A DATA-DRIVEN APPROACH TO QUASI-STATIC ULTRASONIC ELASTICITY IMAGING

BY

CAMERON HOERIG

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Bioengineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Doctoral Committee:

Professor Michael F. Insana, Chair
Professor Emeritus Jamshid Ghaboussi
Professor Stephen Boppart
Professor Brad Sutton

Abstract

Changes in the mechanical properties of soft tissues that accompany pathology can potentially be used as a biomarker for detection and diagnosis of disease. Quasi-static ultrasonic elastography (QUSE) is one of several elasticity imaging techniques sensitive to mechanical contrast and offers a way to visualize the spatiotemporal distribution of material properties within tissues. Unfortunately, QUSE is generally an ill-posed inverse problem. Quantifying the mechanical properties requires measurements of more stress-strain data than can be acquired during a typical ultrasonic (US) imaging exam. Model-based inverse methods attempt to circumvent these limitations in part by estimating the spatial distribution of a pre-defined set of material parameters. As a consequence, model-based methods provide no means for discovering new diagnostically-relevant mechanical properties or for exploring ranges of known model parameters for relevance in a given situation.

We are developing a data-driven approach for quantitative QUSE using the Autoprogressive method (AutoP), which combines artificial neural networks (ANNs) and finite element analysis (FEA). AutoP has previously been used in geotechnical and civil engineering applications to build "soft-computational" models of materials. Using knowledge of object shape and force-displacement measurements, investigators were able to build neural network constitutive models (NNCMs) that accurately describe the behavior of linear, non-linear, and time-dependent materials with no prior constitutive model assumptions. Furthermore, NNCMs provide a means to estimate spatiotemporal stress and strain distributions from force-displacement data.

NNCMs and AutoP offer a fundamentally different approach to QUSE. We first demonstrate that a very sparse sampling of force-displacement data is sufficient for estimating the linear-elastic properties of gelatin phantoms when the interior geometry is known. Then, we introduce Cartesian NNCMs (CaNNCMs), a novel ANN architecture, capable of learning both material property and geometric information. We begin exploring the spatial sampling requirements to reconstruct Young's modulus distributions in both 2-D and 3-D. Moreover, we show how CaNNCMs can be used to estimate the spatial distribution of all stresses and strains and can be directly interrogated to infer the mechanical properties governing measured data. Further development of this method to non-linear and viscoelastic materials may provide a means to discover the mechanical parameters most relevant to clinical elastography.

Acknowledgments

First, I have to thank my adviser, Prof. Michael F. Insana, for his guidance and support during my time here at UIUC. He is also partly to blame for roping me into this project. Over the past several years, we have had many discussions over a broad range of topics — sometimes even about my research — that have helped shape me into a much better scientist. I still don't know what I want to be when I grow up, but he has instilled in me confidence that I have the potential to do well in research. I must also thank Dr. Jamshid Ghaboussi for the insightful discussions and patience. Given that I had no prior experience in computational mechanics, I faced quite the learning curve when starting this research. Fortunately, Dr. Ghaboussi has written a few books on the subject and helped me overcome my initial struggles. Thanks also to my lab mates both past and present, Dr. Khaldoon Altahan, Dr. Yue Wang, Dr. Minwoo Kim, Dr. Sara Bahramian, Yang Zhu, and Yiliang Wang. During the course of my years here, I had the opportunity to mentor many undergraduate students who have helped progress this work in various ways: Yik Tung (Tracy) Ling, Carlos Renteria (who is now enduring his own PhD studies at UIUC), Wendy Reyes, Léo Fabre, Ryan Healey, Pierce Hadley, Stanley Liu, Yoonho Nam, and Sungmin Jang. Lastly, I must thank my parents Pete Hoerig (and Ronda Hoerig) and Christine Billeg, as well as my siblings, Justin, Adrienne (and Seth), Colin, Tyler (and Brittany a.k.a. Lady Fletcher), and Victoria, and my partner Josh, for all their love and support. Especially Josh, who has shared in my highs and helped carry the burden of my lows over the past several years. Further thanks to H. Hashemi, H. Rivaz, E. M. Boctor, M. A Choti, and G. D. Hager for providing the source code for their displacement estimation algorithms.

Research reported in this publication was supported by NCI and NIBIB of the National Institutes of Health under Award Numbers R01 CA168575 and R21 EB023402.

Table of Contents

List of Abbreviations

ANN	Artificial Neural Network
AutoP	Autoprogressive Method
B-mode	Brightness Mode
BP	Backpropagation
CaNNCM	Cartesian Neural Network Constitutive Model
FE	Finite Element
FEA	Finite Element Analysis
F/T	Force/Torque
NNCM	Neural Network Constitutive Model
N-R	Newton-Raphson
MPN	Material Property Network
QUSE	Quasi-static Ultrasonic Elastography
RF	Radio Frequency
RPROP	Resilient Propagation
SN	Spatial Network
US	Ultrasound

Chapter 1

Introduction

Contrast in medical images arises from spatiotemporal variations in the physical properties of tissue. Different imaging modalities apply a particular stimulus and observe the response of the body to reconstruct maps of properties associated with health and disease states. For example, x-ray computed tomography (CT) illuminates the body with x-ray radiation, detects the fraction of transmitted photons, and reconstructs a map of tissue density that reveals inner anatomy. Magnetic resonance imaging (MRI) uses magnetic fields to map the local chemical environment surrounding hydrogen nuclei (protons) [1]. Positron emission tomography (PET) measures the spatiotemporal distribution of blood-born radioactive isotopes that are deposited and concentrate differentially in tissue regions depending on healthy and diseased cell functions [2]. Ultrasonic (US) imaging interrogates the tissue with high frequency sound waves and records the echo signal to reconstruct a map of acoustic impedance, which is determined by spatial fluctuations in tissue density and bulk modulus. Similarly, optical coherence tomography (OCT) observes the backscatter of coherent light to create an image dependent on the light scattering characteristics of tissues.

Changes in mechanical properties are strongly correlated with various tissue pathologies. However, a change in material properties is not always accompanied by morphological changes. As a consequence, the aforementioned imaging modalities are not effective at revealing the mechanical contrast in tissues. Sarvazyan *et al.* [3, 4] compiled measurements of tissue elasticity and showed that the *bulk* modulus of all soft tissues are in the range $\approx 10^9$ kPa whereas the *shear* modulus of soft tissues span the 10^3–10^8 kPa range. Physicians exploit this contrast during manual palpation to literally feel for signs of pathology. One notable use of palpation is in clinical breast exams to aid in the detection of cancerous lesions. Despite its utility, palpation is subjective and lesion detection depends on the skill of the physician in addition to lesion size and depth [5, 6].

Elasticity imaging (or elastography) emerged as a modality to visualize mechanical contrast of soft tissues. As the field progresses, the potential clinical applications continue to grow. For example, the strain profile of an atherosclerotic plaque may reveal if it is a “vulnerable” plaque at risk of rupture, which causes the formation of a thrombus and results in an acute heart attack and often patient death [7, 8, 9]. Increased liver stiffness can be an indicator of liver fibrosis [10, 11, 12, 13] or hepatocellular carcinoma [14, 15, 16]. Measurements of viscosity along with stiffness can increase the diagnostic accuracy of elasticity imaging [17, 18]. Similarly, the stiffness [19, 20, 21], viscoelastic [22, 23, 24, 25], and non-linear [26, 27, 28, 29] properties of focal breast lesions can aid in the differentiation between benign and malignant tumors. Interestingly, cancerous lesions often appear larger on an elastogram than on the corresponding B-mode image [30, 31, 32], potentially due to restructuring of the cellular microenvironment which alters the mechanoenvironment [33], but does not necessarily induce morphological changes visible on US B-mode images [34, 35].

1.1 Elasticity Imaging Techniques

Much like MRI, OCT, etc., elasticity images are formed by observing the tissue response to some applied stimulus. Here, a mechanical load is applied and the time-varying tissue motion is measured. Local displacements are often tracked with US, MRI, or OCT because they are phase-sensitive and can detect very small movements. The general procedure for elasticity imaging is

1) Apply an excitation force to the tissue,

2) Measure the responding tissue motion with US, MRI, or OCT,

3) Infer mechanical parameters from the measured response.

Broadly speaking, elasticity imaging techniques can be categorized as quasi-static or dynamic based on the type of excitation force applied. We cover some of the major advancements in ultrasonic elasticity imaging in the following sections. Mainly, we aim to provide a glimpse into the evolution of ultrasonic elastography techniques and cover how different methods probe the mechanical properties of tissues.

1.1.1 Quasi-static Elastography

Quasi-static ultrasonic elastography (QUSE) was first introduced by Ophir *et al.* in the early 1990s [36]. It is performed by observing tissue motion as compressive forces are slowly applied by a US probe. Quasi-static loading allows time for the force to distribute through and affect the whole contiguous volume, including boundaries, thus the deformation of a whole tissue region can be monitored with a single force stimulus. Internal motion is estimated by recording RF echo frames after each compressive load and operating on said frames with a speckle-tracking algorithm (e.g., [37, 38]). Because all phase information is contained along the axial direction (the direction of the US pulse propagation), axial displacements are most accurately estimated, although some algorithms have been developed to estimate lateral displacements [39, 40, 41]. Computing the gradient of the displacement fields results in a map of the local strains that correspond to stiffness. This type of QUSE is often referred to as strain elastography. Areas of high strain are softer whereas low-strain areas are stiffer. If the compressive force was applied over a large area — which can be achieved by attaching a rigid plate to the face of the US probe — the resulting stress field can be assumed to be roughly uniform. Then, the strains are approximately inversely proportional to Young's modulus (or shear modulus) [42, 43]. Some applications of strain elastography include detection and differentiation of cancerous breast lesions [44, 45], prediction of breast cancer tumor grades [46], and as an aid in differential diagnosis of liver masses [47].

Of course, the stress field is not actually uniform and artifacts caused by violating this assumption make interpreting strain images more difficult. Furthermore, strain is only a qualitative proxy for stiffness and does not provide an absolute measure of elasticity. Section 1.2 addresses methods by which quantitative estimates of material properties are made in QUSE.

Non-linear [48, 49, 50] and viscoelastic [51, 24] material properties can also be estimated in quasi-static elastography. Non-linear properties reveal how the tissue stiffness changes with the applied force. Most tissues exhibit strain-stiffening behavior, but the rates at which the tissues "stiffen" may aid in the differentiation of benign and malignant lesions [52, 27]. Or, a quasi-static load can be applied by the US probe and held constant to observe time-varying tissue deformation. Maintaining a steady force allows one to measure the creep behavior. Conversely, if the US probe is pressed into the tissue surface and held still, the relaxation of dynamics can be measured. Creep or relaxation behavior can be used to infer the viscoelastic properties of soft tissues with potential applications in breast cancer detection [53, 54].

Quasi-static forces need not be applied at the tissue surface. The acoustic radiation force is the phenomenon whereby momentum is transferred from the compressive US wave to a dissipative medium through reflection or absorption [55]. Some investigators exploit this phenomenon to apply internal forces with a "virtual finger". For example, by focusing the US beam to a point on the surface of a sample, Sugimoto *et al.* were able to measure local motion using Doppler ultrasound, from which the elastic properties of the tissue were estimated [56]. Later, Nightingale *et al.* introduced acoustic radiation force impulse (ARFI) imaging whereby the tissue is "pushed" internally and correlation techniques are used to track tissue displacement [57]. Unlike QUSE methods where the excitation force is the US probe pressing into the tissue, ARFI can only measure tissue motion at the site where the "push-pulse" is focused. Therefore, to reconstruct an image, the push-pulse must be raster scanned over a region. Furthermore, the resulting images map the maximum local displacements, not strain. Aside from increasing the acquisition time, repeated push-pulses increase the risk of bioeffects from tissue heating [58] but may be less operator dependent [59]. Research on the potential clinical applications of ARFI include evaluating liver fibrosis [60, 61], characterizing lesions in the breast [62, 63], and facilitating diagnosis of thyroid nodules [64, 65].

Probing the viscoelastic properties with ARFI-based techniques requires an adjustment to the sampling procedure. One method called Viscoelastic Response (VisR) imaging [66] uses two time-delayed push-pulses at each location. Between each pulse, correlation techniques track displacement of the tissue as it relaxes back to its initial state. The time-varying relaxation behavior can then be used to infer the viscoelastic properties. Investigations have begun for utilizing VisR in monitoring progression of muscle fibrosis in Duchenne muscular dystrophy [67, 68]

1.1.2 Dynamic Elastography

Dynamic elasticity imaging methods can be divided into harmonic (narrow-band forces applied) and transient (broad-band forces applied) elastography. Most dynamics methods measure the propagation of shear waves incited by the force stimulus. Fortunately, the shear waves do not interact with the bulk (or compressional) waves used in conventional ultrasonic imaging. As will be described for the various methods below, the shear waves induce tissue motion that can be measured via US imaging and correlation methods.

Harmonic Methods

Sonoelastography was reported in 1988 by Lerner *et al.* as one of the first methods to image the elastic properties of tissues [69, 70]. It is a type of harmonic elastography whereby an external source excites low-frequency harmonic vibrations (20–1000 Hz) within the tissue. Vibrational patterns vary in space based on the variance in material properties and are measured by US pulsed Doppler imaging. Later techniques building upon sonoelastography measure the amplitude and phase of the propagating waves to ascertain the elastic and viscous properties of tissues [71, 72, 73].

Harmonic motion can also be initiated via ultrasonic pulses much like ARFI techniques. Fatemi and Greenleaf introduced vibro-acoustic spectrography that relied on the acoustic radiation force to generate harmonic tissue motion [74]. Two US transducers with coincident foci were driven at two different frequencies, causing the tissue to oscillate locally at the beat frequency. A separate hydrophone tuned to the beat frequency recorded the scattered acoustic signal. Adjusting the difference in frequencies allowed the tissue to be driven and its response measured at various frequencies. Coupled with raster scanning the focal point over the sample, spectroscopic information could be measured within a region to, for example, detect microcalcifications in breast tissue [75, 76].

Crawling wave elastography in a sense combines sonoelastography and vibro-acoustography [77, 78]. Here, external actuators vibrate the tissue at slightly different frequencies to induce an interference pattern. This resulting shear wave propagates slower than the waves induced by the actuators, hence the name “crawling” waves. A US imaging probe acquires RF frames and shear wave velocity is estimated through speckle-tracking methods. Shear wave speed can then be related back to the viscoelastic properties of tissues like skeletal muscle [79, 80].

Transient Methods

Sarvazyan *et al.* first described shear wave elasticity imaging (SWEI) by inducing a shear wave with the acoustic radiation force. Like the harmonic elastography techniques described above, a conventional US imaging probe and speckle-tracking algorithms track the propagation of the shear wave. Shear wave speed is estimated by either measuring the amount of time required for the wave to propagate from its origin to the point of measurement or by tracking two locations within the tissue and measuring how long it takes for the shear wave to propagate from one point to the next. Furthermore, the generated shear wave is assumed to ripple outward within a plane. When observed on a US echo frame, the shear wave would reside within a line of the image frame. Thus, to create an image of shear wave speed, multiple shear waves would have to be generated at several depths in succession.

SWEI was the catalyst for a number of transient ultrasonic elastography methods. Two particular techniques are supersonic shear imaging (SSI) [81] and shear dispersion ultrasound vibrometry [82]. SSI induces a "mach cone" by firing multiple push-pulses along a line at a rate faster than the shear wave speed (hence the name "supersonic"). An ultra high frame-rate US scanner measures the propagation of the shear wave. In this way, SSI is able to measure shear wave speeds throughout a region and estimate viscoelastic properties with a single push-measure sequence [23]. For SDUV, a harmonic shear wave is again induced with a push-pulse and its speed is estimated by measuring the wave at two different points. Modifying the frequency of the generated shear wave allows the dispersion properties of the wave to be measured, from which viscoelastic material properties are estimated [83].

1.2 Model-based Inverse Methods

The preceding discussion of ultrasonic elasticity imaging only briefly introduced some of the developed techniques, what material properties they probe, and some of the relevant clinical applications. However, there was no explanation for how the material properties were estimated. Our aim in this section is to clarify the inverse problem in elastography and review some of the model-based approaches to its solution. Since the remainder of this dissertation describes a quasi-static approach to elasticity imaging, the focus will remain on QUSE.

