0.682, 0.525-0.867) answers. 23

2.8 Example native-resolution and downsampled images. (a) Native echocardiographic images ranged from 300x400 to 768x1024 pixels in resolution, and many contained color, either as color Doppler or as different chroma maps to aid in visualization. (b) The same sample image as in (a), downsampled to the 60-by-80-pixel resolution used as input to the deep learning model. 24

3.1 Deep learning for echocardiography study diagram. (a) Two classification tasks were examined for echocardiography: view classification and left ventricular hypertrophy (LVH) classification. (b) Two different approaches for deep learning models were taken: a supervised pipeline model that performs segmentation (U-Net) before classification (CNN) and a semi-supervised generative adversarial network (GAN) for end-to-end learning. (c) The data, for both view and LVH classification, was split accordingly by study and no test data was utilized in training or validating the model. 28

3.2 An optimal resolution size exists considering performance vs computational time tradeoffs. Plot of validation accuracy and training time per epoch with input resolution. The optimal balance between computational burden and validation accuracy exists at 120x160 resolution. 29

3.3 Figure of image data, labeled map, predicted map and predicted map applied to original image for Field of View (left) and LVH Segmentation (right). Trained segmentation models are able to accurately discern contours in echocardiogram images and output a map over relevant areas. 30

3.4 Field of View (FoV) segmentation and View Classification pipeline. U-Net predicts a segmentation map over the main FoV, which is applied to the input image prior to view classification. FoV segmentation before view classification improved performance of the CNN model from an overall test accuracy of 92.05% to 93.64%. 32

3.5 Normalized Confusion Matrix (left) and Accuracy by Class (right) on test set for ensemble network. 11 out of 15 classes achieved test accuracy above 90%. Classes with highest rate of confusion such as A5C with A4C are consistent with structural similarity overlap between the two classes. 33
3.6 View classification and left ventricular hypertrophy classification pipeline. View segmentation U-Net predicts a segmentation map over the main FoV, which is applied to the input image prior to view classification. Based on the predicted view class, input image is routed to the respective disease classification pipeline. a4c images are routed to a4c segmentation U-Net, which predicts a segmentation map over the left ventricle. This map is applied to the input image prior to LVH disease classification. 34
3.7 Normalized confusion matrix on LVH test set for CNN with segmentation stage 2. CNN model with segmentation was able to classify test images with 91.21% (+/- 0.41) accuracy (specificity 95.70%, sensitivity 76.70%). 35
3.8 View classification performance of the semi-supervised generative adversarial network for varying amounts of labels as input. The model is able to learn from very small amounts of labeled data (approximately 4% of labels kept with the remaining data as unlabeled) to achieve greater than 80% accuracy for view classification. There exists an exponentially asymptotic behavior over number of labeled samples where accuracy gain becomes less prominent. 37
3.9 Performance of semi-supervised GAN for LVH classification in apical 4 chamber echocardiogram images. a) For three separate models trained, the accuracy (top row) and F1-score (bottom row) is plotted vs number of epochs. The model training reliably reaches convergence and continues to fluctuate within a reasonable limit. b) Normalized confusion matrix for an ensemble of the three models. This achieves an F1-score of 0.83 and accuracy of 92.3% (+/- 0.57). 38
3.10 Generated images sampled from the generator network of the semi-supervised GAN during training for LVH classification. For one model, batches of size four are shown containing generated images (top to bottom) after epoch 1, 2, 3, 4, 13. The last row displays generated images from an ensemble of three GAN models. Qualitatively, the model clearly learns and understands the underlying physiological structures in input distribution. 39
3.11 Figure of labeled map, predicted map, labeled and predicted map applied to original image for LVH Segmentation. Even in cases where there is a large difference between the labeled and the predict map, the amount of information loss after applying the predicted map is very limited. 40

3.12 Sample data from echocardiographic studies. a) Sample echocardiogram images at varying resolutions (rows) for three example views (columns). Selection of the optimal resolution is influenced by the tradeoff between classifier performance and computational time. b) Sample apical four chamber echocardiography images with and without Left Ventricular Hypertrophy (LVH). LVH is characterized by the thickening of the left ventricle– a perilous condition increasing the risk of myocardial infarction, stroke, and death. 43

3.13 Modified U-Net architecture used for segmenting relevant visual structures. The architecture consists of a contracting path and a symmetric expanding path– combining high resolution features from the contracting path and upsampled output for precise localization. Pixel-wise softmax is applied on the final output to produce a segmentation map. 46

3.14 Semi-supervised GAN for echocardiogram view and LVH classification. Generator (top) consists of a Gaussian noise layer which gets passed through conv-transpose layers to output images of size 110x110. Discriminator (bottom) downsamples original images with regular strides of two every three layers, resulting in a softmax output for labeled (or supervised) loss and a sigmoid output for unsupervised loss. 48

4.1 Prediction of atherosclerotic vessel stress with machine learning (ML) models trained on finite element method (FEM) simulation data. a) Mechanistic and statistical methods can be used for various continuum mechanics problems. b) Atherosclerosis can cause the buildup of plaque in blood vessels as shown. c) An example idealized geometry of a 2D cross-section of an atherosclerotic vessel. Red- vessel wall, green- fibrous plaque, yellow- lipid pool, blue- calcium deposit, white- lumen. Geometries are randomly generated to create a database of FEM simulations. d) An example stress distribution generated by utilizing FEM. The input and output information are utilized to train ML models for prediction tasks such as maximum von Mises stress. 51

4.2 Model architecture of five approaches taken for training a supervised deep learning model. Approach number ranges from 1 to 5 from top to bottom. Red coloring denotes input data fed to network. Blue coloring denotes output data used for loss calculation. 61

4.3 Deep learning can accurately predict physiologically-relevant indicators (maximum von mises stress and location) for a given vessel geometry and pressure. a) Training and validation loss over 35 epochs displaying convergence and behavior of model training. b) Normalized maximum von mises stress values for the held-out test set and predictions by the ML model. c) 2D heat map of coordinate location of maximum von mises stress over all the held-out test set samples. d) 2D heat map of predicted locations of maximum von mises stress on the held-out test set. 64

4.4 Comparative studies on varying baseline models and training dataset size. a) Deep learning model outperforms other baseline and machine learning strategies. b) Varying amounts of data are utilized for training the same model to examine performance vs data size. 65

5.1 Workflow Schematic: From Biophysics to Graph Neural Networks 70
5.2 MPNet + GRU Performance. a) Classification training statistics b) Regression histograms of predicted vs actual values 76

List of Tables

2.1 Patient Characteristics Table - Comparison of study sample characteristics to clinical echo database. Age (years), Height (cm), Weight (kg), BP blood pressure (mmHg), MAP mean arterial pressure (mmHg), BSA body surface area (m^2), BMI body mass index (kg/m^2), LVMI left ventricular mass index (g/m^2) adjusted for sex, LVEF left ventricular ejection fraction, LVEDVI left ventricular end-diastolic volume index (ml/m2) adjusted for sex, RVSP right ventricular systolic pressure, TAPSE tricuspid annular plane systolic excursion, SD standard deviation, IQR interquartile range. a- p-value. Ns and sample size vary according to availability of different measurements b- p-value. Two-tailed Students t-test, unequal variance c- p-value. Chi-squared test for comparison of proportions . . . 19
2.2 Indications for study sample echocardiograms. a- Defined by echo reports documenting normal four-chamber size and systolic/diastolic function, no chamber hypertrophy or wall motion abnormalities, normal valves with trace or less regurgitation, normal great vessels and estimated right atrial pressure, no pericardial effusion, RVSP<=40, and no other abnormalities, such as atherosclerosis, calcification, pleural effusion, ascites, prostheses, or catheters 22

3.1 Table of evaluation results for resolution study. Validation accuracy plateaus at sub-88% accuracy while training and testing time continue to scale with increasing input resolution. 27
3.2 Table of evaluation results for each model for view classification. Highest performing model by overall test accuracy is an ensemble of three CNN models with field of view segmentation, while transfer learning from Resnet50 and VGG16 models yielded lower test accuracies compared to a single CNN model. 31
3.3 Test accuracies and F1 Scores for LVH classification models. In stage 1, weights for convolution layers were fixed and model was trained over 20 epochs. In stage 2, weights of fully-connected layers were fixed and convolution layers were fine-tuned. 35
3.4 View classification performance of the semi-supervised generative adversarial network for varying amounts of labels as input. 36

4.1 Features for random generation of idealized artery geometry and pressure conditions. N and U denote a normal and uniform distribution respectively. The number of lipids and calcium are also varied as discrete numbers ranging from zero to two deposits. Each ellipse is described by a major axis a, minor axis b, center (u,v), and angle of rotation 52
4.2 Summary of calibrated model parameters in (4.17) obtained from [11, 25]. . . . 57

5.1 Input Representation Features 73
5.2 Prediction Performance of Various Architectures 75

Acknowledgments

In the midst of applying to graduate programs, I distinctly remember sitting in despair at the steps of Stanley Hall late one night– pessimistically thinking to myself that I would never be able to attend such a fine institution such as the University of California, Berkeley. Little did I know soon thereafter I would be accepted into the PhD program at Cal and would start one of the greatest journeys of my life.

First, I am forever thankful to my research advisor and dissertation chair Professor Mohammad R.K. Mofrad. He has been a consistent source of support, guidance, and training. I can proudly say it has been an honor to be his graduate student.

I wish to express my sincere gratitude to my qualification and dissertation committee - Professor Tarek Zohdi, Professor Alyosha Efros, and Professor Ian Holmes. In addition, I gratefully acknowledge the discussions and support of the members of the lab– from past to present PhD students, masters students and the absolutely fantastic undergrads. I am grateful for all the friends I have made both part of the lab and across campus departments.

I would like to thank the Applied Science and Technology program, graduate advisors, and student advisor Ariana Castro. In addition, I would like to thank the UC Dissertation Fellowship for allowing me to focus on my dissertation.

I would like to acknowledge and thank my beautiful, loving family. I can never say enough of the boundless support, love, and guidance that my mother and father have provided me since the day I was born til today. I would be nothing without them– quite figuratively and literally. Likewise for my wonderful sister and brother who I truly love and have been my greatest advocates.

Finally, I acknowledge whole-heartedly my Fatema ...my love, my soul mate, and my best friend. My PhD journey coincided with our marriage journey and I truly could not have done it without you.

Thank you all!

Chapter 1

Introduction

Cardiovascular disease (CVD) presents itself as a major global public health concern. It has historically been the leading cause of death worldwide– equivalent in number to the sum of the following four causes of death [107]. The physical, financial, and emotional burden of its varying types, such as stroke, aneurysms, and coronary artery disease, weighs heavy on the general public and demands innovative research to better understand, predict, and address the disease.

The field of biomechanics yields a unique lens towards the study of human health and disease. Biomechanics, itself, refers to the study of the structure, function, and motion of mechanical aspects of biological systems– ranging from organism-level, tissue-level, to cell and protein-level phenomena. From a mechanical analysis perspective, cardiovascular disease and human biology/medicine is incredibly complex. The heart and vascular system pump oxygenated blood and nutrients across the body. They comprise a network of muscle and connective tissue, from cardiac chambers, valves, arteries, and veins, that efficiently work together and are subject to a variety of stresses and strains from both a solid and fluid mechanics perspective. Although it is a feat of engineering and robust to many a biological shock, the cardiovascular system can be afflicted with disease and present perilous life-threatening symptoms. One can use the field of computational modeling to analyze organ- and tissue-level phenomena to better understand physiological mechanisms and design treatments– the first step of which is to accurately identify and characterize geometry and tissue characteristics.

Advances in medical imaging and acquisition through modalities such as magnetic resonance imaging (MRI), computed tomography (CT), and ultrasound (echocardiography) have enabled us unprecedented access to the inner workings of the human body. Echocardiography in particular is a low-cost, no-radiation modality that is the proverbial next gen stethoscope for cardiologists. Physicians are able to make structural and functional assessments of the patient and prescribe treatments based on the information given. A natural line of computational research is to automate and improve prediction of geometric and dynamics parameters along with clinical classifications/diagnoses. Those predictions can then be used to assist physicians in the ultimate task of patient evaluation and diagnosis. The benefit would be (1) to improve diagnosis accuracy as physicians are routinely subject to burn-out and naturally err and (2) to expand medical access to low-resource areas in terms of trained clinical staff. Another benefit is using predictive algorithms as tools for precision phenotyping that can be applied across large patient sample sizes to find pre-disposing risk factors and develop new standards of patient care altogether.

The structural information characterized by computational algorithms on medical imaging can also be used for further downstream biomechanical analysis methods. The finite element (FE) Method is a numerical method to approximate the solution of partial differential equation (PDE) problems, ubiquitous across nature, into a system of algebraic equations. The FE method along with other corollary approaches for solid mechanics and fluid dynamics simulations have been used extensively to understand the stress-strain behavior of cardiovascular disease to better prevent and address treatment. [108, 91, 11, 109, 47, 41]

While much can be derived from examining cardiovascular disease from a continuum mechanics perspective, the cardiovascular system and its tissue is composed of cells– which are alive, responsive, and adaptable. Cells are comprised of proteins which perform the majority of work in cellular environments. Proteins can be further broken down into amino acids and other molecular sub-divisions/scales. In this study, we stop at the scale of the phenomena that can be modeled using all-atom classical molecular dynamics– a method which describes the Newtonian equations of motion of atoms in a Cartesian system under a defined force field. Under these scales, proteins interact and undergo conformational changes driving the majority of cellular function and structure. Overarching fields have been developed to study these phenomena such as mechanotransduction, a broad class of mechanisms by which cells convert mechanical stimulus into electrochemical activity. These can range from functions such as cell adhesion and differentiation to structures such as the cytoskeletal network of proteins. The root of many cardiovascular diseases are caused by issues at this level as evidenced by numerous studies [44, 13, 44, 53, 101]. Computational methods, including molecular dynamics for biomechanical analyis, can be useful in this cardiovascular disease domain to better understand protein interactions and possibly affect drug design for treatment purposes.

In this study, we specifically are developing machine learning approaches to better predict, understand, and address cardiovascular disease across multiple scales through the lens of a biomechanicist.

Historically, the majority of research efforts in the application of machine learning for healthcare have been through rule-based techniques that replicate the reasoning processes of clinician experts. Deep learning, however, rose to success due to its data-driven perspective and has produced major breakthroughs for notably difficult areas such as image recognition, speech interpretation, and language translation. Deep learning uses multi-layer neural networks that model the relationships between inputs and outputs. These layers, with ”neurons”, are loosely inspired by the visual cortex that change the strengths of synaptic connections between neurons to establish a hierarchy of progressively more complex feature detectors. Problems in biology and medicine, which we limit our subset to cardiovascular disease in this study, are complex and high dimensional. The prediction of an output in health and disease involves complex interactions between input terms. The process of training a machine learning model involves learning those interactions (i.e. weights), which could be an exponentially large search space. The introduction of backpropagation procedure [56] allowed for an elegant way to efficient undergo optimization of neural network parameters.

1.1 Neural Network Architectures

In this section, we will define, at a high-level, the basic neural network architecture types utilized throughout the dissertation. We begin by thinking of a *supervised learning* problem with X as an input space, Y as an output space, and D as the data distribution over $X \times Y$. The goal is to learn a function $f : X \rightarrow Y$ by learning from some sample, observed data from D that can be further divided into training and testing sets. The data is assumed to contain N independent and identically distributed (i.i.d.) samples

$$S = \{(x_i, y_i)\}_{i=1}^{N} \sim D \tag{1.1}$$

with x_i as an input and y_i as the associated, potentially noisy, output. To find and evaluate our function, f, we aim to minimize the test error to find the ideal f^*:

$$f^* = \operatorname*{argmin}_{f \in \mathcal{F}} E_{(x,y)\sim D}[L(f(x), y)] \tag{1.2}$$

where $L(f(x), y)$ is a loss function that penalizes and measures the distance between predicted output $\hat{y} = f(x)$ and actual output (or target) y. In practice, we would restrict the allowable functions f to a relatively small class of functions $\mathcal{F}$, of which we will describe the neural network based variants.

Neural networks are solely biologically *inspired* and consist of *nodes* (also called *neurons* or *units*), activations at each *node*, and directed edges between them. We begin with the **Multi-layer Perceptron** (MLP) architecture, a feed-forward network consisting of a series of hidden layers of nodes that are one-directional from input vector, x, to output vector, y. Each *node* in a hidden layer is connected to every neuron in the previous layer. The value of each node, h_j, is the weighted sum of the values of previous layer nodes passed through an activation function, $e(\cdot)$ as described below:

$$h_j = e_j\left(\sum_{j'} w_{jj'} \cdot h_{j'}\right) \tag{1.3}$$

Where j denotes the current node, j' denotes an incoming node from a previous layer, and associated weighting w. A MLP has multiple layers of hidden nodes connected by weight matrices, $\boldsymbol{W}$ with dimension $k \times j$ where k is the previous layer dimensionality and j is the subsequent layer dimensionality.

The activation function introduces a non-linear property to the network so it can capture the complex relationships between input and output. The choice of activation function can range from the sigmoid function, $\sigma(z) = 1/(1 + e^{-z})$, to the *tanh* function, $\phi(z) = (e^z - e^{-z})/(e^z + e^{-z})$. An activation function commonly seen with high performance across deep learning is the rectified linear unit (ReLU):

$$e(z) = \max(0, z) \tag{1.4}$$

After a series of hidden layers, the final layer, defined as the *output layer* will have a specific output function based on the prediction task at hand. For binary classification or multi-label multi-class classification, typically a point-wise sigmoid is used. For regression, a linear activation is common. For single-label multi-class classification with K classes, the softmax function below is useful as a differentiable max function that normalizes the outputs to one:

$$\hat{y}_j = \frac{e^{h_j}}{\sum_{k=1}^{K} e^{h_k}} \tag{1.5}$$

Training is accomplished by optimizing the weights of the neural network to minimize a chosen objective or loss function, L between the network output, $\hat{\boldsymbol{y}}$, and desired target output $\boldsymbol{y}$. Common choices of loss function for regression include the L1-loss (mean absolute error) or the L2-loss (mean squared error) function:

$$L(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \|\hat{\boldsymbol{y}} - \boldsymbol{y}\|^2 \tag{1.6}$$

For classification, the cross-entropy loss function is a powerful and informative loss function for optimization purposes:

$$L(\hat{\boldsymbol{y}}, \boldsymbol{y}) = -\frac{1}{N}\left(\sum_{i}^{N} y_i \cdot \log \hat{y}_i\right) \tag{1.7}$$

As referred to earlier, backpropagation is the most successful algorithm for training neural networks. It uses the chain rule to calculate the derivative of the loss function, L, with respect to each parameter in the network. The weights are optimized by some form of gradient descent. It is inherently a non-convex optimization problem, but through specific techniques the empirical results suggest sufficiently deep networks are able repeatedly achieve different, yet sufficiently good local optimum [16]. Neural networks are trained well via stochastic gradient descent (SGD) with mini-batches. An example SGD update equation for one sample is described below:

$$\boldsymbol{W} \leftarrow \boldsymbol{W} - \eta \nabla_{\boldsymbol{w}} L \tag{1.8}$$

where η is the learning rate and $\nabla_{\boldsymbol{w}} L$ is the gradient of the loss function with respect to the parameters $\boldsymbol{w}$. In practice, some form of momentum and weight decay is utilized such as in Adam optimization [51].

Recurrent Neural Networks (RNNs) are feed-forward networks tailored for sequential data by the inclusion of recurrent edges that span adjacent time steps. As the acyclic condition is necessary for backpropagation, RNNs can be viewed as a neural network with one layer per time step and shared weights across time steps. At time t, nodes with recurrent edges receive input from the current data sample, $\boldsymbol{x}^{(t)}$, and also from hidden node values $\boldsymbol{h}^{(t-1)}$ in the network's previous state. The output $\hat{\boldsymbol{y}}^{(t)}$ at each time t is calculated given the hidden node values $\boldsymbol{h}^{(t)}$ at time t. Input $\boldsymbol{x}^{(t-1)}$ at time $t-1$ can influence the output $\hat{\boldsymbol{y}}^{(t)}$ at time t and future time steps by recurrent connections. The following equations specify the interactions between layers:

$$\boldsymbol{h}^{(t)} = \sigma\Big(W^{hx}\boldsymbol{x}^{(t)} + W^{hh}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}^{(h)}\Big) \tag{1.9}$$

$$\hat{\boldsymbol{y}}^{(t)} = \text{softmax}\Big(W^{yh}\boldsymbol{h}^{(t)} + \boldsymbol{b}^{(y)}\Big) \tag{1.10}$$

where W^{hx} is the matrix of weights between input and hidden layers, W^{hh} is the matrix of recurrent weights between time steps of hidden layers, and $\boldsymbol{b}$ are bias parameters across nodes.

Convolutional Neural Networks (CNNs) are heavily used in computer vision as they are able to capture the locality of features and utilize *spatial* invariance for effective learning. The inspiration for the CNN was from the mammalian visual cortex which exhibits similar feature hierarchical structure toward image recognition. For 2D images (although the principle can be applied to other dimensionality problems), the input image is passed through a series of *convolutional layers*. Each convolutional layer's parameters consists of a set of learnable filters usually of limited spatial range. These filters are convolved (or move in a sliding window pattern) across the width and height of the input volume:

$$\boldsymbol{h}_i^{(l)} = e\left(\sum_{i \in M_j} \boldsymbol{x}_i^{(l-1)} * \boldsymbol{k}_{ij}^{l} + b_j^{l}\right) \tag{1.11}$$

where l is the neural layer number, $\boldsymbol{k}$ are the filter weights, b is the bias, M_j is the selection of input maps, and e is the activation function.

Intuitively, the network will learn filters that activate on discriminative visual features from edges to more complex patterns. The resulting activation maps are then stacked along the depth dimension of the image and produce an output volume. Typically, there are pooling steps (such as max-pooling and average-pooling) that are placed successively after convolutional layers to progressively reduce the spatial representational size and reduce the amount of parameters/computation. CNNs benefit from weight sharing, and overall are incredibly efficient and effective at learning a hierarchically ordered set of progressively complex features along successive layers.

Aside from the MLP, RNN, and CNN, which are the traditional workhorse architectures to neural networks, we also engage in research with generative adversarial networks and graph neural networks. **Generative Adversarial Networks** (GANs) involve two neural networks, performing as *adversaries*, engaged in a zero-sum game framework. Typically, the generative network learns a mapping from a latent space to a data distribution of interest and the discriminative network distinguishes samples from the true data distribution and the samples produced by the generative network. GANs have made incredible strides especially in generative tasks for visual data. We refer the reader to Chapter 3 for further explanation of GANs and our objective for discriminative tasks.

Lastly, **Graph Neural Networks** describe a fascinating new area of research for non-Euclidean data that are structured as nodes and edges– which many Euclidean domains can reformulated as graphs as well. CNNs and RNNs take advantage of shared weights and spatial and time invariances for efficient learning. For graphical data, progress has been made to apply similar convolutional operations that leverage node/edge permutation invariances. We delve deeper into the definition and formulation of various graph neural network architectures in Chapter 5.

1.2 Dissertation Structure

In this dissertation, we focus on deep learning for diagnosis and biomechanical analysis of cardiovascular disease. We begin our work with one of the most compelling areas for deep learning, diagnosis in medical imaging, then move into biomechanics prediction problems in different scales– as described in the chapters below.

Chapter 2. View Classification in Echocardiography using Deep Learning: Initial Case Study for Cardiovascular Imaging Echocardiography is the most widely-used imaging modalities for cardiology. We begin the dissertation with an initial study of the ability of trained deep neural network models to predict view classification. We perform a thorough study of the methods and compare with human physician performance.

Chapter 3. Data-Efficient Supervised and Semi-supervised Learning Towards Automated Diagnosis The next chapter engages in developing and studying data-efficient supervised learning techniques, with pipeline classification and segmentation networks, as compared to end-to-end semi-supervised learning techniques, specifically utilizing generative adversarial networks. The study of these methods extends past echocardiography specifically, however we focus on two prediction use-cases: view and left ventricular hypertrophy classification.

Chapter 4. Bridging Finite Element Modeling and Machine Learning for Atherosclerosis We then move into explicitly biomechanical phenomena by building machine learning models that can predict peak von Mises stress in idealized arteries as a potential indicator for plaque rupture due to atherosclerosis. We venture into the bridge between numerical methods such as the finite element method with the statistical methods of machine learning.

Chapter 5. Bridging Molecular Dynamics Modeling and Graph Neural Networks for Calponin Domain Conformational Mechanics In spirit of the above bridge between two modeling paradigms, we then engage in more fundamental biology of cardiovascular disease by predicting protein biomechanical behavior. We utilize molecular dynamics simulations of calponin homology domain stretching force-response. Calponin homology domains in actinin are cytoskeletal proteins that formulate the basic science behind cardiac muscle and function. If we can better predict mechanosensitivity, we can better understand disease and treat it. This project also has the dual purpose of advancing graph neural networks through a reliable benchmark dataset.

Chapter 2

View Classification in Echocardiography using Deep Learning: Initial Case Study for Cardiovascular Imaging

2.1 Introduction

Imaging is a critical part of medical diagnosis. Interpreting medical images typically requires extensive training and practice and is a complex and time-intensive process. Deep learning, specifically using convolutional neural networks (CNNs), is a cutting-edge machine learning technique that has proven *unreasonably* [48] successful at learning patterns in images and has shown great promise helping experts with image-based diagnosis in radiology, pathology, and dermatology, for example, in detecting the boundaries of organs in computed tomography and magnetic-resonance images, flagging suspicious regions on tissue biopsies, and classifying photographs of benign vs. malignant skin lesions. [24, 34, 61] However, deep learning has not yet been widely applied to echocardiography, a noninvasive, relatively inexpensive, radiation-free imaging modality that is an indispensable part of modern cardiology. [23]

A transthoracic echocardiogram (TTE) consists of scores of video clips, still images, and Doppler recordings measured from over a dozen different acquisition angles, offering complementary views of the hearts complex anatomy. The majority of the acquired information is represented as video clips; only pulsed-wave Doppler (PW), continuous-wave Doppler (CW), and m-mode recordings are represented exclusively as single images. Determining the view is the essential first step in interpreting an echocardiogram. [112] This step is non-trivial, not least because several views differ only subtly from each other. In principle, a CNN can be trained to classify views, requiring only a training set of labeled images from which to learn; given a new image, a well-trained model should then be able determine the view almost instantaneously. The versatility of training in deep learning represents a significant advantage over earlier machine-learning methods, which have sometimes been applied to echocardiography. Previous methods often require time-consuming and operator-dependent manual selection and annotation of features (e.g. manually tracing the outline of the heart) in each of a large number of training images, and are out-performed by deep learning on complex, high-dimensional problems, such as image recognition. [49, 52, 74, 77, 92]

To assist echocardiographers and improve use of echocardiography for precision medicine, we tested whether supervised deep learning with CNNs can be used to automatically classify views without requiring prior manual feature selection. We report a model that achieves nearly 98 percent overall test accuracy based on a variety of video and still-image view-classification tasks.

To achieve translational impact in medicine, novel computational models must not just achieve high accuracy but must also address clinical relevance. We did this in three main ways. First, we used randomly selected, real-world echocardiograms to train our model, including a variety of patient variables, echocardiographic indications and pathologies, technical qualities, and multiple vendors to ensure that our deep learning model would be clinically relevant. Second, deep learning approaches are often considered data hungry; we sought to achieve high accuracy on view classification with minimal data. Third, deep-learning models are sometimes considered black boxes because their internal workings are at first glance obscure. To address this issue, we used several methods to look inside our model to show that classification depends on human-recognizable clinical features within images.

Taken together, these results suggest that our approach may be useful in helping echocardiographers improve their accuracy, efficiency, and workflow and provide a foundation for high-throughput analysis of echocardiographic data.

2.2 Results

Deep learning achieves expert-level view classification

We designed and trained a convolutional neural network (CNN) (Figure 2.1) to recognize 15 different standard echocardiographic views, 12 from b-mode (video and still image) and three from pulsed-wave Doppler (PW), continuous-wave Doppler (CW), and m-mode (still image) recordings (Figure 2.2), using a training and validation set of over 200,000 images (240 studies) and a test set of over 20,000 images (27 studies). To maintain sample independence, each echocardiogram was from a different patient, and training, validation and test sets did not overlap by patient or study (Figure 2.1b). These images covered a range of natural echocardiographic variation with patient variables (Table 2.1) and indications for imaging (Table 2.2) that represented our overall clinical database, and they included differences in zoom, depth, focus, sector width, gain, chroma map, systole/diastole, angulation, image quality, and use of 3D, color Doppler, dual mode, strain, and LV contrast (Figure 2.3). Clustering analyses showed that the neural network could sort heterogeneous input images into groups according to view (Figure 2.4).

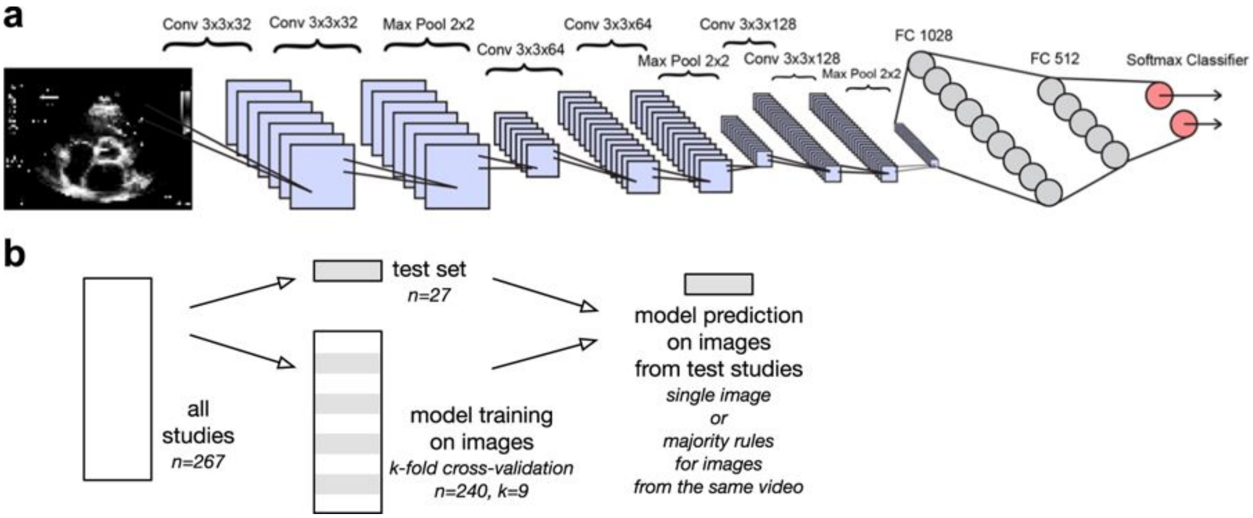

Figure 2.1: Convolutional neural net architecture for image classification. a The neural network algorithm used for classification included six convolutional layers and two fully-connected layers of 1028 and 512 nodes, respectively. The softmax classifier (pink circles) consisted of up to 15 nodes, depending on the classification task at hand. b Training, validation, and test data were split by study, and test data was not used for training or validating the model. The model was trained to classify images, with video classification as a majority rules vote on related image frames. Conv convolutional layer, Max Pool max pooling layer, FC fully connected layer

The model achieved an average overall test accuracy of 97.8 percent on videos (F-score 0.964s.d. 0.035) and 100 percent accuracy on seven of the 12 video views (Figure 2.5a). CW, PW, and m-mode categories, which always appeared in echocardiograms as still images, had 98, 83, and 99 percent accuracies, respectively (Figure 2.5b). Classification of test images by the trained model took an average of 21ms per image on a standard laptop.