Recall that QUSE is performed by slowly pressing the US probe into the tissue surface while acquiring RF echo frames. In addition to estimating internal motion with speckle-tracking techniques, displacement of the US probe and quantitative measures of the applied force can be collected. Furthermore, the surface shape of the tissue may be known. Thus, each data set acquired during QUSE at most contains the time-varying 1) force applied by the US probe, 2) motion of the probe, 3) internal deformation of a tissue volume sampled within a finite plane, and 4) the external geometry of the tissue being imaged (e.g., the shape of the breast or abdomen).

One can define the *forward* problem as: given the force applied by the US probe and distribution of the material properties, what are the displacements of the probe and body? The forward problem is governed by well-defined physical laws. From the conservation of linear momentum, stresses $\boldsymbol{\sigma}$ throughout a body in static equilibrium can be related to deforming forces $\boldsymbol{f}$ [84]:

$$\nabla \cdot \boldsymbol{\sigma} + \boldsymbol{f} = 0, \tag{1.1}$$

$$\tag{1.2}$$

where ∇ is the del operator. Often in QUSE the body forces comprising $\boldsymbol{f}$ are neglected and the term is removed. For an elastic body, the material properties are described by the stiffness tensor $\boldsymbol{C}$ relating stresses to strains $\boldsymbol{\varepsilon}$:

$$\varepsilon_{ij} = \frac{1}{2}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}) \tag{1.3}$$

$$\sigma_{ij} = C_{ijkl}\varepsilon_{kl}, \tag{1.4}$$

where u refers to displacement, the tensors are expressed in component form, $i, j = 1, 2, 3$, and repeated indices imply summation (e,g., $A_{ik}B_{kj} = A_{i1}B_{1j} + A_{i2}B_{2j} + A_{i3}B_{3j}$). A total of 81 components are contained within $\boldsymbol{C}$, although major and minor symmetry of the tensor reduces it to a maximum of 21 independent parameters.

Knowledge of the governing laws and kinematic relationships are used in formulating the *inverse* problem. One can state the inverse problem simply: given a set of force-displacement estimates and overall object shape, reconstruct the spatial distribution of mechanical properties. Alternatively, what is the spatial distribution of the 21 components of $\boldsymbol{C}$?

QUSE is generally an ill-posed inverse problem because we cannot normally acquire all of the data necessary to solve for material properties exactly. Current solutions take a model-based (or knowledge-driven) approach, where the mechanical properties of tissues are defined by parameters of a constitutive model relating stresses and strains. This problem is ill-posed in part due to the presence of measurement noise and limited force-displacement sampling from which stress-strain behavior is determined. Some strain information can be computed as spatial derivatives of the displacements, but stresses cannot be calculated from force data without knowing the object's geometry, boundary conditions, and material properties.

Simplifying assumptions are adopted in model-based techniques to help overcome the ill-posed nature of the inverse problem. Most often the tissue is assumed to be linear-elastic, isotropic, (nearly) incompressible, and under small strain. These materials are defined by the two Lamé parameters μ and λ:

$$\sigma_{ij} = 2\mu\varepsilon_{ij} + \lambda\varepsilon_{kk}\delta_{ij} \tag{1.5}$$

where δ_{ij} is the Kronecker delta and $\varepsilon_{kk} = \mathrm{tr}(\boldsymbol{\varepsilon})$. The Lamé parameters are related to the Young's modulus E and Poisson's ratio ν [85]:

$$\mu = \frac{E}{2(1+\nu)}, \qquad \lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}. \tag{1.6}$$

Soft tissues are primarily composed of water and thus assumed to be incompressible ($\nu = 0.5$) leaving only the shear modulus μ to be estimated.

Stresses cannot typically be measured and therefore the stress term in (1.1) is replaced with (1.5). Making said replacement, neglecting body forces, and expressing strain in terms of displacements in (1.3), the resulting equation is [86]

$$\nabla \cdot \mu \nabla \boldsymbol{u} + \nabla (\lambda + \mu) \nabla \cdot \boldsymbol{u} = 0. \tag{1.7}$$

Equation (1.7) relates 3-D displacement and the two Lamé parameters whereas internal displacements in QUSE are estimated only within a finite plane. Due to this limitation, the inverse problem is often reduced to 2-D via a plane-stress or plane-strain approximation. In one of the first inverse schemes for elasticity imaging, Skovoroda *et al.* [87] used a plane-strain approximation and rearranged the PDE in (1.7) to arrive at

$$\frac{\partial^2 \mu \varepsilon_{xy}}{\partial x^2} - \frac{\partial^2 \mu \varepsilon_{xy}}{\partial y^2} + 2 \frac{\partial^2 \mu \varepsilon_{yy}}{\partial x \partial y} = 0. \tag{1.8}$$

Solving (1.8) for the spatial distribution of the shear modulus is a *direct* inverse method. In this case, a solution can only be found if the shear modulus is known on the boundary of the ROI. Furthermore, the existence of second-order derivatives greatly amplifies noise in the estimated strain fields, causing the inverse to be unstable. Smoothing operations can reduce the effects of noise at the cost of decreased resolution.

Barbone and Bamber [88] proved that displacement measurements both within and on the boundary are insufficient for uniquely determining the shear modulus distribution of the material. In fact, they provided examples where the exact same displacement fields can arise in materials with different shear modulus distributions. They also conclude that, lacking information of the stress distribution, surface force measurements must be incorporated into the inverse problem to uniquely estimate the shear modulus, a fact later corroborated by Tyagi *et al.* [89]. Barbone and Gokhale further demonstrated that a significant amount of prior information of the shear modulus must be known in order to uniquely reconstruct a map of its distribution [90]. They also proved that the shear modulus can be determined up to a multiplicative constant if multiple independent displacement fields are measured. Given these results, investigators have sought ways to overcome the ill-posed inverse by including prior information in different ways.

A different direct inverse approach was developed by Zhu *et al.* that incorporates surface force information in the solution [91]. Their method was based on finite element analysis (FEA) techniques to construct a linear system of equations. Their method was able to produce absolute estimates of shear modulus from a single displacement field due to the incorporated force measurement. However, their approach was still sensitive to noise and because of technical limitations, they were unable to provide examples of Young's modulus reconstructions with experimentally acquired measurement data.

Kallel and Bertrand introduced *iterative* inverse methods in 1996 [92]. Generally speaking, iterative methods express the inverse problem as an objective function to be minimized by seeking a pre-defined set of model parameters at position $\boldsymbol{x}$, $\boldsymbol{\theta}(\boldsymbol{x})$,

$$\boldsymbol{\theta}(\boldsymbol{x}) = \underset{\hat{\boldsymbol{\theta}}(\boldsymbol{x}) \, \in \, \mathbb{R}}{\operatorname{argmin}} \sum_{n=1}^{N_p} \sum_{k=1}^{N_d} \mathrm{f_u}(\boldsymbol{u}_k^n\{\hat{\boldsymbol{\theta}}(\boldsymbol{x})\}, \hat{\boldsymbol{u}}_k^n) + g(\hat{\boldsymbol{\theta}}(\boldsymbol{x})), \tag{1.9}$$

where N_p refers to the number of measured data sets (i.e., number of compressive loads), N_d is the number of measured displacements in each of the N_p sets, $\hat{\boldsymbol{u}}_k^n$ are the measured displacement vectors, $\boldsymbol{u}_k^n$ are displacements estimated via a forward problem (e.g., via FEA), and $\mathrm{f_u}$ is often the L_2 norm of their difference. A regularization term $g(\cdot)$ may be added to improve the conditioning of the inverse problem. For the cases where the tissue is assumed to be linear-elastic, $\boldsymbol{\theta}(\boldsymbol{x}) = \mu(\boldsymbol{x})$. Direct methods are limited in accuracy and resolution by the least accurate component of displacement, which is the lateral displacement in QUSE, and the implied continuity for both strain and modulus means they have difficulty capturing sharp transitions. Compared to direct solution methods, iterative approaches are typically more robust to measurement noise, make no implicit assumptions on continuity (unless enforced via regularization), and have generally performed better at the cost of computational complexity (i.e., direct methods typically are much faster).

The iterative approach developed by Kallel and Bertrand minimized (1.9) via Newton-Raphson iterations and utilized Tikhonov regularization, but assumed uniformity in the stress field because only displacement data was available. Later, Doyley *et al.* evaluated a similar iterative algorithm and compared the results when only simulated displacement or boundary stress information was available [93]. They defined the objective function

$$\pi = \int_{\Omega} (U^1 - U^1_{\text{meas}})^2 \mathrm{d}\Omega, \tag{1.10}$$

where U^1_{meas} are the measured displacements and U^1 are the displacements computed in a forward problem using (1.5) as the constitutive model. Instead of a regularization term, they included spatial filtering in the Young's modulus estimate. They found the resolution of the reconstructed shear modulus distribution was better when using only displacements, but solving the problem with only stress data allowed the magnitude of the shear modulus to be estimated. Samani *et al.* developed a method whereby stresses were iteratively estimated via FEA from displacement data and paired with strains computed from full 3-D displacement data (acquired from MRI) to estimate the relative Young's modulus distributions [94]. They later adapted the method for QUSE [95]. Again, they could achieve fairly high resolution elastograms under controlled experimental settings, but only obtain the relative modulus values.

For imaging the non-linear elastic properties of tissues, Oberai *et al.* replaced the linear constitutive equation in (1.5) with the Veronda-Westmann model [50]:

$$W = \mu_0 \left(\frac{e^{\gamma(I_1 - 3)} - 1}{\gamma} - \frac{I_2 - 3}{2} \right). \tag{1.11}$$

Equation 1.11 is a strain-energy function describing a hyperelastic material, hence the different form compared to (1.5). Terms I_1 and I_2 are the first and second invariants of the Cauchy-Green strain tensor, μ_0 is the shear modulus at zero strain, and γ is the non-linear parameter. They defined an objective function using two measured displacement fields: U^1_{meas} acquired under small strain (in the linear-elastic regime) and U^2_{meas} under large strain (in the non-linear regime):

$$\pi = \frac{1}{2} \int_{\Omega} \left(w_1 (U^1 - U^1_{\text{meas}}) + w_2 (U^2 - U^2_{\text{meas}}) \right) \mathrm{d}\Omega + \frac{\alpha_\mu}{2} \mathcal{R}(\mu_0) + \frac{\alpha_\gamma}{2} \mathcal{R}(\gamma). \tag{1.12}$$

U^1 and U^2 are the displacements computed in a forward problem using (1.11) as the constitutive model, w_1 and w_2 are weighting terms, $\mathcal{R}$ is the total-variation diminishing regularization functional that penalizes large changes in the corresponding parameter without penalizing steepness, and α_μ and α_γ are regularization parameters. They found that while reconstructing images of the non-linear parameter γ was difficult, it was independent of the local strain state. Meaning γ is independent of the force loading and, theoretically, can be estimated quantitatively without any information of force BCs or stress fields. The caveat is that this non-linear parameter is specific to the Veronda-Westmann model and is not immune to modeling errors. However, in a later study using this inverse method, Goenezen *et al.* were able to correctly classify nine out of ten tumors as benign or malignant from the non-linear parameter alone [96].

Richards *et al.* developed a method for 3-D QUSE using a 2-D linear US array [97]. A sample was confined by plates in contact with its top and bottom surfaces. The top plate had a section removed to serve as an acoustic window. 3-D RF echo data was acquired by scanning the US probe along the elevational direction within this window. A small compressive load was applied by moving the bottom plate toward to the top, after which a new 3-D RF data set was acquired. The authors minimized an objective function similar to (1.12) except they used only the U^1 term, the sample was assumed to be linear-elastic, and they introduced a new regularization term based on the assumption of incompressibility. Even though they were able to reconstruct accurate relative shear modulus volumes, they were still unable to estimate modulus magnitudes.

Finally, methods to characterize the viscoelastic properties of tissues typically adopt a linear-viscoelastic model [54, 51, 24]. For example, Bayat *et al.* [54] created an automated method using a ramp-and-hold force to monitor the creep behavior of tissue. Using a standard linear model, they found that contrast in the instantaneous and long-term modulus values between benign and malignant tumors was insignificant, but the retardation time was significantly smaller for cancerous lesions. Interestingly, their system measures the force applied and thus they were able to quantify the modulus values. However, like with the non-linear parameter estimated by Oberai *et al.*, the results are subject to modeling errors. In particular, the sample is assumed to be under uniaxial stress with a precise loading curve and fully described by the three parameters of a standard linear model. Any discrepancy between model and measurement introduces errors into the parameter estimates.

1.3 The Innovation and Outline of Dissertation

The common theme with the inverse methods is that a lack of information inhibits accurate estimation on material parameters. Quantitative shear modulus estimates require prior knowledge of 1) the shear modulus somewhere in the body, 2) the spatial distribution of stress, or 3) compressive forces applied at the boundary. There are currently no methods to estimate stress, so many model-based methods introduce regularization terms or *a priori* information to reduce sensitivity to noise or improve resolution, but as Lanczos said, "A lack of information cannot be remedied by any mathematical trickery" [98]. Even for parameters not dependent on the stress distribution (e.g., non-linear parameter or retardation constant) or cases when sufficient information is available to estimate material parameters exactly, modeling errors are unavoidable unless the assumed constitutive model perfectly describes the tissue. Fundamentally, model-based methods provide no means for discovering new diagnostically-relevant mechanical properties or for exploring ranges of known model parameters for relevance in a given situation.

The goal of this dissertation is to develop the foundation for a quantitative, data-driven QUSE method by replacing the classic mathematically defined constitutive model with a neural network constitutive model (NNCM). Contrary to model-based inverse methods, NNCMs learn a non-parametric mapping of strain to stress from force-displacement measurements using the Autoprogressive method (AutoP). As we will describe in the next chapter, NNCMs are deeply intertwined with FEA, ensuring these data-driven models not only characterize the mechanical behavior encoded in the force-displacement measurements, but also obey fundamental physical laws.

Perhaps the most notable advancement of NNCMs to QUSE is the ability to reconstruct maps of all strains *and stresses* from noisy force-displacement measurements with no *a priori* constitutive model assumption. Access to full-field stresses and strains allows one to estimate material parameters from any mechanical model *after* information is extracted from measurement data. Thus, we believe NNCMs may eventually allow for discovery of material parameters most relevant for aiding in the differential diagnosis of disease.

NNCMs and AutoP have been used in many civil and geotechnical engineering applications to develop soft-computational model of linear, non-linear, viscoelastic, and rate-dependent materials using only boundary force-displacement measurements and known object geometry [99, 100, 101, 102, 103, 104, 105, 106, 107]. In this dissertation, we lay the foundation for data-driven QUSE and therefore limit the investigation to linear-elastic materials. The accomplishments of this dissertation are

- The first adaptation of NNCMs and AutoP for QUSE (Chapter 3),
- Developed a new NNCM architecture to learn both material property and geometric information in 2-D (Chapters 4–5),
- Extended NNCMs to 3-D imaging (Chapter 6),
- Began investigations into the feasibility of free-hand QUSE (Chapter 6).

Chapter 2 is a brief introduction to artificial neural networks, finite element analysis, and the Autoprogressive method. Later chapters (and earlier works on AutoP) describe the steps in AutoP, but do not effectively explain why NNCMs learn stress-stress behavior from force-displacement data in AutoP. The goal of this chapter is to better illustrate how ANNs and FEA come together to build data-driven material models.

The first application of NNCMs and AutoP for elasticity imaging is described in Chapter 3. We demonstrate the ability of NNCMs to learn the linear-elastic material properties of gelatin phantoms under quasi-static loading with a very sparse sampling of displacement data when the interior geometry is known. Chapter 3 is a modified version of our paper published in *Biomechanics and Modeling in Mechanobiology* [108].

Cartesian neural network constitutive models (CaNNCMs) are introduced in Chapter 4. Unlike NNCMs discussed in Chapter 3, CaNNCMs learn both material property and geometric information. Chapter 4 develops the theory of CaNNCMs and describes how spatial information is extracted from stress-strain data to train these networks. Chapter 4 is a modified version of our technical report submitted to arXiv [109].