On single still images drawn from all 15 views, the model achieved an average overall accuracy of 91.7 percent (F-score 0.904s.d. 0.058) (Figure 2.5b), compared to an average of 79.4 percent (range, 70.284.0; n=4 subjects) for board-certified echocardiograpers classifying a subset of the same test images (one-sample t-test, p=0.03) (Figure 2.5c). Associated areas under the curve (AUCs) for still-image model prediction by view category ranged from 0.985 to 1.00 (mean 0.996; Figure 2.5f). For the 8.3 percent of test images that the model misclassified, its second-best guessthe view with the second-highest probabilitywas the correct one in 67.0 percent of cases (5.3 percent of test images; Figure 2.5e). Therefore, 97.3 percent of test still-images were classified correctly when considering the models top two guesses.

Accuracy was highest for views with more training data (e.g. apical four-chamber) and views that are most visually distinct from the others (e.g. m-mode). Accuracy was lowest for views that were clinically similar to other views, such as apical three-chamber (which

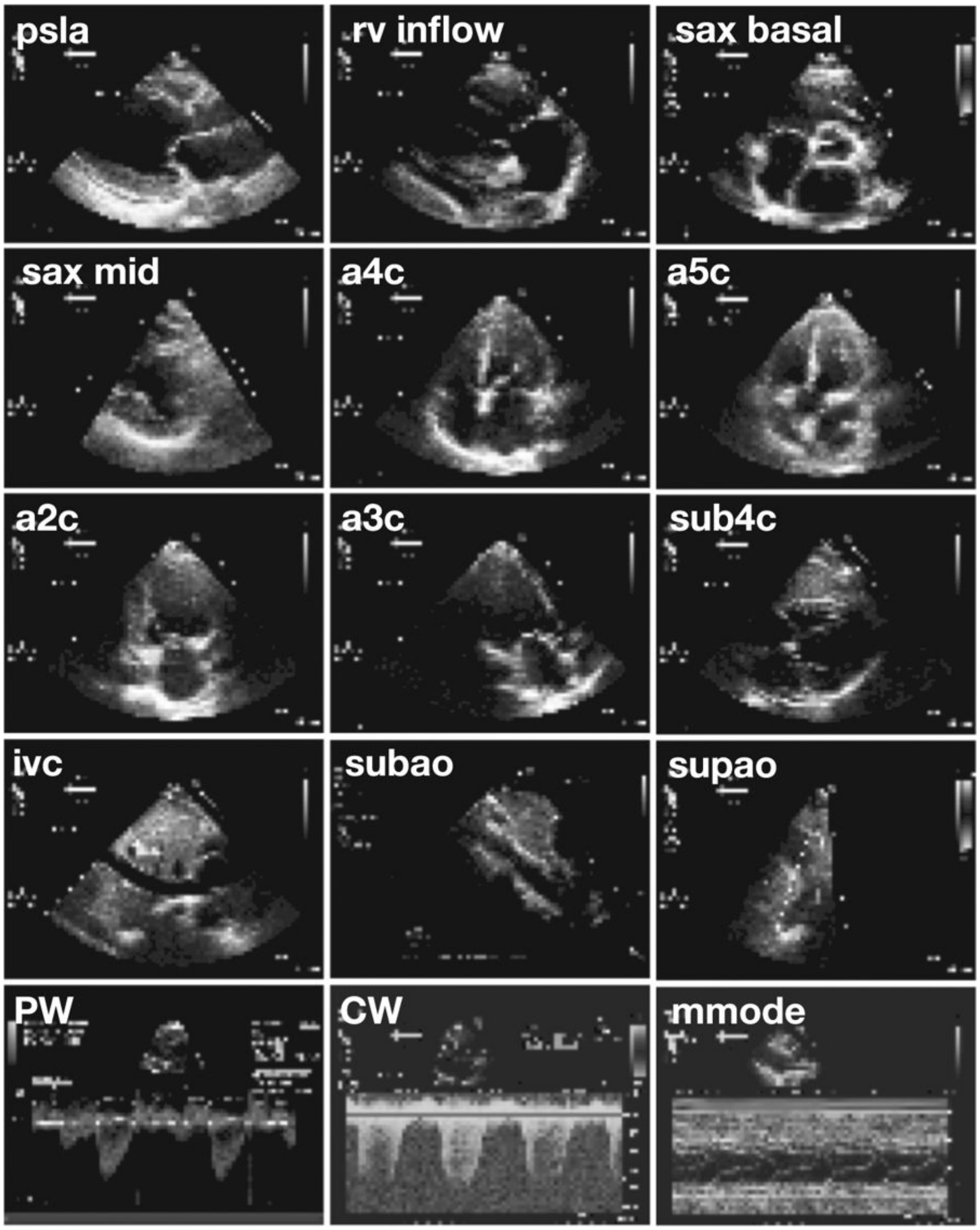

Figure 2.2: Sample input images. Views classified included parasternal long axis (psla), right ventricular inflow (rv inflow), basal short axis (sax basal), short axis at mid or mitral level (sax mid), apical four-chamber (a4c), apical five chamber (a5c), apical two chamber (a2c), apical three chamber/apical long axis (a3c), subcostal four-chamber (sub4c), subcostal inferior vena cava (ivc), subcostal/abdominal aorta (subao), suprasternal aorta/aortic arch (supao), pulsed-wave Doppler (PW), continuous-wave Doppler (CW), and m-mode (mmode). Note that these images are the actual resolution of input data to the deep learning algorithm

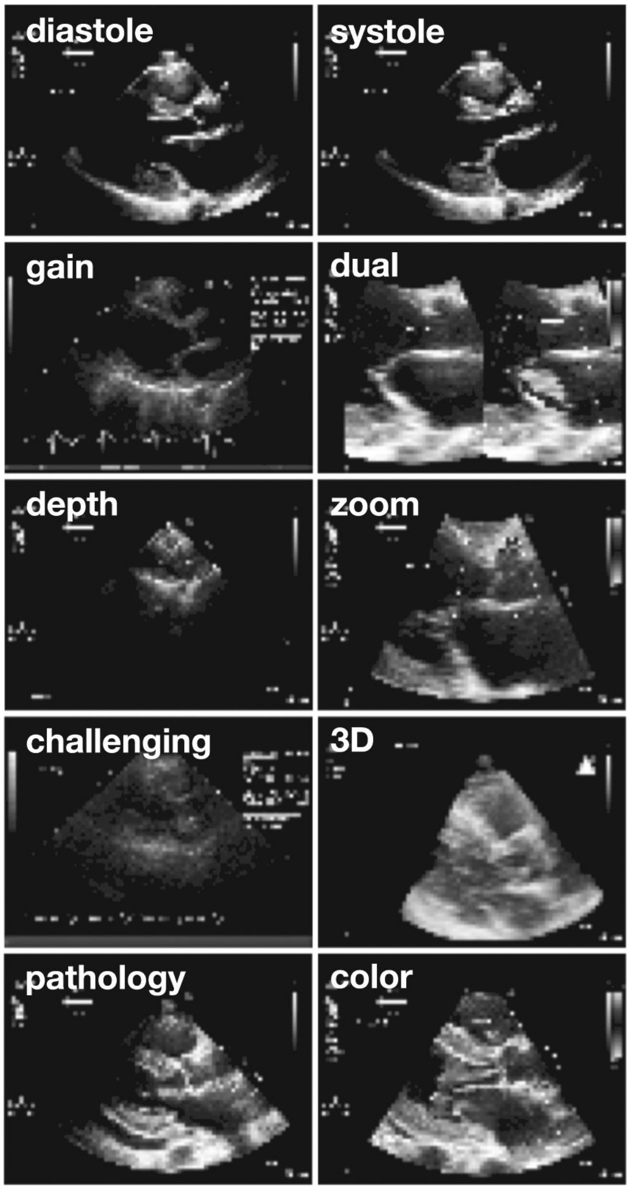

Figure 2.3: Natural variations in input data. In addition to applying data augmentation algorithms, we included in each category a range of images representing the natural variation seen in real-life echocardiography. The parasternal long-axis view is shown here for example. Variations include a range of timepoints spanning diastole and systole, differences in gain or chroma map, use of dual-mode acquisition, differences in depth and zoom, technically challenging images, use of 3D acquisition, a range of pathologies (seen here, concentric left ventricular hypertrophy and pericardial effusion), and use of color Doppler, as well as differences in angulation, sector width, and use of LV contrast. Note that these images are the actual resolution of input data to the deep learning algorithm

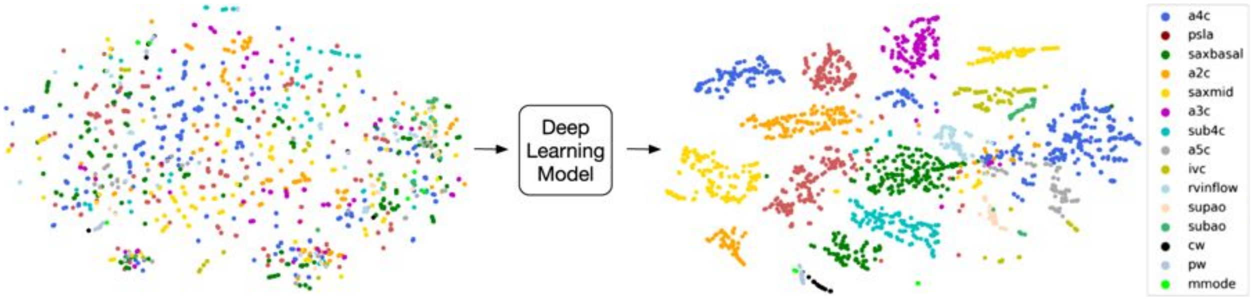

Figure 2.4: Deep learning model simultaneously distinguishes among 15 standard echocardiographic views. We developed a deep-learning method to classify among standard echocardiographic views, represented here by t-SNE clustering analysis of image classification. On the left, t-SNE clustering of input echocardiogram images. Each image is plotted in 4800-dimensional space according to the number of pixels, and projected to two-dimensional space for visualization purposes. Different colored dots represent different view classes (see legend in figure). Prior to neural network analysis, input data does not cluster into clear groups. On the right, data as processed through the last fully connected layer of the neural network are again represented in two-dimensional space, showing organization into clusters according to view category. Abbreviations: a4c apical 4 chamber, psla parasternal long axis, saxbasal short axis basal, a2c apical 2 chamber, saxmid short axis mid/mitral, a3c apical 3 chamber, sub4c subcostal 4 chamber, a5c apical 5 chamber, ivc subcostal ivc, rvinflow right ventricular inflow, supao suprasternal aorta/aortic arch, subao subcostal/abdominal aorta, cw continuous-wave Doppler, pw pulsed-wave Doppler, mmode m-mode recording

can be confused for apical two-chamber) and apical four-chamber (vs. apical five-chamber), or views in which multiple view-defining structures can be seen in the same image, such as subcostal IVC vs. subcostal four-chamber. As expected, training on randomly labeled still images achieved an accuracy (6.9 percent) commensurate with random guessing (6.7 percent, the probability of guessing the correct one out of 15 views by chance).

Model classification is based on cardiac image regions

To understand whether classification is based on clinically relevant features, such as heart chambers and valves, or on confounding or statistical features that might be clearer to a machine than a human, such as fiducial markings, border regions, or fraction of white pixels, we performed occlusion experiments by measuring prediction performance on test images on which we masked clinically relevant features with different shapes. Overall test accuracy fell significantly with masking of the heart but not other parts of the image, consistent with this region being important to the model (Figure 2.6a). In addition, saliency mapping, which identifies the input pixels that are most important to the models assignment of a particular classification, revealed that structures that would be important to defining the view to a

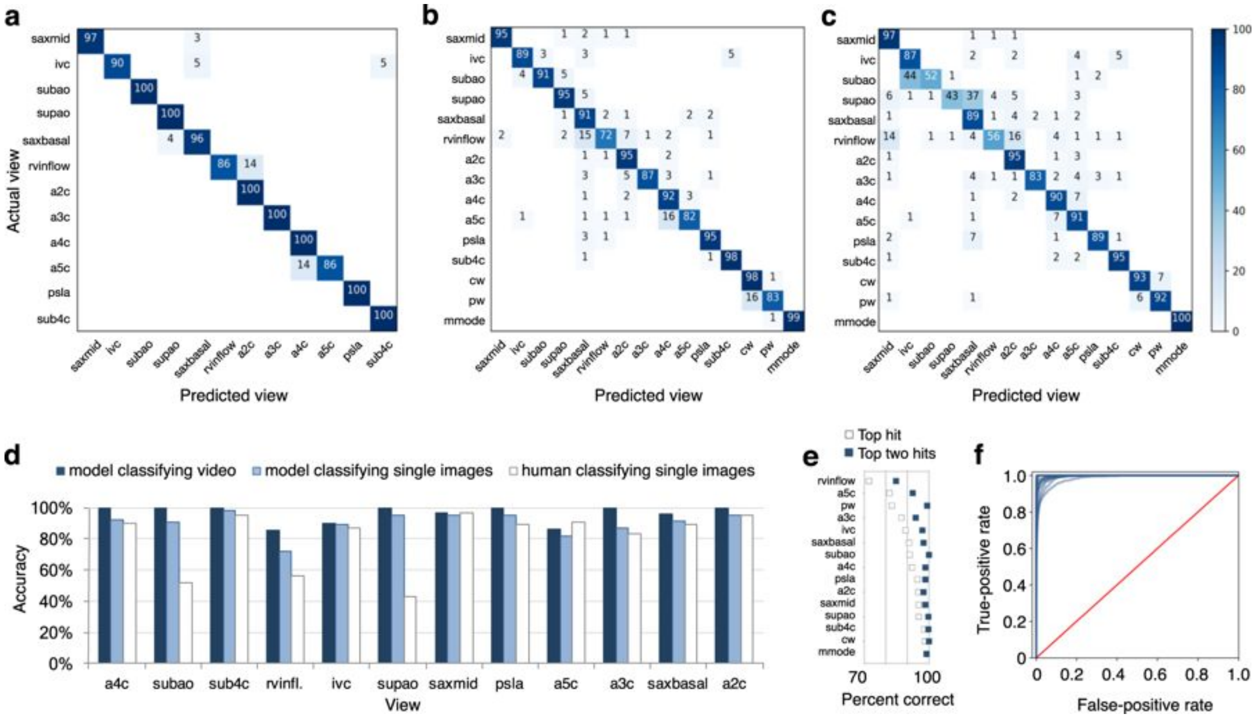

Figure 2.5: Echocardiogram view classification by deep-learning model. Confusional matrices showing actual view labels on y-axis, and neural network-predicted view labels on the x-axis by view category for video classification (a) and still-image classification (b) compared with a representative board-certified echocardiographer (c). Reading across true-label rows, the numbers in the boxes represent the percentage of labels predicted for each category. Color intensity corresponds to percentage, see heatmap on far right; the white background indicates zero percent. Categories are clustered according to areas of the most confusion. Rows may not add up to 100 percent due to rounding. d Comparison of accuracy by view category for deep-learning-assisted video classification, still-image classification, and still-image classification by a representative echocardiographer. e A comparison of percent of images correctly predicted by view category, when considering the models highest-probability top hit (white boxes) vs. its top two hits (blue boxes). f Receiver operating characteristic curves for view categories were very similar, with AUCs ranging from 0.985 to 1.00 (mean 0.996). Abbreviations: saxmid short axis mid/mitral, ivc subcostal ivc, subao subcostal/abdominal aorta, supao suprasternal aorta/aortic arch, saxbasal short axis basal, rvinflow right ventricular inflow, a2c apical 2 chamber, a3c apical 3 chamber, a4c apical 4 chamber, a5c apical 5 chamber, psla parasternal long axis, sub4c subcostal 4 chamber

human expert were also the ones that contributed most to the models classification (Figure 2.6b).

2.3 Discussion

View classification is the essential first step in interpreting echocardiograms. Previous attempts to use machine learning to assist with view classification required laborious manual annotation, failed to distinguish among more than a few views at a time, used only textbook-quality images for training, exhibited low accuracy, or were tied to a specific equipment vendor, limitations unsuitable for general practice. [49, 52, 74, 77, 92, 27, 79] In contrast, we report here a single, vendor-agnostic deep-learning model that correctly classifies all types of echocardiogram recordings (b-mode, m-mode, and Doppler; still images and videos) from all acquisition points relevant to a full standard transthoracic echocardiogram (parasternal, apical, subcostal, and suprasternal), at accuracies that exceed those of board-certified echocardiographers given the same task. Furthermore, the echocardiograms used in this study were drawn randomly from real echocardiograms acquired for clinical purposes, from patients with a range of ages, sizes, and hemodynamics; for a range of indications; and including a range of pathologies, such as low left ventricular ejection fraction, left ventricular hypertrophy, valve disease, pulmonary hypertension, pericardial effusion. Training data also included the natural variation in echocardiographic acquisition of each view, including variations in technical quality. By avoiding limited or idealized training subsets, our model is broadly applicable to clinical practice, although of course a larger training set would likely capture still more echocardiographic variability.

Because deep networks like CNNs usually include large numbers of (highly correlated) parameters (which describe the weights of connections among the nodes in the network), it is usually difficult to understand a models decision-making by simple inspection. For life-or-death decisions, such as in medicine or self-driving cars, this issue can breed suspicion and has legal ramifications that can slow adoption. Occlusion testing and saliency mapping help address these concerns by getting inside the black box. In our model, these techniques show that classification depends on the same features that echocardiographers use to reach their conclusions. For example, the maps shown in Figure 2.6b for a short-axis-mid view and a suprasternal aorta view, respectively, each trace the basic outlines of their corresponding input view. In the future, applying these approaches to intermediate layers may prove interesting to more precisely define the similarities, or differences, in how humans and models move from features to conclusions. For now, it is reassuring that our model considers the same features that human experts do in classifying views.

This similarity also explains the occasional misclassifications of single images, which most often involved views that can look similar to human eyes (Figures 2.2e, f, g, h, j, k and 2.5). These include adjacent views in echocardiographic acquisition, where a slight difference in the angle of the sonographers wrist can change the view, resulting in confusion of an apical three-chamber view for an apical two-chamber view or an apical five-chamber for apical four-

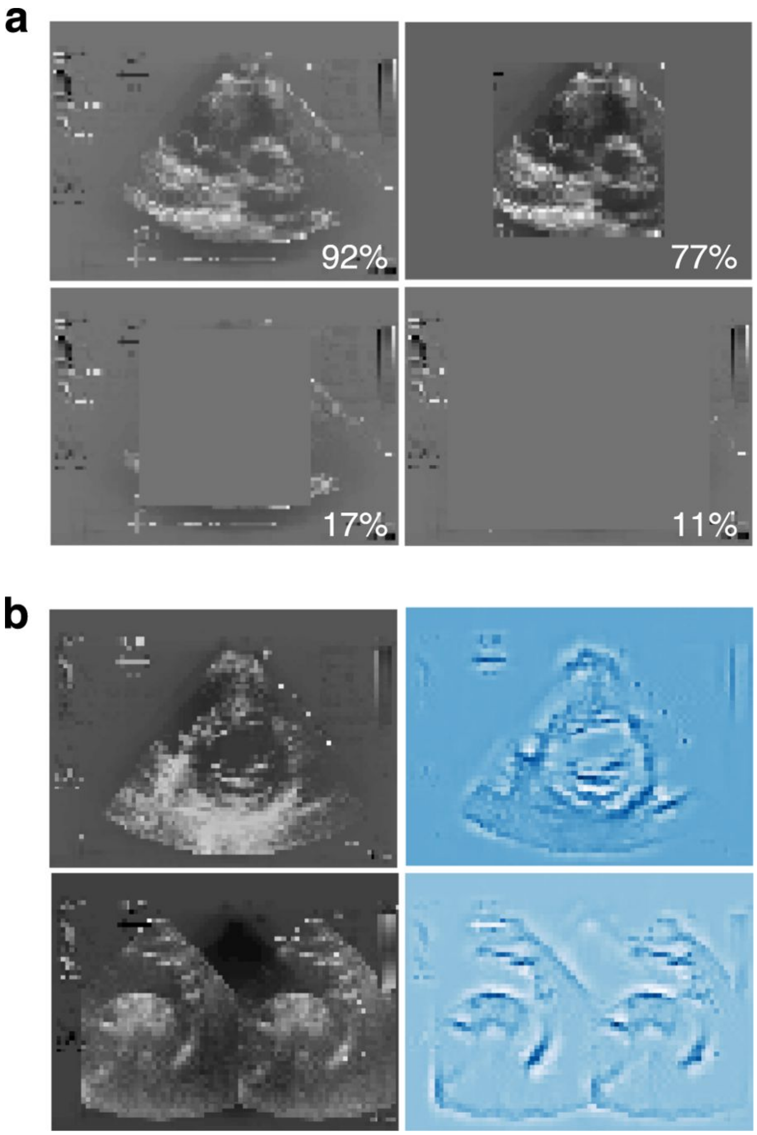

Figure 2.6: Visualization of decision-making by neural network. a Occlusion experiments. All test images (a short axis basal sample image is shown here) were modified with grey masking of different shapes and sizes as shown, and test accuracy predicted for the test set based on each different modification. Masking that covered cardiac structures resulted in the poorest predictions. b Saliency maps. The input pixels weighted most heavily in the neural networks classification decision for two example images (left; suprasternal aorta/aortic arch and short axis mid/mitral input examples shown) were calculated and plotted. The most important pixels (right) make an outline of structures clinically relevant to the view shown

chamber; as well as views in which two view-defining structures may be seen in the same image, such as the IVC seen in a subcostal four-chamber view. A low-velocity PW signal can look similar to a faint CW signal. In fact, the only misclassification made by our model without an obvious explanation of this sort was that of the right ventricular inflow view for short-axis basal; of note, the right ventricular inflow view was also very challenging for human echocardiographers to distinguish (with 5157 percent accuracy). We note in the confusion matrices that misclassification of certain views for one another was non-symmetrical; for example, PW images were confused with CW, but CW images were almost never mistaken for PW (Figure 2.5b). In this case, as mentioned above, this asymmetry makes clinical sense; however, more training and test data can be used to explore this phenomenon further and refine accuracies for these categories. Because classification of videos is based on multiple images, and error decays exponentially with the number of images, misclassification of videos was very rare (2 percent; Figure 2.5d). We also noted that the models confidence in its choice (the probability assigned to a view classification for a particular image) affected performance; where confidence was higher, accuracy was also higher (Figure 2.7). Therefore, communicating the models confidence for each classification should further benefit users.

Finally, our approach had two unexpected advantages related to efficiency, practicability, and cost-effectiveness. First was the perhaps surprising effectiveness of a simple majority vote in classification of videos. Video analysis can be a complex undertaking that involves non-trivial tasks, such as frame-to-frame color variation and object tracking. We have demonstrated that view classification, at least, can be done much more efficiently and cost-effectively, reducing coding and training time. Moving beyond view classification, it will be interesting to see what other clinically actionable information can be extracted from (collections of) still images. Second, in removing color and in standardizing the sizes and shapes of videos and still images for training, we discovered that we could downsamplei.e., shrinkimages appreciably without losing accuracy. This allowed for a 9699 percent savings in file size (vs. 300-by-400- to 1024-by-768-pixel images; Figure 2.8), and corresponding gains in the cost and speed of training and of classifying new samples at deployment. While human echocardiographers routinely classify views, they appear to require full-resolution, native video data to do so with high accuracy. With less input data, the model outperformed overall human accuracy (and speed: 32s vs. hours to classify the same 1500-image test sample). We note the potential implications for telemedicine and global health, including in resource-poor regions of the United States, of requiring storage and transmission of smaller files (though decentralized use of the model can also come through transmission of the model, which is a small file), and of embracing older echo machines that may image with lower resolution.

Echocardiography is essential to diagnosis and management for virtually every cardiac disease. In this study, we have demonstrated the application of deep learning to echocardiography view classification that classified 15 major TTE views with expert-level quality. We purposely used a training set that reflected a wide range of clinical and physiological variations, demonstrating applicability to real-world data. We found that our model uses some of the same features in echocardiograms that human experts use to make their decisions.

Looking forward, our model can be expanded to classify additional sub-categories of echocardiographic view (e.g. to distinguish among different CW, PW, and m-mode acquisitions), as well as diseases, work that has foundational utility for research, for clinical practice, and for training the next generation of echocardiographers.

2.4 Methods

Dataset

All datasets were obtained and de-identified, with waived consent in compliance with the Institutional Review Board (IRB) at the University of California, San Francisco (UCSF). Methods were performed in accordance with relevant regulations and guidelines. Two-hundred sixty-seven echocardiographic studies from different patients and performed between 2000 and 2017 were selected at random from UCSFs clinical database. These studies included men and women (49.4 and 50.6 percent, respectively) ages 2096 (median age, 56; mode, 63) with a range of body types (25.8 percent obese), which can affect technical quality of TTE (Table 2.1), and included indications and pathologies that are representative of the uses of echocardiography in current clinical practice (Table 2.2). Studies were carried out using echocardiograms acquired with equipment from several manufacturers (e.g., GE, Philips, Siemens).

Data Pre-processing

DICOM-formatted echocardiogram videos and still images were stripped of identifying metadata, anonymized by zeroing out all pixels that contained identifying information, labeled by view by a board-certified echocardiographer with access to native-resolution and video data, then split into constituent frames and converted into standardized 6080-pixel monochrome images, resulting in 834,267 images. Fifteen views were selected for multi-category classification, covering the majority used in the field. Views classified included parasternal long axis, right ventricular inflow, basal short axis (aortic valve level), short axis at mid (papillary muscle) or mitral level, apical four-chamber, apical five chamber, apical two chamber, apical three chamber (apical long axis), subcostal four-chamber, subcostal inferior vena cava (IVC), subcostal abdominal aorta, suprasternal aortic arch, pulsed-wave Doppler, continuous-wave Doppler, and m-mode. For the purposes of this study, CW Doppler, PW Doppler, and m-mode recordings from different acquisition points were considered part of the same view, e.g. m-mode of the aortic valve, mitral valve, left ventricle, and right ventricular annulus were all considered part of the m-mode view. For each view, we included images with a range of natural echocardiographic variation, such as differences in zoom, depth, focus, sector width, gain, chroma map, systole/diastole, angulation, image quality, and use of 3D, color Doppler, dual mode, strain, and left-ventricular (LV) contrast, to capture the range of variation normally seen by echocardiographers.

Demographics	**Study**	**sample**		**Clinical**	**Echo**	**Database**^{a}	
	Mean	**SD**	**IQR**	**Mean**	**SD**	**IQR**	**p**^{b}
Age	56.1	16.6	22.5	58.5	16.8	23	0.5
Height	170	11.6	16.5	169	11	17.8	0.8
Weight	77	20.5	31.5	77	22	26.3	0.9
Systolic BP	127	19	20.3	126	22	28	1
Diastolic BP	70	123	13.3	70	12	17	0.5
MAP	88.9	13.4	18.3	88.6	13.9	13.8	0.7
BSA	1.87	0.27	0.44	1.82	0.56	0.37	0.8
BMI	26.6	6.1	10.2	27.1	6.8	7.4	0.9

Demographics	**%**	**N**	**Sample size**	**Percent**	**N**	**Sample size**	**p**^{b}
Female	50.6	135	267	49.5	79,460	159,503	0.7
Male	49.4	132	267	50.5	80,043	159,503	0.7
Obese	25.8	69	267	25.1	25,770	102,669	0.8

Pathology	**%**	**N**	**Sample size**	**Percent**	**N**	**Sample size**	**p**^{c}
LVMI>normal	32.8	67	204	39.2	34,056	86,878	0.06
LVEF<55%	21.7	58	267	20.3	18,432	90,798	0.6
LVEDVI>normal	16.9	45	238	46.8	10,677	90,375	0.3
RVSP>40mmHg	10.9	29	267	14.6	10,774	73,795	0.09
TAPSE<1.6cm	7.84	8	102	10.6	1768	16,679	0.4

Table 2.1: Patient Characteristics Table - Comparison of study sample characteristics to clinical echo database. Age (years), Height (cm), Weight (kg), BP blood pressure (mmHg), MAP mean arterial pressure (mmHg), BSA body surface area (m^2), BMI body mass index (kg/m^2), LVMI left ventricular mass index (g/m^2) adjusted for sex, LVEF left ventricular ejection fraction, LVEDVI left ventricular end-diastolic volume index (ml/m2) adjusted for sex, RVSP right ventricular systolic pressure, TAPSE tricuspid annular plane systolic excursion, SD standard deviation, IQR interquartile range. a- p-value. Ns and sample size vary according to availability of different measurements b- p-value. Two-tailed Students t-test, unequal variance c- p-value. Chi-squared test for comparison of proportions

A subset of 223,787 images from 15 views were randomly split using Python into training, validation, and test datasets in approximately an 80:10:10 ratio. Each dataset contained images from separate echocardiographic studies, to maintain sample independence. The number of images in training, validation, and test datasets were 180,294, 21,747, and 21,746 images, respectively (corresponding to 213, 27, and 27 different studies in each set). The validation dataset was used for model selection and parameter fine-tuning. The test dataset was used for performance evaluation of the final trained and validated model. For training, 256-shade greyscale pixel values were scaled from [0,255] to [0,1] and the mean over the training data was subtracted from each dataset, as is standard in image-recognition tasks. Also as per standard practice, data were augmented at run-time by randomly applying rotations of up to 10 degrees, width and height shifts of up to a tenth of total length, zooms of up to 0.08, shears of up to 0.03, and vertical/horizontal flips. Training and validation datasets in which view labels were randomized were used as a negative control.

Model architecture and training

Our neural network architecture was designed in Python using the Tensorflow, Theano, and Keras packages, drawing inspiration from the VGG-16 network, which won the Imagenet challenge in 2014. [1, 15, 96, 86] Our model utilized a series of small 33 convolutional filters connected with max-pooling layers over 22 windows. Dropout was utilized in training for both the convolutional and fully connected layers to prevent overfitting. In addition to dropout for regularization, batch normalization was used before neuron activations, which led to faster training and increased accuracy. Activation functions were mainly rectified linear units (ReLU) with the exception of the softmax classifier layer. Training was performed over 45 epochs using an adaptive learning-rate decay for RMSprop optimization. k-fold cross-validation (k=9) was used to randomly vary which images were in the training and validation sets, to make use of all available data for training and to select the optimal weights at each epoch. Batches of 64 samples at a time were used for gradient calculation. Convergence plots of training and validation accuracy by epoch confirmed that the model was not overfitting. The training method was robust, with three separate trainings of the 223,787 images resulting in overall test accuracies above 97 percent. Training was performed on Amazons EC2 platform with a GPU instance g2.2xlarge and took about 18h. Testing was performed on a laptop computer (Intel i5-3320M CPU @ 2.60GHzx4 with 16GB RAM); it took a total of 32s to predict 1500 images, yielding an average of 21ms per image. Code availability: VGG-16 is publically available on Github.

Model evaluation

Several metrics were used over the test dataset for performance evaluation. Overall accuracy was calculated as the number of correctly classified images as a fraction of the total number of images. Average accuracy was calculated as the average over all views of per-view accuracy. F-score was calculated in standard fashion as twice the harmonic mean of precision (positive predictive value) and recall (sensitivity). Receiver operator characteristic (ROC) curves were plotted in the standard way as true-positive fraction (y-axis) against false-positive fraction (x-axis) and the associated area under curve (AUC) was calculated. Confusion matrices were calculated and plotted as heat maps to visualize performance of multi-view classifiers and their associated errors. Single test images were classified according to the view with the highest probability. Test videos were classified by simple majority vote on multiple images from a given video.

The basis for the models classification decisions was explored using t-distributed stochastic neighbor embedding (t-SNE) dimensionality reduction [62] of raw pixels and of the last fully connected layer output for each sample. Occlusion experiments were performed by masking test images with bounding boxes of different shapes, then submitting them to the model for label prediction. Saliency maps were created using guided backpropogation, which keeps the model weights fixed and computes the gradient of the models output for a given image.

Comparison to human experts

Echocardiogram test-image classification by board-certified echocardiographers was approved by the UCSF Human Research Protection Program and Institutional Review Board. Each board-certified echocardiographer gave informed consent and was given a randomly selected subset of 1500 60-by-80 pixel images, 100 of each view, drawn from the same low-resolution test set given to the model, and performance compared using the relevant metrics above.

Study sample indication	**Percent**	**N**
Heart failure/cardiomyopathy	24	64
Arrhythmia	11.6	31
Chemotherapy	10.9	29
Valve disease	10.5	28
Preoperative exam	7.9	21
Dyspnea	6.4	17
Coronary artery disease	6	16
Stroke	6	16
Syncope	5.2	14
Rule out endocarditis	4.9	13
Pulmonary HTN	4.5	12
Hypertension	3.7	10
Pericardial effusion	3.4	9
Murmur	3	8
Palpitations	3	8
Aortic aneurysm	2.6	7
Congenital heart disease	2.6	7
Lung disease	1.9	5
Edema	1.5	4
Hypotension	1.5	4
Cardiac arrest	0.4	1
Heart transplant	0.4	1

Number of normal studies[a]	**Percent**	**N**
Normal studies	10.9	29

Table 2.2: Indications for study sample echocardiograms. a- Defined by echo reports documenting normal four-chamber size and systolic/diastolic function, no chamber hypertrophy or wall motion abnormalities, normal valves with trace or less regurgitation, normal great vessels and estimated right atrial pressure, no pericardial effusion, RVSP<=40, and no other abnormalities, such as atherosclerosis, calcification, pleural effusion, ascites, prostheses, or catheters

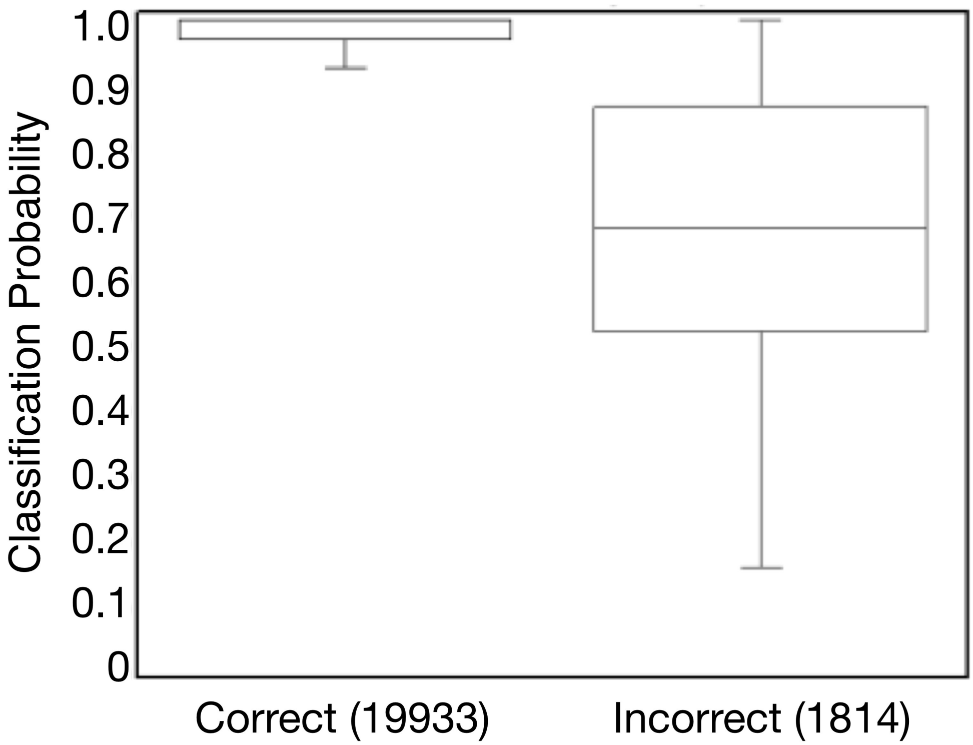

Figure 2.7: Confidence for first and second guesses on image classification. Box plot summarizing probabilities assigned to correct and incorrect images in the test set using the single highest probability to classify the test image. Confidence for correct answers was higher than for incorrect answers. Median, interquartile range for correct (0.999, 0.970-1.00) and incorrect (0.682, 0.525-0.867) answers.

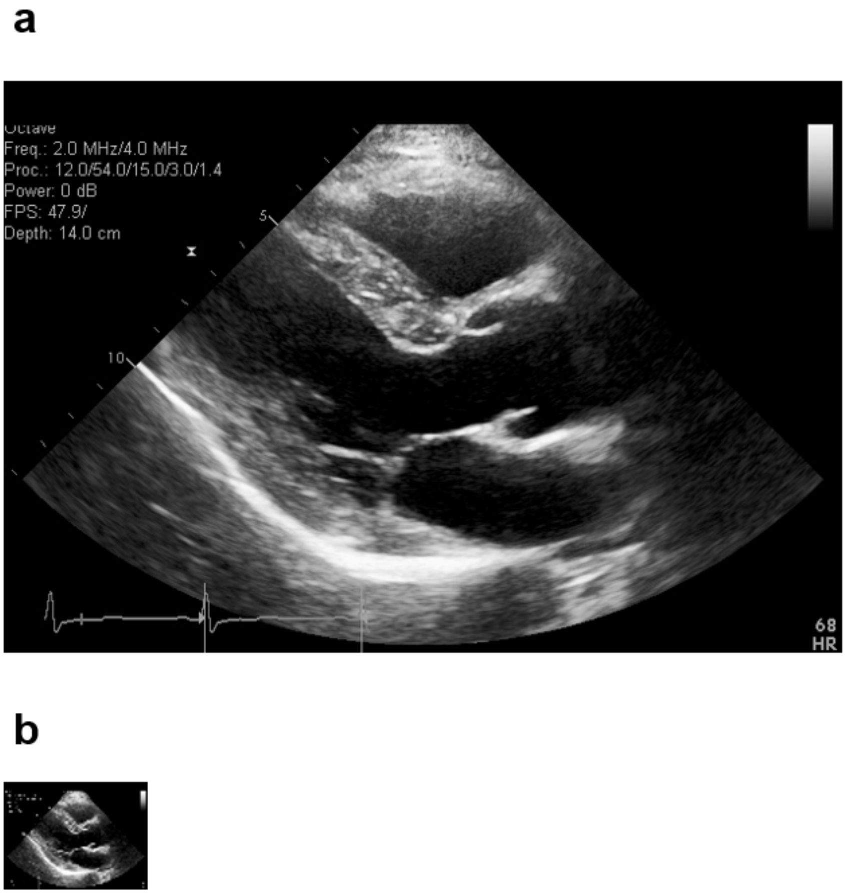

Figure 2.8: Example native-resolution and downsampled images. (a) Native echocardiographic images ranged from 300x400 to 768x1024 pixels in resolution, and many contained color, either as color Doppler or as different chroma maps to aid in visualization. (b) The same sample image as in (a), downsampled to the 60-by-80-pixel resolution used as input to the deep learning model.

Chapter 3

Data-Efficient Supervised and Semi-supervised Learning Towards Automated Diagnosis

3.1 Introduction

With the improved quality and accessibility of both medical imaging equipment and effective healthcare policy, medical imaging has become an increasingly critical step in modern healthcare diagnostics and procedures. Interpretation of medical imagery requires specialized training and is a time-intensive process. Machine learning and computer vision techniques provide an avenue to augment insights, improve accuracy, and optimize workload time for interpretation. Traditional machine learning techniques in medical imaging involve matching of features hand-engineered by domain experts, a laborious process with limited scope and effectiveness [98, 22]. Recent advances in deep learning [54, 97, 55], a data-driven approach, and the increasing accessibility of powerful graphical processing units (GPUs) [17, 3] have made the automation of image-based diagnosis insights possible. Researchers have succeeded in applying deep learning techniques in radiology, cardiology, and dermatology [61, 67, 64], including detecting pneumonia from chest X-rays [82] and classifying images of benign versus malignant skin lesions [24].

While deep learning holds great promise in automating the task of medical diagnosis, there remains a set of unique challenges to be resolved before it can be deployed in practice at scale [61]. Specifically, deep learning algorithms require massive amounts of labeled data to achieve human-level classification performance. Due to privacy laws, differing standards across the healthcare industry and the lack of medical system integration, medical data is less available compared to other fields of computer vision. In addition, medical datasets suffer from class imbalances as certain conditions occur much less frequently than others. Labeling of medical images is complex and requires the time of medical professionals, making it significantly more expensive compared to other computer vision tasks. In addition, there is a limit to the effectiveness of current algorithms in processing high-resolution medical images. As a result, there is a need to identify an optimal balance between resolution size and computational burden. Lastly, medical images often contain other metadata that may be irrelevant for the classification task, leading to less than optimal performance from deep learning models that are unable to filter out this extra information [49].

In this paper, we address the above challenges by developing data-efficient training methodologies in the domain of transthoracic echocardiograms (TTEs) classification, the most ubiquitous, versatile, and cost-effective cardiac imaging modality available [76]. Currently, studies have shown surprising rates of echocardiographic assessment inaccuracy can be up to 30% of echo reports [45] and echocardiographic quality inadequacy in 24% of imaging studies. Automating interpretation of echocardiograms with trained deep learning classifiers can significantly lower cost, improve quality, and augment cardiologists in making faster and more accurate diagnoses. TTE consists of video clips, still images, and doppler measurements recorded from over a dozen different viewing orientations [116]. Determining the view is the essential initial step in interpreting an echocardiogram and establishing quality of recorded medical imaging. This step is challenging due to subtle inter-view differences and the variability within a single view [49]. In addition, from different view orientations further symptoms can be visually identified, such as the enlargement of chambers and thickening of ventricular walls before final prognosis. Left ventricular hypertrophy (LVH), a condition commonly associated with heart disease and hypertension, can be identified by the thickness and relative size of the left ventricle from echocardiograms [75].

Previously [66], we explored supervised learning techniques for effective classification of echocardiogram view orientations, reporting a single image classification accuracy of 91.7% over 15 view classes. View classification is an important initial step in terms of a proof-of-concept to gain confidence that deep learning networks are performing accurately by learning relevant visual features, on-its-own as an assistant to sonographers for quality assurance, and as a precursor toward prediction of clinical diagnoses; as if an algorithm can learn what anatomical structure it is looking at, it can conceivably then learn patterns in what is defined as normal vs abnormal. We expand the previous work [66] by examining both supervised and semi-supervised techniques for classification of view orientations, segmentation of relevant structures in echocardiogram images, and classification of left ventricular hypertrophy with the goal of automating cardiovascular disease prediction as captured in figure 3.1. Our expansion of work provides an expansion of the depth and breadth of both algorithms and applications in deep learning for echocardiography and medical imaging as a whole. We train convolutional neural networks (CNNs) on input resolutions of varying sizes and found an optimal balance between classifier performance and computational burden at 120x160 pixels. Performance of CNNs trained with default weight initialization were compared with transfer learning from Resnet50 [37] and VGG16 [96], models that were pre-trained on the ImageNet [20, 86] dataset.

We report an accuracy of 94.4% on the same test set, using an ensemble of CNNs with a single U-Net [84] for field of view segmentation. We applied the same techniques on a limited training dataset for single image LVH classification, obtaining a test accuracy of 91.2%. We show that generative adversarial networks (GANs) [33], adapted for semi-supervised learning, can achieve better results than conventional CNNs in settings where labeled data is limited, achieving a test accuracy of 92.3% for LVH classification. Together, we present a working CNN system capable of accurately classifying left ventricular hypertrophy from a single echocardiogram image and a GAN system for automating disease predictions in data-limited settings.

3.2 Results

An optimum balance exists between resolution size and computational burden

We examined the effect of resolution size on accuracy performance and computational time. Ideally, there is an optimal point where the resolution is efficiently small without losing excessive structural information. For a subset of the echocardiographic data, we observed that with our VGG16-like architecture, validation accuracy plateaus at sub-88% accuracy for resolution above 60x80. As shown in figure 3.2 and table 3.1, computation time scales with increasing input resolution. For convolution layers, we observed that filter sizes of 3x3, same padding, stride of 1 pixel, and pooling size of 2x2 works well generally across all resolutions, while maximum pooling works better than average pooling layers for resolutions above 60x80 pixels.

Resolution	Training Accuracy	Validation Accuracy	Training Time / Epoch
240x320	94.49%	87.92% (+/- 0.06)	781s
120x160	98.43%	87.82 (+/- 0.02)	226s
60x80	99.10%	87.04% (+/- 0.05)	104s
30x40	93.57%	82.30% (+/- 0.01)	79s
15x20	84.69%	73.36% (+/- 0.02)	55s

Table 3.1: Table of evaluation results for resolution study. Validation accuracy plateaus at sub-88% accuracy while training and testing time continue to scale with increasing input resolution.

With only a few samples, custom deep learning segmenters can be trained to identify focus areas

Learning segmentation maps allows for focusing only the relevant pixels for subsequent classification. Our trained segmentation models are able to achieve satisfactory results with minimal labeling. The segmentation results for both Field of View and LVH segmentation are shown in figure 3.3. For View Segmentation, our model reported a pixel-wise cross-entropy

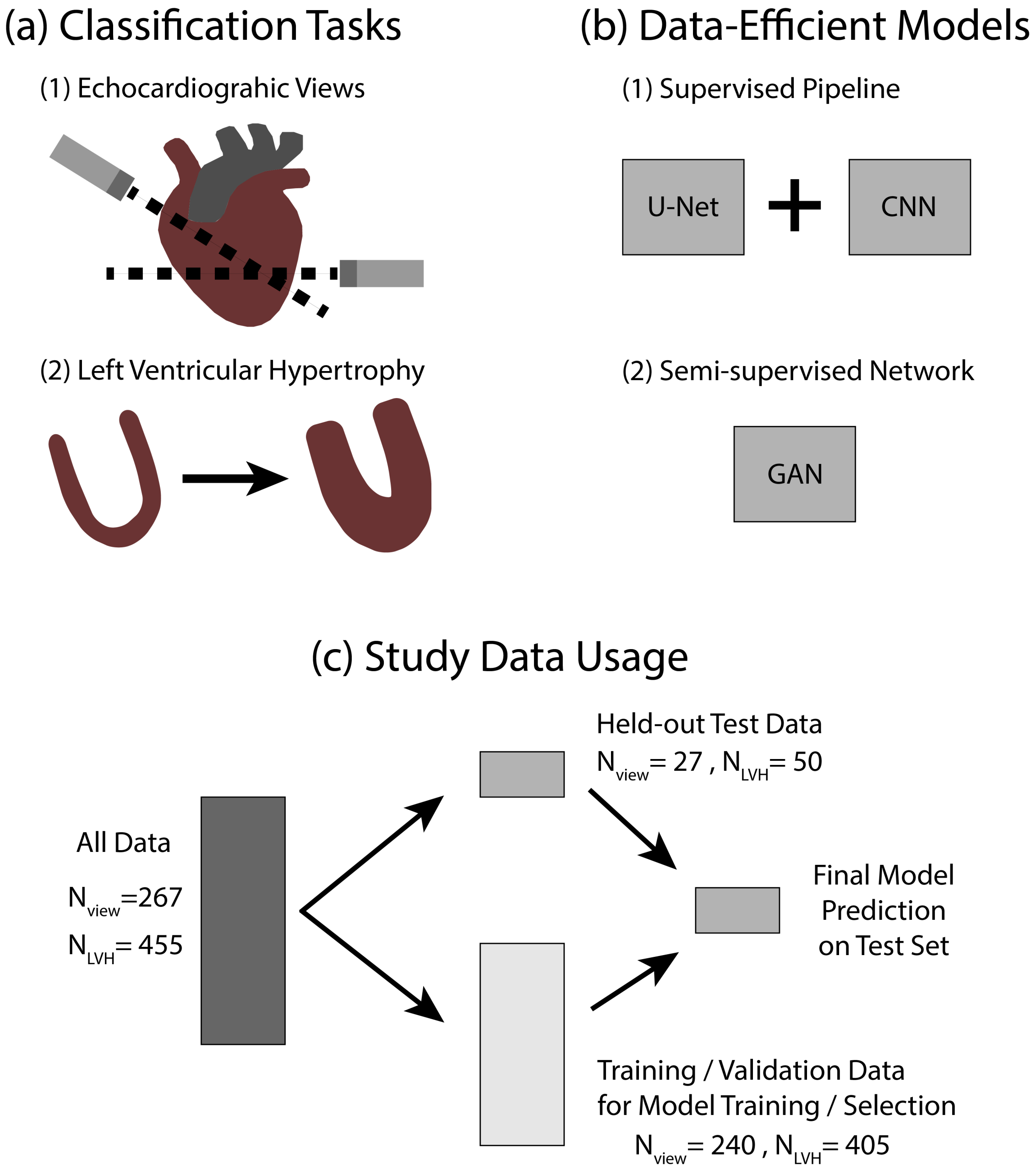
(a) Classification Tasks
(1) Echocardiograhic Views
(2) Left Ventricular Hypertrophy
(b) Data-Efficient Models
(1) Supervised Pipeline
U-Net
+
CNN
(2) Semi-supervised Network
GAN
(c) Study Data Usage
All Data
N_{view}=267
N_{LVH}= 455
Held-out Test Data
N_{view}= 27 , N_{LVH}= 50
Final Model Prediction on Test Set
Training / Validation Data for Model Training / Selection
N_{view}= 240 , N_{LVH}= 405

Figure 3.1: Deep learning for echocardiography study diagram. (a) Two classification tasks were examined for echocardiography: view classification and left ventricular hypertrophy (LVH) classification. (b) Two different approaches for deep learning models were taken: a supervised pipeline model that performs segmentation (U-Net) before classification (CNN) and a semi-supervised generative adversarial network (GAN) for end-to-end learning. (c) The data, for both view and LVH classification, was split accordingly by study and no test data was utilized in training or validating the model.

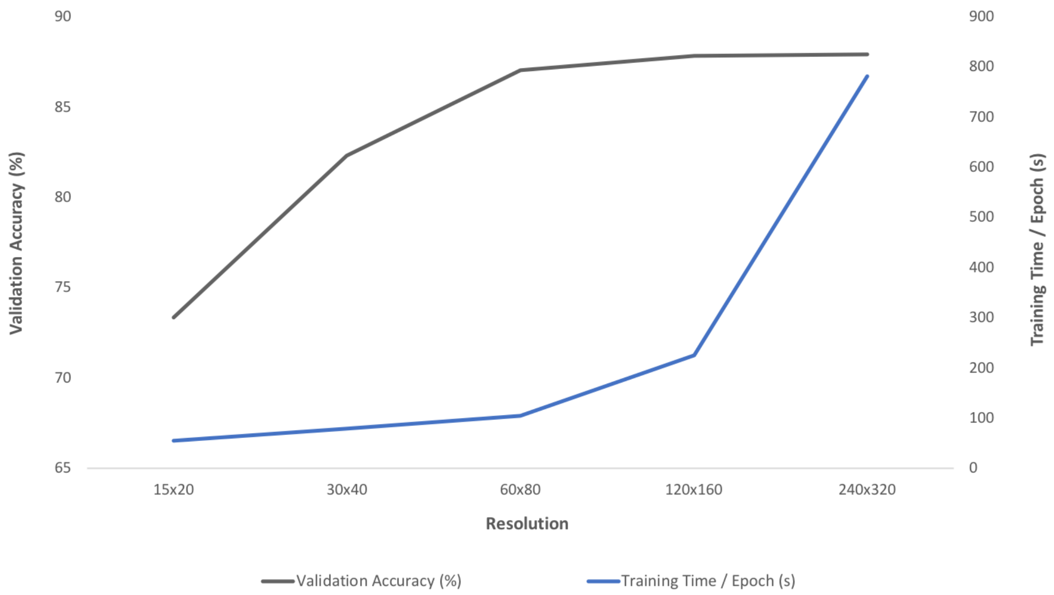

Figure 3.2: An optimal resolution size exists considering performance vs computational time tradeoffs. Plot of validation accuracy and training time per epoch with input resolution. The optimal balance between computational burden and validation accuracy exists at 120x160 resolution.

loss of 0.3984 on the test set of 32 images. From visually inspecting the segmentation on our test set, segmentation of single mode images matches very closely to the labeled map, while segmentation of volumetric and dual mode is less consistent.

For Left Ventricle Segmentation, our model reported a pixel-wise cross-entropy loss of 0.1926 on the test set of 50 images. From visually inspecting the segmentation on our test set, segmentation of left ventricle is less tight compared to the labeled segmentation maps. Segmentation of images in dual mode is less consistent. While images in dual mode were labeled with a single mask around the left view, the model predicted segmentation maps around the left ventricle in both views.

Field of view segmentation before classification achieves the highest reported accuracy for echocardiographic view classification

For echocardiographic 15-view classification, our CNN model without segmentation achieved an overall test accuracy of 92.05%. With the field of view segmentation prior to view classification pipeline as shown in figure 3.4, the same network architecture achieved an overall test accuracy of 93.64%. We found that average pooling layer outperforms maximum pooling layers for input images with Field of View segmentation. The best-performing model in

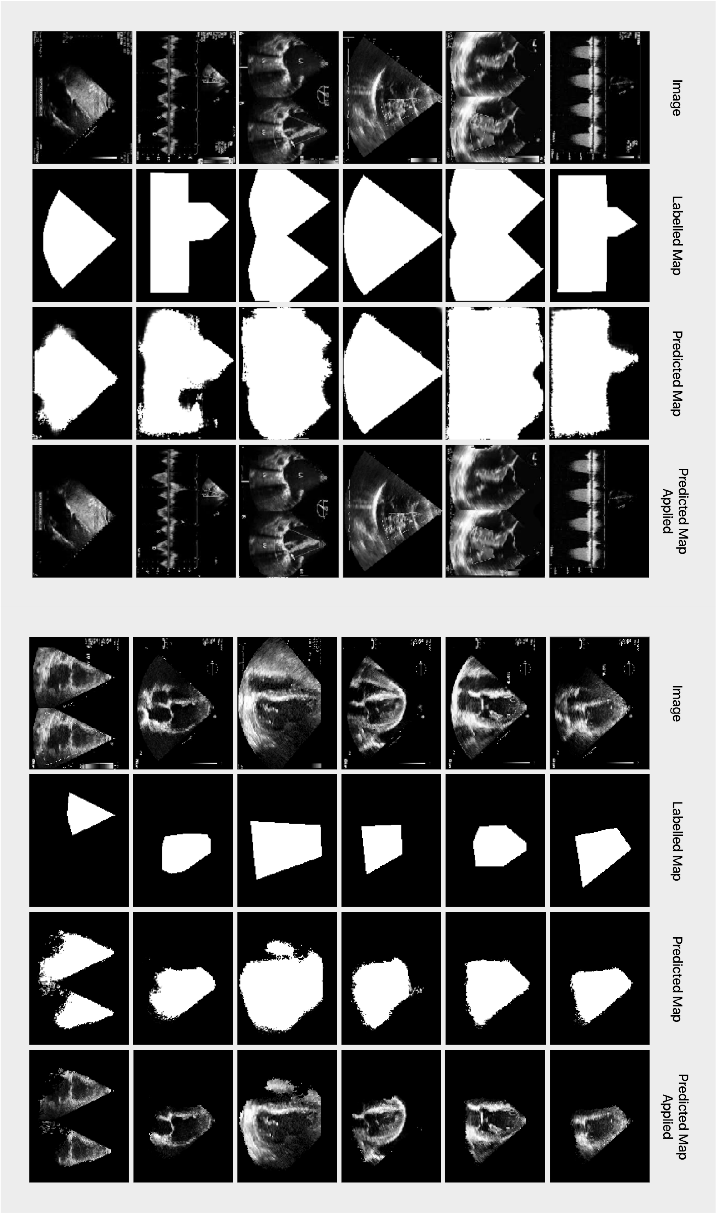

Figure 3.3: Figure of image data, labeled map, predicted map and predicted map applied to original image for Field of View (left) and LVH Segmentation (right). Trained segmentation models are able to accurately discern contours in echocardiogram images and output a map over relevant areas.

our experiment is an ensemble of 3 CNNs with Segmentation that achieved an overall test accuracy of 94.40%. The resulting performance of each model is tabulated in table 3.2. The normalized confusion matrix and accuracy by individual class for the ensemble is reported in figure 3.5. Both Resnet50 and VGG16 models achieved lower overall test accuracy of 91.36% and 83.67% respectively, despite having a more complex and deeper architecture.

As segmentation is performed in the preprocessing step, test and training time corresponds with the size of the network architecture, transfer learning with Resnet50 took more than 8 times the amount of time per epoch compared to the CNN models.

Model	Train Accuracy	Test Accuracy	Training Time / Epoch
CNN	95.39%	92.05% (+/- 0.72)	919s
CNN with Segmentation	95.18%	93.64% (+/- 0.61)	903s
Resnet50*	92.61%	91.36% (+/- 1.43)	7482s
VGG16*	98.15%	83.67% (+/- 2.68)	4559s (second stage)
Ensemble	-	94.40% (+/- 0.52)	-

Table 3.2: Table of evaluation results for each model for view classification. Highest performing model by overall test accuracy is an ensemble of three CNN models with field of view segmentation, while transfer learning from Resnet50 and VGG16 models yielded lower test accuracies compared to a single CNN model.

Left ventricle segmentation and transfer learning enables efficient classification of LVH using convolutional neural networks

For left ventricle hypertrophy classification, a pipeline model of segmenters and classifiers was developed as shown in figure 3.6. As shown in table 3.3, after the first stage training our convolutional neural network model with image segmentation pipeline achieved an overall test accuracy of 81.318% and F1 Score of 0.5952 on our test set of 182 images. We evaluated the model after the second stage training, and an overall test accuracy of 91.208% and an F1 Score of 0.8139 were measured. The normalized confusion matrix is reported in figure 3.7. CNN with segmentation outperformed our CNN network on both test accuracy and F1 score. In addition, we attempted training the CNN network with default initialization but the network failed to learn well and accuracy was very low.

Semi-supervised generative adversarial networks enable even higher classification accuracy for scenarios with large sets of unlabeled data.

The advantage of semi-supervised GANs is the utilization of all available data, whether labeled or unlabeled. For the view classification task, we studied the effect of training on

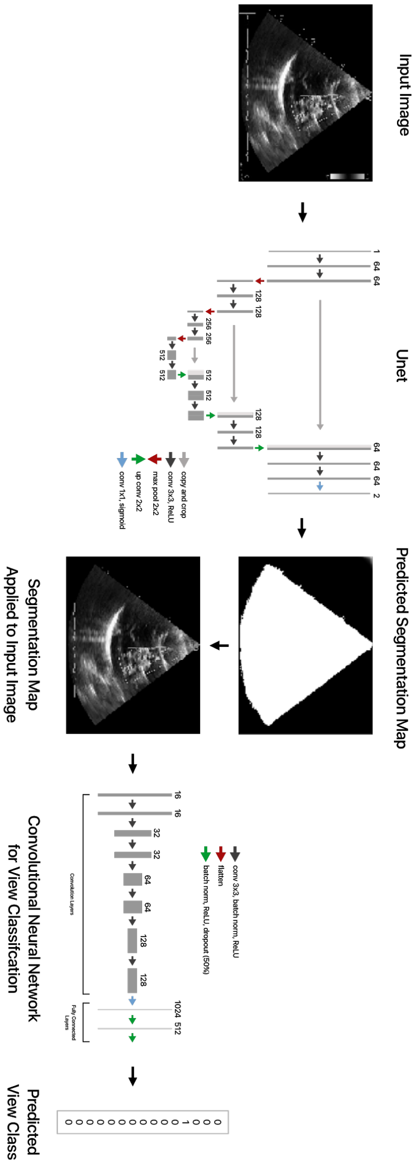

Figure 3.4: Field of View (FoV) segmentation and View Classification pipeline. U-Net predicts a segmentation map over the main FoV, which is applied to the input image prior to view classification. FoV segmentation before view classification improved performance of the CNN model from an overall test accuracy of 92.05% to 93.64%.