The implementation of CaNNCMs in AutoP is covered in Chapter 5. Force-displacement measurements acquired from simulated phantoms, simulated RF echo frames, and experimentally on gelatin phantoms are used to test the ability of CaNNCMs to learn the spatially varying Young's modulus distribution of different models. Chapter 5 is a modified version of the manuscript titled *Data-driven Elasticity Imaging Using Cartesian Neural Network Constitutive Models and the Autoprogressive Method* submitted for review to *IEEE Transactions on Medical Imaging*.

Chapter 6 extends the capabilities of CaNNCMs to 3-D imaging. We first explore the abilities of 3-D CaNNCMs to reconstruct the Young's modulus within a volume based on the number of planes in which displacement data is available. Then, we describe a modification to accommodate objects under large deformation and begin investigations into the feasibility of free-hand 3-D QUSE.

Chapter 2

A Primer on the Autoprogressive Method

Neural network constitutive models and the Autoprogressive method are a fundamentally different approach to the inverse problem in elastography. While some of the intuition obtained from model-based inverse methods has analogues in this data-driven approach, a high-level understanding of the tools used in AutoP can help reveal where the differences reside. The purpose of this chapter is to introduce neural network constitutive models, present a high-level view of FEA, and begin to explore the way in which AutoP combines the two in order to build data-driven models from force-displacement measurement data. A more detailed description of the steps taken in AutoP is given in Chapter 3 and is further expanded upon in Chapter 5 after the introduction of Cartesian NNCMs.

2.1 Artificial Neural Networks

Artificial neural networks are an information-based machine learning technique often used to solve complex problems. Loosely based on biological neural networks, ANNs consist of a network of "neurons", or nodes, interconnected to exchange information and compute an output based on a given input stimulus. Several characteristics inherent to ANNs make them desirable as a problem-solving technique, including their massive parallelism, ability to learn and generalize, inherent nonlinearity, adaptability, robustness, and ability to handle noisy data [110, 111]. Because of these properties, ANNs are able to accurately model complex behaviors and relationships even when provided with noisy data. Perhaps the most important ability of ANNs for the subject of informational modeling is their ability to model phenomena only understood through measurement or observation. In fact, ANNs are "universal approximators" and can represent any function [112]. From

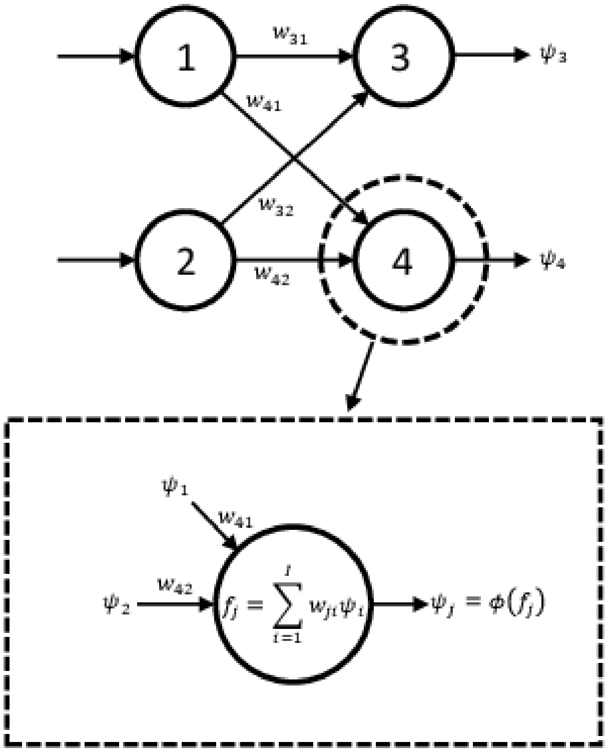


Figure 2.1: Example of a simple two-layer, feed-forward ANN. Output from nodes are weighted and summed to form the input to a node in the succeeding layer. The input signal is passed to an activation function, the output of which becomes the output of the node.

a set of noisy measurements, ANNs can generalize and learn the underlying relationships between chosen inputs and outputs.

Many different types of neural network topologies exist; for the purpose of data-driven elasticity imaging, our interest lies only in feed-forward networks. Figure 2.1 contains an example of a simple feed-forward, two-layer ANN. Nodes in the first layer compute outputs ψ_i that feed the nodes in the second layer. Connection weights w_{ji} multiply the output of node i in layer $N-1$ before it reaches node j in layer N. Input to a node, referred to as the input signal f_j, is computed by summing the output of all feeding nodes multiplied by the corresponding connection weights. Node output is found by supplying the input signal to an activation function $\phi(f_j)$ defined for the jth node.

ANNs "learn" by adjusting the connection weights between nodes to minimize the error between the output computed by the ANN and the target output specified by training data. Input-output relationships described by ANNs are the result of the connectivity of the network, node activation functions, and connection weights. The first two are typically static, meaning the connection weights must change for the ANN to adjust its response. The type of learning used for NNCMs is a form of supervised learning where a known output is given for every training input. Many supervised learning techniques are based on the backpropagation algorithm, a form of gradient descent optimization, to adjust the connection weights based on the error between ANN output and expected response [113, 114]. Specific training algorithms and parameters are described in later chapters as they pertain to the networks in question.

2.1.1 Neural Network Constitutive Models

Constitutive models of materials relate stresses in a body to deformation through a kinematic description and material parameters. For example, we previously defined a linear-elastic, isotropic material by the constitutive model

$$\begin{aligned} \mathrm{f}_{\varepsilon} &: \boldsymbol{\varepsilon} \rightarrow \boldsymbol{\sigma} \\ \mathrm{f}_{\varepsilon}(\boldsymbol{\varepsilon}) &= 2\mu\boldsymbol{\varepsilon} + \lambda\,\mathrm{tr}(\boldsymbol{\varepsilon})\boldsymbol{I}. \end{aligned} \tag{2.1}$$

The terms $\boldsymbol{\varepsilon}$ and $\mathrm{tr}(\boldsymbol{\varepsilon})$ are the kinematic description whereas μ and λ are material parameters. These parameters provide a convenient way of summarizing mechanical behavior such as stiffness (e.g., Young's modulus), nonlinearity, time-dependence, and rate-dependence. Hence, material parameters are estimated in QUSE to reconstruct images summarizing tissue mechanical properties.

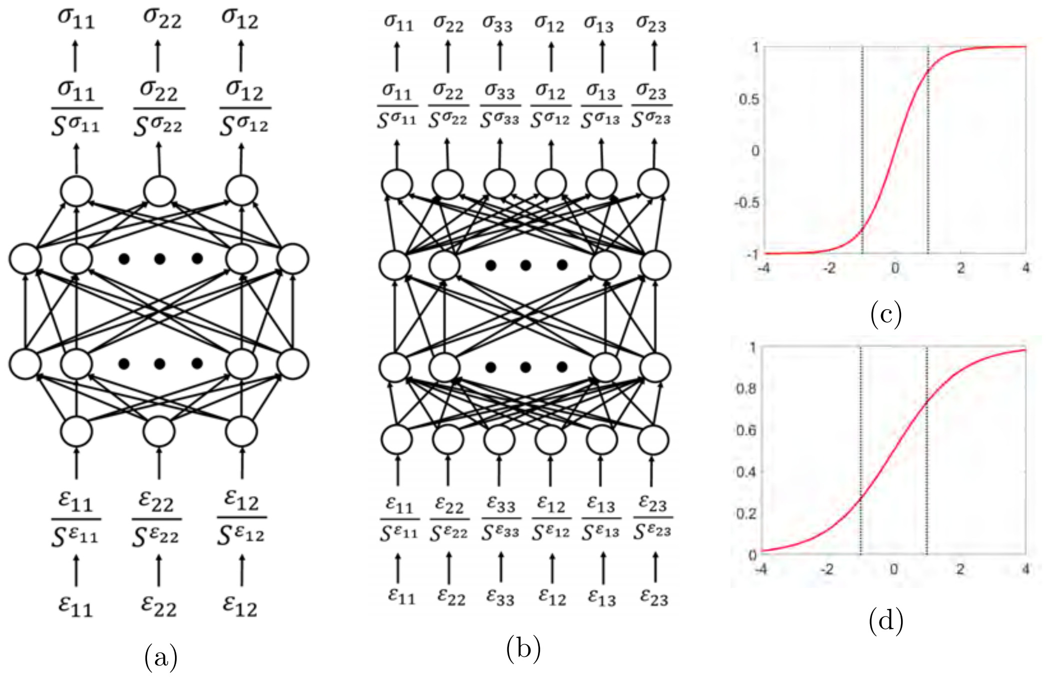
A NNCM is a "soft computational" description of material properties. Like conventional constitutive models, a NNCM accepts strain at the input and returns stress at the output:

$$NNCM : \boldsymbol{\varepsilon} \rightarrow \boldsymbol{\sigma}. \tag{2.2}$$

There is no explicit kinematic description — aside from the form of strain at the input — nor material parameters. Instead, NNCMs exploit the "universal approximation" capabilities of ANNs to learn a mapping from strain to stress. A NNCM can therefore be trained to represent the mechanical properties of any material without prior knowledge of the underlying constitutive model.

A full expression for computing stress is given by (2.3) after introducing the NNCM architecture.

Figs. 2.2a and 2.2b illustrate two different NNCMs that will be encountered in the following chapters. Each network is comprised of an input layer, two hidden layers, and an output layer. On the left is the NNCM architecture used for 2-D materials[1] and the network on the right is capable of characterizing 3-D materials. Strains input to the 2-D network are first scaled by the strain scaling vector $\boldsymbol{S}^{\varepsilon} = [S^{\varepsilon_{11}} \ S^{\varepsilon_{22}} \ S^{\varepsilon_{12}}]$ (for 3-D, $\boldsymbol{S}^{\varepsilon} = [S^{\varepsilon_{11}} \ S^{\varepsilon_{22}} \ S^{\varepsilon_{33}} \ S^{\varepsilon_{12}} \ S^{\varepsilon_{13}} \ S^{\varepsilon_{23}}]$). Similarly, stresses returned by the network are scaled by the stress scaling vector $\boldsymbol{S}^{\sigma}$.


σ_{11} σ_{22} σ_{12}
$\frac{\sigma_{11}}{S^{\sigma_{11}}}$ $\frac{\sigma_{22}}{S^{\sigma_{22}}}$ $\frac{\sigma_{12}}{S^{\sigma_{12}}}$
$\frac{\varepsilon_{11}}{S^{\varepsilon_{11}}}$ $\frac{\varepsilon_{22}}{S^{\varepsilon_{22}}}$ $\frac{\varepsilon_{12}}{S^{\varepsilon_{12}}}$
ε_{11} ε_{22} ε_{12}
σ_{11} σ_{22} σ_{33} σ_{12} σ_{13} σ_{23}
$\frac{\sigma_{11}}{S^{\sigma_{11}}}$ $\frac{\sigma_{22}}{S^{\sigma_{22}}}$ $\frac{\sigma_{33}}{S^{\sigma_{33}}}$ $\frac{\sigma_{12}}{S^{\sigma_{12}}}$ $\frac{\sigma_{13}}{S^{\sigma_{13}}}$ $\frac{\sigma_{23}}{S^{\sigma_{23}}}$
$\frac{\varepsilon_{11}}{S^{\varepsilon_{11}}}$ $\frac{\varepsilon_{22}}{S^{\varepsilon_{22}}}$ $\frac{\varepsilon_{33}}{S^{\varepsilon_{33}}}$ $\frac{\varepsilon_{12}}{S^{\varepsilon_{12}}}$ $\frac{\varepsilon_{13}}{S^{\varepsilon_{13}}}$ $\frac{\varepsilon_{23}}{S^{\varepsilon_{23}}}$
ε_{11} ε_{22} ε_{33} ε_{12} ε_{13} ε_{23}
1 0.5 0 -0.5 -1
-4 -2 0 2 4
1 0.8 0.6 0.4 0.2 0
-4 -2 0 2 4

(a) (b) (c) (d)

Figure 2.2: (a) 2-D NNCM (b) 3-D NNCM (c) Hyperbolic tangent activation function (d) Logistic activation function

Output stress is computed by providing a strain vector at the input and propagating the signal forward through the network:

$$\sigma_i = S^{\sigma_i}\phi_i\Bigg(\sum_{c=1}^{N_{h_2}} w_{ic}\phi_c\Bigg(\sum_{b=1}^{N_{h_1}} w_{cb}\phi_b\Bigg(\sum_{a=1}^{N_i} w_{ba}\phi_j\Big(\frac{\varepsilon_j}{S^{\varepsilon_j}}\Big)\Bigg)\Bigg)\Bigg), \tag{2.3}$$

where σ_i is the value computed by the ith output node. The values f_i, f_c,

[1] The 2-D networks approximate plane-stress conditions and thus only accept/return the lateral, axial, and shear components of stress/strain.

and f_b are the activations of nodes in the output layer, second hidden layer, and first hidden layer, respectively. N_{h_2} is the number of nodes in the second hidden layer, N_{h_1} is the number of nodes in the first hidden layer, and N_i specifies the number of nodes in the input layer. Weights from node p in layer $N-1$ to node q in layer N are denoted as w_{qp}.

The NNCMs shown in Fig. 2.2 use a hyperbolic tangent as the activation function of the nodes ($\phi(\cdot) \triangleq \tanh(\cdot)$). A new NNCM architecture introduced in Chapter 4 utilizes the logistic activation function for a portion of the network. Both of these activation functions are shown in Figs. 2.2c-2.2d.

There are two important characteristics of these functions to be noted. Both 1) have bounded outputs and 2) drastically lose sensitivity as the input extends beyond ± 1. With respect to the first point, no output node can produce a value $\geq \pm 1$. Therefore, if the NNCM is to learn values of stress outside of this range, they must first be scaled, hence the output scaling values $S^{\sigma_{11}}$–$S^{\sigma_{23}}$. In regard to the second point, even though there is no bound on the range of inputs to the network, dividing the input values with $S^{\varepsilon_{11}}$–$S^{\varepsilon_{23}}$ to keep the strains with ± 1 helps ensure the NNCM remains sensitive to changes at the input.

Scaling the input and output vectors can do more than keeping the values within given bounds; scaling can also increase the range of values closer to the bounds. Strains induced during QUSE are usually contained within a fairly small range. For instance, if all the strains were < 0.01, all of the input data would be packed into a small range. If the goal is to maintain sensitivity to changes at the input, then it would be logical to scale the strains so that they range closer to ± 1. Similarly, for the case where the stresses are all small and tightly bound, it could be beneficial to scale the output values closer to ± 1.

While NNCMs are capable of characterizing the material properties of any object given appropriate inputs and outputs, the necessary training data must be available. Measurements acquired in QUSE are forces and displacements whereas NNCMs learn a stress-strain relationship. We therefore need a method to convert force-displacement measurements to stresses and strains. Finite element analysis is used for this purpose. However, the solution of a FEA requires knowledge of the underlying material property distribution, which is the problem trying to be solved with NNCMs. FEA is briefly reviewed in the next section followed by an overview of AutoP describing how the combination of NNCMs and FEA can be used to learn material properties from force-displacement data.

2.2 Finite Element Analysis

Many texts rigorously cover FEA procedures (e.g., [115, 84]) and form the basis for this section. A detailed description of FEA is not necessary to understand AutoP. Rather, we aim to provide a high-level view and point out where NNCMs may enter into the formulation.

The governing equation for an elastic continuum (under small strain) in static equilibrium relating internal stresses $\boldsymbol{\sigma}$ to forces $\boldsymbol{f}$ given boundary conditions (BCs) $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{t}}$ is

$$\nabla \cdot \boldsymbol{\sigma} + \boldsymbol{f} = 0 \tag{2.4}$$

$$\boldsymbol{u} = \hat{\boldsymbol{u}} \text{ on } \Gamma_g \tag{2.5}$$

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = \hat{\boldsymbol{t}} \text{ on } \Gamma_h, \tag{2.6}$$

where $\hat{\boldsymbol{u}}$ are displacements imposed on the boundary Γ_g and $\hat{\boldsymbol{t}}$ are surface tractions applied along the boundary Γ_h. Typically, $\hat{\boldsymbol{u}}$ are imposed on the surface but can include internal displacements. Note that Γ_g and Γ_h comprise the full boundary ($\Gamma_{total} = \Gamma_g + \Gamma_h$) and do not overlap ($\Gamma_g + \Gamma_h = \varnothing$).

Consider the simple model in Fig. 2.3 that is a simplification of the experimental phantom measurements described in the next chapter. The bottom surface of the phantom is "pinned" and cannot move. A series of N_p concentrated force loads $\hat{\boldsymbol{p}}^{i,n}$ are applied to the top surface, where the superscripts i and n indicate node number and load increment. Experimentally, we apply the surface forces and measure some of the resulting displacements $\hat{\boldsymbol{u}}^{i,n}$. Only surface displacement measurements are shown in Fig. 2.3, but internal displacements can also be measured (e.g., at nodes 4, 5, and/or 6).

In a FEA, BCs are applied and displacements for all nodes in the mesh are computed. Deformation of the body results an in internal resisting force vector $\boldsymbol{I}^n$. Surface tractions, concentrated force loads, and body forces $\boldsymbol{f}$ comprise the vector $\boldsymbol{P}^n$ of forces acting on the body. The goal of the FEA

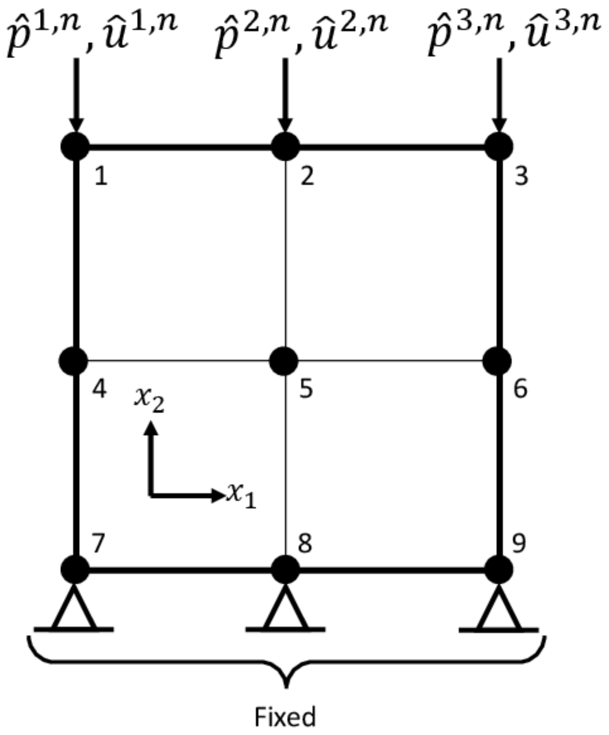

$\hat{p}^{1,n}, \hat{u}^{1,n}$
$\hat{p}^{2,n}, \hat{u}^{2,n}$
$\hat{p}^{3,n}, \hat{u}^{3,n}$
1
2
3
4
5
6
7
8
9
x_2
x_1
Fixed

Figure 2.3: Example FE model.

is to find the displacement of the nodes that balance $\boldsymbol{I}^n$ and $\boldsymbol{P}^n$:

$$\boldsymbol{P}^n = \boldsymbol{I}^n. \tag{2.7}$$

Suppose the phantom exhibits non-linear mechanical properties $N(u)$ illustrated by the red curve in Fig. 2.4a. One method of solving the non-linear FEA is through a Taylor expansion of (2.7) and Newton-Raphson (N-R) iterations. $\boldsymbol{P}^n$ remains constant while the tangent stiffness matrix $\boldsymbol{K}_i^n$ and vector of internal resisting forces are updated at the end of each N-R iteration. The following set of equations details the solution procedure for the

ith N-R iteration. Fig. 2.4a diagrams the process.

$$\boldsymbol{P}^n = \boldsymbol{I}^n_{i+1} \tag{2.8}$$

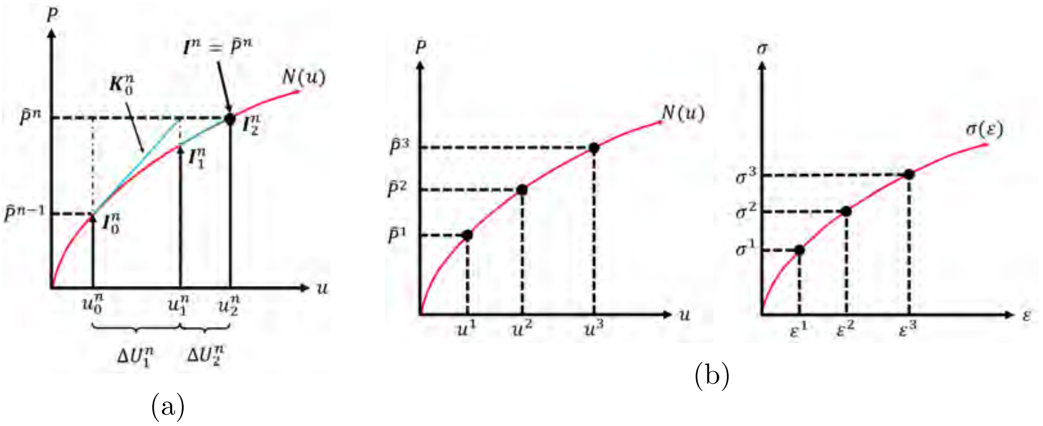
$$\boldsymbol{I}^n_{i+1} = \boldsymbol{I}^n_i + \boldsymbol{K}^n_i \Delta \boldsymbol{U}^n_{i+1} \tag{2.9}$$

$$\boldsymbol{K}^n_i \Delta \boldsymbol{U}^n_{i+1} = \boldsymbol{P}^n - \boldsymbol{I}^n_i \tag{2.10}$$

$$\boldsymbol{K}^n_i = \sum_{e=1}^{N_e} \int_{\Omega_e} \boldsymbol{B}^T_e \boldsymbol{D}^n \boldsymbol{B}_e \mathrm{d}\Omega_e \tag{2.11}$$

$$\boldsymbol{I}^n_i = \sum_{e=1}^{N_e} \int_{\Omega_e} \boldsymbol{B}^T_e \boldsymbol{\sigma}^n_i \mathrm{d}\Omega_e \tag{2.12}$$

Equation 2.10 is derived by replacing the internal force term in (2.8) with (2.9). The tangent stiffness matrix in (2.11) is assembled from all N_e elements in the mesh. $\boldsymbol{B}^T_e$ is the strain-displacement matrix defined for each element and is determined by the mesh. A constitutive model is used to compute the stiffness matrix $\boldsymbol{D}^n$ and element stress vector $\boldsymbol{\sigma}^n_i$.[2]



(a)

(b)

Figure 2.4: (a) Diagram of the N-R iterations to solve a FEA. (b) Points in the force-displacement domain map to points in the stress-strain domain.

Equations (2.11) and (2.12) are the entry point for a NNCM in a FEA. We have already defined how to compute stress from a NNCM in (2.3). The stiffness matrix can also be computed from the connection weights and activations of the network. Appendix D contains the derivation of $\boldsymbol{D}^n$. Con-

[2] Integrals in eqs. 2.11 and 2.12 are computed numerically with Gauss quadrature. Each integral is calculated by evaluating it at a predetermined number of *integration points* (or Gauss points) and summing the results. Different element types have different numbers of integration points. Furthermore, stresses and strains are evaluated at the integration points. The number of stress-strain data pairs discussed in later chapters is thus determined by the quantity and type of elements in the mesh.

sequently, the classic constitutive model can be replaced by a NNCM in FEA.

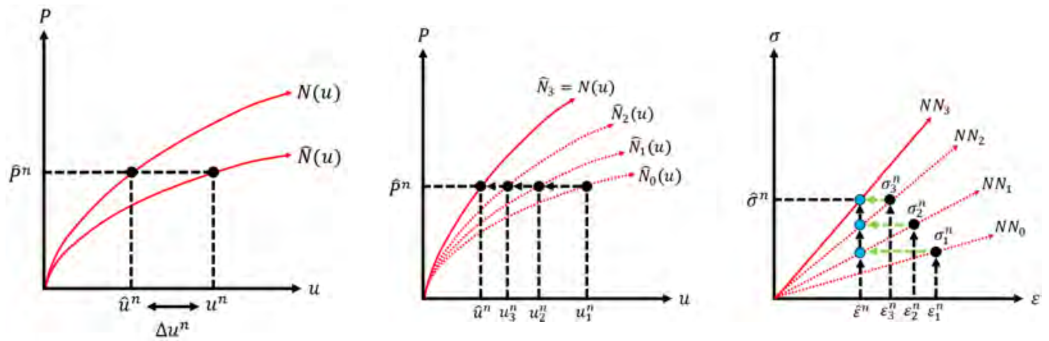
Returning to the object in Fig. 2.3, applying force loads $\hat{\boldsymbol{P}}^1$–$\hat{\boldsymbol{P}}^3$ and solving (2.10) results in a mesh displacements $\boldsymbol{u}^1$–$\boldsymbol{u}^3$. Similarly, $\hat{\boldsymbol{u}}^{i,n}$ can be applied in the FEA (without applying the surface forces) to compute the displacements of the remaining nodes. No matter how the object is being loaded, mesh deformations computed in a FEA must satisfy *stress equilibrium* defined in (2.4) relating stresses to forces and *compatibility* requirements relating displacements to strains. One simple definition of compatibility is that elements in the mesh cannot separate or overlap. As will be explained in the next section, AutoP takes advantage of these requirements to train NNCMs.

Stresses and strains can be computed from the solution of a FEA. Strains can be calculated from displacements through kinematic relationships like that defined in (1.3). Then, using a constitutive model (or NNCM), stresses are computed from the strains. Thus, the points in the force-displacement domain (Fig. 2.4b, left) correspond to points in the stress-strain domain (Fig. 2.4b, right). In other words, stresses and strains (the data we want to train NNCMs) can be computed from mesh deformations caused by force and/or displacement loads (the data we measure).

2.3 Overview of the Autoprogressive Method

Measurement data are in the form of forces and displacements, FEA can convert force-displacement measurements to stresses and strains, and NNCMs learn stress-strain behavior. AutoP exploits these facts to build data-driven models that gradually learn the stress-strain behavior encoded in the force-displacement measurements. This section is intended to explain why AutoP works and differentiate our data-driven method from model-based approaches.

Recall that Fig. 2.3 is the scenario where forces $\hat{\boldsymbol{p}}^{i,n}$ were applied and the displacements at the same nodes were measured. Let the true material properties of the phantom be described by $N(u)$ in Fig. 2.5a. Applying the measured forces to the FE model and using $N(u)$ to solve (2.11) and (2.12) will return the measured displacements $\hat{\boldsymbol{u}}^{i,n}$ (along with the displacements of nodes 4–6). However, because the material properties are unknown, we start with an initial guess $\hat{N}(u)$. Solving the FEA with $\hat{N}(u)$ after applying force BCs would result in the displacements $\boldsymbol{u}^n$. The difference between the measured displacements and those computed with the initial guess of material properties is $\Delta\boldsymbol{u}^n$. The goal of AutoP is to adjust $\hat{N}(u)$ to minimize $\Delta\boldsymbol{u}^n$ (i.e., minimize the objective function in (1.9)). Simply put, change $\hat{N}(u)$ so that it matches $N(u)$.



(a)

(b)

Figure 2.5: (a) Measured displacements $\hat{u}^n$ and displacements computed in a FEA u^n resulting from an applied force will differ when the material properties $\hat{N}(u)$ defined by an NNCM do not match the true material properties $N(u)$ of the object. (b) During training in AutoP, the NNCM gradually learns the true material properties. Blue points representative stress-strain data used to train the NNCM.

Notice the force and displacement measurements in Fig. 2.3 are along the same boundary. Generally speaking, there will be some overlap in boundaries Γ_g and Γ_h for the measurement data. The FEA formulation forbids force and displacement BCs to be defined simultaneously on the same boundary. We therefore apply the measurements in separate FEAs to compute mesh deformation in response to each set of BCs. Define FEA$_\sigma$ as the FEA where measured forces are applied to the mesh and FEA$_\varepsilon$ as the FEA where measured displacements are imposed. Displacement BCs for the fixed bottom boundary are applied in both FEA$_\sigma$ and FEA$_\varepsilon$.

Let NN_0 in Fig. 2.5 be the stress-strain relationship described by a pretrained NNCM[3] and $\hat{N}_0(u)$ be the corresponding force-displacement relationship. In the first AutoP iteration, measured forces $\hat{\boldsymbol{P}}^n$ are applied in FEA$_\sigma$ and displacements $\boldsymbol{u}_1^n$ (subscripts indicate AutoP iteration) are computed

[3] NNCMs are pretrained with stress-strain data generated from the linear-elastic model in (2.1). Meaning, the initial "guess" is a linear-elastic material with a specified Young's modulus and Poisson's ratio. Elastic pretraining is described in further detail in the next chapter.

using NN_0 in the solution of the FEA. These displacements correspond to strains which in turn can be used to compute stresses $\boldsymbol{\sigma}_1^n$ throughout the object with NN_0. Given that equilibrium requirements relate stresses to forces, stresses computed in FEA$_\sigma$ are assumed to be physically consistent with "true" stresses that we are trying to estimate. Thus, from measured forces we obtain an estimate of stress.

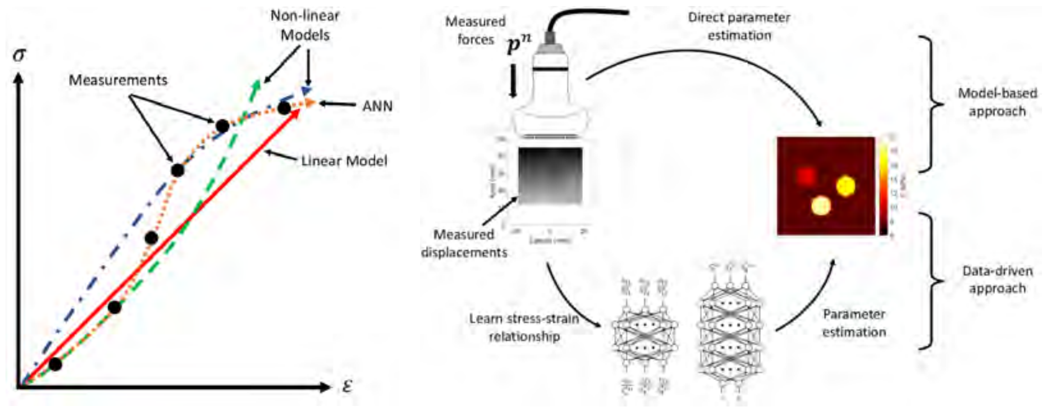
In FEA$_\varepsilon$, we apply the corresponding displacement measurements $\hat{\boldsymbol{u}}^n$ and again solve the FEA using NN_0 as the material model. This time, the displacements in the mesh are used to compute strains $\hat{\boldsymbol{\varepsilon}}^n$.[4] $\boldsymbol{\sigma}_1^n$ are paired with $\hat{\boldsymbol{\varepsilon}}^n$ and are used to retrain the NNCM: $NN_0 \rightarrow NN_1$. Mechanical behavior characterized by NN_1 more closely resembles the true properties of the material than NN_0.

The diagram in Fig. 2.5b helps explain what is happening in each AutoP iteration that causes the NNCM to converge to the true material properties. If $\hat{N}_0(u)$ does not match $N(u)$, pairing stresses from FEA$_\sigma$ and strains from FEA$_\varepsilon$ creates a point (blue dot) that does not reside on the curve created by NN_0 (black dot). In this example, the NNCM must "become stiffer" in order to reconcile the stresses and strains computed in FEA$_\sigma$ and FEA$_\varepsilon$, respectively. Through repeated AutoP iterations, the blue and black dots in Fig. 2.5b eventually overlap, at which point $\hat{N}(u) = N(u)$ and consequently minimize $\Delta\boldsymbol{u}^n$

The important aspects of AutoP are 1) the training data comes from the solution of FEAs so the material properties learned by the network must obey physical laws, 2) the NNCM participates in *generating its own training data* and therefore must reconcile stresses and strains resulting from the force-displacement measurements, and 3) NNCMs are flexible enough to learn any stress-strain behavior.

Fig. 2.6a highlights point (3). Measurements (black dots) are governed by the material properties of an object. Model-based methods adopt a constitutive model to describe the measurements. For example, assuming the material properties are linear-elastic (red line) or non-linear (green dashed line or blue dash-dotted line). Different models can capture various aspects

[4] The diagram suggests that applying measured displacements to the FEA will always result in the same strain. Unless displacement BCs are given for every node in mesh, $\hat{\varepsilon}^n$ will also change. However, for the homogeneous model considered in this example where displacements are applied along the entire top boundary, the strains compute in FEA$_\varepsilon$ will change very little.