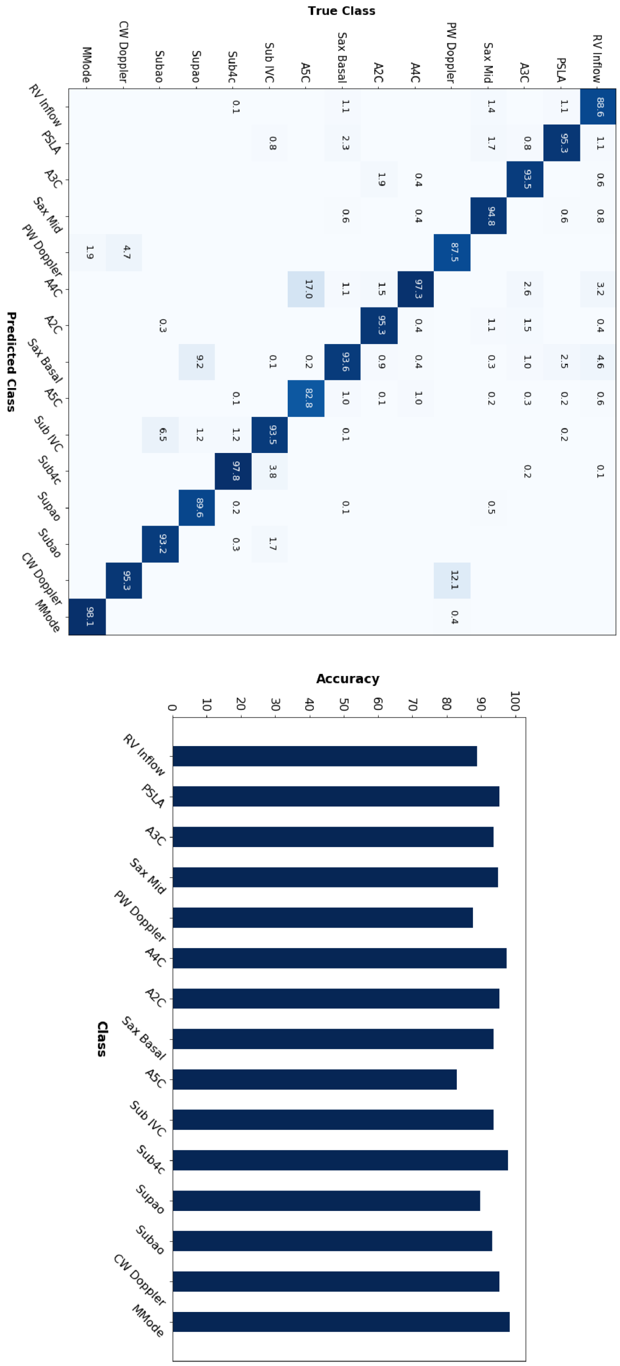

Figure 3.5: Normalized Confusion Matrix (left) and Accuracy by Class (right) on test set for ensemble network. 11 out of 15 classes achieved test accuracy above 90%. Classes with highest rate of confusion such as A5C with A4C are consistent with structural similarity overlap between the two classes.

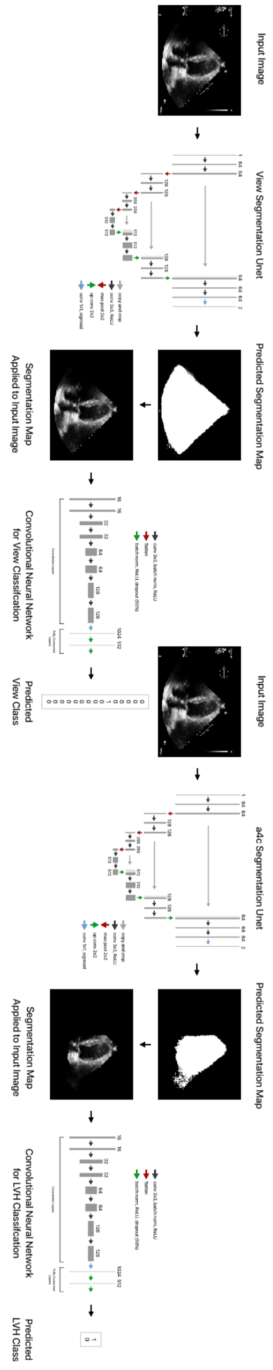
Input Image
View Segmentation Unet
Predicted Segmentation Map
Segmentation Map Applied to Input Image
Convolutional Neural Network for View Classification
Predicted View Class
Input Image
a4c Segmentation Unet
Predicted Segmentation Map
Segmentation Map Applied to Input Image
Convolutional Neural Network for LVH Classification
Predicted LVH Class

Figure 3.6: View classification and left ventricular hypertrophy classification pipeline. View segmentation U-Net predicts a segmentation map over the main FoV, which is applied to the input image prior to view classification. Based on the predicted view class, input image is routed to the respective disease classification pipeline. a4c images are routed to a4c segmentation U-Net, which predicts a segmentation map over the left ventricle. This map is applied to the input image prior to LVH disease classification.

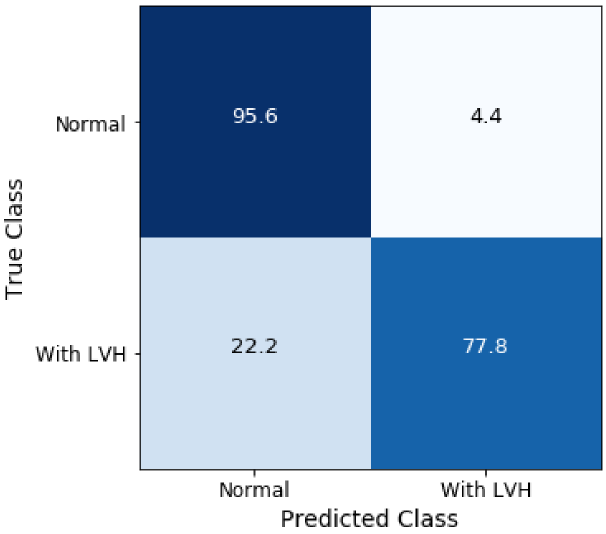

Figure 3.7: Normalized confusion matrix on LVH test set for CNN with segmentation stage 2. CNN model with segmentation was able to classify test images with 91.21% (+/- 0.41) accuracy (specificity 95.70%, sensitivity 76.70%).

varying amounts of labeled data by artificially apportioning part of the data as labeled and the remaining as unlabeled. The accuracy on the same test set for the various training scenarios is presented in figure 3.8 and table 3.4. As observed, the relationship between increasing number of labels and model accuracy is highly exponential. With less than 4% of the data, the model is able to achieve greater than 80% accuracy.

For the LVH classification task, we have the scenario of a small labeled data set (2000 samples) but access to a large unlabeled data set (76000). We assume the distribution of LVH in the unlabeled data is as probably more severely unbalanced than the labeled data, yet it does not affect our training methodology. In figure 3.9, we trained three separate models on the LVH dataset and compared the accuracies and F1 scores. We also plot the confusion matrix for an ensemble model comprising the three models that were trained on the LVH

Model	Test Accuracy	F1 Score
CNN (Stage 1)	84.07% (+/- 1.23)	0.6027
CNN (Stage 2)	87.91% (+/- 0.57)	0.7381
CNN with Segmentation (Stage 1)	81.32% (+/- 0.98)	0.5952
CNN with Segmentation (Stage 2)	91.21% (+/- 0.41)	0.8139

Table 3.3: Test accuracies and F1 Scores for LVH classification models. In stage 1, weights for convolution layers were fixed and model was trained over 20 epochs. In stage 2, weights of fully-connected layers were fixed and convolution layers were fine-tuned.

dataset. The accuracy rates and F1 scores for this task are higher for semi-supervised GANs than for the pipeline model technique described above.

Figure 3.10 shows the progression of results from the generator which was used in the GAN training process to perform semi-supervised learning. We observe the quality of generated samples improves after a few epochs and that medically-relevant structures are discernable by the time of convergence as opposed to random noise. This affirms that the model is capturing and understanding the relevant features that comprise our data distribution.

3.3 Discussion

Deep learning has revolutionized the development of automated algorithms across multiple computer vision tasks [55]. For each domain, there are specific considerations that become more relevant in the development of deep learning models. In particular, medical imagingacross modalitieswill often come in a DICOM format and have either high resolution images or varying resolution images per acquisition. Pre-processing the data to standardize the resolution is a prerequisite step before building deep learning models with non-trivial implications downstream. Keeping all pixel information can be superfluous and lead to excessive number of weights to learn and computation time. Downsampling is usually performed on input images while attempting to avoid loss of visual features that can be used to distinguish between multiple classes. Our results show that for a computer vision domain and task, we can quantify a study of ideal resolution size. For our study, there exists a critical resolution at around 60x80 pixels which provides the minimal amount of visual information necessary for accurate view classification. Naturally, the ideal amount of downsampling will vary based on the size and complexity of visual structures for a particular task. However in general, the computation time increases rapidly with increasing input resolution as the network architecture increases in depth and convolution per layer quadruples. We also found that the model architecture with highest validation accuracy varies across each resolution. For 240x320 resolution, a filter size of 7 for first two convolution layers outperforms a filter size

Labels per class	Test Accuracy
1	16%
5	28%
15	43%
30	51%
90	65%
270	78%
810	82%
21732	88%

Table 3.4: View classification performance of the semi-supervised generative adversarial network for varying amounts of labels as input.

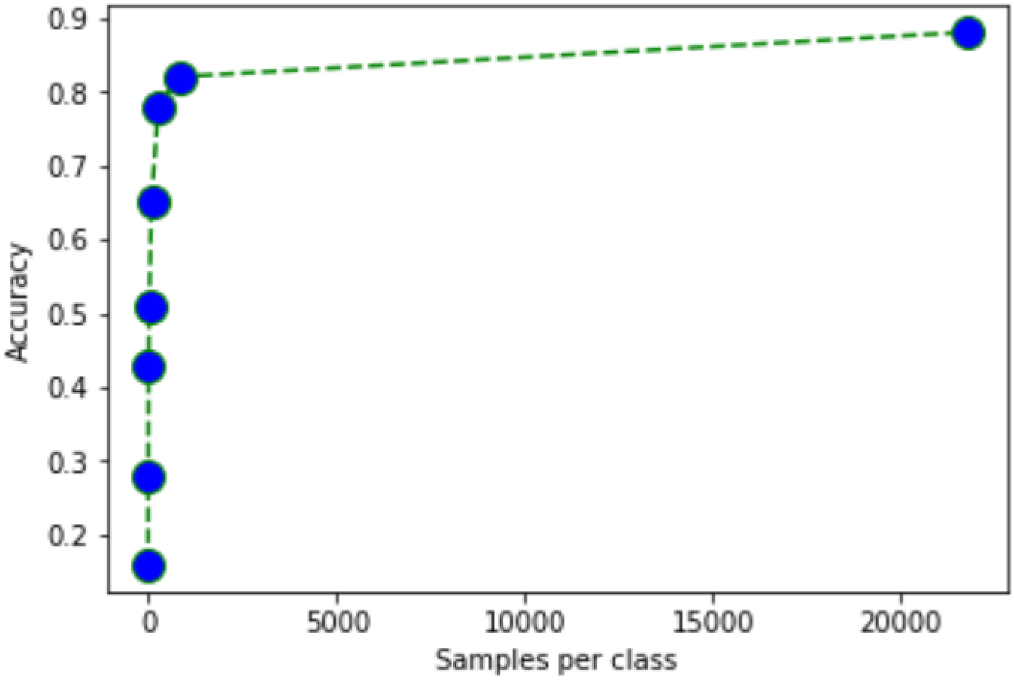

Figure 3.8: View classification performance of the semi-supervised generative adversarial network for varying amounts of labels as input. The model is able to learn from very small amounts of labeled data (approximately 4% of labels kept with the remaining data as unlabeled) to achieve greater than 80% accuracy for view classification. There exists an exponentially asymptotic behavior over number of labeled samples where accuracy gain becomes less prominent.

of 3 as a 3x3 patch of the input image is contains insufficient visual information for effective classification. In the varying resolution study, our training set had less than a quarter of the available training data to allow for rapid experimentation. While we attempted to identify the best architecture for each resolution, there remains scope for further optimization.

Our segmentation model effectively removes visual features that are less relevant for View and LVH classification. Utilizing segmentation before classification provides an elegant method to localize the attention of predictive models to pixels with relevant visual features. This is enabled and made practically feasible by the fact that only a small sample of labeled segmentation masks are required for training. These masking subtasks can be abstracted away to lower-cost labeling labor as well. Lastly, we observed that even in cases where there is a large difference between the labeled and the predict map, there is very limited amount of information loss after the predicted map is applied as shown in figure 3.11. The U-Net tends not to mask over areas where it is less confident of.

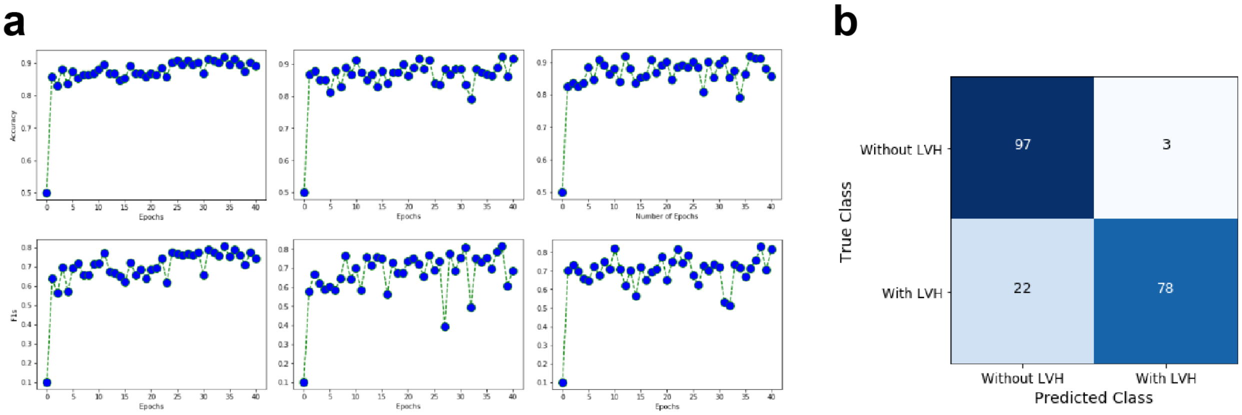

Figure 3.9: Performance of semi-supervised GAN for LVH classification in apical 4 chamber echocardiogram images. a) For three separate models trained, the accuracy (top row) and F1-score (bottom row) is plotted vs number of epochs. The model training reliably reaches convergence and continues to fluctuate within a reasonable limit. b) Normalized confusion matrix for an ensemble of the three models. This achieves an F1-score of 0.83 and accuracy of 92.3% (+/- 0.57).

There is an improvement (+1.59%) in overall test accuracy with the segmentation pipeline compared to the initial CNN model, which confirms our initial hypothesis that the removal of auxiliary information using the U-Net segmentation model helps to simplify the view classification problem. While end-to-end learning is growing in popularity with the increasing size of datasets and network depth, having a sequential pipeline of different neural network models each performing a heuristic-based classification task that simplifies the classification problem for the next network can be more effective depending on the problem domain. We view this as the primary reason behind the improvement of view classification accuracy from our previous study [66].

Transfer learning from pre-trained VGG16 and Resnet50 models was computationally expensive and failed to match the accuracy of the simpler CNN model with default weight initialization. While transfer learning has been used successfully in the domain of disease classification [94], this strengthens our view that transfer learning from models pre-trained on the ImageNet dataset may not be as effective or computationally feasible for datasets that with significant structural differences. Lastly, as with previously published results [21], we were able to exact accuracy improvements on our test dataset by applying an ensemble technique to average predictions of multiple models.

In our left ventricular hypertrophy study, we observed that using transfer learning techniques from the view classification model enables the neural network to learn more effectively compared to default weight initialization. Applying left ventricle segmentation to the apical 4 chamber images produced an even larger improvement in overall test accuracy compared to our view classification study, corresponding to the larger area of image removed during

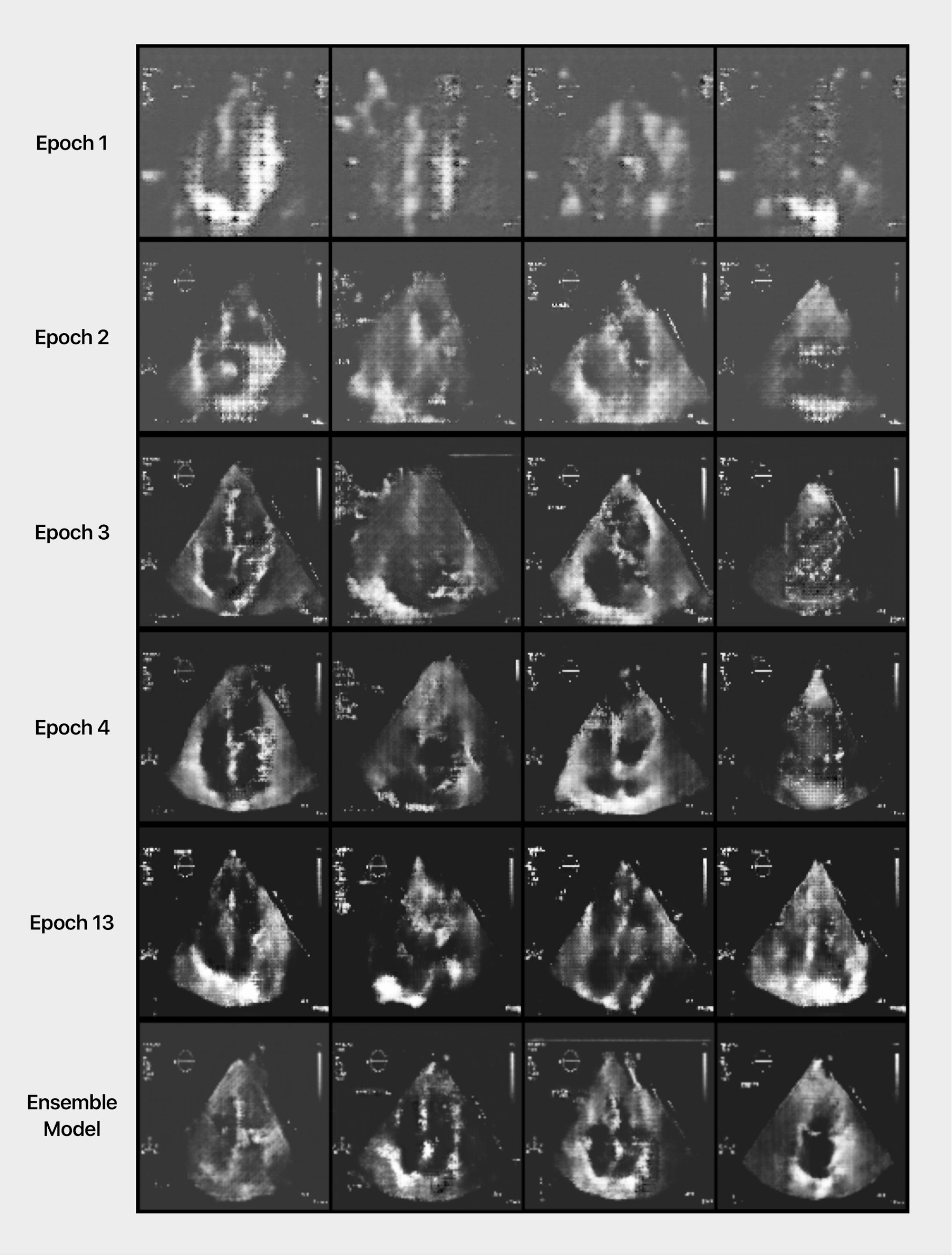

Figure 3.10: Generated images sampled from the generator network of the semi-supervised GAN during training for LVH classification. For one model, batches of size four are shown containing generated images (top to bottom) after epoch 1, 2, 3, 4, 13. The last row displays generated images from an ensemble of three GAN models. Qualitatively, the model clearly learns and understands the underlying physiological structures in input distribution.

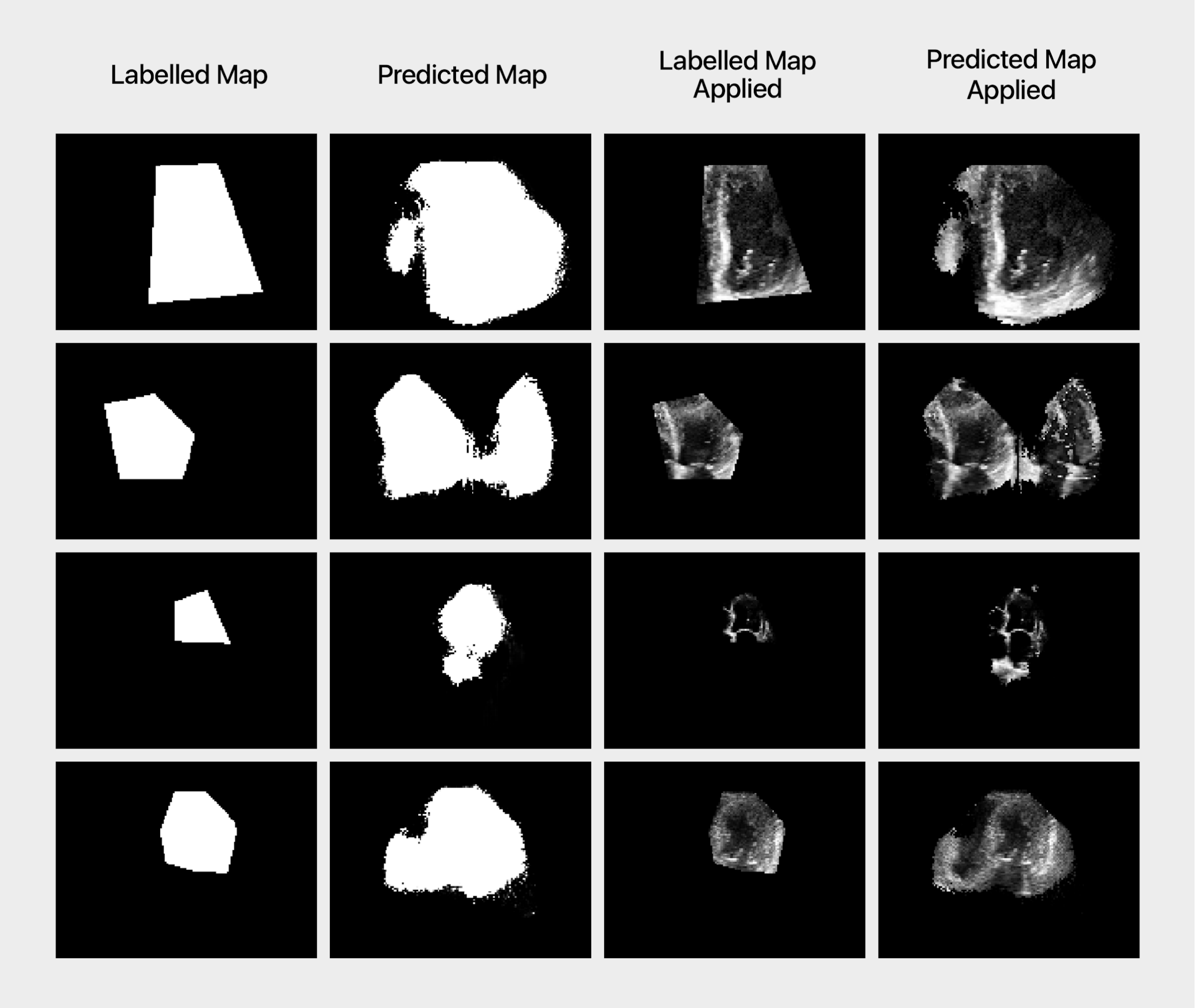

Figure 3.11: Figure of labeled map, predicted map, labeled and predicted map applied to original image for LVH Segmentation. Even in cases where there is a large difference between the labeled and the predict map, the amount of information loss after applying the predicted map is very limited.

the segmentation step.

While the final model achieved 91.2% (+/- 0.41) accuracy with 95.7% specificity and 76.7% sensitivity, it is within reasonable limits given that there were 4 times the amount of training data for the former. The success of our LVH classification model demonstrates that deep learning models adapted for echocardiogram datasets can generalize well from view classification to disease classification even with a small training set.

In addition to a pipeline deep learning model with solely supervised classifiers, semi-supervised GAN models were explored as a generalizable approach that leverages learning from both labeled and unlabeled data. For computer vision tasks in medical imaging, we often have scenarios where there is a larger unlabeled dataset and only a portion of the data can be accurately and cost-effectively ground-truthed.

The semi-supervised GAN model was trained and tested on the view classification problem first as we could designate varying proportions of data for labeled vs unlabeled to observe the effect on classification performance. The results show that semi-supervised GANs require an order of magnitude less labeled data (4% of total data) to achieve adequate performance. We also saw that the GAN still performs well in asymmetrically distributed categories for our view classification. These results provided motivation to train for the LVH classification task where only a small portion of data existed with LVH labels (2200 samples). For the LVH classification task, the semi-supervised GAN is able to learn from both the labeled and unlabeled data, even with highly unbalanced classes, to achieve an accuracy of 92.3% (+/- 0.57), with specificity of 97.0% and sensitivity of 79.1%.

The semi-supervised GAN loss function is formulated to account for contributions from unlabeled, labeled, and generated samples. On a high level, the GAN through the sigmoid real/fake loss for unlabeled samples becomes better at training filters which can identify salient features of an image while the labeled samples are utilized mostly for implicitly training the layers which performs the final classification. GANs also implicitly are performing data augmentation as the generator converges – producing more realistic images which allow the GAN to explore probability spaces of the data not covered in the original training data. Future areas of exploration include more robust loss functions, conditioning the generated noise, and normalization techniques [72, 73].

One of the most important aspects of our study is the usage of GANs for biomedical image classification tasks. Although the particular hyper-parameters and implementation details may vary outside this dataset and prediction task, the overall model considerations in addition to the data and experimental pipeline is applicable across medical domains and tasks. We shed light on the development and improvements of semi-supervised GANs, relevant performance metrics, and functional intuitions as to promote its application both within and beyond echocardiography and cardiology. To do so, we experiment with two different prediction tasks in addition to a comparative analysis with reference to traditional deep learning architectures such as a CNN.

As with all studies, there are natural limitations to our work. It is important to be stated that although we focus on improving image classification tasks in echocardiography, image classification is only one aspect of clinical diagnosis as clinicians utilize other forms of cognition to both understand and treat a patients illness. We view this line of work as not a replacement of the clinician but as an assistant for relevant tasks that a clinician performs. Also, it is also worth noting that our sample sizes are limited due to practical reasonswhich is a common medical imaging issue that beckons for data-efficient, generalizable techniques presented in this study. We look forward to future work and validation that is conducted on even larger sample sizes that include both more patients and also variations in acquisition (i.e. different institutions, field of views, and more).

Also important to note regarding the use of GANs for biomedical imaging, researchers and practitioners should exercise caution when using GANs for image synthesis in the medical domain. Clinical decisions on images synthesized by GANs should require a better understand of GANs underlying mechanics. For example in [18], the authors show how distribution matching losses can produced translated images that will lead to misdiagnosis of medical conditions.

To conclude, the focus of our study is on data-efficient deep learning models for classification tasks in medical imaging. Initially, we investigate the trade-off between resolution size and computational burden. We then explore two main techniques: (1) supervised pipeline models to first extract relevant structures then pass through a CNN classifier (2) semi-supervised GAN models for end-to-end training. With customized segmentation models and the extra effort to label additional segmentation maps, the first method is able to achieve the highest reported accuracy for view classification in cardiac ultrasound imaging and relatively high performance for LVH classification. Often in practice, labeled data in medical imaging is scarce, locked by privacy or regulatory concerns, or expensive to annotate. Utilizing both labeled and unlabeled data and with a generalizable end-to-end training strategy, the second method is able to achieve high performance for LVH classification. We provide avenues and trade-offs for practitioners and researchers with the aforementioned techniques for their computer vision tasks in medical imaging.

3.4 Methods

Echocardiographic Data

All datasets were obtained and de-identified, with waived consent in compliance with the Institutional Review Board (IRB) at the University of California, San Francisco (UCSF). Methods were performed in accordance with relevant regulations and guidelines. Echocardiographic studies were extracted then videos and still images were stripped of identifying metadata and flattened to individual frames. Data comprised of studies between 2000 and 2017 at random from UCSFs clinical database. These studies included men and women (49.4 and 50.6 percent, respectively) ages 2096 (median age, 56; mode, 63) with a range of body types (25.8 percent obese) and were acquired with equipment from several manufacturers (eg. GE, Philips, Siemens). For view classification, the number of studies taken were N=267 and fifteen common views were utilized including: parasternal long axis, right ventricular inflow, basal short axis (aortic valve level), short axis at mid (papillary muscle) or mitral level, apical two chamber, apical three chamber (apical long axis), apical four-chamber, apical five chamber, subcostal four-chamber, subcostal inferior vena cava (IVC), subcostal abdominal aorta, suprasternal aortic arch, pulsed-wave Doppler, continuous-wave Doppler, and m-mode. For LVH classification, the number of studies taken were N=455 and the diastolic frames in the apical 4 chamber view were selected and comprised of the following clinical labels: normal, moderate-to-severe asymmetric LVH, severe asymmetric LVH, moderate concentric LVH, moderate-to-severe concentric LVH, severe LVH. Further information regarding source data can be found in the previous literature [66].

Varying Resolution Study

We divided our subset of 103102 images into a training set containing 75937 images, a validation set containing 11696 images and a test set containing 15469 images, in an approximate (74:11:15) split. There is no patient overlap between the training, validation and test set. Downsizing of images originally in 600x800 pixels to the image resolutions of 240x320, 120x160 60x80, 30x40 and 15x20 pixels were completed using Scipys imresize function with the nearest interpolation mode. Figure 3.12a displays sample images from each resolution for three classes.