(a) (b)

Figure 2.6: (a) Model-based methods may discard important information pertaining to tissue material properties if an inappropriate mechanical model is selected. On the other hand, NNCMs are theoretically flexible enough to characterize any type of mechanical behavior. (b) Our data-driven approach does not directly estimate material parameters from measurement data, as it done in model-based inverse methods. Instead, material parameters from any constitutive model can be estimated from NNCMs after AutoP training. This ability may allow for discovery of parameters most useful for diagnostic imaging.

of the material behavior. When solving the inverse problem, the form of the model is fixed and the parameters are adjusted to minimize $\Delta\boldsymbol{u}^n$. Unfortunately, adopting the incorrect model will result is modeling errors, as stated back in Sec. 1.3, and may discard potentially relevant clinical information. E.g., a linear-elastic model cannot describe any non-linear properties.

On the other hand, NNCMs (orange dotted line) have the ability to learn material properties without discarding any information. We do not adjust parameters to minimize $\Delta\boldsymbol{u}^n$; rather, we *adjust the function itself.* Fig. 2.6a summarizes the fundamental difference between model-based inverse methods and our data-driven approach. The former directly estimates parameters of a pre-selected constitutive model whereas NNCMs learn a non-parametric description of the stress-strain relationship governing the data. A NNCM can then be tested against any number of constitutive models to estimate the relevant parameters, creating a way to discover which parameters best describe the material properties inherent to the data and are most clinically relevant.

Chapter 3

Neural Network Constitutive Models of Gelatin Phantoms

3.1 Introduction

Medical elasticity imaging encompasses a broad range of techniques for imaging mechanical properties of biological tissues. Standard approaches to patient imaging begin by tracking displacements resulting from a weak mechanical stimulus applied to the tissue being imaged. Combining time-varying displacement information with measurements of or assumptions about the associated forces into a collection of linear equations, the elements of a constitutive matrix are estimated.

Elasticity imaging reveals how tissue stiffness and/or viscosity vary with position, time, and applied force. These basic mechanical properties can indicate regions of inflammation, edema, hypertrophy, and fibrosis that accompany the presence of disease processes [116]. Conventional approaches show great promise for cancer imaging in particular. However, simplifying assumptions, which enter the analysis as tissue models, can miss information about the structural complexity of most tissues. These assumptions are necessary to formulate a properly constrained inverse problem for mechanical parameter estimation. Violating the assumptions of a tissue model can distort the description of the mechanical environment needed for diagnostic decision making. Unjustified assumptions also limit the vast possibilities for learning more about the role of mechanobiology in revealing disease processes. Furthermore, simple models in common use generally discard some of the information gathered during an imaging exam.

This report describes a different approach to elasticity imaging that re-

The majority of this chapter is reproduced from [108]. Reprinted by permission from Springer Nature: Biomechanics and Modeling in Mechanobiology. ”An information-based machine learning approach to elasticity imaging”, Cameron Hoerig, Jamshid Ghaboussi, and Michael F. Insana, ©2018 quires no assumptions about tissue mechanical properties, including linearity and isotropy. Our initial focus is on quasi-static methods that generate full stress and strain maps from just a few force-displacement measurements and knowledge of object shape. Estimation of constituent-matrix elements for elasticity imaging occurs retrospectively, once the stress and strain fields have been accurately estimated.

The technique accepts a time-series of surface force-displacement measurements made while pressing a rigid ultrasound transducer into the medium surface. Simultaneously, the transducer records radiofrequency echo data used to estimate a time series of displacements at select points in the medium. These force-displacement data are input into the Autoprogressive (AutoP) method [117] that employs two finite-element analyses (FEAs) to exploit (a) equilibrium conditions relating forces and stresses and (b) compatibility requirements relating displacements and strains. We will show how these FEA processes occur simultaneously and iteratively by relating them to each other through one or more artificial neural networks (ANNs) that learn and record material properties. The ANNs take the place of the constitutive matrix, relieving us of the need for simplifying assumptions. The material properties learned by the ANNs during training with AutoP are stored in the distributed connection weights. Later, and without further measurements, the informational model can be probed to find imaging parameters in the range of mechanical stimuli that were used to train the model.

An important part of quasi-static deformation analysis is that a mechanical stimulus applied to the medium may be sensed at every contiguous location in that medium. Consequently, a few well-positioned measurements made during quasi-static deformation can result in a complete mechanical description, even for large deformations in nonlinear media. Combining FEA and ANNs in AutoP allows the material properties to be extracted from a sparse sampling of force-displacement measurements.

For the first time, complete and accurate images of all stress and strain fields throughout an object resulting from an applied force can be described from experimental measurements. The advancement to the art is in combining a few strategically-placed measurements with mechanical principles and information-based machine learning tools to solve the inverse problem without making any initial assumptions of the underlying constitutive model. Furthermore, the trained ANNs are not used to classify or interpret images;

instead, they are used to construct images of material properties. Our informational models usher in a new approach to implementing ANNs for medical imaging applications.

Despite some similarities with previous reconstruction methods in elasticity imaging that use FEA to iteratively solve the inverse problem[92, 89], we do not apply mathematical inverse methods to directly estimate a constitutive matrix. Existing iterative schemes still require the assumptions of a constitutive model with a compact feature set. The ANNs in AutoP replace the constitutive matrix, eliminating the need for assumptions about material properties and boundary values.

In this introductory report, we consider relatively simple physical phantoms with known properties to demonstrate the method. Heterogeneous phantoms composed of linear-elastic gelatin gels with known shapes and stiffnesses were built to develop and validate these methods. An ultrasonic linear array is used to compress phantoms under plane-stress conditions. We expand upon our previous reports [118, 119] and show that all stress and strain fields for 2-D and 3-D measurement geometries, along with the spatially varying Young's modulus, can be accurately reconstructed even for locations outside the region sampled ultrasonically.

3.2 Methods

A constitutive model relates output stress to input strain (or vice-versa) through mathematical equations based on known physical principles and experimentally observed behavior. For systems too complex to have a complete reductionist description, informational models can describe input-output relationships via machine learning techniques and training involving repeated exposure to high-quality experimental data. In the following section, we describe AutoP as applied to data from tissue-like gelatin phantoms.

3.2.1 Force-displacement Measurements

The same phantom manufacture and experimental measurement techniques were used in this study and those described in the next two chapters. Imaging phantoms were constructed using a simple gelatin formulation of water, gelatin powder, and corn starch that acts as a scattering agent [120]. In this study, each phantom was a 50x50x50 mm^3 cube of a soft gel (≈9 kPa) with one or three embedded stiff cylindrical inclusions (≈22-26 kPa), 10 mm in diameter and 50 mm long. Dimensions of the phantoms and placement of the inclusions were tightly controlled via machined molds. By pouring the gelatin mixture for the inclusions after the background material solidified, we were able to create a tightly coupled bond between the background and inclusions. Macro-indentation methods [121], described in the next section, were applied to independently estimate the Young's modulus of each gelatin material. The long axis of each inclusion was oriented along the x_3 axis (out of plane of Fig. 3.1). Each phantom component is a linear-elastic, incompressible material with a gelatin powder concentration that determines the gel stiffness.

The basic measurement technique involves pressing an ultrasound linear-array transducer into the phantom surface in a series of discrete steps (Fig. 3.1). The US probe face was positioned flush with the top surface of the phantom in the (x_1, x_3) plane that are, respectively, the lateral and elevational axes of the transducer. A plastic housing used for positioning encased the US probe, creating a phantom contact area of 43x50 mm^2. A 6-axis force-torque transducer (ATI Industrial Automation, Apex, NC) coupled the ultrasound probe housing to a motion controller capable of positioning the probe with sub-millimeter accuracy (not shown). The bottom surface of the phantom was fixed to a rigid base so it did not slip as the probe compressed the top phantom surface. Coupling gel was applied to the top surface and provided free-slip contact between the probe and phantom. The remaining four surfaces of the phantom were unconstrained during loading.

In this study, only a downward compressive force was slowly applied along the axis of the ultrasound beam (x_2 axis). A radiofrequency (RF) echo-signal frame was acquired after each load increment using a Siemens Sonoline Antares ultrasound system (Siemens Healthcare USA, Mountainview, CA) with a VF10-5 linear-array probe at 8 MHz center frequency. At the same time, the axial surface force generated by the US probe was measured by the force-torque transducer and the probe position recorded via the motion controller. Application of a speckle-tracking algorithm [41] provided an estimate of internal displacements within the RF frames (see Appendix B for an evaluation of the accuracy of the speckle-tracking algorithm). One series

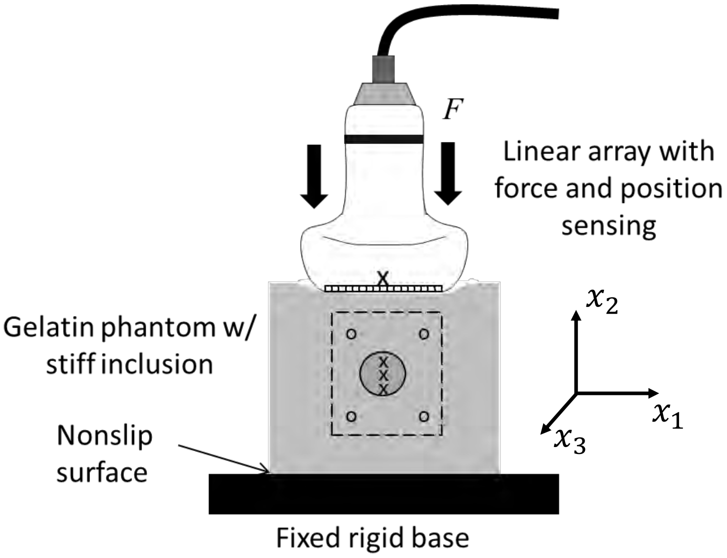

F
Linear array with force and position sensing
Gelatin phantom w/ stiff inclusion
Nonslip surface
Fixed rigid base
x_2
x_1
x_3

Figure 3.1: Diagram of the experimental setup using a single-inclusion gelatin phantom scanned along the cross section of the inclusion. A centrally-positioned ultrasound probe coupled to a force/torque transducer (not shown) is maneuvered by a 3-D positioning system. The phantom is placed on a fixed non-slip base and is compressed from above with surface force p as an RF echo frame (dashed line region) is acquired. Speckle-tracking applied to echo signals measures displacements within the RF echo frame. Locations marked '×' indicate positions where displacements were applied in FEA$_\varepsilon$ for training. These displacements are also used for convergence testing. Locations marked '∘' are examples of locations where additional displacement data can be used for FEA$_\varepsilon$ and/or convergence testing.

of load increments provided a set of surface forces and probe displacements that were measured directly and a corresponding set of displacements within the phantoms which required speckle tracking to estimate. These data were then used as inputs to AutoP to build informational models of the gelatin phantoms. Information regarding the geometry of the phantoms was also necessary to develop finite-element meshes used in AutoP. External boundaries were known, but the shape and location of internal structures must be assumed or estimated. In Sections 3.2.5 and 3.2.6, we investigate cases where the inclusion boundaries were modeled exactly or were estimated via manual segmentation.

3.2.2 Macroindentation Measurements

A small amount of each gelatin material was poured into a separate cylindrical mold while manufacturing the phantom to undergo macroindentation testing. All indentation measurements in this dissertation were performed using a TA.XTplus Texture Analyzer (Stable Micro Systems Ltd., Surrey, U.K) with a 1 kg load cell and a 2.5 mm radius stainless steel spherical indenter tip. A force-displacement curve was generated by slowly pressing the spherical probe into the surface of the gelatin at a rate of 0.05 mm/s, up to 1.5mm total. Space was left at the top of the cylindrical mold and filled with water before the indentation test began to remove static electric forces between the probe and the gelatin. Without the water, the surface of the gelatin may "jump" up to the probe before contact is made, resulting in a negative force reading. Adhesion between the probe and gelatin was limited with the use of a lubricant (Pol-Ease 2300 Release Agent, Polytek Development Corp., Easton, PA).

Under the assumption of no adhesion and elastic bodies, a Hertzian contact model between a sphere and a half-space was used to estimate the Young's modulus of the gelatin, as was done in [121]:

$$F = \frac{4}{3}E^{*}\sqrt{R}\sqrt{d^{3}} \tag{3.1}$$

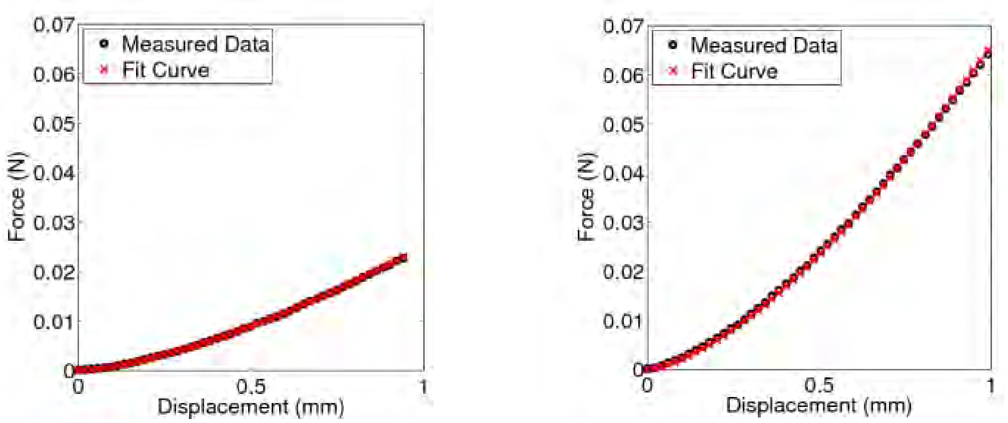
and E^{*} is defined as

$$\frac{1}{E^{*}} = \frac{1-\nu_{sphere}^{2}}{E_{sphere}} + \frac{1-\nu_{gelatin}^{2}}{E_{gelatin}} \tag{3.2}$$

In (3.1), R and d correspond to the radius and displacement of the indenter, respectively, and F is the measured force. E^{*} is a function of the Young's and Poisson's ratio of the spherical indenter (E_{sphere} and ν_{sphere}, respectively) and of the gelatin ($E_{gelatin}$ and $\nu_{gelatin}$). The large value of the Young's modulus for the spherical indenter allows the corresponding term in (3.2) to be neglected. Then, putting 3.2 into 3.1, the contact model reduces to

$$F = \frac{16}{9}E_{gelatin}^{2}\sqrt{R}\sqrt{d^{3}} \tag{3.3}$$

An example of the force-displacement curve generated during indentation tests on the gelatin materials comprising Phantom 1 can be seen in Figure 3.2. The curve for the soft background is plotted in Fig. 3.2a whereas Fig. 3.2b is a plot of the curve for a stiff inclusion. Fitting each measured curve (solid black line) to the contact model in (3.3) produced a Young's modulus estimate for the material. Estimated modulus values could then be used to create estimated force-displacement curves for the indentation (dotted red line).



(a) (b)

Figure 3.2: Examples of the force-displacement curve generated during a spherical indentation test. The solid black line in each figure is the measured result whereas the red dotted line is the data fit to the Hertzian contact model described in (3.3). Both measurements are from indentation tests on the gelatin materials that compose Phantom 1 of this study. (a) Spherical indentation on soft background gelatin, $E = 9.16$ kPa. (b) Spherical indentation on stiff inclusion gelatin, $E = 22.9$ kPa.

3.2.3 Autoprogressive Method

Overview

At the core of our approach to create empirical models of mechanical behavior is the Autoprogressive Method developed by [117]. The power of AutoP lies in its ability to generate stress-strain information from force-displacement measurements. AutoP makes use of standard FEA methods (ABAQUS 6.13 finite-element software) to iteratively generate stress and strain data as artificial neural networks learn material properties.