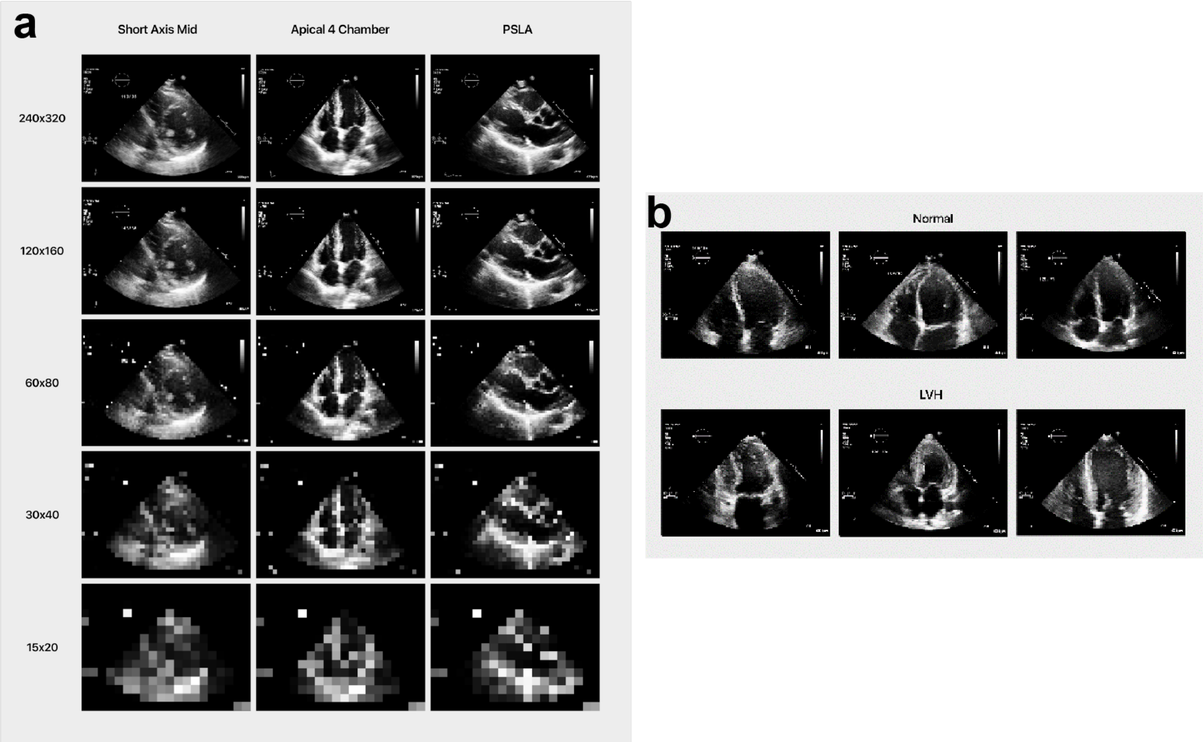

Figure 3.12: Sample data from echocardiographic studies. a) Sample echocardiogram images at varying resolutions (rows) for three example views (columns). Selection of the optimal resolution is influenced by the tradeoff between classifier performance and computational time. b) Sample apical four chamber echocardiography images with and without Left Ventricular Hypertrophy (LVH). LVH is characterized by the thickening of the left ventricle– a perilous condition increasing the risk of myocardial infarction, stroke, and death.

Our neural network architectures were designed in Python using the Keras library with TensorFlow backend [1], primarily based on the VGG16 Network [96], which won the ImageNet Challenge in 2014 [20, 86]. Our architecture consists of multiple convolution layers, followed by fully connected layers. We applied Batch Normalization [42] and rectified linear activations [32] after each layer. Softmax activation was applied to the final fully connected layer for classification over the 15 output classes. Dropout [97] and L2 Regularization were added in the fully-connected layers to prevent overfitting.

We further experimented with various architectures for each resolution and the architectures with the highest validation accuracy. For convolution layers, filter sizes of 3x3, same padding, stride of 1 pixel, and pooling size of 2x2 were used for each resolution except 240x320, for which a filter size of 7x7 were used for the first two convolution layers. Maximum Pooling were used for resolutions 60x80 pixels and above, and Average Pooling for resolutions below 60x80 pixels.

Data augmentation was applied during training with up to 10 degrees rotation and 10% height and weight shifts. Adam optimizer with default parameters was used to minimize the categorical cross-entropy loss. Learning rate was set to 0.02, with decay per epoch of 0.85. Early stopping was applied once validation loss stops decreasing for two consecutive epochs. Training was performed on Nvidia GTX1080Ti GPUs, with batch sizes of varying sizes used for the different resolutions to maximize GPU memory usage. We evaluated our final models by computing the overall validation accuracy, average time to train and validate per epoch.

Relevant Structure Segmentation

From our varying resolution study, we selected 120x160 as the optimal resolution for our subsequent experiments. It provided an ideal balance between accuracy and computational time as discussed in Results.

We employed relevant structure segmentation as preprocessing for removal of irrelevant details in the images to simplify the classification task. We trained a convolutional neural network on two different datasets for segmentation of the main field of view (FoV) in echocardiogram images prior to view classification, and segmentation of the left ventricle in apical 4 chamber (a4c) images prior to left ventricular hypertrophy classification.

Our field of view segmentation dataset contains 433 images- inclusive of image frames from various 2D views, doppler, and m-mode echocardiograms. Images were converted to grayscale and downsampled to 120x160 pixels using Scipys imresize function with the nearest interpolation mode. The data were divided by class into a training set consisting of 411 images and a test set consisting of 32 images for evaluation of the models performance.

Masks were drawn on the outline of the field of view containing the medical image with an in-house labeling tool that the user selects vertex points to form a polygon shape. For doppler and m-mode echocardiograms, masks were drawn around the general shape of the waveforms and field of view. The labeling tool is an interactive polygon editor built with matplotlib that sets pixels inside the mask to 1 and pixels outside to 0.

Our left ventricle segmentation dataset contains 720 images sampled randomly from the apical 4 chamber (a4c) view dataset. Some apical 4 chamber views might be cropped and the left ventricle may not be visible. Therefore, for Relevant Structure Segmentation for LVH only, we did not include images that failed to show the left ventricle. Important to emphasize, no filtering was performed for the final LVH test set images or calculation of performance metrics on the final goal of LVH classification. Images were downsampled to 120x160 pixels using Scipys imresize function with the nearest interpolation mode. Images were divided into a training set consisting of 670 images and a test set consisting of 50 images. Using the aforementioned labeling tool, masks were drawn to create an approximate polygon around the left ventricle epicardium. Masks were drawn around the left ventricle using the labeling tool. For images in dual mode, a single mask was drawn around the left ventricle in the left-most view.

Our image segmentation model is based on the U-Net architecture [84] and is shown in the figure 3.13. Modifications were made to the original architecture for adaptation to the 120x160 pixels resolution of our dataset. These include changes to filter sizes, removal of convolution layers with 1024 filters and the addition of Dropout before the first up-sampling convolution layer.

The model was trained over 50 epochs with a learning rate of 0.0001 and per epoch decay of 0.93 using the Adam optimizer. We evaluated the model by computing the pixel-wise loss on the test set. We also visually inspected the U-Nets segmentation on the test set to further verify the segmentation performance.

Echocardiographic View Classification

Our dataset consists of 347726 echocardiogram images, of which 325980 images were in the training set. The original test set used in the previous study [66] containing 21746 images were retained for this experiment.

Images were downsampled to 120x160 pixels using Scipys imresize function with the nearest interpolation mode for the convolutional neural network (CNN) experiments. Images used for Resnet50 and VGG16 were resized and copied over all three channels to fit the pre-trained models input dimensions of 224x224x3.

In this experiment, we trained the following models to compare the effectiveness of relevant structure segmentation for preprocessing, transfer learning and ensembling: CNN model trained using the original images, CNN model trained using images with FoV Masking, Transfer Learning from Resnet50 and VGG16 models using images with FoV Masking and an Ensemble of 3 CNN models trained using images with FoV Masking. We retained the same architecture for 120x160 resolution as used in the varying resolution study, with the exception of replacing Max Pooling layers with Average Pooling layers for the CNN trained with FoV Masking.

Resnet50 and VGG16 models from the Keras Library which were pre-trained on the ImageNet dataset were used. The fully-connected layers from the Resnet50 model were removed and replaced with a batch normalization layer followed by a fully connected layer

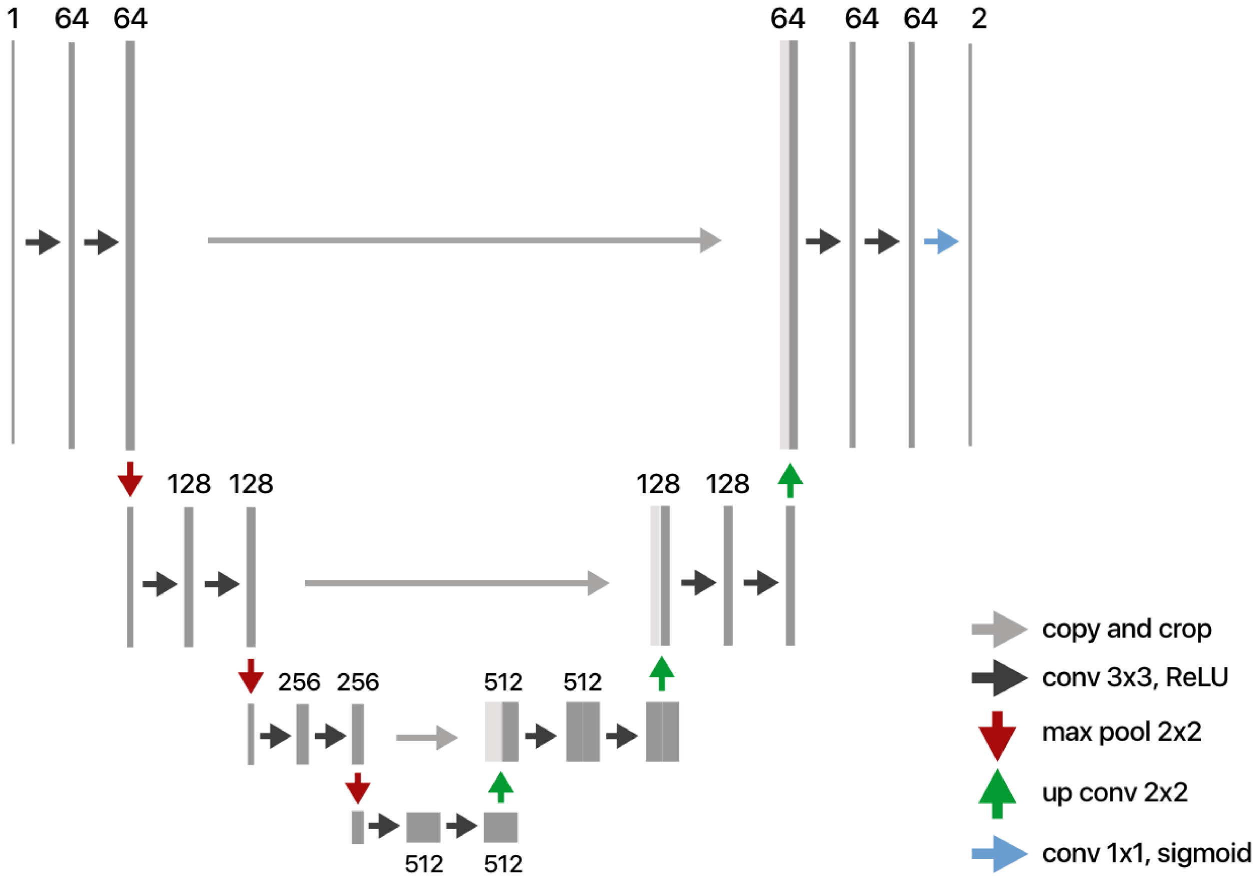

Figure 3.13: Modified U-Net architecture used for segmenting relevant visual structures. The architecture consists of a contracting path and a symmetric expanding path– combining high resolution features from the contracting path and upsampled output for precise localization. Pixel-wise softmax is applied on the final output to produce a segmentation map.

with softmax activation. For VGG16, we replaced the fully-connected layers with the same fully-connected layers layers with batch normalization and L2 regularization, followed by a fully connected layer with softmax activation. Our ensemble consists of 3 CNN models trained individually using images with FoV Masking, with the predictions output averaged.

Adam optimization with default parameters and early stopping were used for the following experiments. During training of the CNN models, learning rate was set to 0.02 with decay per epoch of 0.85. Training data was augmented with up to 10 degrees rotation and 10% height and weight shifts.

For the pre-trained models, training data was augmented with up to 5 degrees rotation, 10% height and 15% weight shifts. Fine-tuning for Resnet50 was performed end-to-end with a learning rate of 0.01 and decay per epoch of 0.96.

For VGG16, training was divided into two stages. In the first stage, weights from the convolution layers were frozen and the fully connected layers with Xavier initialization were trained till convergence with a learning rate of 0.02 and decay per epoch of 0.9. In the second stage, weights from the convolution layers were fine-tuned along with the fully connected layers at a lower learning rate of 0.001 and decay of 0.9 per epoch.

We evaluated the various models by computing the overall test accuracy across 15 classes, training and test time per epoch, test accuracy by class, confusion matrix, and F1 score, which is the harmonic mean of precision and recall. Confidence intervals were computed by the bootstrapping technique with replacement for 95% intervals.

Left Ventricular Hypertrophy Classification

The dataset consists of 2269 images from the apical 4 chamber (a4c) view. The first two frames of an echocardiographic study were selected to ensure only diastolic phase. To formulate this as a binary classification problem as in figure 3.12b, we chose images with different labels for Left Ventricular Hypertrophy (LVH) to form a single class with 462 images. 1807 images of normal a4c views formed the other class. The ratio of images without LVH to images with LVH is approximately 4:1.

Image were divided into a training set consisting of 1890 images, validation set of 172 images and a test set of 207 images. Images from the same patient were placed in the same split. For preprocessing, images were downsampled to 120x160 pixels and masked with the predicted output from the left ventricle segmentation model in our previous experiment.

We retained the same architecture for 120x160 resolution as used in the varying resolution study, with the exception of replacing Max Pooling layers with Average Pooling layers. L2 regularization of 0.03 and Dropout of 0.4 was applied to the fully connected layers.

We applied a two-stage transfer learning method for the training of this model. Weights for convolution layers were initialized with weights from the CNN model trained using images with FoV Masking. Fully connected layers were initialized with Xavier uniform initialization.

In the first stage, weights of convolution layers were fixed and the model was trained over 20 epochs to minimize binary cross-entropy loss, with a learning rate of 0.025 and decay per epoch of 0.85. In the second stage, weights of fully-connected layers were fixed and convolution layers were fine-tuned with a learning rate of 0.001 and decay per epoch of 0.9. Early stopping was applied once validation loss stops decreasing for two consecutive epochs.

We evaluated the model by computing the test accuracy and F1 score, which is the harmonic mean of the precision and recall. Confidence intervals were computed by the bootstrapping technique with replacement for 95% intervals.

Semi-supervised Generative Adversarial Networks

As shown in 3.14, the GAN makes use of two neural networks: a generator which attempts to generate realistic images and a discriminator which discriminates between real (optionally including specified classes) and fake [33].The same semi-supervised GAN architecture is used for both the view classification task and the left ventricular hypertrophy classification task.

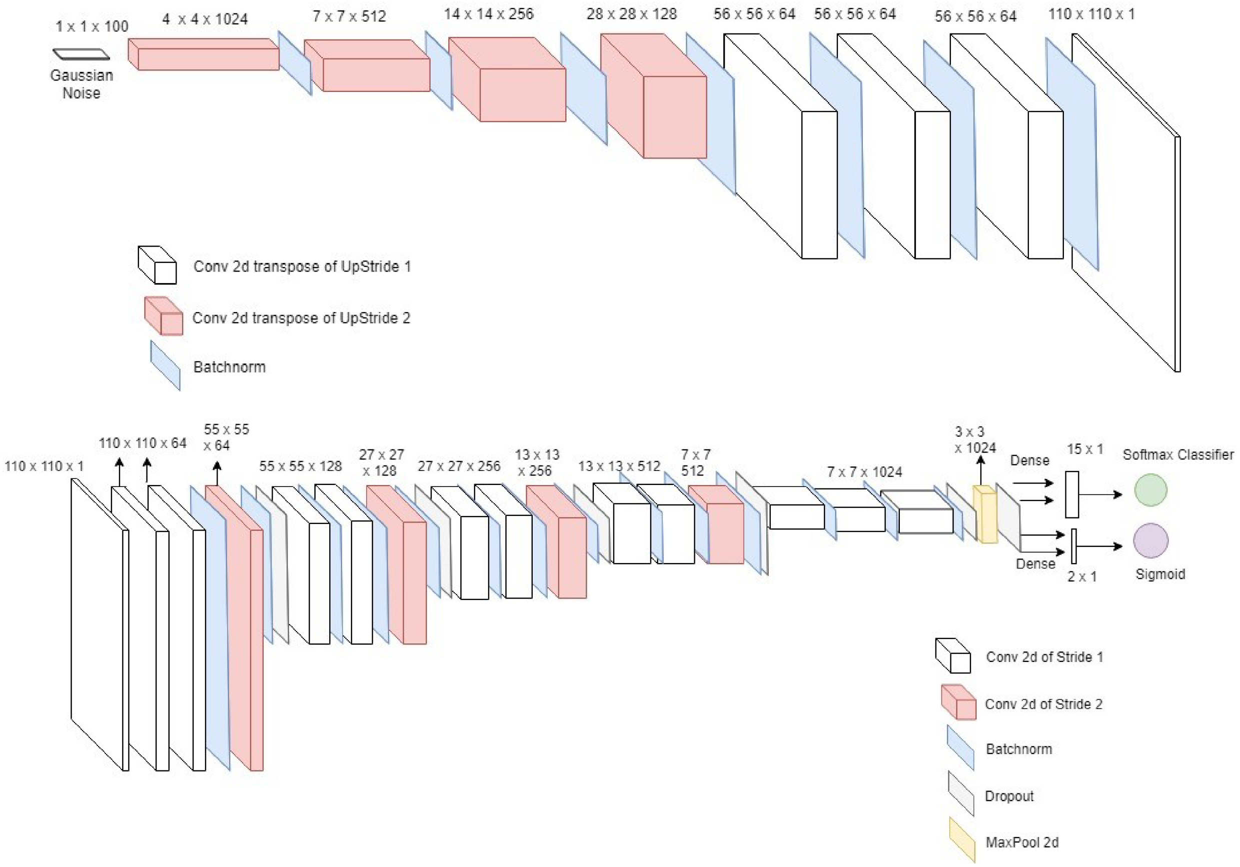

Figure 3.14: Semi-supervised GAN for echocardiogram view and LVH classification. Generator (top) consists of a Gaussian noise layer which gets passed through conv-transpose layers to output images of size 110x110. Discriminator (bottom) downsamples original images with regular strides of two every three layers, resulting in a softmax output for labeled (or supervised) loss and a sigmoid output for unsupervised loss.

The discriminator model is structured in blocks of three convolutions comprising two convolutional layers of stride 1 and one convolutional layer of downstride 2. All except the last convolution layer make use of 3x3 convolutional filters, batch normalization layer after each convolution layer and dropout layer after every downstride convolution layer. Leaky RELUs are used as activation functions for all the layers and the last classification layers which use Softmax and Sigmoid.

The generator model consists of seven deconvolutional layers each of stride two and it works by progressively upsampling from the gaussian noise layer where the model begins. Each of these layers use either a 3x3 or 4x4 convolutional filter and a batch normalization layer. All intermediate activations consist of ReLUs while the very last layer makes use of Tanh activation to output the final image.

Training GANs for semi-supervised learning involves performing three passes through the discriminator and one pass through the generator at each iteration. To compute the discriminator loss a labeled image is passed through the discriminator to assess a cross-entropy loss. Then an unlabeled image and a fake image are passed through the discriminator for both of which we compute binary cross entropy losses. All three of these losses are summed together and are used to perform a single step of backpropagation through the discriminator. The losses were inspired by previous literature [88] and detailed below:

$$L = L_{supervised} + L_{unsupervised} + L_{generated} \tag{3.1}$$

$$L_{supervised} = -E_{x,y \sim p_{data}(x,y)}[\log(p_{model}(y|x, y < K))] \tag{3.2}$$

$$L_{unsupervised} = -E_{x \sim p_{data}(x)}[\log(1 - p_{model}(y = K + 1|x))] \tag{3.3}$$

$$L_{generated} = -E_{x \sim G}[\log(p_{model}(y = K + 1|x))] \tag{3.4}$$

To compute the generator loss, we generate a fake image from the generator which we pass through the discriminator and compute the loss by using a mean square error loss between the second last layer of the discriminator for our fake image and for an unlabeled image.

The data used for view classification was the same as the models outlined above, including the size of the splits. The only differences were that the image was downsized to 110x110 pixels and that the split between unlabeled data and labeled data was changed for different models so as to determine the relationship between labels and accuracy. One epoch was defined as the number of unlabeled images since this was always going to be the larger value. This however meant that in each epoch the discriminator ends up going through multiple epochs of the labeled images.

For LVH classification problem we used the same labeled data consisting of 2269 images which were divided into a training set consisting of 1915 images, validation set of 172 images and a test set of 182 images. Images from the same patient were placed in the same split. The only difference was that we also used 76404 unlabeled images (from the apical 4 chamber data in the view classification dataset) on top of the labeled ones to bolster the semi-supervised GAN. For preprocessing, these images were downsampled to 110x110 pixels.

Training was performed using the Adam optimizer set at default values for both of these tasks. A learning rate of 0.0003 with no decay mechanism. Training a GAN using 2 GTX 1080Ti GPUs training was on the order of five hours.

To evaluate the performance of semi-supervised GAN for view classification we only made use of the accuracy rate of the discriminator model at different epochs. To evaluate the performance of the LVH model we use accuracy rate of three models trained on all of the data as described above. We also use the F1 scores of these models and plot them on a separate curve. For the highest performing model, we used confusion matrices to better illustrate how the model performs on different categories.

Chapter 4

Bridging Finite Element Modeling and Machine Learning for Atherosclerosis

4.1 Introduction

Advances in computational methods and infrastructure have enabled researchers to develop powerful models across multiple domains including, but not limited, to biology and medicine. Broadly speaking, computational models are developed within two main paradigms: mechanistic and statistical modeling. Mechanistic models assume some a priori knowledge on the behavior of physical processes. We broadly define molecular dynamics, computational fluid dynamics, and finite element method techniques under mechanistic modeling as the governing equations are explicitly given and the solution is approximated [63, 4]. Statistical modeling, particularly machine learning, needs not any hard-coded behavioral assumptions and treats the system as a black-box. Machine learning algorithms, in this context, aim to learn the mapping between input and output. One of the most powerful classes of models are deep neural networks which are well-suited for highly non-linear data and have demonstrated high performance in natural language processing and computer vision [55, 84, 96, 100, 50]. The intersection of these two modeling paradigms is of emerging interest [8, 36, 81, 69].

In this study as shown in figure 4.1a, we explore the intersection between mechanistic and statistical modeling of a solid mechanics problem utilizing finite element simulations and machine learning models. Our case study is the stress analysis of arterial walls under atherosclerosis (figure 4.1b), one of the leading causes of disease worldwide. Prediction of plaque rupture through examining arterial stress distributions could potentially lead to life-saving treatment and prevention outcomes.

By creating thousands of finite element (FE) simulations of the highly non-linear behavior of a hyperelastic arterial cross-section, we attempt to answer how well can machine learning models learn the underlying mapping that is determined via an FE procedure? In addition,

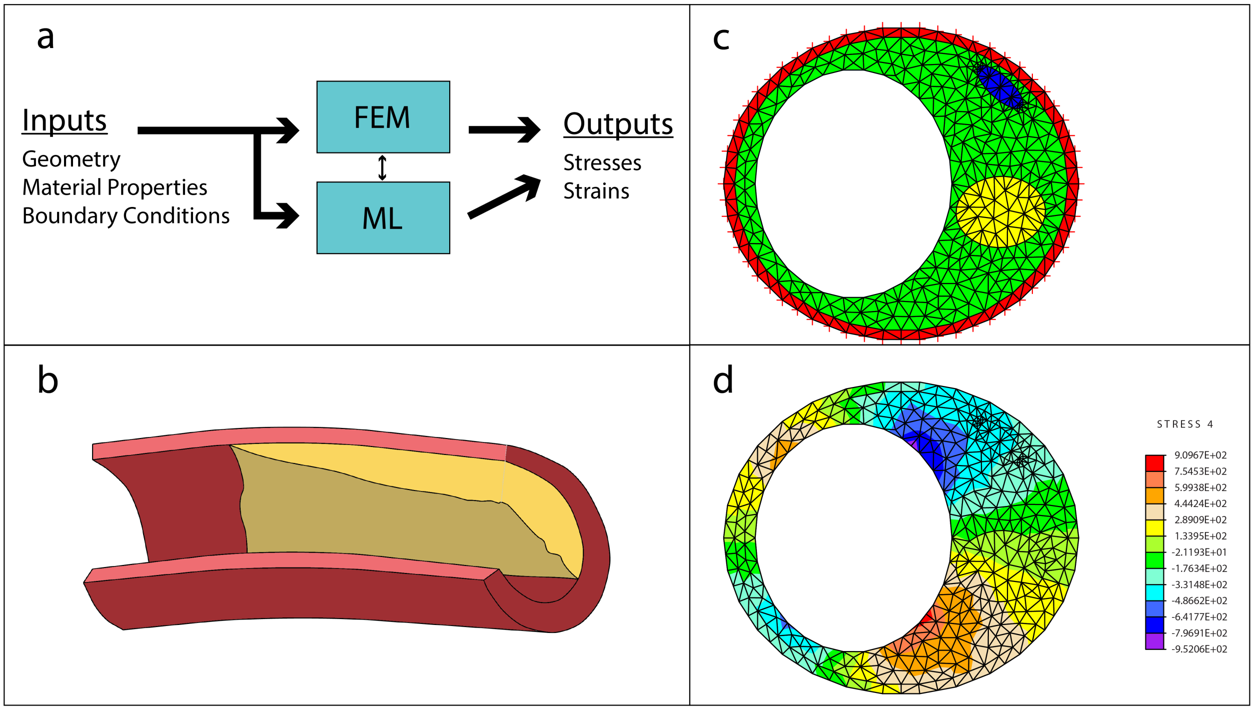

Figure 4.1: Prediction of atherosclerotic vessel stress with machine learning (ML) models trained on finite element method (FEM) simulation data. a) Mechanistic and statistical methods can be used for various continuum mechanics problems. b) Atherosclerosis can cause the buildup of plaque in blood vessels as shown. c) An example idealized geometry of a 2D cross-section of an atherosclerotic vessel. Red- vessel wall, green- fibrous plaque, yellow- lipid pool, blue- calcium deposit, white- lumen. Geometries are randomly generated to create a database of FEM simulations. d) An example stress distribution generated by utilizing FEM. The input and output information are utilized to train ML models for prediction tasks such as maximum von Mises stress.

we hope to gain intuitions behind if machine learning can learn the stress-strain relationship and constitutive behavior via the functional representation presented by FE or some higher-level correlation.

Lastly, this research work could have implications across both the computational and clinical fields. FE simulations require trained specialists and are computationally intensive. Trained machine learning models, particularly for multi-task learning [85], could enable generalizable mechanics simulations. In addition, the replacement of certain FE parts with machine learning models could enable faster computation times especially for multi-scale problems that require nested FE simulations [110]. For the clinical side, deep learning has shown promise in visual recognition tasks in healthcare [65, 66, 67]. Recent work in shape modeling and FE estimation with machine learning has shown promise for aortic aneurysms and collagenous tissues [58, 59, 57]. For atherosclerotic vessels, automated acquisition with intravascular techniques of tissue geometry, composition, and arterial pressure that is provided as input to a trained deep learning model could enable accurate prediction of real-time stress distribution. Clinicians can use this information during patient examination, evaluation, and treatment for potential areas of high rupture risk.

4.2 Methods

Parametric Vessel Model

Our training database consists of finite element simulations of various arterial geometries and boundary conditions. The first step was to develop a 2D parametric model to rapidly create various artery geometries. As depicted in figure 4.1c, our main simplifying assumption is that the cross-section consists of a series of ellipses. The number of lipid and calcium deposits, fluid pressure, and ellipse specifications were randomly varied according to table 4.1.

Model	μ [MPa]	K [MPa]	C_{1m} [Pa]	C_{2m} [-]
Arterial wall	0.334	16.67	26447.5	8.365
Plaque	0.334	16.67	5105.3	13.0
Lipid	0.334	16.67	50.0	5.0
Calcium	0.334	16.67	18804.5	20.0

Feature	Sym	a [mm]	b [mm]	u [mm]	v [mm]	θ [Rad]
Outer wall	A_{out}	$\mathcal{N}(4, 0.5)$	$\mathcal{N}(3.7, 0.3)$	0	0	0
Inner wall	A_{in}	$\mathcal{N}(1.07, 0.02)a_{A_{out}}$	$\frac{a_{A_{in}}}{a_{A_{out}}}b_{A_{out}}$	0	0	0
Lumen	L	$\mathcal{N}(1.9, 0.2)$	$\mathcal{N}(2.3, 0.2)$	$\mathcal{N}(-1, 0.2)$	0	0
Lipid	F	$\mathcal{N}(0.5, 0.4)$	$\mathcal{N}(0.5, 0.2)$	$\mathcal{U}(u_1^F, u_2^F)$	$\mathcal{U}(-v_1^F, v_1^F i)$	$\mathcal{U}(0, \pi)$
Calcium	C	$\mathcal{N}(0.5, 0.4)$	$\mathcal{N}(0.5, 0.2)$	$\mathcal{U}(u_1^C, u_2^C)$	$\mathcal{U}(-v_1^C, v_1^C i)$	$\mathcal{U}(0, \pi)$

Table 4.1: Features for random generation of idealized artery geometry and pressure conditions. N and U denote a normal and uniform distribution respectively. The number of lipids and calcium are also varied as discrete numbers ranging from zero to two deposits. Each ellipse is described by a major axis a, minor axis b, center (u,v), and angle of rotation .

The parameters were sampled from normal distributions, $\mathcal{N}(m, \sigma)$ for a mean m and variance σ^2 for a random variable X, and uniform distributions, $\mathcal{U}(i, j)$ for $X \in [i, j]$. The range of the distributions are based on patient-informed physiological parameters present in literature [11, 39]. Lastly, we define the following two terms for Table 1 where a and b refer to major and minor axes respectively:

$$u_1^{F,C} := u_L + a_L + \max(a_{F,C}, b_{F,C}) \tag{4.1}$$

$$u_2^{F,C} := a_{A_{in}} - \max(a_{F,C}, b_{F,C}) \tag{4.2}$$

$$v_1^{F,C} := b_{A_{in}} \sqrt{1 - \frac{u_1^{F,C\,2}}{a_{A_{in}}}} - \max(a_{F,C}, b_{F,C}) \tag{4.3}$$

In the generation of arterial geometries, it is plausible that a particular configuration could be physically infeasible. If such a configuration is detected, we simply discard the geometry and sample randomly again. There are three sources of infeasible geometries:

1. An ellipse is outside of the inner wall, A_{in},
2. An ellipse is inside the lumen, L,
3. Two or more ellipses intersect/collide.

The first two are trivial to detect. For the last scenario, we make use of support vector machine (SVM) to check if two ellipses are disjoint. We know that the ellipses do not collide if and only if the SVM loss is zero as there exists a perfectly separating linear hyperplane between the two classes defined by a pair of ellipses.