A finite-element (FE) mesh is constructed that models the data acquisition process. For our measurements described in later sections, the FE mesh comprised a rigid, rectangular model of the US probe in contact with a square (or cube for 3-D) mesh of a phantom. Then, measured forces and displacements are split and applied to the FE model to solve two different FEAs: FEA$_\sigma$ and FEA$_\varepsilon$. Solving FEA$_\sigma$ allows us to estimate the stresses throughout the model in response to an applied force. Similarly, solving FEA$_\varepsilon$ provides an estimate of the strains resulting from an applied displacement. Because forces and displacements are both measured for each experiment, the strains provided by FEA$_\varepsilon$ should be consistent with the stresses found in FEA$_\sigma$ when the ANNs that relate the two FEAs correctly represent material properties. Iteratively solving FEA$_\sigma$ and FEA$_\varepsilon$ while adjusting ANN weights produces increasingly accurate estimates of the stress and strain existing in the medium at the time forces and displacements were measured. Through the course of AutoP, the ANNs learn the material properties based on the stress-strain behavior estimated by the FEAs.

ANNs are selected as the machine learning component of AutoP because of their inherent nonlinearity, robustness, and ability to learn complex behaviors [110, 122]. Their highly parallel structure accumulates information as it becomes available experimentally. A general concern when using ANNs is determining the size of the network to be used. ANNs with too many nodes in each layer have a capacity exceeding that necessary to learn mechanical behavior and can generate overfitting errors. On the other hand, if there are too few nodes per layer, the ANN does not have enough capacity to store the necessary information. We implemented nested adaptive neural networks [110] to minimize these issues.

Creating Finite-element Meshes

Training with AutoP begins with the development of a FE model of the data acquisition process. In the FE model, probe-phantom contact and external boundary conditions are set to mimic the experimental setup; for compressions of the gelatin phantoms, the bottom edge of the phantom is fixed, the contact surface between the ultrasound probe and phantom is assumed frictionless, and the remaining sides are free. In this initial technique, one ANN is used for each region that could have distinct material properties. Here, only two ANNs were used: NN$_{soft}$, which describes the material properties of the soft background gelatin material, and NN$_{stiff}$, which characterizes the stiff inclusion(s). Only one ANN is needed for the inclusions in the three-inclusion model because each is composed of the same gelatin mixture. If a different material was used for each inclusion, three different ANNs would be required, bringing the total to four. Shape estimation is required for meshing and to determine the number of ANNs applied.

Elastic Pre-training

Once the FE mesh is created and the ANNs are assigned to the mesh elements, each ANN material model is pre-trained with stress-strain data generated using linear-elastic equations. This phase is for ANN initialization only to avoid non-physical behavior that emerges from random connection weights. Note that the model selected for pre-training is not important. During the course of training with AutoP, the linear-elastic behavior learned during pre-training is "overwritten" as the ANNs learn the mechanical behavior of a specific object. We will show that the ANNs can be pre-trained using data generated with an incorrect assumption of the Young's modulus (i.e., different than the value estimated from macro-indentation measurements) but will learn the stress-strain behavior governing the measured force-displacement data, allowing us to accurately estimate the Young's modulus after training with AutoP.

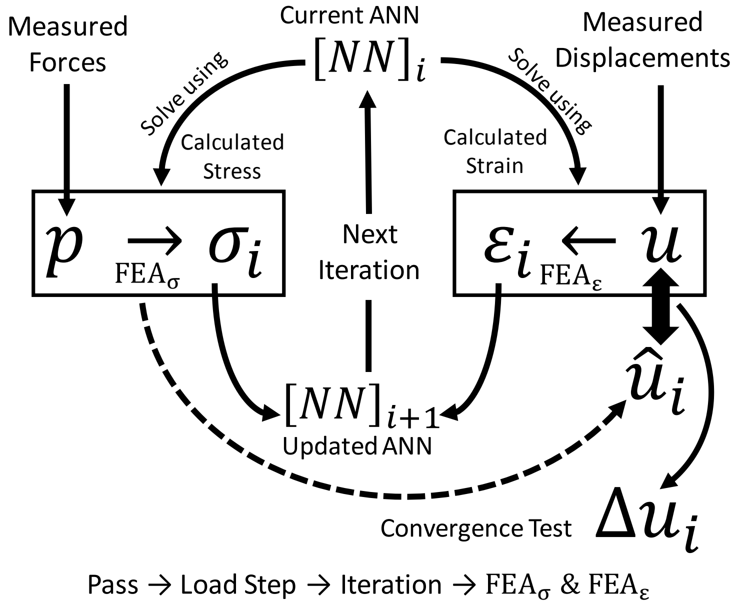


Figure 3.3: Flow diagram of the Autoprogressive method. Measured forces and displacements from one load increment are supplied to finite element analyses $\text{FEA}_\sigma$and FEA_ε. In the i^{th} iteration, current version of the ANNs, $[NN]_i$, are used as the element material models to solve the FEAs, producing estimates of the stress (σ_i) and strain (ε_i) vectors throughout the model. Stress-strain pairs are collected and used to update the connection weights of the ANNs, producing the updated network $[NN]_{i+1}$. This process iterates, and each time displacement estimates $\hat{u}_i$ from FEA_σ are compared with measured displacements u_i to compute error vectors Δu_i. If displacement errors are above set thresholds, the training iterations continue using the same force-displacement data. Once errors fall below the thresholds, training for the i^{th} iteration has converged. This training process is performed for all force-displacement data sets to complete the first pass and may be repeated for the same or new data in a second pass.

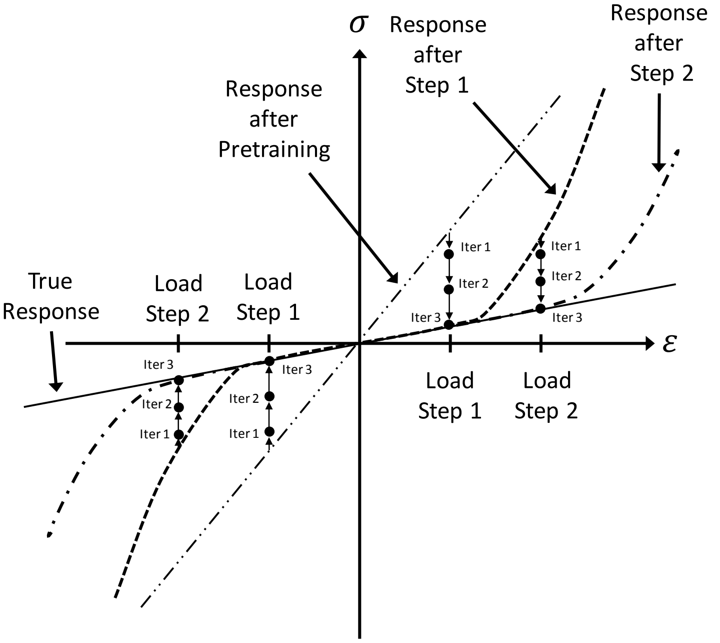


Figure 3.4: Example of the evolution of the mechanical behavior described by an ANN during training with AutoP. After pretraining (-··-), the stress-strain response of the ANN is significantly different than the true mechanical behavior (——). Performing training iterations using force-displacement data from the first load increment (Load Step 1) gradually moves the stress-strain curve produced by the ANN toward the true solution. After completing training iterations using the data from the first load increment, the ANN accurately describes the mechanics only over the strain range produced for the first step (- - -). The valid range of the ANN is increased by continuing training through load step 2 (-·-·).

AutoP Training Iterations

AutoP training iterations, diagrammed in Fig. 3.3, begin after the pretraining phase. Force-displacement data acquired during the first load step (first compression of the gelatin phantom) are applied to the FE model to solve FEA$_\sigma$ and FEA$_\varepsilon$. Measured forces are applied in FEA$_\sigma$ and the FEA is solved using the current state of the ANNs ($[NN]_i$) as the element material models. In the solution of the FEA, equilibrium conditions - which relate forces and stresses - are satisfied, producing an estimate of the stresses throughout the model of the phantom. In the second analysis, FEA$_\varepsilon$, measured displacements are applied and the FEA solved once again using the current state of the ANNs as the element material models. Compatibility conditions, which relate displacements to strains, are satisfied in FEA$_\varepsilon$, producing an estimate of the strains throughout the phantom model. After solving both FEAs, stress vectors computed in FEA$_\sigma$ are paired with their corresponding strain vectors computed in FEA$_\varepsilon$ to create a set of stress-strain training pairs used to update the connections weights of the ANNs ($[NN]_i \rightarrow [NN]_{i+1}$).

Training iterations are performed using the same force-displacement data from the first load step until a predefined maximum number of iterations is reached or the ANNs converge. Convergence of the ANN material models is checked by using the computed displacements from FEA$_\sigma$ ($\hat{u}_i$) and the measured displacements provided in FEA$_\varepsilon$ (u_i) to calculate a vector of displacement errors using Eq. (3.4).

$$\Delta u_i = |u_i - \hat{u}_i| \tag{3.4}$$

Note that the total number of computed displacement errors matches the number of displacements provided in FEA$_\varepsilon$. In Section 3.3.1, we discuss the effect of including additional displacements not used in FEA$_\varepsilon$. These displacement errors are used to calculate two values:

$$c_{max} = \frac{\max(\Delta u_i)}{|\max(u_i)|} \tag{3.5}$$

$$c_{\mu} = \frac{\text{mean}(\Delta u_i)}{|\max(u_i)|} \tag{3.6}$$

If $c_{max} < C^n_{max}$ and $c_\mu < C^n_\mu$, where C^n_{max} and C^n_μ are predefined criteria for the n^{th} load step, convergence has been achieved for the first load step. AutoP iterations are repeated in the same manner using each set of force-displacement data for all load steps. This constitutes one pass of AutoP. Multiple passes may be performed using the same or new data.

The stress-strain response of an ANN during the course of training is visualized in Fig. 3.4. In this example, the true linear-elastic behavior of an imaginary 1-D material (—) is to be learned by an ANN that has been pre-trained (-··-). Force-displacement data from the first load step (Load Step 1) are used in AutoP iterations, causing the stress-strain response of the ANN to gradually move toward the true behavior. Convergence is achieved after several iterations, implying the mechanical behavior described by the ANN matches the true behavior over the stress-strain range produced in the first load step (- - -). Outside of this range, the ANN response becomes nonlinear and diverges from the true response. Performing additional AutoP iterations with force-displacement measurements from other load steps extends the range over which the ANN and the true behavior match (-·-·). Note that if the true mechanical behavior was nonlinear, the inherent nonlinear behavior of ANNs would allow this method to learn those responses from the data.

Mechanical Parameter Estimation

The output of AutoP is one or more trained ANNs that accurately describe the mechanical behavior of every material in the target object over the stress-strain range resulting from the applied loads (e.g., compressions by the US probe). A fully trained ANN essentially replaces the constitutive matrix to compute an output stress vector from an input strain vector as illustrated in Fig. 3.5. We can then conduct numerical experiments where we apply any force (or displacement) within the training range to estimate all components of stress and strain. This data may be applied to a constitutive model, e.g., Kelvin-Voigt, and estimate an elastic modulus or other mechanical parameter that compose the image. This is a profound step away from current elastography methods that require a model to be assumed at the beginning in order to derive a properly constrained inverse problem. With our method, we can explore the parameter space and determine which parameter best fits the mechanical behavior *after* the stress-strain behavior is learned by the ANNs. In section 3.3.1, we will show a method of interrogating a trained ANN material model to determine an appropriate constitutive model for parameter estimation.

3.2.4 Experimental Design

Testing of this new method of modeling the mechanical properties of materials was performed on gelatin phantoms using the data acquisition scheme described in Section 3.2.1. A total of four phantoms were created having background gelatin materials of nominally the same stiffness and one or three stiffer inclusions. Phantoms 1 and 4 contained a single inclusion centered in the (x_1, x_2) plane. Phantom 2 contained a single non-centrally positioned inclusion, and Phantom 3 embeds three inclusions. Example B-mode images of the four phantoms are provided in Fig. 3.6.

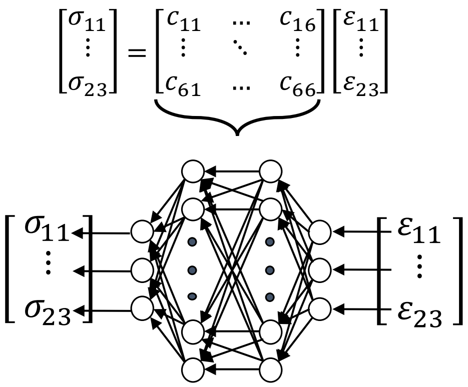


Figure 3.5: After training with AutoP, an ANN material model is able to compute an output stress vector in response to an input strain, essentially replacing the constitutive matrix. The same information described by the components of the constitutive matrix is contained non-parametrically in the connection weights of the ANN. Using the ANN in a computational experimental like a FEA allows the full stress and strain vectors to be estimated throughout an entire material in response to an externally applied force. Access to estimates of all components of stress and strain allows mechanical parameters from any constitutive model to be estimated.

In the following sections, we will describe specific details regarding data acquisition and AutoP training for models under measurement geometries. Regardless of the model, AutoP was used to train feed-forward, fully connected ANNs with hyperbolic tangent activation functions. Each ANN contained two hidden layers and resilient propagation [123] was chosen as the method to update ANN connection weights. The adaptive ANN architecture helped alleviate issues of ANN size. The beginning and ending sizes of the hidden layers in the ANNs varied and will be specified in the following sections.

A total of 30 compressions (load increments) were performed on each phantom. Total probe displacement was 1.5 mm for Phantoms 1-3 (3% of the 50mm phantom height) and 2.5 mm for Phantom 4 (5% of phantom height). Axial forces measured between 12-29 N. The total phantom deformation was kept small in these preliminary studies to ensure linear-elastic response of the phantoms. In future studies, we will show this restriction can be lifted. We trained ANNs with AutoP using 10 nearly equally spaced load steps that spanned the entire range of the applied load, including the first and last load increments. This means force-displacement data from load increments 1, 4, 7, . . . , 26, 30 were used in AutoP. Small load increments were used to minimize artifacts caused by decorrelation and out-of-plane motion when applying the speckle-tracking algorithm for internal displacement estimation. In FEA$_\sigma$, measured forces were applied as concentrated loads to the top nodes of US probe in the FE meshes. Measured probe displacements were applied to the top nodes of probe (same nodes forces were applied) and estimated internal displacements were given at three points within each inclusion in FEA$_{\varepsilon}$. For example, the highlighted points in Fig. 3.6 indicate the specific locations where interior displacements were applied.

In the elastic pre-training phase, a set of random stress-strain vectors were generated using the plane-stress approximation (for 2-D models) or the constitutive matrix for a Hookean solid (for 3-D models). Young's moduli that differed by at least 30% from the values estimated during macro-indentation were chosen to create the pre-training pairs. For example, the Young's modulus of the background gelatin material in Phantom 1 was estimated to be 9.16 kPa from indentation testing whereas a value of 4 kPa was chosen for pre-training NN$_{soft}$. We chose values of Young's moduli that significantly differed from true values to show that our informational models are capable of producing accurate estimates of material properties when initialized to an incorrect state.

After gathering the stress-strain training pairs in an iteration of AutoP, we adjusted the data in two ways. First, the mean and standard deviation of each strain component was computed. We then removed any training pairs with a strain component falling outside of the mean ± one standard deviation range. Preliminary tests indicated that limiting the variation of new information to be learned by the ANNs improved their ability to learn by reducing the impact of outlier data during the weight update stage.

Second, frame invariance (i.e., objectivity) was enforced. Frame invariance means the stress-strain response learned by the ANNs is independent of the chosen axes; that is, if the coordinate system was rotated $\frac{\pi}{2}$ (observed from a different perspective), the resulting stress and strain distributions are unchanged. Objectivity is a fundamental concept in mechanics and requires no assumption of the underlying constitutive behavior. By swapping the normal stress and strain components in each training pair, the ANNs learn frame invariance. Enforcing frame invariance doubled (or tripled in the case of 3-D) the training data obtained in each iteration of AutoP. We must stress that frame invariance is built into the training data to ensure the ANNs learn the fundamental principle, not to increase the total amount of training data. Furthermore, the stress-strain vectors could be rotated by any arbitrary angle in range $0 - 2\pi$; the angle $\frac{\pi}{2}$ was chosen because it did not require any matrix multiplications, therefore reducing the computational load.

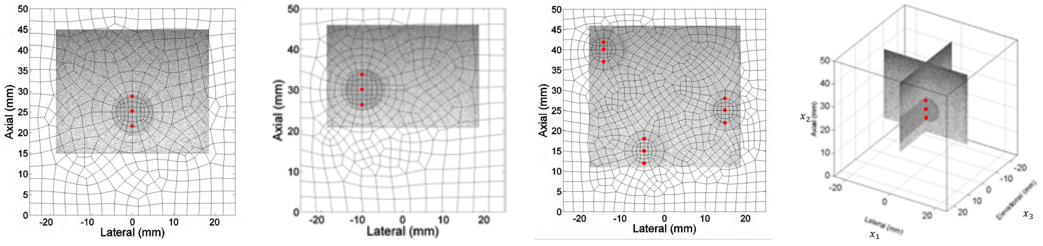
A forward FEA, identified as FEA^n_{NN}, was performed at the end of each training step. This was effectively FEA_σ with the exception that all 10 force measurements were applied. Results from FEA^n_{NN} allowed us to retrospectively analyze the state of the ANNs during AutoP. Stresses and strains computed in FEA^{10}_{NN} - performed upon the completion of AutoP - are of particular interest as they are produced by the fully trained ANNs and will be referenced in Section 3.3.

3.2.5 2-D Models with Known Internal Geometry

Informational models were first developed for 2-D models of Phantoms 1-3 under the assumption that the internal geometry was known. For this, the FE meshes modeled the size and locations of the inclusions determined by the phantom molds, as illustrated in Figs. 3.6a-3.6c. Each FE mesh aligns with the B-mode image within 1mm error, verified by comparing the boundary of the inclusions in the B-mode and mesh. A plane-stress approximation was used in these 2-D models of the phantoms. Therefore, axial, lateral, and shear strains and stresses were used as the input and output of the ANNs, respectively. Four-node, quadrilateral, generalized plane-stress elements (CPS4 in ABAQUS 6.13) were used to create the FE meshes of the phantoms. The US probe was meshed using the same element type, although it was modeled as a rigid body.

Additional data were acquired on Phantoms 2 and 3 by rotating the phantoms about the x_3 axis and applying compressive loads in the same way. New FE meshes were created to model the rotated versions of the phantoms. These meshes were virtually the same as those in Figs. 3.6b and 3.6c, albeit rotated 270° or 90°, respectively.

AutoP training was performed over two passes at ten load steps each. Up to three training iterations were performed in load steps 1-3 and up to five iterations in the remaining steps. Convergence criteria (described in Section 3.2.3) were set as $C^n_{max} = 0.5$ and $C^n_\mu = 0.1$ for all load steps. Displacement errors (Δu_i) were computed only for those nodes provided displacement information in FEA_ε (i.e., only for the nodes highlighted with dots



(a) (b) (c) (d)

Figure 3.6: FE meshes overlayed on B-mode images of (a) Phantom 1, (b) Phantom 2, and (c) Phantom 3. The highlighted nodes in (a)-(c) indicate locations where displacement information was provided during training with AutoP. (d) Orientations of the B-mode images in the 3-D model. For developing 3-D ANN material models, we once again used only three internal displacements. Displacements were provided along the line of intersection of the two B-mode images and given at the same points for both US probe orientations.

in Fig. 3.6).

ANNs were initialized to have three nodes in each hidden layer. Elastic pre-training was performed by generating 100 random stress-strain vectors for an incompressible material (Poisson's ratio $\nu = 0.5$) as described in Section 3.2.3. As AutoP progressed, one node was added to each hidden layer during the second training iteration in load steps 4 and 5 in the first pass, resulting in ANNs that had five nodes per hidden layer at the completion of training.

Two important aspects of our new method are that informational models are 1) better able to learn mechanical behavior when provided with diverse training data and 2) able to update their knowledge as new information becomes available. In AutoP, one way to produce a richer set of training data is to apply a variety of spatial load distributions to the same phantom. For Phantom 1, we acquired data with the US probe centrally positioned in the (x_1, x_2) plane (data set 1, previously described) and with the probe located 4 mm off-center along the x_1 axis (data set 2). Data set 2 was acquired in the same manner as data set 1. From these two data sets on Phantom 1, we explored four training scenarios described as cases in Table 3.1. Cases (a) and (b) compared with cases (c) and (d) explores 1) how the diversity of training data affects modulus estimation accuracy as compared to macro-indentation testing and 2) the ability of ANN material models to incorporate new information and its affect on modulus estimates.

Table 3.1: Six different training scenarios for AutoP. These are the cases referred to in Table 3.2 for Phantoms 1 and 3 (Fig. 3.6). Data set 1 are measurements for a centrally-located US probe (Fig. 3.1). Data set 2 are measurements for an off-center US probe location.

Cases for Phantom 1, Known Geometry	
Case (a)	training using data set 1 only
Case (b)	training using data set 2 only
Case (c)	training using data set 1 and then retraining on set 2
Case (d)	training using both sets simultaneously
Cases for Phantoms 1 and 3, Estimated Geometry	
Case (e)	Good segmentation - boundaries closely followed
Case (f)	Poor segmentation - large errors in boundary estimation

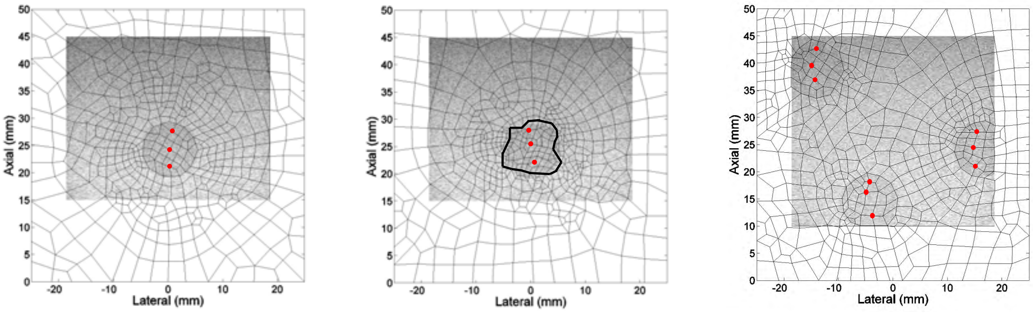
3.2.6 2-D Models with Coarsely Estimated Internal Geometry

Imaging in a clinical setting precludes prior knowledge of the exact internal geometry. Instead, images obtained during data acquisition must be segmented to determine boundaries of each unique material in the object. We tested the ability of our informational models to learn the mechanical properties of the gelatin materials when geometry errors are present in the FE models.

B-mode images of Phantoms 1 and 3 were manually segmented to create the FE meshes used for training. Two cases of segmentation were performed and are described in Table 3.1. In case (e), the boundaries of the inclusion were closely followed, producing a FE mesh with internal boundaries very similar to the meshes in Fig. 3.6. Conversely, case (f) is a segmentation producing large errors in estimating the edge of the inclusion. We note that while the shape of the inclusion was a poor representation of the true geometry in case (f), the general location of the inclusion was accurate.

Training cases (e) and (f) were performed for Phantom 1, while only case (e) was performed for Phantom 3. This latter case poses an interesting problem. The B-mode image does not contain enough information to determine the full shape of all three inclusions. During the segmentation process, the edges of the inclusions were accurate within the imaged region, but parts of each inclusion were cut off and assumed to be part of the background material. Meshes produced by manual segmentation are displayed in Fig. 3.7.

AutoP training was performed with the same data used for Phantoms 1 and 3 with known geometry. Cases (e) and (f) of Phantom 1 were trained



(a) (b) (c)

Figure 3.7: FE meshes produced by manually segmenting B-mode images. (a) Segmentation of Phantom 1 closely matching the internal geometry (case (e)). The RMS error between the true inclusion boundary and the estimated boundary was 0.29mm. (b) Poor segmentation of Phantom 1 where the inclusion boundaries significantly differ from the true boundaries (case (f), RMSE = 0.76mm). The thick, black line indicates the boundary of the inclusion in the mesh. (c) Segmentation of Phantom 3 (case (e), RMSE = 0.76mm). The B-mode image does not capture the full shape of the inclusions. The boundaries of the inclusion within the B-mode are accurately captured, but inclusion boundaries outside of the imaged region are cut off.

in the same manner described in Section 3.2.5 with the exception of ANN sizes and a slight adjustment to pre-training. ANNs began with five nodes per hidden layer and a single node was added to each hidden layer in the fourth load step. Pre-training was performed using 1000 random stress-strain vectors instead of 100.

Case (e) of Phantom 3 required further adjustments to the ANN sizes and convergence criteria. For this training scenario, we initialized the ANNs with eight nodes per hidden layer and added a single node to each hidden layer in the sixth load step. Convergence criteria were set as $C^n_{max} = 0.5$ and $C^n_{\mu} = 0.15$ for load steps 1-3 and $C^n_{max} = 0.2$ and $C^n_{\mu} = 0.08$ for the remaining training steps.

The aforementioned changes to AutoP training were found through preliminary testing. With increased complexity in the stress and strain distributions caused by more complex shapes, it is likely the ANNs with larger hidden layers (i.e., larger capacity for information) were required to accurately describe the stress-strain relationship.

3.2.7 3-D Models with Known Interior Geometry

Our informational modeling approach was extended to 3-D models. For these tests, we returned to the assumption that the interior geometry was known. Data acquired on Phantom 4 was used to train ANN material models relating a six-component output stress vector (three normal components, three shear) to a six-component input strain vector.

Two data sets were acquired on Phantom 4. The first data set was obtained using the measurement setup described above (Fig. 3.1). A second set of compressions was performed where the US probe was rotated 90° about the x_2 axis. B-mode images from both data sets are displayed in Fig. 3.6d. Rotating only the probe allowed us to use the same FE mesh for the gelatin phantom and simply adjusting the orientation of the probe model. General 4-node, tetrahedral element (C3D4 in ABAQUS 6.13) were used to mesh the phantom, whereas the probe comprised 8-node, cubic elements (C3D8) and was modeled as a rigid body. Measured forces were applied as distributed surface loads on the top surface of the probe in FEA$_\sigma$.

We trained ANN material models in separate instances of AutoP with the two different data sets. Once again, interior displacements were applied at three locations within the inclusion in FEA$_\varepsilon$. These points resided along the x_2 axis centered on the (x_1, x_3) plane and are highlighted in Fig. 3.6d. Displacements were applied to the same nodes for both data sets.

ANNs were pre-trained with 8000 random stress-strain vectors computed using a model of a 3-D Hookean, nearly incompressible ($\nu = 0.49$) solid. Each ANN started with eight nodes per hidden layer that was increased to nine in the sixth load step.

3.2.8 Rabbit Kidney Embedded in Gelatin Cube

A rabbit kidney was excised and suspended in the phantom mold, after which a gelatin mixture and manufactured and poured into the mold following the same process described in Sec. 3.2.1. Force-displacement data were acquired in the same manner outlined for the gelatin phantoms. Constructing the FE mesh for the 2-D model was done by segmenting a single B-mode image plane that was acquired with the US probe centered on the phantom in both the lateral and elevational axes. The resulting mesh is displayed in Fig. 3.8a. Highlighted nodes indicate the locations where displacement data was applied in FEA$_\varepsilon$.

Two different tests were performed with the 2-D model. In the first test, a total of four ANNs were assigned to the segmented regions. Fig. 3.8b

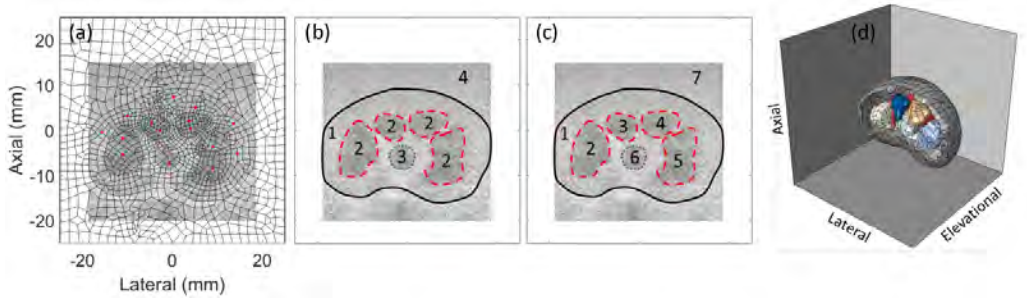


Figure 3.8: (a) 2-D FE mesh created by manually segmenting B-mode image of kidney. (b) Four ANNs were assigned to the segmented regions. Numbers in each region identify the corresponding ANN. (c) Annotated regions for model with seven ANNs. (d) 3-D model of the kidney generated by manually segmenting B-mode images acquired in planes separated by 1 mm along the elevational (x_3) axis.

specifies the number of the ANN characterizing each region. One network was assigned to the background gelatin, one to the cortex of the kidney, one to all four visible medullae, and one for the small area near the hylar region. For the second test, four different ANNs were assigned to each meduallae, as shown in Fig. 3.8c, for a total of seven ANNs. AutoP training parameters matched those defined for the 2-D gelatin phantoms with estimated geometry.

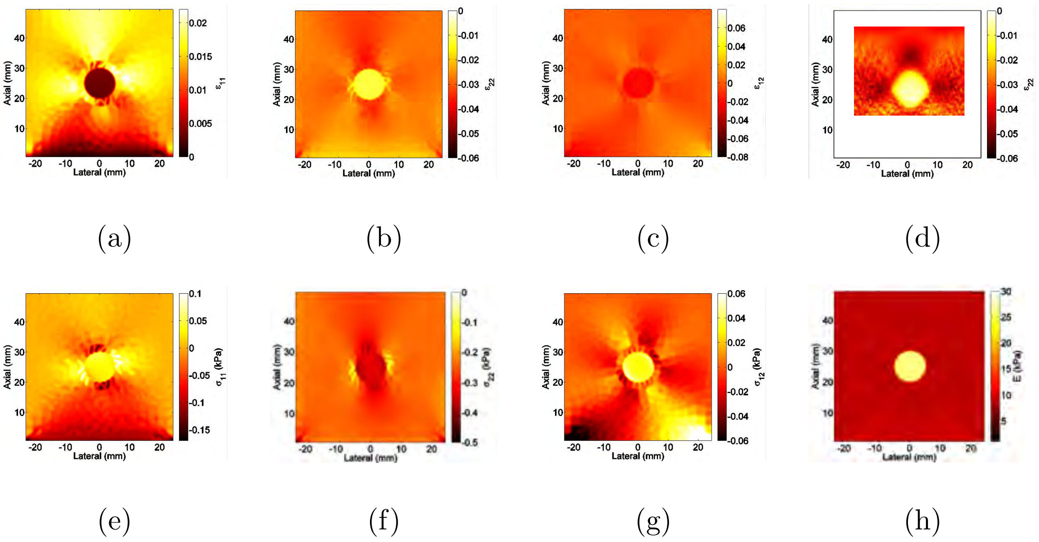
A third test trained four ANNs to characterize a 3-D model of the kidney. Constructing the FE mesh was more difficult and time consuming because the internal geometry was not known *a priori*. To create the mesh, a series of B-mode images were acquired in 1 mm increments along the elevational axis. Each B-mode image was manually segmented. From this set of segmented images, a full 3-d model was created, shown in Fig. 3.8d. Due to the more complex geometry, 4-node tetrahedral elements were required for generating the FE mesh instead of 8-node hexahedral elements used for the 3-D gelatin phantom model. AutoP training parameters were the same as those defined for the 3-D gelatin phantom.

ANNs were assigned to the regions in the same manner shown in Fig. 3.8b, albeit the larger red region visible in Fig. 3.8d was also characterized by ANN 2. Displacements from a single US scan plane were used in FEA$_\varepsilon$, distributed in the plane in a similar manner shown in Fig. 3.8a.

3.3 Results

3.3.1 2-D Phantoms with Known Geometry

Fig. 3.9 displays the reconstructed stress- and strain-vector maps for Phantom 1 case (d) computed in FEA$^{10}_{NN}$. Images labeled axial, lateral, or shear strains (top row) and stresses (bottom row) are relative to the orientation of the ultrasound beam along the vertical x_2 axis. Again, the ANN material models were trained with AutoP using only surface force and displacement measurements and three internal displacement measurements within the inclusion.