Finite Element Method Simulation Database

The tissue is modeled within a classical continuum mechanics framework wherein we seek to solve the governing equations of motion for a body (manifold with boundaries) subject to boundary conditions (tractions and displacements). We define a one parameter (time t) family of finite deformation maps $\phi_t : R^3 \mapsto R^3$ of a hyperelastic body from a reference configuration ($\mathcal{B}_0$) to a current configuration ($\mathcal{B}_t$). Namely, $\mathcal{B}_t = \phi_t(\mathcal{B}_0)$. [25, 39]

We define the deformation gradient $\boldsymbol{F} = \nabla\phi$. Note that $\boldsymbol{F}$ is a 2-point tensor mapping vectors from the reference manifold to the deformed manifold. We define the left and right Cauchy-Green deformation tensors by $\boldsymbol{C} = \boldsymbol{F}^T\boldsymbol{F}$ and $\boldsymbol{b} = \boldsymbol{F}\boldsymbol{F}^T$, respectively. We further define the associated strain tensors: the Green-Lagrange strain tensor $\boldsymbol{E} = \frac{1}{2}(\boldsymbol{C} - \boldsymbol{I})$ for the reference configuration and the Almansi strain tensor $\boldsymbol{e} = \frac{1}{2}(\boldsymbol{I} - \boldsymbol{b}^{-1})$ for the current configuration. $\boldsymbol{I}$ is the identity tensor for vectors in R^3.

The equilibrium deformation map at time t is the one that minimizes the potential energy (Π) of the elastic system subject to conservative traction loading $\bar{\boldsymbol{t}}_t$:

$$\phi_t^{eq} = \arg\inf_{\phi_t} \Pi(\phi_t; \bar{\boldsymbol{t}}_t). \tag{4.4}$$

Under the assumption of hyperelasticity, the 1st Piola-Kirchhoff stress of the system, $\boldsymbol{P}$, is obtained from the Helmholtz free energy, $\hat{\psi}$, of the material:

$$\boldsymbol{P} = \frac{\partial\hat{\psi}}{\partial\boldsymbol{F}}. \tag{4.5}$$

Although the solution to (4.4) is in general not unique, polyconvexity (as in [5]) of the energy function guarantees the existence of a solution. We solve the problem with a standard Finite Element (FE) numerical procedure. Our challenge is to specify $\hat{\psi}$ such that the FE model is consistent with the observed experimental response.

We can further define two additional stress tensors: the 2nd Piola-Kirchhoff stress tensor $\boldsymbol{S}$ and the Cauchy stress tensor $\boldsymbol{T}$. The relationship between the stress tensors is

$$\boldsymbol{T} = J^{-1}\boldsymbol{F}\boldsymbol{P} = J^{-1}\boldsymbol{F}\boldsymbol{S}\boldsymbol{F}^T = \boldsymbol{T}^T, \tag{4.6}$$

where $J = \det(\boldsymbol{F})$ is the Jacobian (of the deformation gradient). A quantity of interest in the subsequent analysis is the von Mises (VM) stress:

$$\nu^2 = \frac{3}{2}\boldsymbol{V} \cdot \boldsymbol{V}, \tag{4.7}$$

where

$$\boldsymbol{V} = \boldsymbol{T} - \frac{tr(\boldsymbol{T})}{3}\boldsymbol{I}, \tag{4.8}$$

is the deviatoric stress, and $tr(\boldsymbol{T}) = \sum_i T_{ii}$ is the trace.

We use an FE approach to solve (4.4). Let $\partial\mathcal{B}_u$ and $\partial\mathcal{B}_t$ denote the partitions of the boundary ($\partial\mathcal{B}_0$) of the body, $\mathcal{B}_0$, where deformation and tractions are imposed, respectively, with $\partial\mathcal{B}_u \cap \partial\mathcal{B}_t = \emptyset$, $\overline{\partial\mathcal{B}_u \cup \partial\mathcal{B}_t} = \partial\mathcal{B}_0$. Equation (4.4) is solved by satisfying the weak form statement:

Find

$$\phi \in \mathcal{S} := \{\phi \mid \phi = \bar{\phi} \text{ on } \partial\mathcal{B}_u\},$$

such that

$$\int_{\mathcal{B}_0} \boldsymbol{P} \cdot \nabla(\delta\phi)\, dV = \int_{\mathcal{B}_0} \boldsymbol{B} \cdot \delta\phi\, dV + \int_{\partial\mathcal{B}_t} \bar{\boldsymbol{t}} \cdot \delta\phi\, dA, \tag{4.9}$$

$$\forall\delta\phi \in \mathcal{V} := \{\delta\phi \mid \delta\phi = \boldsymbol{0} \text{ on } \mathcal{B}_u\},$$

where ρ_0 is the material density in $\mathcal{B}_0$ and we assume there is no body force $\boldsymbol{B}$. The FE solution begins with a tessellation of the domain (figure 4.1) into a finite set of discrete nodes and elements. Letting the superscript g denote discretized parameters, and $N^A(\boldsymbol{x})$ denote interpolating shape functions in each element, we construct a Galerkin discretization as:

$$\mathcal{B}_0^h = \mathbf{A}(\mathcal{B}_0^e), \quad \boldsymbol{u}^g = \sum_{A=1}^{n_n} N^A \boldsymbol{u}_A, \quad \delta\boldsymbol{u}^g = \sum_{A=1}^{n_n} N^A \delta\boldsymbol{u}_A, \tag{4.10}$$

where e indexes the n_{el} elements in the domain, $\boldsymbol{u}$ denotes displacements, $\delta\boldsymbol{u}$ denotes variational displacements, $\boldsymbol{u}_A$ denotes nodal displacements indexed by A over n_n nodes per element, and $\mathbf{A}$ is the assembly operator [115] over every node element. Substituting (4.10) into (4.9) leads to the nonlinear (static) equilibrium equations:

$$\boldsymbol{R}(\boldsymbol{u}_t) = \boldsymbol{f}_t - \mathbf{A}(\int_{\mathcal{B}_0^e} \nabla \boldsymbol{N}_e^T \boldsymbol{P}_e \; dV_e) = 0, \tag{4.11}$$

where $\boldsymbol{R}$ is the residual for a state of displacements $\boldsymbol{u}_t$ at time t, which must be in equilibrium with the applied nodal forces $\boldsymbol{f}_t$ at time t, and $\nabla \boldsymbol{N}$ is the matrix formed from derivatives of the shape functions $N^A(\boldsymbol{X})$ with respect to $\boldsymbol{X}$. The reader is referred to Zienkiewicz and Taylor [115] for a comprehensive treatment of the FE procedure.

We use an iterative Newton-Rhapson approach to solve (4.11). Given an initial state $\boldsymbol{u}_t^0$, the update equations are

$$\boldsymbol{u}_t^{k+1} \leftarrow \boldsymbol{u}_t^k - \boldsymbol{K}_T^{-1}(\boldsymbol{u}_t^k)\boldsymbol{f}_t, \tag{4.12}$$

where

$$\boldsymbol{K}_T = \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}} = \mathbf{A}(\boldsymbol{k}_{e,\mathrm{mat}} + \boldsymbol{k}_{e,\mathrm{geom}}), \tag{4.13}$$

is the linearized tangent stiffness. The element material stiffness is

$$\boldsymbol{k}_{e,\mathrm{mat}} = \int_{\mathcal{B}_e} \bar{\nabla} \boldsymbol{N}_e^T \boldsymbol{c} \bar{\nabla} \boldsymbol{N}_e \; dV, \tag{4.14}$$

where, $\bar{\nabla}$ is the gradient operator with respect to the spatial manifold, and $\boldsymbol{c}$ is the spatial material tangent, defined as

$$c_{ijkl} = \frac{1}{J} F_{iA} F_{jB} F_{kC} F_{lD} \mathcal{C}_{ABCD}, \tag{4.15}$$

the push-forward of the material tangent $\boldsymbol{\mathcal{C}} = 2\partial \boldsymbol{S}/\partial \boldsymbol{C}$. The element geometric stiffness is

$$\boldsymbol{k}_{e,\mathrm{geom}}^{AB} = \left(\int_{\mathcal{B}_e} N_{,i}^A T_{ij} N_{,j}^B \; dV \right), \tag{4.16}$$

where summation convention is implied, with lower-case subscripts indicating the spatial coordinates, subscript commas indicating partial differentiation, upper-case subscripts indicating the reference coordinates, and upper-case superscripts indicating nodal numbers. Note that the integrals for the stiffnesses are taken over the deformed element. The iterations are carried out until a stopping criterion, such as the satisfaction of (4.11) within some tolerance. For the Newton-Rhapson strategy to converge, the initial guess must be in the neighborhood of the solution. This requirement poses an issue for the highly nonlinear AV tissue, particularly in the low stiffness regime.

To address this problem, we apply the load incrementally and adaptively. We start with a small load factor α_t ($\boldsymbol{f}_t = \alpha_t \boldsymbol{f}_0$) and adjust the factor heuristically based on the number of iterations (n_i) it takes for (4.12) to converge ($\alpha_t \propto n_i^{-1}$). In this manner, we are able to circumvent the use of unreasonably small load factors (i.e., excessive computational time) during the entire load path. If a load factor is too large and the Newton-Rhapson algorithm diverges, we appropriately scale the load factor down.

For each individual layer we choose a Helmholtz free energy $\psi(I_1, J_4, J) := \hat{\psi}(\boldsymbol{F})$ of the form

$$\begin{aligned}\psi = C_{1m}\big\{ \exp\big[C_{2m}(I_1-3)\big]-1\big\} + \sum_{i=1}^{n_f}\frac{C_{1f}}{2C_{2f}}\Big\{ \exp\big[C_{2f}(J_4^i-1)_+^3\big]-1\Big\} \\ +c_1(I_1-3)+c_2(J^2-1)+c_3\ln(J).\end{aligned} \tag{4.17}$$

The first term on the right hand side is a Fung-like [25] isotropic term, where $I_1 = tr(\mathbf{F}^T\mathbf{F})$ is the first invariant, and C_{1m}, C_{2m} are material parameters. The second term is a directional term in the spirit of Holzapfel [39, 40] to account for the n_f collagen fiber directions, where $J_4^i = tr(\boldsymbol{C}\boldsymbol{M}_i)$ is the first mixed invarient for the fiber direction indexed by i, with $\boldsymbol{M}_i = \boldsymbol{m}_i \otimes \boldsymbol{m}_i$, $||\boldsymbol{m}_i||_2 = 1$, as the rank-1 structure tensor. C_{1f}, C_{2f} are material parameters and $(x)_+ := \max(x, 0)$ guarantees that the fibers do not take compressive load. The last three terms represent a Neo-hookean ground substance [35] with parameters c_1, c_2, c_3.

To establish the polyconvexity of (4.17), we turn to Appendices B and C of Schröder and Neff [89], namely, Lemmas B.9, C.2, and C.4. A necessary and sufficient condition for polyconvexity is then

$$C_{1m}, C_{2m}, C_{1f}, C_{2f}, c_1, c_2, -c_3 > 0. \tag{4.18}$$

Note that for the case $C_{1m}, C_{2m}, C_{1f}, C_{2f} = 0$, e.g., the spongiosa, the material is Neo-hookean, which is indeed polyconvex.

We impose that our material model, in the infinitesimal strain limit, recovers an isotropic linear elastic model.[1] Define the linearization operator as:

$$Lin\boldsymbol{f}(\boldsymbol{y})[\boldsymbol{u}] := \boldsymbol{f}(\boldsymbol{y}) + D\boldsymbol{f}(\boldsymbol{y})[\boldsymbol{u}], \tag{4.19}$$

where $D\boldsymbol{f}[\boldsymbol{u}]$ denotes the Fréchet differential in direction $\boldsymbol{u}$. We then linearize about $\mathbf{0}$ and impose that

$$Lin\boldsymbol{T}(\mathbf{0})[\boldsymbol{u}] = \boldsymbol{\sigma}_{\ell in}, \tag{4.20}$$

where $\boldsymbol{T}$ is the Cauchy stress tensor and $\boldsymbol{\sigma}_{\ell in}$ is the infinitesimal stress tensor. Equation (4.20) leads to the following conditions:

$$c_1 = \mu/2 - C_{1m}C_{2m}, \quad c_2 = K/4 - \mu/6 - C_{1m}C_{2m}^2, \quad c_3 = 2C_{1m}C_{2m}^2 - K/2 - 2\mu/3, \tag{4.21}$$

where μ and K are the infinitesimal-strain shear and bulk moduli, respectively. The stress and tangent are given by

---

[1] By construction, the fibers have negligible contribution in the infinitesimal regime.

$$\boldsymbol{T} = 2c_1\frac{\boldsymbol{b}}{J} + 2C_{1m}C_{2m}A_m\frac{\boldsymbol{b}}{J} + \sum_{i}^{n_f} 3C_{1f}(J_4^i - 1)_+^2 A_f \boldsymbol{b}_m/J + (2c_2 J + c_3/J)\boldsymbol{I}, \tag{4.22}$$

where, $A_f := \exp\left[C_{2m}(I_1 - 3)\right]$, $A_m := \exp\left[C_{2f}(J_4^i - 1)_+^3\right]$, and $\boldsymbol{b}_m = \boldsymbol{F}\boldsymbol{M}\boldsymbol{F}^T$. The material tangent (in the spatial configuration) is:

$$\begin{aligned} \boldsymbol{c} = \frac{1}{J}\Bigg[& 4c_2 J^2 \boldsymbol{I} \otimes \boldsymbol{I} - 2(2c_2 J^2 + c_3) I \\ & + 4C_{1m}C_{2m}^2 A_m \boldsymbol{b} \otimes \boldsymbol{b} \\ & + 6C_{1f}A_f\big(2(J_4 - 1)_+ + \sum_{i}^{n_f} 3C_{2f}(J_4^i - 1)_+^4\big)\boldsymbol{b}_m \otimes \boldsymbol{b}_m\Bigg], \end{aligned} \tag{4.23}$$

where $\boldsymbol{I}$ is the order-2 3×3 identity tensor and $(I)_{ijkl} = \frac{1}{2}(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk})$, ($\delta_{ij}$ is the Kronecker delta).

The last term in (4.17) represent the embedded fibers in a the material which we neglect in this analysis. Table 4.2 presents the material parameters used [11, 25]:

Model	μ [MPa]	K [MPa]	C_{1m} [Pa]	C_{2m} [-]
Arterial wall	0.334	16.67	26447.5	8.365
Plaque	0.334	16.67	5105.3	13.0
Lipid	0.334	16.67	50.0	5.0
Calcium	0.334	16.67	18804.5	20.0

Table 4.2: Summary of calibrated model parameters in (4.17) obtained from [11, 25].

Figure 4.1c shows a typical FE mesh. We use a 3-node constant strain element [115] discretized by the tessellation discussed above. A static pressure load is applied to the boundary elements in the lumen to simulate the effect of blood pressure. Nodes on the outside boundaries are restrained from displacement. All boundary nodes were detected with a boundary algorithm from DistMesh [80]. Solution of the problem was carried out with FEAP software [102]. Figure 4.1d shows a typical converged stress distribution.

Statistical Modeling

Data Preprocessing

From the database of FEM simulations, the input space included parameters describing geometry and arterial pressure and the output space included the pointwise stress distribution. We were interested in using arterial images and von Mises (VM) stress distributions in predicting max VM stress and location. Thus, we converted the parameters into 256x256 arterial images. For the stress distribution heatmaps, we used cubic interpolation from the finite element method data points, and clipped the VM values at 2000 (all values greater than 2000 were set equal to 2000). For each simulation, we then calculated the maximum VM stress (clipped at 5000), and the corresponding location, which was converted from cartesian coordinates to polar coordinates centered on the lumen.

Using an 80-20 split, the training and held-out test set had 9,737 and 2,435 simulations respectively. We further split the training data by randomly sampling 20% of the training data for the validation set. The validation subset was created to evaluate performance during model architecture and hyperparameter selection. The final test set evaluation was on predictions from selected models trained on the full training set. In addition, the training set was rotationally augmented by a random multiple of 45 degrees six separate times, producing a total of 58,422 not necessarily unique training data points.

The last pre-processing step was to normalize all the inputs and outputs from zero to one to ensure quicker convergence. The minimum and maximum were computed based off of the training set and those values were used to normalize both the training and testing set.

Loss Functions and Performance Metrics

The loss functions and metrics used in this experiment were the

1. mean absolute percentage error on maximum VM stress,
2. mean absolute error of polar distance (i.e. radius from lumen center),
3. a modified mean absolute error function for the angle of the point of maximum VM stress with respect to the center of the lumen.

For top-predicted point comparison between predicted max VM stress and actual max VM stress, the performance metric is equivalent to the loss function. As there could be multiple, disparate point-locations with VM stress values similar to the peak VM stress value, we also created soft metrics for post-processing. These incorporated stress points were utilized by measuring the mean absolute error of the top four VM stress polar coordinates that are within 5% of the maximum VM stress. We formally define the metrics below.

Let σ_{iv} be the max VM stress, $\hat{\sigma}_{iv}$ the predicted max VM stress, (l_{ix}, l_{iy}) the center of the lumen, (x_i, y_i) the position of maximum VM stress, and $(\hat{x}_i, \hat{y}_i)$ the predicted position of maximum VM stress. The polar angle loss, θ_{loss}, polar distance loss, r_{loss}, and stress value loss, σ_{loss} are defined below:

$$\begin{aligned}\theta_{loss} = \frac{1}{n}\sum_{i=1}^{n} \min\Bigg\{ &\left|\operatorname{arctan2}\left(\frac{\hat{y}_i - l_{iy}}{\hat{x}_i - l_{ix}}\right) - \operatorname{arctan2}\left(\frac{y_i - l_{iy}}{x_i - l_{ix}}\right)\right|, \\ &2\pi - \left|\operatorname{arctan2}\left(\frac{y_i - l_{iy}}{x_i - l_{ix}}\right) - \operatorname{arctan2}\left(\frac{\hat{y}_i - l_{iy}}{\hat{x}_i - l_{ix}}\right)\right|\end{aligned} \tag{4.24}$$

$$r_{loss} = \frac{1}{n}\sum_{i=1}^{n}\left|\sqrt{(\hat{y}_i - l_y)^2 + (\hat{x}_i - l_x)^2} - \sqrt{(y_i - l_{iy})^2 + (x_i - l_{ix})^2}\right| \tag{4.25}$$

$$\sigma_{loss} = \frac{1}{n}\sum_{i=1}^{n}\frac{|\hat{\sigma_v} - \sigma_v|}{\sigma_v} \tag{4.26}$$

For the “soft” metrics, let (x_i^j, y_i^j) be the position of j-th largest VM value where $j \in [1, 2, 3, 4]$. We define the following:

$$\begin{aligned} \theta_i^j = \min\{ &\left|\operatorname{arctan2}\left(\frac{\hat{y}_i - l_{iy}}{\hat{x}_i - l_{ix}}\right) - \operatorname{arctan2}\left(\frac{y_i^j - l_{iy}}{x_i^j - l_{ix}}\right)\right|, \\ &2\pi - \left|\operatorname{arctan2}\left(\frac{y_i^j - l_{iy}}{x_{ij}^j - l_{ix}}\right) - \operatorname{arctan2}\left(\frac{\hat{y}_i - l_{iy}}{\hat{x}_i - l_{ix}}\right)\right| \end{aligned} \tag{4.27}$$

$$r_i^j = \left|\sqrt{(\hat{y}_i - l_y)^2 + (\hat{x}_i - l_x)^2} - \sqrt{(y_i^j - l_{iy})^2 + (x_i^j - l_{ix})^2}\right| \tag{4.28}$$

The soft metrics for polar angle, θ_{soft}, and polar distance, r_{soft}, are defined as follows:

$$\theta_{soft} = \frac{1}{n}\sum_{i=1}^{n}\min_j \theta_i^j \tag{4.29}$$

$$r_{soft} = \frac{1}{n}\sum_{i=1}^{n}\min_j r_i^j \tag{4.30}$$

Lastly, we also calculated a relative Euclidean distance error metric as to show the average Euclidean distance error relative to the size of the artery. This metric is an average of the distance from the peak stress location to the predicted peak stress location normalized by major axis length of the arterial wall. The soft metric corollary was computed as well.

Model Approaches

We used Tensorflow and Keras to create the model architecture [1, 15]. Each model used rectified linear units (ReLU) for activation functions, and dropout layers were placed directly after any fully connected layers or convolutional layers (in image-based models) to prevent overfitting. Dropout layers were at uniform intervals between 0.1 and 0.5 in all of the models [97]. The adam optimizer was used to speed up convergence of our models [51]. We used mean absolute error as the loss for polar distance, mean squared error for max VM stress, and a custom angle loss function for polar angle as defined in the above subsection. The custom loss function was created to take into account the periodicity of circle angles and behaves similarly to mean absolute error. Lastly, a separate neural network was created to predict the max VM stress with an identical architecture to the neural network for the polar distance and angle.

The following descriptions are the final models we created after the model selection phase which consisted on evaluation on a validation set before testing on the test set. The approaches are graphically summarized in figure 4.2.

Approach 1 uses the parameterized ellipses in the form of 34 numbers and predicts polar distance and angle of the location of max VM stress. The model architecture involved three sets of a fully connected layer followed by a dropout layer. After the first fully connected and dropout layer, the model bifurcates into two branches: one for polar distance and one for polar angle. Training ran for 35 epochs.

Approach 2s model used the arterial image and pressure to predict the original parameters, polar distance and polar angle. The architecture involved 3x3 convolutional layers, 2x2 max pooling layers and a dropout layer, followed by three branches of three sets of a fully connected and a dropout layer. Then, we performed transfer learning on the model by first training it to predict the original parameters, and then eventually learning the polar distance and angle. Training ran for 15 epochs.

Approach 3 predicts the polar distance and angle based off of the arterial image. The architecture involves two sets of 3x3 convolutional layers, 2x2 max pooling layers and a dropout layer, followed by two branches of three sets of fully connected layers and a dropout layer. Training ran for 15 epochs.

Approach 4s model attempts to predict polar distance and angle from the original parameters, using the heatmap as an intermediary. The architecture involves three sets of fully connected and dropout layers, then bifurcated into three branches, two of which have three more sets of a fully connected and dropout layer. The last branch uses upsampling and convolutional layers to recreate the heatmap. Transfer learning was used to first learn the heatmap and then the polar distance and angle. Training ran for 30 epochs.

Using arterial images, Approach 5s model predicts polar distance and angle. The architecture involves 3x3 convolutional layers, 2x2 max pooling layers, and a dropout layer, followed by two sets of fully connected and dropout layers. The model then bifurcates into three branches, two of which have three more sets of fully connected and dropout layers followed by either the polar distance or angle output layer. The last branch uses upsampling and convolutional layers to recreate the heatmap. Finally, transfer learning was used to learn the heatmap first and the polar distance and angle second. Training ran for 30 epochs.

Comparative Study of Predictive Models

One goal was to evaluate different prediction model techniques of varying complexity to determine whether deep learning was the most appropriate model for this problem. The three baselines we were interested in were random guess, mean guess, and support vector machine regression (SVR). Random guess uniformly guesses between the minimum and maximum value, while mean guess only guesses the average value of the training set. Lastly, we experimented with various kernels for SVR, and came to the conclusion that radial basis

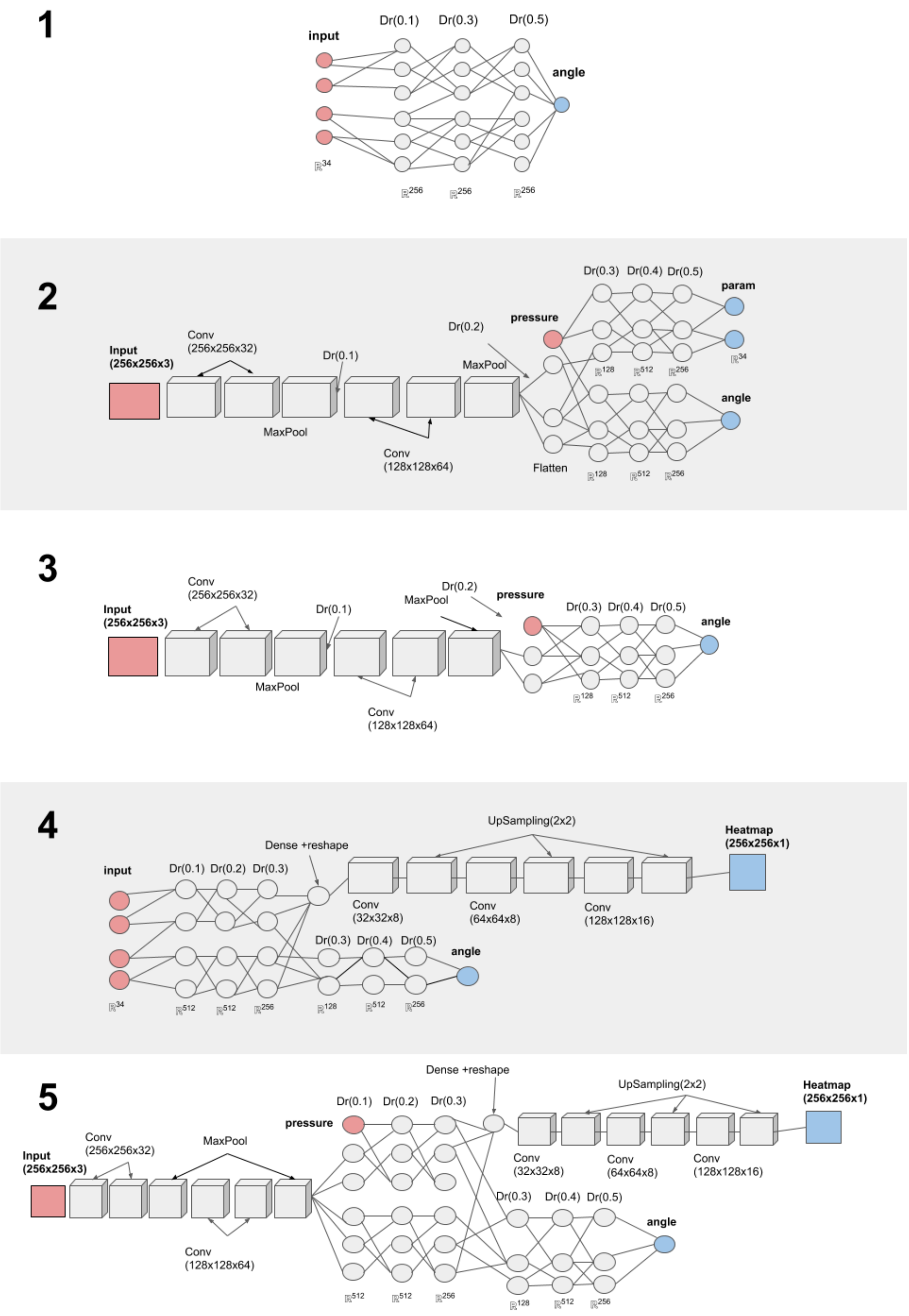

Figure 4.2: Model architecture of five approaches taken for training a supervised deep learning model. Approach number ranges from 1 to 5 from top to bottom. Red coloring denotes input data fed to network. Blue coloring denotes output data used for loss calculation.

function (rbf) kernel was the most effective. The final metrics were calculated on the testing set.

Feature Importance Study

The purpose of this study was to evaluate the correlated relationships between various input features (like lipids and calcium) and the labels (maximum VM stress and location).

Our first adjusted feature was (1) pressure, which we replaced with uniform noise with min and max values determined by the min and max values of our training data. We then adjusted our (2) lipid pool representation, which we replaced with zeros, and then applied the same function for (3) calcium deposits. We then randomly (4) switched our lipid pool and calcium deposit representations, and lastly (5) replaced both lipid pools and calcium deposits with zeros. The deep learning model was trained with each of the adjusted training sets and then evaluated using the same metrics on the unchanged testing set.

Amount of Data Study

Deep learning is a data-intensive process whereby inference performance scales with amount of training data. To determine whether more data is needed to optimize performance, we adjusted the training set by using various amounts of data (increments of 20%), trained the same model with the adjusted training set, and evaluated the model using the same metrics described above.

4.3 Results

Generation of finite element simulation results as training data

A parametric model was developed to efficiently generate multiple idealized geometries of atherosclerotic plaques as shown in figure 4.1c. Topologically feasible configurations were assessed by using a support vector machine to check for collisions. These geometries along with material model properties and boundary conditions were then modeled with the finite element method as described in the Methods. We were able to successfully generate 12000 simulations of aforementioned conditions. Figure 4.1d shows an example distribution. A set of matrices were then formed for the machine learning model. The inputs were either an array of parameters or image-based matrices. The outputs were label matrices of various mechanics metrics of interest such as maximum von mises stress. Intermediary outputs such as the 2D von mises stress distribution were also created. These were then split to an appropriate training/validation and held-out test sets for machine learning model selection and evaluation. Full details are described in the Methods.

Deep learning can capture a statistical model for prediction tasks in solid mechanics

Our best model (Approach 3), which estimates location and VM stress from the arterial image, predicts maximum VM stress to within 10% of the maximum VM stress and 0.86 radians within the true location. The loss graph from figure 4.3a shows that the training and validation error converge near each other, implying overfitting is negligible. Figure 4.3b shows the distribution of predictions for maximum VM stress is similar to the actual values. Figure 4.3c-d demonstrates the maximum VM stress predictions have a very similar distribution, and the predicted VM location looks similar albeit more conservative than the test-set locations.

Data representation is important and affects learning ability

As described in Table 3, Approach 3, which estimates the maximum VM stress and location from the arterial image, and Approach 5, which estimates the labels from the arterial image using the VM heatmap as an intermediary, were the best performing models for predicting maximum VM stress location. Approach 5 predicted the polar distance better than Approach 3 with an average peak polar distance error of 0.157 mm and soft error of 0.156 mm while Approach 3 scored 0.160 mm and 0.159 mm on the peak and soft metrics respectively. On the other hand, Approach 3 predicted polar angle better with the peak angle metric of 0.857 radians and a soft metric of 0.598 radians compared to the 0.876 and 0.609 radians of Approach 5.

The approaches that used the arterial images as an input (2, 3, and 5) performed better at predicting max VM location than those that used the parametric geometry model parameters (Table 1) as an input. Approaches 2, 3 and 5 perform between 0.85 radians and 0.89 radians for the peak angle metric and 0.15 to 0.19 mm for the peak distance metric, while Approaches 1 and 4 perform between 0.91 and 0.93 radians and 0.21 and 0.23 mm respectively. Only Approach 1, which estimated the maximum VM location and stress from the 34 parameters, performed better than Approach 3 for VM Magnitude. Approach 1 had a mean absolute percent error of 9.86%, with Approach 3 trailing behind at 10.42%.

Feature importance studies can shed light on potential determinants

As tabulated in Table 4, when Approach 1 was trained with a dataset with random pressure values instead of the true pressure values, the VM Magnitude metric jumped from 9.86% to 23.98%, while the polar distance and angle metric stayed the same. This confirms our intuition that input arterial pressure is the greatest contributing factor to max von mises stress among input parameters.

Randomly switching lipid and calcium information had a negligible influence on the VM Magnitude metric, but increased the mean absolute error for the peak and soft metrics

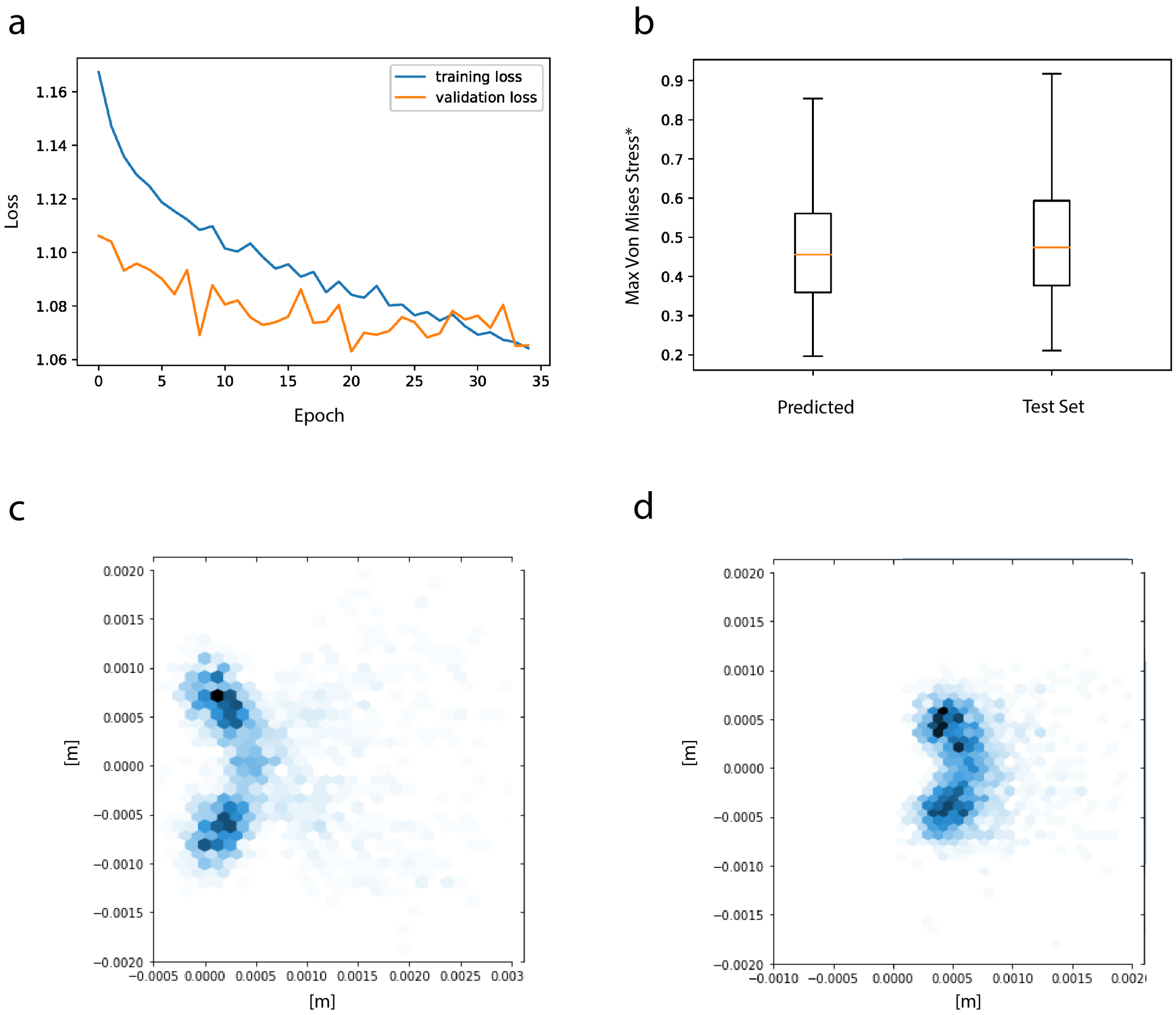

Figure 4.3: Deep learning can accurately predict physiologically-relevant indicators (maximum von mises stress and location) for a given vessel geometry and pressure. a) Training and validation loss over 35 epochs displaying convergence and behavior of model training. b) Normalized maximum von mises stress values for the held-out test set and predictions by the ML model. c) 2D heat map of coordinate location of maximum von mises stress over all the held-out test set samples. d) 2D heat map of predicted locations of maximum von mises stress on the held-out test set.

by 0.014 radians and 0.05 radians respectively. Furthermore, removing calcium from the parameters had a larger influence on the accuracy of the VM location than removing lipids. Removing lipids increased the peak mean absolute error from 0.915 to 0.993 radians and the soft error from 0.663 to 0.804 radians, while removing calcium from the parameters increased the peak mean absolute error to 0.965 radians and the soft error to 0.778 radians.

However, lipids have a larger influence on VM magnitude than calcium. Removing lipids increased the mean absolute percentage from 9.86% to 16.2% while calcium raised it to only 12.5%. Removing both lipids and calcium had a larger influence on all three statistics than removing only lipids or only calcium as expected.

Deep Learning outperforms other prediction models and its performance scales with amount of data

As shown in figure 4.4a, the deep learning model outperforms other predictive baseline strategies such as mean guess and support vector machine regression. The decrease in error is consistent across both maximum VM stress and location.

In figure 4.4b, we examine the effect of training dataset size on performance. Between 20% and 60% of the original dataset, the metric for peak angle decreases linearly but between 60% and 100%, the slope decreases. For the location of max VM stress, the metric does not seem to have qualitatively fully converged, meaning there is still room for additional training data to increase model performance. For the maximum VM stress value, the metric does seem to have likely converged as the difference between 80% and 100% of the data is about 0.05%. Additional training data will likely have a greater effect on location prediction performance.

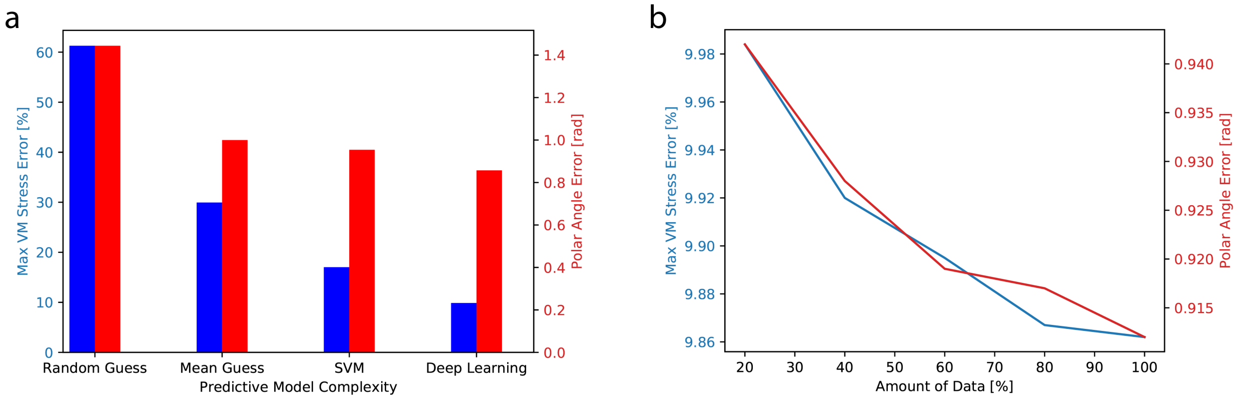

Figure 4.4: Comparative studies on varying baseline models and training dataset size. a) Deep learning model outperforms other baseline and machine learning strategies. b) Varying amounts of data are utilized for training the same model to examine performance vs data size.

4.4 Discussion

As training deep learning models can be data-intensive, our parametric model enabled us to generate a large dataset of topologically feasible geometries. This allowed the model to witness a large variation of different scenarios with their respective outputs. We were able to successfully parallelize our pipeline to both generate geometries and solve the finite element calculations to develop a sizeable database of input-output pairs. Our data could be further improved by adding different shape distributions as well as random noise distortions to the shape lines. The next step would be to include real patient data, i.e. from intravascular ultrasound [71], as either testing and/or additional training data. The choice of real versus simulated geometries is not exclusive. The incorporation of real and synthetic data is an active area of research that could be especially relevant when attempting to learn physics mechanisms [95].

Consistent with breakthroughs in the application of machine learning (ML) to multiple domains, deep learning seems to learn an underlying functional mapping that is solved by FEM with minimal assumptions regarding its structure. Our prediction task, as opposed to object recognition or other canonical ML tasks, is unique in that the machine learning model needs to learn the underlying physics at play. We are encouraged to see that our models are able to perform reasonably well at learning physics, particularly the prediction of maximum von Mises stress. Our best performing models are able to predict the peak von Mises stress magnitude with an average error less than 10%, peak stress polar angle with respect to lumen center by 34.3 degrees, and a peak stress relative Euclidean distance error of 12.2% with respect to the vessel axis. The performance of stated techniques also scale with amount of training data which can be efficiently augmented with our finite element simulation generation pipeline to improve stress prediction performance even further.

The exact mapping of solid mechanics is difficult to interpret with deep learning models. Some progress is being made with respect to model interpretability as a whole [12]. Particularly interesting would be investigating the manner and patterns learned by the ML model. It could potentially shed light on whether the model is learning a similar approximation to the partial differential equations at-hand. It might also reveal the model has learned differently by developing an implicit coarse-grain model or employing a fast, nearest neighbor search by implicit memory [114].

In our study however, we were able to gain confidence in our model by a couple different practices and results. We appropriately divided the data into training/validation and held-out test sets with sample sizes larger than the thousands. Our training and validation loss curves do not exhibit large gaps, indicating a lack of observable overfitting. The comparison of our model performance with other baseline prediction models is satisfactory. Lastly, our feature importance study generally affirms intuitions behind causal determinants.

Our results demonstrate the importance of data representation in effective model training. The model approaches utilized different input representations and intermediary outputs. Based on the prediction task and metric of interest, the choice of type of input data can vary. Overall, vision-based approaches seemed to outperform parameter-input approaches for maximum von Mises stress location prediction. Convolutional neural networks are especially well-suited for computer vision and learn both low- and high-level patterns in the training data.

The clinical importance of this study exists in two dimensions: automation for precision medicine and exploration of clinical insights. The subject of our study is regarding atherosclerosis, one of the largest health concerns globally. Precision medicine could provide a unique avenue to help with clinical evaluation and treatment. More specifically, the evaluation of patient-specific stress distributions could be informative for patient risk assessment. Physicians can currently use invasive and non-invasive techniques to characterize geometry and material properties of arteries [11]. However, it is not scalable to require a specialist to formulate finite element simulations for each patient. Our study, by successfully predicting maximum von Mises stress for arterial geometries, lays the groundwork to utilize deep learning for automatic stress distribution analysis while the patient is being seen by a physician. This type of information delivered to the physician in real-time could greatly improve assessment and treatment for conditions such as plaque rupture.

In addition, deep learning can enable the exploration of clinical insights gleaned from pattern recognition across large data samples. In our feature importance study, we see intuitive correlations such as between arterial pressure and maximum von mises stress. However, we also observe that the presence of lipid pool information, in comparison to calcium deposits, has a greater effect on maximum von Mises stress and a lesser effect on the location of the maximum point. This could be put into context by experimental results such as [41]. Deep learning models can potentially be utilized to develop better insights on the nature of disease itself.

In this work, we attempt to bridge the gap between finite element and machine learning modeling by using a clinical use-case of prediction of maximum von Mises stress for atherosclerotic arteries. In essence, we utilize deep learning to learn the underlying physics of our solid mechanics problem as determined by the finite element method. We demonstrate the strength of this approach on a widespread biomechanics problem with clear impact. We hope to lay the groundwork for further theoretical and applied investigation in this domain.

Chapter 5

Bridging Molecular Dynamics Modeling and Graph Neural Networks for CH Domain Conformational Mechanics

5.1 Introduction

The primary sequence of a well-ordered protein encodes its structural information as well as its functional properties. Proteins undergo dynamic shifts in conformation to perform their designated tasks. The growth of computational resources and simulation algorithms allow for studying molecular trajectories at atomic resolution commonplace. However, because of the heterogeneity of bio-molecules and their chemical environments, computational models can suffer in performance when tasked with predicting molecular conformations, physical and chemical properties, and protein-protein interaction networks. Along with the growing excitement in machine learning in tangent fields, there is emerging interest in data-driven approaches to studying biological function at scales ranging from biochemical properties to tissues mechanics [30, 7, 90, 28, 65, 113, 83].

Mechanosensitive proteins are central to a plethora of biological phenomena [26]. Our goal in this paper is to bridge the primary sequence of proteins to their intrinsic force observables while undergoing large-scale conformational changes. To accomplish this, we generated a molecular dynamics database of trajectories of two calponin homology domain mutants being pulling apart [Figure 5.1] by Steered Molecular Dynamics (SMD) [43]. SMD is a technique that allows for biasing the interaction potential between explicit regions in a simulation; in our case, two globular domains, CH1 and CH2, are gradually pulled apart during the simulation. While these simulations are computationally expensive and require careful consideration for proper physical dynamics, they provide information at the atomic resolutions about the sensitivity of the roles mutations play in response to mechanical loads in proteins. Our aim to cast protein structures as graphical models allows us to take the first imperative steps towards building relationships between sequences and physical observables that can be computed with more expensive techniques such as molecular dynamics. We show that the force between the two CH domains, a global property over a long simulated trajectory, can accurately be predicted using graph neural networks.

5.2 Methods

Generating a database of point-mutated protein dynamics

To generate models to predict protein dynamics, we established the first ProDyn dataset with high-throughput *in silico* mutagenesis experiments of proteins undergoing large-scale conformational changes. The generated trajectories span 2020 unique point-mutations along two calponin homology domains, CH1 and CH2, connected by a tether that comprise a total of 224 residues. The data per simulation was processed to yield a graphical input representation of residue interactions in terms of node and edge features, in addition to an output representation for force profile classification and regression. We then construct and compare conventional and graph neural network architectures to predict the force characteristics of our biophysical system.

Molecular Dynamics - A Primer

Although the continuum assumption can be applied to mechanics at the organ, tissue, and even (in some applications) the cellular scales, it does not hold as we approach molecular scale phenomena. Molecular dynamics (MD) simulations, which are rapidly progressing due to increasingly powerful computational capabilities, must be used to analyze molecular-scale interactions. An MD simulation is a system of Newtonian equations of motion, where the Newtonian equation is special case of the Lagrangian equation of motion for mass points in a Cartesian system. The fundamental equation relating the force in terms of accelerations can be expressed as the gradient of the potential energy as follows:

$$\boldsymbol{F}_i = m_i \ddot{\boldsymbol{r}}_i = -\nabla U_i, \tag{5.1}$$

where $\boldsymbol{F}_i$, m_i, $\boldsymbol{r}_i$, and U_i are the force, mass, position and potential of particle i, respectively. Therefore, to calculate the forces acting on the atoms, the potential energy of the system must be known. The non-bonded potential energy terms are traditionally considered for the pair potential, $U(r_i, r_j) = U(r_{ij})$, and neglected for higher order interactions. One of the most commonly used potentials is the Lennard-Jones potential with two parameters: σ, finite diameter at which the inter-particle potential is zero, and ε, the potential well depth:

$$U_{i,j} = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]. \tag{5.2}$$

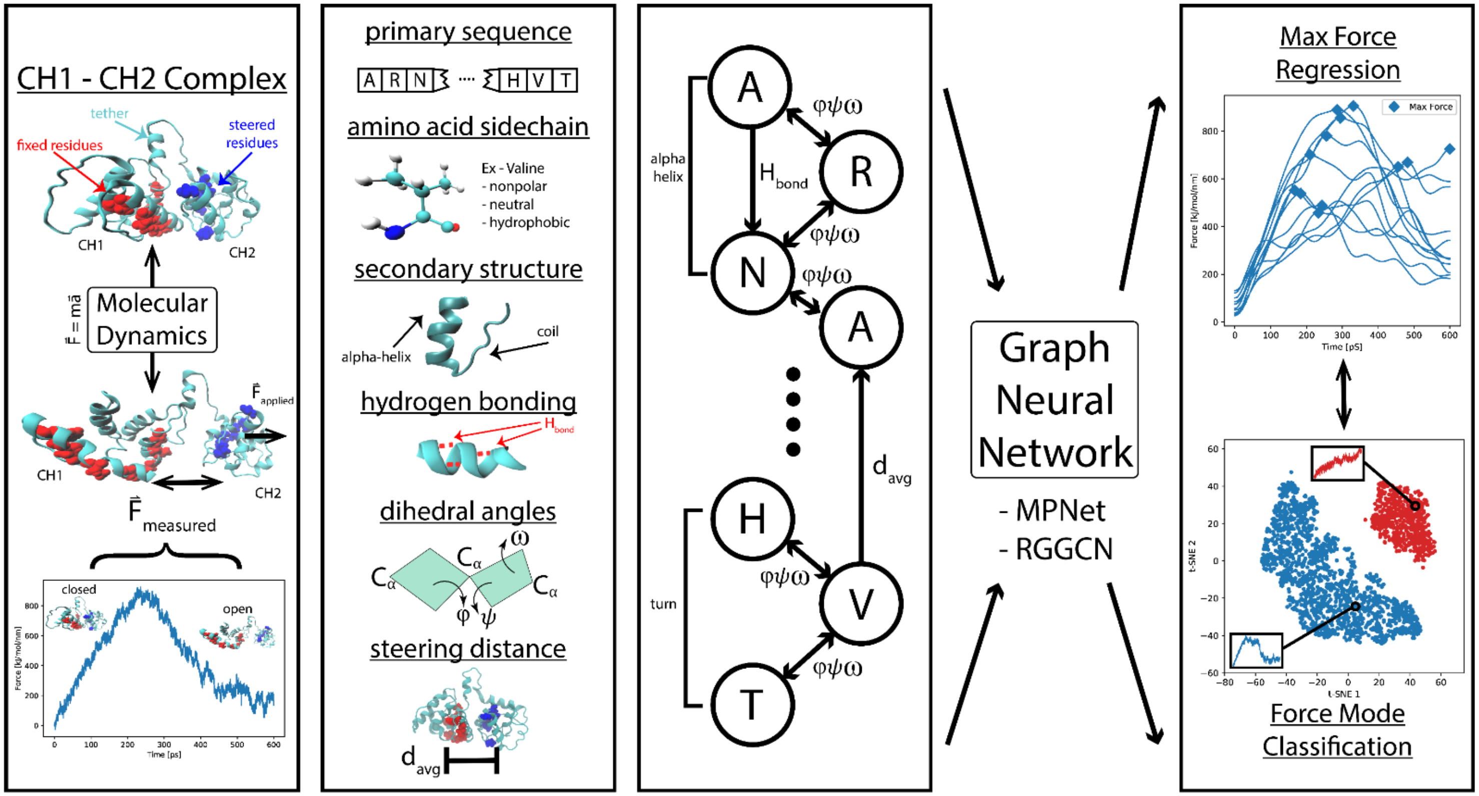

Figure 5.1: Workflow Schematic: From Biophysics to Graph Neural Networks

In addition, there are a variety of bonding potentials that describe the intramolecular interactions. Force fields have been constructed to describe the complex interactions between atoms through the parameterization of potential functions via experimental data or *ab initio* and semi-empirical quantum mechanical calculations. The functional form of a force field such as AMBER [19] or CHARMM [10] can be described by the following terms:

$$U_{field} = U_{bond} + U_{angle} + U_{torsion} + U_{improper} + U_{vdW} + U_{elec}, \tag{5.3}$$

where each term is representative of bond energies, angle, torsional effects, van der Waal's, electrostatic. For a system composed of atoms with positions, potential energy, and kinetic energy (derived from atomic momenta), the classical equations of motion will be a set of coupled ordinary differential equations. Methods exist, such as the original Verlet, velocity Verlet, and predictor-corrector algorithms, that perform the step-by-step numerical integration. [78, 2, 104] With the proper selection of parameters and algorithm, MD techniques are able to accurately reproduce macroscopic properties of the system.

Mutagenesis and Steered Molecular Dynamics

The creation of the mutation library was primarily informed by a BLOSUM95 matrix displaying frequencies of swaps between pairs of residues in closely-related protein homologues [38]. In addition, an alanine scan over most of the sequence was made to complement work done by saturation mutagenesis experiments [111]. To augment our data set beyond this, a random sample of positions were mutated to random residues with no further bias.

To generate the database of single point-mutated calponin homology domain trajectories, all-atom molecular dynamics was performed. Experimental procedures were largely adapted from previous work documenting the SMD experimental protocol for the wild-type CH1-CH2 domain [93]. The wild type structure from Shams et al. was modified at single residue locations using a MODELLER (https://salilab.org/modeller/) script that swaps residues, optimizes dihedral angles, reduces bad contacts with conjugate gradient steps, and performs a few steps of molecular dynamics to allow the swapped residue to relax [87].

All subsequent molecular dynamics was performed with GROMACS [103, 60, 6] using the CHARMM27 all-atom force-field. Pulling simulations were performed in a 11.5 x 11.5 x 13 nm3 box solvated with SPC216 water molecules and 100 mM KCL. The system was minimized with steepest-descent to remove bad contacts until the total energy was less than 1000.0 kJ/mol/nm. Periodic boundary conditions were set in all 3 directions, a Verlet cutoff scheme truncated non-bonded long-range interactions at 1.2 nm, and PME electrostatics with Fourier spacing of 0.12 nm was used. Following minimization, each system was equilibrated for 100 ps in NVT followed by 100 ps of NPT, both using 2 fs time steps and a target temperature of 300K and 1 bar, respectively. The equilibration was closely monitored for each mutation to ensure that the system was properly sampling the correct equilibrium distribution before steering experiments were performed.

SMD experiments were performed by anchoring down and pulling on a selected group of heavy carbon atoms on the interface between the CH1 and CH2. To maintain this experimental control, residues that include these atoms were not mutated in our experiments. All other parameters of our simulations is held constant except for a single point mutation that is unique to each simulation. The reaction coordinate distance is defined here as the distance between the center of masses of the fixed and steered residues. Steering was performed by moving an umbrella potential with stiffness 1255 kJ mol-1 nm-2 at a rate of 2 nm/ps along a reaction coordinate defined by the axis connecting the center of masses of the fixed and steered atoms. Temperature coupling was performed by a Nose-Hoover thermostat targeting 310K and coupling was maintained at 1 bar with an isotropic Parrinello-Rahman piston for the duration of the pulling experiment. Throughout the 600 ps pulling experiment, readouts of the reaction coordinate distance and the force between the fixed and steered atoms were measured. The maximum force applied between the domains as well as the entirety of the force profile are the observables of interest predicted with our neural networks. Figure 5.1 shows the force profiles can be vary dramatically in shape for different mutations, while remaining relatively stationary for different initial conditions per mutation.

Data Pre-processing

A molecular dynamics (MD) simulation under a given force field outputs a trajectory file that includes type, position, and velocity information for every atom in the system for each time step. There are several prediction problems of interest that can be encapsulated by our data. In this study, our specific objective is to predict the force characteristics for a given protein initial state (i.e. residue information and interactions pre-steering).

The input representation is a directed graph, G, with node features, x_v and edge features, e_{vw}. As shown in table 5.1, node features comprise of the residue type, residue properties, and initial secondary structure type as calculated by mdtraj [70]. Edge features currently consist of pair-wise dihedral angles, Kabsch-Sander (K-S) hydrogen bonds [46], and the initial distance between steered and fixed residues.

The output representation, a graph-level target of protein mechanosensitivity, is of two types: regression and classification. Each SMD simulation exhibits a characteristic force response over time of the simulation. The regression target is a 2 dimensional vector encoding maximum pulling force magnitude and maximum pulling force time-point respectively. The classification target is a one-hot encoded 2-dimensional vector that encodes a force mode category. The force mode categories were ascertained via spectral k-means (k=2) clustering on the force-response graphs in a lower dimensional t-SNE [62] derived space as shown in Figure 5.1. There emerged two clearly well-separated clusters which we utilized as categories for graph-level classification. Lastly, all continous data was scaled to [0,1], and the entire dataset was split randomly into a 404 sample held-out test set and 1616 sample 80-20 training/validation split. For the classification task, we compute the accuracy and macro-averaged F1-score as performance metrics. For the regression task, we calculate the mean absolute error (MAE) for maximum force magnitude and maximum force time-point.

Feature	Node/Edge	Type	Index	Examples
Residue Type	Node	One-hot	[0 : 20]	ALA, GSP, ALY, ...
Residue Secondary Structure	Node	One-hot	[21 : 26]	alpha-helix, turn, ...
Residue Properties	Node	Multi-label	[27 : 43]	acidic, polar, ...
Dihedral-ϕ	Edge	Continuous	[0]	$0 - 2\pi$
Dihedral-ψ	Edge	Continuous	[1]	$0 - 2\pi$
Dihedral-ω	Edge	Continuous	[2]	$0 - 2\pi$
Hydrogen Bond	Edge	Continuous	[3]	K-S energy
Steer-to-Fixed Residue Dist	Edge	Continuous	[4]	average distance

Table 5.1: Input Representation Features

Baseline Conventional Deep Learning Models

As a comparison to graph-based neural networks described below, conventional multi-layer perceptron (MLP) and gated recurrent unit (GRU) were trained [14]. The graph node features x_v were provided as input either by concatenation or sequentially. For the MLP architectures with one vector input of concatenated features, the number of layers [1-5], activation functions [ReLU, eLU], nodes per layer [128-512], dropout regularization [0.0-0.5], and loss functions [L1, L2, logcosh, cross-entropy] were experimented with. For the GRU+MLP architecture with inputs features passed sequentially through the GRU, we experimented with different GRU implementations- varying hidden dimensions [16-64], stacking [1-2], bi-directionality, regularization [via dropout], and outputs [last layer depth-wise vs time-wise].

Neural Message Passing Network Model

In line with [29], we developed various neural message passing network (MPNet) architectures. Our models learn to compute a function of the entire graph by learning features, invariant to graph isomorphisms, through a message passing algorithm. We define the hidden state of each node in the graph by h_v^t with $h_v^0 = x_v$ where v denotes residue/node number and t denotes the cycle number, and the directed edge from node w to v as $e_{v \leftarrow w}$. The message passing scheme runs for a total of T cycles which can be interpreted as the message-passing neighborhood size. During each cycle, messages are passed between a neighboring node and the current node based on the edge direction. There is a unique message function, M, that computes the message given the current node features, neighboring node features, and edge features which is then aggregated into one message update per node, m_v^t. This message is then used to update the hidden node state by the function, U. The final hidden states of all nodes are then passed through a readout function, R, which yields an output layer of the dimension of the sample target. To formalize, for each message passing cycle the node states are updated in the following fashion:

$$m_v^t = \frac{1}{|N(v)|} \sum_{w \in N(v)} M(h_v^t, h_w^t, e_{v \leftarrow w}) \tag{5.4}$$

$$h_v^{t+1} = U(h_v^t, m_v^t) = h_v^t + m_v^t \tag{5.5}$$

where $N(v)$ denotes the neighbors of v in graph G. The message function, M, is a 3-layer MLP with 0.4 dropout, 256 hidden layer dimensions, and output layer of the same dimension as node features. After T cycles, the readout function R is applied as follows:

$$\hat{y} = R(\{h_v^T | v \in G\}) \tag{5.6}$$

We did not see a significant increase in performance for increasing T cycles and present our MP models for T=1 in Results. For the choice of R, we experimented with a max-pooling operation over the feature dimension and also feeding the nodes in primary sequence order to a GRU. The final output time-wise of the GRU is fed to a 2-layer MLP with 0.5 dropout. The final loss function, $L(\hat{y}, y)$ was either a L1-norm for regression or softmax+cross-entropy for classification. Models were trained using stochastic gradient descent with ADAM [51] and batch size of 25.

Residual Gated Graph ConvNet Model

Residual Gated Graph ConvNets (RGGCN) is a method introduced by [9] that combines the vanilla graph ConvNet [99] and the edge gating mechanism [68] with residual connections for problems involving variable graphs. We extend this method by incorporating edge features, such that each graph convolution layer follows the layer-wise propagation rule:

$$h_i^{\ell+1} = \mathrm{ReLU}\left(U^\ell h_i^\ell + \sum_{j \to i} \eta_{j \to i} \odot \left(V^\ell h_j^\ell + D^\ell e_{j \to i} \right) \right)$$

where h_i^ℓ denote features of node i at layer l, $e_{j \to i}$ denote features from edge $j \to i$, and edge gates $\eta_{j \to i} = \sigma\left(A^\ell h_i^\ell + B^\ell h_j^\ell + C^\ell e_{j \to i} \right)$. U, V, A, B, C and D are learnable parameters.

In our experiments, we used a 4-layer RGGCN and applied max pooling as our readout function. For classification, our model had 50 hidden dimensions and a single linear layer. For regression, our model had 256 hidden dimensions with a 2-layer MLP. Batch normalization was employed, as with residual connections between successive convolution layers.

Our models were initialized using Glorot initialization [31]. We used the Adam SGD optimizer with an initial learning rate of 0.0005, and computed loss using binary cross-entropy for classification, and L1 for regression. Dropout rate was set to 0.4 for both node features and edge gates in classification, and only for the 2-layer MLP in regression.

5.3 Results and Discussion

One of our main achievements is creating a database that is uniquely well-positioned to benchmark the rapid development and evaluation of novel graph neural network architectures on both a graph-level classification and regression task. As shown in table 5.2, our baseline non-graph-based deep learning models– both (1) variations of concatenating features and processing through an MLP and (2) sequentially feeding primary sequence-ordered nodes through a GRU – were unable to learn effectively and predicted tightly near the majority/mean target. Within our data size scale, the complexity of many biophysical phenomena, through initial states and short/long-range bonded and non-bonded interactions partially captured in this study, is too difficult to train effectively unconstrained via an MLP or recurrently via a GRU.

Model	Accuracy	F1-Score	Force$_{mag}$ MAE	Force$_{time}$ MAE
Conventional	70.54	0.414	89.68	111.39
MPNet + Maxpool	77.09	0.673	85.39	92.44
MPNet + GRU	**86.63**	0.839	**81.59**	**76.99**
RGGCN + Maxpool	**86.39**	**0.904**	89.00	100.52

Table 5.2: Prediction Performance of Various Architectures

Our goal was to experiment with 2 types of graphical neural networks: our own designed MPNet and a state-of-the-art graph technique such as the RGGCN. Both graph neural network architectures, MPNet and RGGCN, capture the graphical structure, information communication, and structural invariances to learn an effective node embedding for classification/regression tasks. In end-to-end training after node embedding, a readout function is applied that is agnostic to the number of nodes/residues. Among functions invariant to graph isomorphism, we observed that maxpool outperformed average pool. As proteins have multiple hierarchies of structure, we can view the MPNet with GRU readout as encoding the protein structure hierarchy with the GRU applied after message passing cycle, as to allow message passing interactions to be communicated between nodes before the recurrent primary structure network. The choice of GRU readout is likely more effective for protein-like graph systems; the RGGCN+Maxpool would be more generalizable otherwise. As shown in table 5.2, all graph neural networks significantly outperform conventional deep learning techniques. The RGGCN+Maxpool and MPNet+GRU perform well on the classification task at around 86%. For the regression task, the MPNet+GRU is able to learn with a MAE of 81.59 $kJmol^{-1}nm^{-1}$ MAE on max force magnitude prediction and 76.99 $psec$ MAE on max force time prediction. To put into context the performance to the natural stochasticity of the MD system, we ran duplicate SMD simulations and observed the mean standard deviation of the max force mag was 57.94. Also, in figure 5.2, we show the well-behaved convergence of the classification model training in addition to the resemblant output statistics of the true and predicted targets for regression.

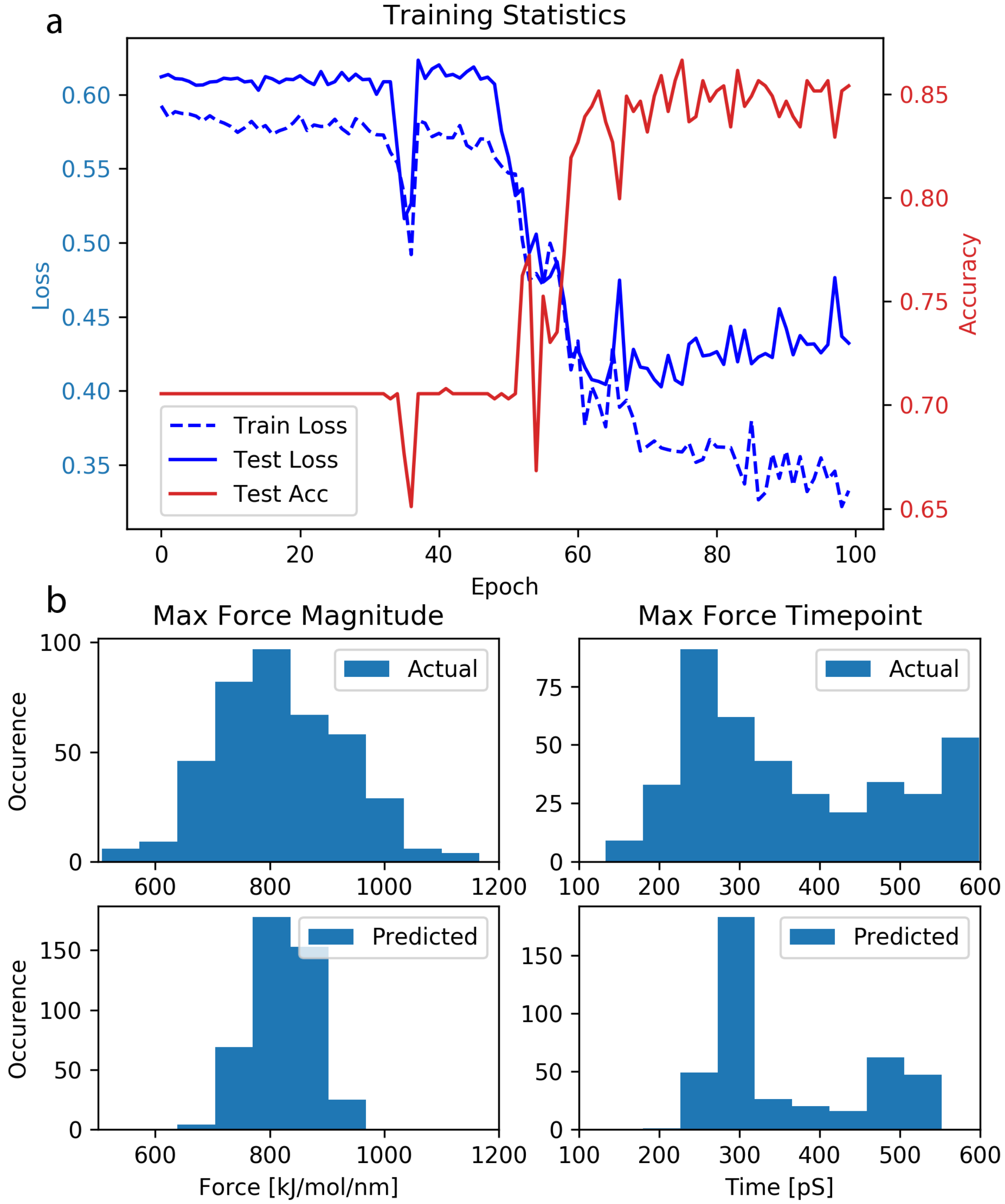

Figure 5.2: MPNet + GRU Performance. a) Classification training statistics b) Regression histograms of predicted vs actual values

Future work can include learning edge states in addition to node states, set2vec as readout function [106] or GRU as an update function, and experimenting with unsupervised graph embedding techniques such as [105]. From the biophysical perspective, we can expand the choice of node/edge features, try different protein systems, or formulate a plethora of pertinent biophysical prediction tasks.

To conclude, the progress of geometric deep learning techniques is inextricably linked to the quality of benchmark datasets– analagous to the impact of benchmark datasets for vision and language. We view our work as laying the groundwork for the intersection of graph neural network and biophysics researchers for the advancement of both fields.

Chapter 6

Conclusion

6.1 Synopsis

The nature of the work above is inherently interdisciplinary. We start by addressing clinical assessment and diagnosis via the development of computer vision algorithms for medical imaging, specifically echocardiography. The initial study validated our ability to use deep learning to predict cardiac anatomical orientation quickly and accurately with greater than physician-level performance. The subsequent study delves deeper into the comparison of state-of-the-art data-efficient and semi-supervised techniques that can be applied for patient diagnosis and disease characterization purposes generally across medical imaging. Clinically, we extended the work to be able to predict left ventricular hypertrophy.

We then move into addressing atherosclerosis and plaque rupture prediction via stress analysis with deep learning. Traditionally, numerical methods such as the finite element method are utilized to be able to perform mechanical analysis. The drawbacks of these techniques for clinical adoption to have real-time mechanical analysis available for clinicians are the computational constraints and non-automated specialized work required to deliver accurate predictions. We attempt to bridge the fields of machine learning and finite element (FE) analysis to learn the underlying mapping described the FE process, and more clinically predict maximum von Mises stress and associated factors as an potential indicator for plaque rupture risk.

Lastly, a *deep* understanding of cardiovascular disease necessitates the exploration of phenomena occurring at the cellular and molecular scales. Our last endeavor engages in utilizing graph neural networks, in the paradigm of geometric deep learning, to predict the biomechanical behavior of mechanosensitive proteins that are normally computationally modeled via molecular dynamics simulations. The advancement of techniques to model proteins, such as the calponin domains responsible for cytoskeletal integrity, can enable us to better understand the nature of cardiovascular disease and assist in the development of therapeutics by communities such as in protein engineering.

6.2 Future Work

There are a multitude of natural extension and future developments of the body of work presented in this study– ranging from clinical to applied/engineering to scientific. I will limit the discussion to more direct extensions:

- **Expansion of clinical diagnosis prediction tasks for echocardiography and other medical imaging domains.** Deep learning for echocardiography is uniquely positioned for high impact due to the wide-adoption, ease-of-use, and scale of datasets. In terms of further extensions of work, there are a range of prediction tasks relevant within echocardiography, i.e. to characterize ejection fraction, pericardial effusion, cardiac output, and valvular disease, that are well-positioned for deep learning algorithms. In addition, this work can be extended both in terms of other imaging modalities (such as magnetic resonance imaging (MRI) and computed tomography (CT)) and other types of cardiovascular disease (such as coronary artery or abdominal/thoracic aneurysms).

- **Integration with bioinformatics.** The body of work in bioinformatics and its use of machine learning is well-documenting and vast. A particularly exciting bridge between that work and ours, would be to combining -omics (genomics, proteomics, etc) data with precision phenotyping with deep learning. Many of the current day healthcare guidelines are established from limited study samples or are not granular enough. From a pattern matching perspective on high dimensional data, deep learning is uniquely positioned to be applied to the integration of these two data sources (informatics and biophysical broadly speaking) could reveal altogether new insights on cardiovascular disease.

- **Computer vision advancement for anomaly detection and limited labeled data.** In general, application-ended needs can help steer core discipline / theory research. In this case, medical imaging and automated diagnosis have pertinent needs that overlap with core machine learning / computer vision goals in (1) anomaly detection and (2) limited labeled data training.

- **Full deep learning pipelines from imaging to mechanical analysis.** To extend the atherosclerosis finite element work, there should be efforts to develop a full deep learning pipeline from imaging to mechanical analysis. Using our specific use-case as an example, intravascular imaging could be fed to a model for geometry and material property characterization. The parameters extracted could then be fed to a trained model for stress distribution prediction. This endeavor requires sufficient data and labeling for the multiple prediction tasks and ideally could be trained end-to-end.

- **Bridging traditional computational modeling techniques for biomechanics with ML.** A theme in the last two chapters is bridging computational modeling with machine learning. The extension of this work could enable high-throughput simulations (of both finite element and molecular dynamics simulations) for experimentalists (like clinicians or protein engineers). More specific to finite element, there could be further development in enabling multi-scale FEM which suffers from computational inefficiency. Further development of machine learning for molecular dynamics simulations could also alleviate computational inefficiency concerns to enable longer timescales of simulation.

- **Advancement of graph neural networks.** The merging of graph theory with deep learning particularly for the use of non-Euclidean data is an emerging field. The availability of benchmark datasets across other domains from computer vision to speech recognition enabled major breakthroughs in deep learning. Biophysical simulations are inherently structured in a graphical fashion with nodes and edges which make it well-positioned as an interesting data domain to help advance graph neural network techniques.

6.3 Closure

Machine learning and neural network research are by no means a new field of study – with roots in the mid to late twentieth century. The proliferation of "big", open datasets, advances in computer hardware, and technical refinements in neural networks enabled the rise of deep learning and renewed interest in advancing the field of neural networks. While we try to steer away from excessive hype, unchecked inflated expectations, and excessive fear around deep learning, it is undeniable that deep learning will have a lasting impact (if not already) on multiple industries and alter the way we interact with the world.

In this body of work, I aimed to apply and advance deep learning for the benefit of mankind and human health. Healthcare as a whole is a behemoth of an industry that is ripe for technological innovation. More specifically, cardiovascular disease is a complex, pertinent public health issue that can be understood and addressed through various means. With the lens of a biomechanicist, I hope to lay the groundwork for deep learning approaches to better understand and address cardiovascular disease.

Bibliography

[1] Martin Abadi et al. “Tensorflow: A system for large-scale machine learning”. In: *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. 2016, pp. 265–283.

[2] Hans C Andersen. “Rattle: A fffdfffdfffdvelocityfffdfffdfffd version of the shake algorithm for molecular dynamics calculations”. In: *Journal of Computational Physics* 52.1 (1983), pp. 24–34.

[3] Wenjia Bai et al. “Human-level CMR image analysis with deep fully convolutional networks”. In: (2017).

[4] Ahmed A Bakhaty, Sanjay Govindjee, and Mohammad RK Mofrad. “Consistent trilayer biomechanical modeling of aortic valve leaflet tissue”. In: *Journal of Biomechanics* 61 (2017), pp. 1–10.

[5] John M Ball. “Convexity conditions and existence theorems in nonlinear elasticity”. In: *Archive for rational mechanics and Analysis* 63.4 (1976), pp. 337–403.

[6] Herman JC Berendsen, David van der Spoel, and Rudi van Drunen. “GROMACS: a message-passing parallel molecular dynamics implementation”. In: *Computer physics communications* 91.1-3 (1995), pp. 43–56.

[7] V. Botu and R. Ramprasad. “Learning scheme to predict atomic forces and accelerate materials simulations”. In: *Phys. Rev. B* 92 (9 Sept. 2015), p. 094306. DOI: `10.1103/PhysRevB.92.094306`. URL: `https://link.aps.org/doi/10.1103/PhysRevB.92.094306`.

[8] Venkatesh Botu and Rampi Ramprasad. “Adaptive machine learning framework to accelerate ab initio molecular dynamics”. In: *International Journal of Quantum Chemistry* 115.16 (2015), pp. 1074–1083.

[9] Xavier Bresson and Thomas Laurent. “Residual Gated Graph ConvNets”. In: *arXiv preprint arXiv:1711.07553* (2017).

[10] Bernard R Brooks et al. “CHARMM: a program for macromolecular energy, minimization, and dynamics calculations”. In: *Journal of computational chemistry* 4.2 (1983), pp. 187–217.

[11] Alexandra H Chau et al. “Mechanical analysis of atherosclerotic plaques based on optical coherence tomography”. In: *Annals of biomedical engineering* 32.11 (2004), pp. 1494–1503.

[12] Xi Chen et al. “Infogan: Interpretable representation learning by information maximizing generative adversarial nets”. In: *Advances in neural information processing systems*. 2016, pp. 2172–2180.

[13] Kenneth R Chien. “Genomic circuits and the integrative biology of cardiac diseases”. In: *Nature* 407.6801 (2000), p. 227.

[14] Kyunghyun Cho et al. “Learning phrase representations using RNN encoder-decoder for statistical machine translation”. In: *arXiv preprint arXiv:1406.1078* (2014).

[15] Francois Chollet et al. *Keras*. `https://keras.io`. 2015.

[16] Anna Choromanska et al. “The loss surfaces of multilayer networks”. In: *Artificial Intelligence and Statistics*. 2015, pp. 192–204.

[17] Adam Coates et al. “Deep learning with COTS HPC systems”. In: *International conference on machine learning*. 2013, pp. 1337–1345.

[18] Joseph Paul Cohen, Margaux Luck, and Sina Honari. “Distribution matching losses can hallucinate features in medical image translation”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2018, pp. 529–536.

[19] Wendy D Cornell et al. “A second generation force field for the simulation of proteins, nucleic acids, and organic molecules”. In: *Journal of the American Chemical Society* 117.19 (1995), pp. 5179–5197.

[20] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[21] Thomas G Dietterich. “Ensemble methods in machine learning”. In: *International workshop on multiple classifier systems*. Springer. 2000, pp. 1–15.

[22] Kunio Doi. “Computer-aided diagnosis in medical imaging: historical review, current status and future potential”. In: *Computerized medical imaging and graphics* 31.4-5 (2007), pp. 198–211.

[23] Pamela S Douglas et al. “ACCF/ ASE/ AHA/ ASNC/ HFSA/ HRS/ SCAI/ SCCM/ SCCT/ SCMR 2011 appropriate use criteria for echocardiography: a report of the American College of Cardiology Foundation Appropriate Use Criteria Task Force, American Society of Echocardiography, American Heart Association, American Society of Nuclear Cardiology, Heart Failure Society of America, Heart Rhythm Society, Society for Cardiovascular Angiography and Interventions, Society of Critical Care Medicine, Society of Cardiovascular Computed Tomography, and Society for Cardiovascular Magnetic Resonance Endorsed by the American College of Chest Physicians”. In: *Journal of the American College of Cardiology* 57.9 (2011), pp. 1126–1166.

[24] Andre Esteva et al. "Dermatologist-level classification of skin cancer with deep neural networks". In: *Nature* 542.7639 (2017), p. 115.

[25] YC Fung. "Biomechanical aspects of growth and tissue engineering". In: *Biomechanics.* Springer, 1990, pp. 499–546.

[26] Vitold E Galkin et al. "Opening of tandem calponin homology domains regulates their affinity for F-actin". In: *Nature structural & molecular biology* 17.5 (2010), p. 614.

[27] Xiaohong Gao et al. "A fused deep learning architecture for viewpoint classification of echocardiography". In: *Information Fusion* 36 (2017), pp. 103–113.

[28] Michael Gastegger, Jörg Behler, and Philipp Marquetand. "Machine learning molecular dynamics for the simulation of infrared spectra". In: *Chemical science* 8.10 (2017), pp. 6924–6935.

[29] Justin Gilmer et al. "Neural message passing for quantum chemistry". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70.* JMLR. org. 2017, pp. 1263–1272.

[30] Dariya S Glazer, Randall J Radmer, and Russ B Altman. "Combining molecular dynamics and machine learning to improve protein function recognition". In: *Biocomputing 2008.* World Scientific, 2008, pp. 332–343.

[31] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics.* 2010, pp. 249–256.

[32] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. "Deep sparse rectifier neural networks". In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics.* 2011, pp. 315–323.

[33] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems.* 2014, pp. 2672–2680.

[34] Varun Gulshan et al. "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs". In: *Jama* 316.22 (2016), pp. 2402–2410.

[35] Namrata Gundiah, Mark B Ratcliffe, and Lisa A Pruitt. "Determination of strain energy function for arterial elastin: experiments using histology and mechanical tests". In: *Journal of biomechanics* 40.3 (2007), pp. 586–594.

[36] Jiequn Han, Arnulf Jentzen, and E Weinan. "Solving high-dimensional partial differential equations using deep learning". In: *Proceedings of the National Academy of Sciences* 115.34 (2018), pp. 8505–8510.

[37] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2016, pp. 770–778.

[38] Steven Henikoff and Jorja G Henikoff. “Amino acid substitution matrices from protein blocks”. In: *Proceedings of the National Academy of Sciences* 89.22 (1992), pp. 10915–10919.

[39] Gerhard A Holzapfel, Thomas C Gasser, and Ray W Ogden. “A new constitutive framework for arterial wall mechanics and a comparative study of material models”. In: *Journal of elasticity and the physical science of solids* 61.1-3 (2000), pp. 1–48.

[40] Gerhard A Holzapfel and Ray W Ogden. “Constitutive modelling of arteries”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 466.2118 (2010), pp. 1551–1597.

[41] Hayden Huang et al. “The impact of calcification on the biomechanical stability of atherosclerotic plaques”. In: *Circulation* 103.8 (2001), pp. 1051–1056.

[42] Sergey Ioffe and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *arXiv preprint arXiv:1502.03167* (2015).

[43] Barry Isralewitz, Mu Gao, and Klaus Schulten. “Steered molecular dynamics and mechanical functions of proteins”. In: *Current opinion in structural biology* 11.2 (2001), pp. 224–230.

[44] Diana E Jaalouk and Jan Lammerding. “Mechanotransduction gone awry”. In: *Nature reviews Molecular cell biology* 10.1 (2009), p. 63.

[45] Fuad M Jan, Elizabeth Thompson, and Kiran Sagar. “moderated Session 2: impact of Physician Training on Interpretation of Echocardiograms and Health Care Costs”. In: *Journal of the American Society of Echocardiography* 23.5 (2010), B47.

[46] Wolfgang Kabsch and Christian Sander. “Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features”. In: *Biopolymers: Original Research on Biomolecules* 22.12 (1983), pp. 2577–2637.

[47] Alireza Karimi et al. “Study of plaque vulnerability in coronary artery using Mooney–Rivlin model: a combination of finite element and experimental method”. In: *Biomedical Engineering: Applications, Basis and Communications* 26.01 (2014), p. 1450013.

[48] Andrej Karpathy. “The unreasonable effectiveness of recurrent neural networks”. In: *Andrej Karpathy blog* 21 (2015).

[49] Hanan Khamis et al. “Automatic apical view classification of echocardiograms using a discriminative learning dictionary”. In: *Medical image analysis* 36 (2017), pp. 15–21.

[50] Yoon Kim et al. “Character-aware neural language models”. In: *Thirtieth AAAI Conference on Artificial Intelligence*. 2016.

[51] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[52] Christian Knackstedt et al. “Fully automated versus standard tracking of left ventricular ejection fraction and longitudinal strain: the FAST-EFs Multicenter Study”. In: *Journal of the American College of Cardiology* 66.13 (2015), pp. 1456–1466.

[53] Ralph Knöll, Masahiko Hoshijima, and Kenneth Chien. “Cardiac mechanotransduction and implications for heart disease”. In: *Journal of molecular medicine* 81.12 (2003), pp. 750–756.

[54] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[55] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *nature* 521.7553 (2015), p. 436.

[56] Yann LeCun et al. “A theoretical framework for back-propagation”. In: *Proceedings of the 1988 connectionist models summer school*. Vol. 1. CMU, Pittsburgh, Pa: Morgan Kaufmann. 1988, pp. 21–28.

[57] Liang Liang, Minliang Liu, and Wei Sun. “A deep learning approach to estimate chemically-treated collagenous tissue nonlinear anisotropic stress-strain responses from microscopy images”. In: *Acta biomaterialia* 63 (2017), pp. 227–235.

[58] Liang Liang et al. “A machine learning approach as a surrogate of finite element analysis–based inverse method to estimate the zero-pressure geometry of human thoracic aorta”. In: *International journal for numerical methods in biomedical engineering* 34.8 (2018), e3103.

[59] Liang Liang et al. “A machine learning approach to investigate the relationship between shape features and numerically predicted risk of ascending aortic aneurysm”. In: *Biomechanics and modeling in mechanobiology* 16.5 (2017), pp. 1519–1533.

[60] Erik Lindahl, Berk Hess, and David Van Der Spoel. “GROMACS 3.0: a package for molecular simulation and trajectory analysis”. In: *Molecular modeling annual* 7.8 (2001), pp. 306–317.

[61] Geert Litjens et al. “A survey on deep learning in medical image analysis”. In: *Medical image analysis* 42 (2017), pp. 60–88.

[62] Laurens van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE”. In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.

[63] Ali Madani, Kiavash Garakani, and Mohammad RK Mofrad. “Molecular mechanics of Staphylococcus aureus adhesin, CNA, and the inhibition of bacterial adhesion by stretching collagen”. In: *PloS one* 12.6 (2017), e0179601.

[64] Ali Madani et al. “Chest x-ray generation and data augmentation for cardiovascular abnormality classification”. In: *Medical Imaging 2018: Image Processing*. Vol. 10574. International Society for Optics and Photonics. 2018, p. 105741M.

[65] Ali Madani et al. “Deep echocardiography: data-efficient supervised and semi-supervised deep learning towards automated diagnosis of cardiac disease”. In: *npj Digital Medicine* 1.1 (2018), p. 59.

[66] Ali Madani et al. “Fast and accurate view classification of echocardiograms using deep learning”. In: *npj Digital Medicine* 1.1 (2018), p. 6.

[67] Ali Madani et al. “Semi-supervised learning with generative adversarial networks for chest X-ray classification with ability of data domain adaptation”. In: *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. IEEE. 2018, pp. 1038–1042.

[68] Diego Marcheggiani and Ivan Titov. “Encoding sentences with graph convolutional networks for semantic role labeling”. In: *arXiv preprint arXiv:1703.04826* (2017).

[69] Francisco Martínez-Martínez et al. “A finite element-based machine learning approach for modeling the mechanical behavior of the breast tissues under compression in real-time”. In: *Computers in biology and medicine* 90 (2017), pp. 116–124.

[70] Robert T. McGibbon et al. “MDTraj: A Modern Open Library for the Analysis of Molecular Dynamics Trajectories”. In: *Biophysical Journal* 109.8 (2015), pp. 1528–1532. DOI: `10.1016/j.bpj.2015.08.015`.

[71] Gary S Mintz et al. “American College of Cardiology clinical expert consensus document on standards for acquisition, measurement and reporting of intravascular ultrasound studies (ivus): A report of the american college of cardiology task force on clinical expert consensus documents developed in collaboration with the european society of cardiology endorsed by the society of cardiac angiography and interventions”. In: *Journal of the American College of Cardiology* 37.5 (2001), pp. 1478–1492.

[72] Mehdi Mirza and Simon Osindero. “Conditional generative adversarial nets”. In: *arXiv preprint arXiv:1411.1784* (2014).

[73] Takeru Miyato et al. “Spectral normalization for generative adversarial networks”. In: *arXiv preprint arXiv:1802.05957* (2018).

[74] Sukrit Narula et al. “Machine-learning algorithms to automate morphological and functional assessments in 2D echocardiography”. In: *Journal of the American College of Cardiology* 68.21 (2016), pp. 2287–2295.

[75] Francesca Negri et al. “Left ventricular geometry and diastolic function in the hypertensive heart: impact of age”. In: *Blood pressure* 22.1 (2013), pp. 1–8.

[76] Alexander Papolos et al. “US hospital use of echocardiography: insights from the nationwide inpatient sample”. In: *Journal of the American College of Cardiology* 67.5 (2016), pp. 502–511.

[77] JinHyeong Park et al. “AutoGate: fast and automatic Doppler gate localization in B-mode echocardiogram”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2008, pp. 230–237.

[78] M Germana Paterlini and David M Ferguson. "Constant temperature simulations using the Langevin equation with velocity Verlet integration". In: *Chemical Physics* 236.1-3 (1998), pp. 243–252.

[79] Otávio AB Penatti et al. "Mid-level image representations for real-time heart view plane classification of echocardiograms". In: *Computers in biology and medicine* 66 (2015), pp. 66–81.

[80] Per-Olof Persson and Gilbert Strang. "A simple mesh generator in MATLAB". In: *SIAM review* 46.2 (2004), pp. 329–345.

[81] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. "Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations". In: *arXiv preprint arXiv:1711.10566* (2017).

[82] Pranav Rajpurkar et al. "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning". In: *arXiv preprint arXiv:1711.05225* (2017).

[83] Raghunathan Ramakrishnan et al. "Electronic spectra from TDDFT and machine learning in chemical space". In: *The Journal of chemical physics* 143.8 (2015), p. 084111.

[84] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[85] Sebastian Ruder. "An overview of multi-task learning in deep neural networks". In: *arXiv preprint arXiv:1706.05098* (2017).

[86] Olga Russakovsky et al. "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115.3 (2015), pp. 211–252.

[87] Andrej Šali and Tom L Blundell. "Comparative protein modelling by satisfaction of spatial restraints". In: *Journal of molecular biology* 234.3 (1993), pp. 779–815.

[88] Tim Salimans et al. "Improved techniques for training gans". In: *Advances in neural information processing systems*. 2016, pp. 2234–2242.

[89] Jorg Schroder and Patrizio Neff. "Invariant formulation of hyperelastic transverse isotropy based on polyconvex free energy functions". In: *International journal of solids and structures* 40.2 (2003), pp. 401–445.

[90] Kristof T Schütt et al. "Quantum-chemical insights from deep tensor neural networks". In: *Nature communications* 8 (2017), p. 13890.

[91] Christine M Scotti et al. "Wall stress and flow dynamics in abdominal aortic aneurysms: finite element analysis vs. fluid–structure interaction". In: *Computer methods in biomechanics and biomedical engineering* 11.3 (2008), pp. 301–322.

[92] Partho P Sengupta et al. "Cognitive machine-learning algorithm for cardiac imaging: a pilot study for differentiating constrictive pericarditis from restrictive cardiomyopathy". In: *Circulation: Cardiovascular Imaging* 9.6 (2016), e004330.

[93] Hengameh Shams et al. "Dynamic regulation of α-actininfffdfffdfffds calponin homology domains on F-actin". In: *Biophysical journal* 110.6 (2016), pp. 1444–1455.

[94] Hoo-Chang Shin et al. "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning". In: *IEEE transactions on medical imaging* 35.5 (2016), pp. 1285–1298.

[95] Ashish Shrivastava et al. "Learning from simulated and unsupervised images through adversarial training". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2107–2116.

[96] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[97] Nitish Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting". In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[98] John Stoitsis et al. "Computer aided diagnosis based on medical image processing and artificial intelligence methods". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 569.2 (2006), pp. 591–595.

[99] Sainbayar Sukhbaatar, Rob Fergus, et al. "Learning multiagent communication with backpropagation". In: *Advances in Neural Information Processing Systems*. 2016, pp. 2244–2252.

[100] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. "Sequence to sequence learning with neural networks". In: *Advances in neural information processing systems*. 2014, pp. 3104–3112.

[101] Pasi Tavi et al. "Cardiac mechanotransduction: from sensing to disease and treatment". In: *Trends in pharmacological sciences* 22.5 (2001), pp. 254–260.

[102] Robert L Taylor. *FEAP-A finite element analysis program*. 2014.

[103] David Van Der Spoel et al. "GROMACS: fast, flexible, and free". In: *Journal of computational chemistry* 26.16 (2005), pp. 1701–1718.

[104] WF Van Gunsteren and HJC Berendsen. "Algorithms for macromolecular dynamics and constraint dynamics". In: *Molecular Physics* 34.5 (1977), pp. 1311–1327.

[105] Petar Veličković et al. "Deep graph infomax". In: *arXiv preprint arXiv:1809.10341* (2018).

[106] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. "Order matters: Sequence to sequence for sets". In: *arXiv preprint arXiv:1511.06391* (2015).

[107] Theo Vos et al. "Global, regional, and national incidence, prevalence, and years lived with disability for 328 diseases and injuries for 195 countries, 1990–2016: a systematic analysis for the Global Burden of Disease Study 2016". In: *The Lancet* 390.10100 (2017), pp. 1211–1259.

[108] Joseph C Walker et al. "MRI-based finite-element analysis of left ventricular aneurysm". In: *American Journal of Physiology-Heart and Circulatory Physiology* 289.2 (2005), H692–H700.

[109] Hiroshi Watanabe et al. "Multiphysics simulation of left ventricular filling dynamics using fluid-structure interaction finite element method". In: *Biophysical journal* 87.3 (2004), pp. 2074–2085.

[110] Eli J Weinberg and Mohammad Reza Kaazempur Mofrad. "Transient, three-dimensional, multiscale simulations of the human aortic valve". In: *Cardiovascular Engineering* 7.4 (2007), pp. 140–155.

[111] Gregory A Weiss et al. "Rapid mapping of protein functional epitopes by combinatorial alanine scanning". In: *Proceedings of the National Academy of Sciences* 97.16 (2000), pp. 8950–8954.

[112] Gill Wharton et al. "A minimum dataset for a standard adult transthoracic echocardiogram: a guideline protocol from the British Society of Echocardiography". In: *Echo Research and Practice* 2.1 (2015), G9–G24.

[113] Zhenqin Wu et al. "MoleculeNet: a benchmark for molecular machine learning". In: *Chemical science* 9.2 (2018), pp. 513–530.

[114] Chiyuan Zhang et al. "Understanding deep learning requires rethinking generalization". In: *arXiv preprint arXiv:1611.03530* (2016).

[115] Olgierd Cecil Zienkiewicz, Robert Leroy Taylor, and Robert Leroy Taylor. *The finite element method: solid mechanics*. Vol. 2. Butterworth-heinemann, 2000.

[116] William A Zoghbi et al. "Recommendations for noninvasive evaluation of native valvular regurgitation: a report from the American Society of Echocardiography developed in collaboration with the Society for Cardiovascular Magnetic Resonance". In: *Journal of the American Society of Echocardiography* 30.4 (2017), pp. 303–371.