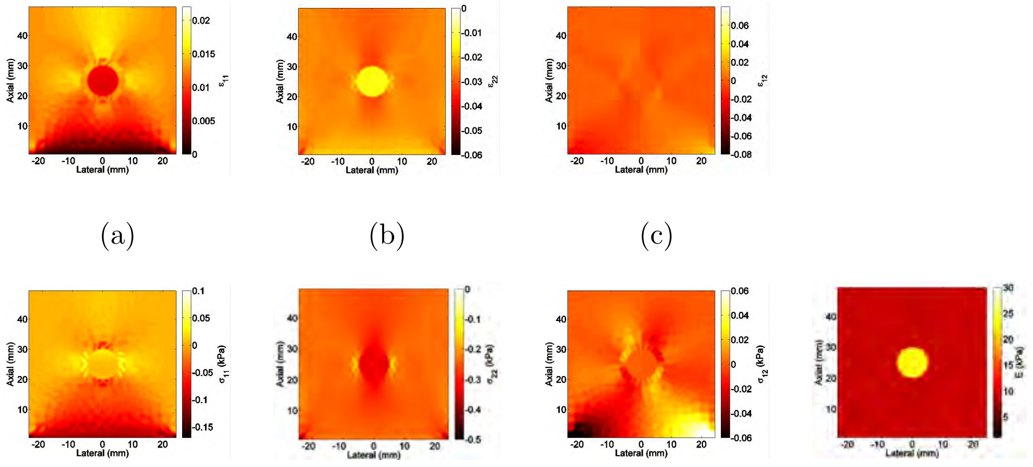
(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 3.9: Stress and strain fields computed in FEA$^{10}_{NN}$ for Phantom 1, case (d). Trained ANNs were used as the element material models to solve a forward FEA problem to produce estimates of the full stress and strain vectors. The results displayed were interpolated to a uniform grid. (a) Lateral strain. (b) Axial strain. (c) Shear strain. (d) Axial strain computed from speckle-tracking applied to the acquired RF frames. (e) Lateral stress. (f) Axial stress. (g) Shear stress. (g) Young's modulus distribution $E(x_1, x_2)$ computed using Equation 3.7.

Fig. 3.10 stress and strain images corresponding one to one with those in Fig. 3.9. The material properties used to compute these results were obtained from indentation measurements, so we refer to them as the FEA$_{ind}$ results. They were generated using conventional FEA forward modeling under the same loading and boundary conditions as FEA$^{10}_{NN}$ for Phantom 1. The background and inclusion elements assume a linear-elastic material model with Poisson's ratio $\nu = 0.5$ and Young's modulus values $E = 9.16$ kPa and 22.9 kPa, respectively.

A direct comparison between strain estimates from conventional quasi-



(a) (b) (c)

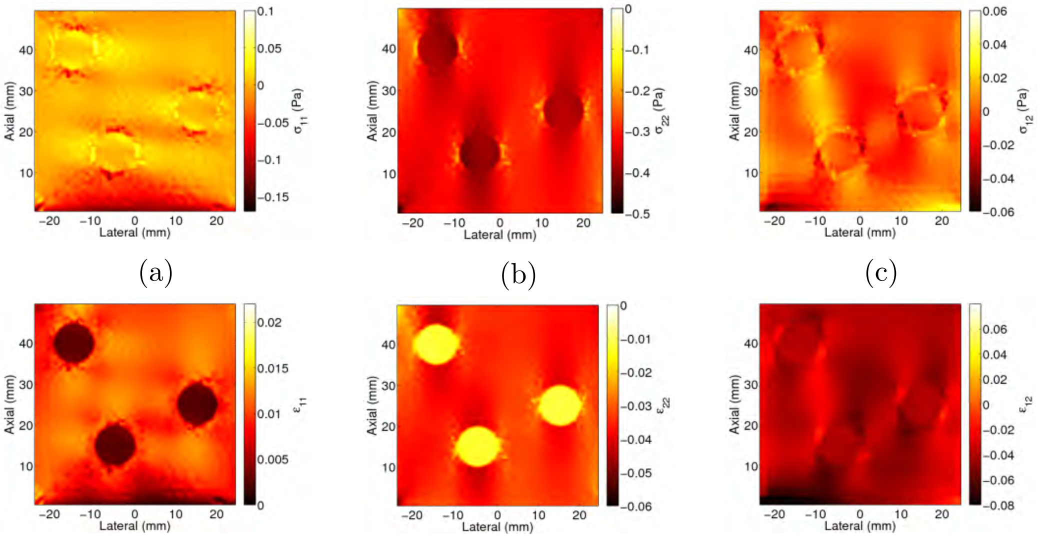
(d) (e) (f) (g)

Figure 3.10: FEA$_{ind}$ estimates of the stress and strain fields for Phantom 1 under plane-stress (CPS4 elements in ABAQUS) using Young's modulus values estimated from macro-indentation tests. A linear-elastic material model replaced each ANN for the mesh in Fig. 3.9 to solve FEA$_{ind}$ and produce these baseline estimates. These are the forward FEA estimates that we use to compare with the AutoP trained models shown in Fig. 3.9. (a) Lateral strain. (b) Axial strain. (c) Shear strain. (d) Lateral stress. (e) Axial stress. (f) Shear stress. (g) Young's modulus image $E(x_1, x_2)$ computed using Equation 3.7.

static elasticity imaging modalities can be made with Figs. 3.9b and 3.9d. The axial strain image shown in Fig. 3.9d was calculated using a least-squares strain estimator on the displacements computed from the speckle-tracking algorithm [41]. Conventional methods are limited to only the imaged region and are prone to artifacts, like the darker region around the inclusion.

Results shown in Figs. 3.9 and 3.10 are the outputs from FE analyses. The difference is that in Fig. 3.9 the element material properties were modeled by ANNs trained with AutoP using force-displacement data acquired on Phantom 1, whereas in Fig. 3.10 an ideal linear-elastic model was chosen using Young's modulus values estimated from indentation testing of Phantom 1 materials. Imposing a nonslip condition at the base of the phantom and a free-slip condition at the top surface, we were able to add stress and strain features in the image plane that should not and did not appear in the modulus image given a uniform background material. For example, the large variations in lateral strain (Fig. 3.9a) and stress (Fig. 3.9e) do not appear in the Young's modulus image. Note that the experimental conditions for the data in Fig. 3.9 are not exactly the same as the ideal conditions assumed in the modeled results of Fig. 3.10. Therefore, we expect the corresponding pairs of stress-strain maps to be similar but not an exact match.

Figs. 3.11 and 3.12 provide the same comparison of stresses and strains, but for those computed by ANNs trained with data from Phantom 3, case (a) and the corresponding ideal linear-elastic model. The Young's modulus reconstructed by the stress-strain maps in Fig. 3.11 is displayed in Fig. 3.13b. Given that only axial forces and displacements were provided in FEA$_\sigma$ and FEA$_\varepsilon$, it is not surprising that the axial stresses and strains estimated by the ANNs closely resemble those computed in the forward FEA using an ideal linear-elastic model. Estimates of the lateral and shear components are not as accurate, the potential reason for this mismatch is discussed later.

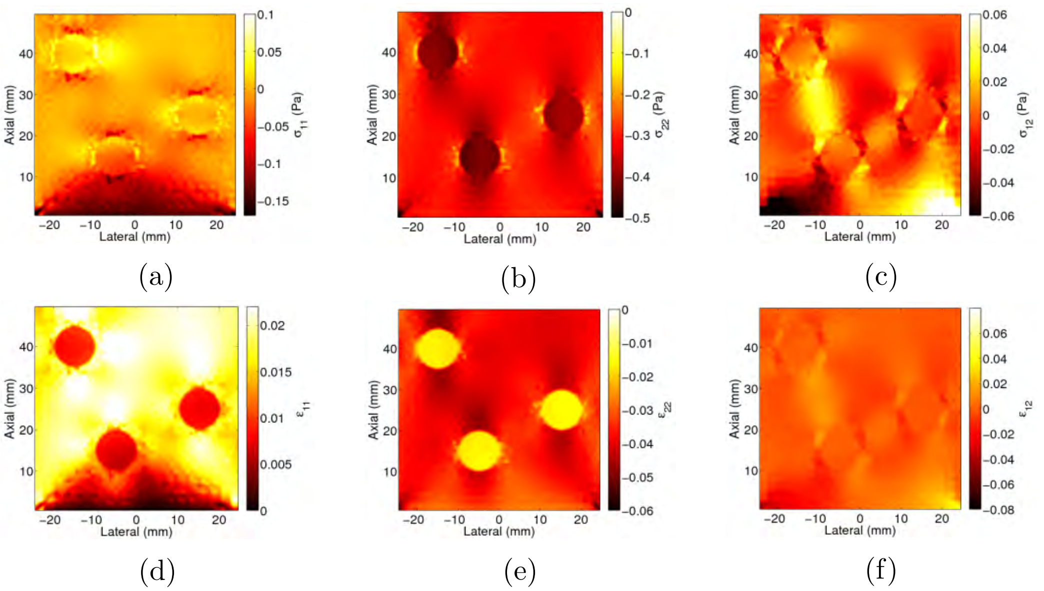


(a) (b) (c)

(d) (e) (f)

Figure 3.11: Images of the stress and strain fields computed in a forward FEA using ANNs trained with measurement data from Phantom 3. (a) Lateral stress (b) Axial stress (c) shear stress (d) Lateral strain (e) Axial strain (f) shear strain

An essential test for trained ANN material models is their ability to accurately estimate stresses and strains in an object in response to an applied load. Comparing Figs. 3.9 and 3.10 we see modest differences between measured and modeled normal stresses σ_{11}, σ_{22} and normal strains $\varepsilon_{11}, \varepsilon_{22}$. Close agreement suggests that measuring axial forces and displacements at the surface and at a few points within the inclusions enabled ANNs trained with AutoP to learn to reliably predict normal stress and strains throughout a 5×5 cm^2 area of a linear-elastic material with a spatial resolution on par with conventional FE modeling. *To the best of our knowledge, this is the first time that all relevant stress fields have been measured with this degree of accuracy and resolution without assuming the underlying mechanical properties,*



(a) (b) (c)

(d) (e) (f)

Figure 3.12: Images of the stress and strain fields computed during a forward FEA with a linear-elastic material model. Young's modulus values were the same as those estimated via macro-indentation measurements on gelatin material comprising Phantom 3. (a) Lateral stress (b) Axial stress (c) shear stress (d) Lateral strain (e) Axial strain (f) shear strain

even for this relatively simple object.

Table 3.2: Young's modulus values and displacement errors for the four phantoms studied. Estimated values were calculated from modulus images. Indentation measurements were made on samples for comparison. Cases refer to training scenarios described in Table 3.1. For Phantoms 1-3, rotation angles refer to the orientation of the phantoms illustrated in Fig. 3.6. Rotation angle for Phantom 4 refers to the rotation of the US probe about the x_2 axis away from the (x_1, x_2) plane. Displacement errors are the mean of c_μ (Eq. 3.6) computed from FEA_{NN}^{10} over all ten load increments for displacements at all nodes in the imaged region.

Young's Modulus Measurements of Phantom Components					Displacement
{Phan. #} (Case) [Rot. Angle]	ANN Estimated (kPa)		Indentation Meas. (kPa)		Errors
	Background	Inclusion	Back.	Inc.	c_μ/c_{max}
2-D Models, Known Geometry					
{1} (a) [0 deg]	8.01 ± .5599	19.35 ± .2022	9.16	22.9	0.17 ± 0.13 / 0.58
{1} (b) [0 deg]	11.01 ± .1189	27.71 ± .5141			0.12 ± 0.10 / 0.48
{1} (c) [0 deg]	10.18 ± .3408	22.15 ± .3711			0.16 ± 0.11 / 0.55
{1} (d) [0 deg]	9.06 ± .1433	22.39 ± .3883			0.19 ± 0.14 / 0.65
{2} (a) [0 deg]	8.57 ± .5596	28.84 ± .1770	8.95	26.87	0.17 ± 0.10 / 0.42
{2} (a) [270 deg]	9.04 ± .0561	26.19 ± .1315			0.12 ± 0.11 / 0.45
{3} (a) [0 deg]	7.99 ± .2755	26.65 ± .2105	8.00	24.58	0.19 ± 0.07 / 0.35
{3} (a) [90 deg]	9.62 ± .1132	25.58 ± .3166			0.11 ± 0.12 / 0.68
2-D Models, Estimated Geometry					
{1} (e) [0 deg]	9.01 ± 0.10	26.65 ± 0.21	9.16	22.9	0.17 ± 0.13 / 0.70
{1} (f) [0 deg]	9.18 ± 0.10	22.42 ± 2.28			0.16 ± 0.11 / 0.63
{3} (e) [0 deg]	10.98 ± 0.47	24.24 ± 0.38	8.00	24.58	0.22 ± 0.17 / 0.81
3-D Models, Known Geometry					
{4} (a) [0 deg]	11.02 ± 0.01	25.22 ± 0.05	11.37	23.25	0.10 ± 0.10 / 0.51
{4} (a) [90 deg]	11.02 ± 0.01	25.23 ± 0.06			0.14 ± 0.12 / 0.46

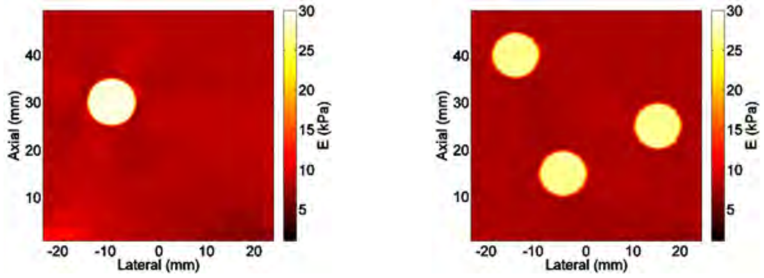
To compute the Young's modulus for these 2-D models, we used the incompressible result $\nu = 0.5$ and the equation

$$E(x_1, x_2) = \frac{(1-\nu^2)\sigma_{22}(x_1, x_2)}{\nu\varepsilon_{11}(x_1, x_2) + \varepsilon_{22}(x_1, x_2)} \tag{3.7}$$

at each integration point the FE mesh, providing point-wise estimate of the Young's modulus, $E(x_1, x_2)$. In Eq. (3.7), $\sigma_{22}(x_1, x_2), \varepsilon_{11}(x_1, x_2)$, and $\varepsilon_{22}(x_1, x_2)$ are the axial stress, lateral strain, and axial strain, respectively, at location (x_1, x_2).This resulted in the modulus images labeled (h) Fig. 3.9 and (g) in Fig. 3.10. Because we supplied axial forces and displacements to AutoP during training and have incompressible materials, the axial and lateral stress and strain estimates are expected to be more accurate than shear estimates. Consequently, the second row of the plane-stress equation, which relates axial stresses to axial and lateral strains, was selected to estimate $E(x_1, x_2)$.

Table 3.2 lists the estimated values (from AutoP) and predicted values (from indentation) of Young's modulus for the four phantoms. The same table contains results where AutoP was trained using different orientations of the phantoms and different data sets for the known geometry cases. Additionally, Young's modulus estimates for 2-D models with estimated internal geometry and 3-D models with known geometry are included. Mean values measured from modulus images for Phantom 1-3 agreed with indentation estimates of the moduli within 20%. Fig. 3.13 are the Young's modulus images for Phantoms 2 and 3 computed using their respective stress and strain estimates produced in FEA_{NN}^{10}. Described later is another method for Young's modulus estimation from trained ANNs that was used for the 3-D models.

In addition to evaluating the accuracy of the Young's modulus estimated from trained ANNs, we also wanted to ensure the deformations predicted by the ANNs in FEA_{NN}^{10} were comparable to the measured displacements. Axial displacements at all nodes in the imaged region were compared those estimated via speckle-tracking. Errors reported in the last column of Table 3.2 are c_μ ± one standard deviation (Eq. 3.6) and c_{max} (Eq. 3.5) over all ten load increments. That is, displacements were estimated in FEA_{NN}^{10} and c_μ and c_{max} were computed using all N nodes in the imaged region. A total of ten displacements were computed for each node and both errors were calculated using the $10 * N$ values. The large standard deviation is attributed to larger displacement errors occurring during the first load increments where displacements are small. For example, in Phantom 1 case (a), c_μ calculated for FEA$^{10}_{NN}$ is 0.49 ± 0.04 using data from the first load increment only but decreases to 0.11 ± 0.05 when data from only the last load increment is considered. Similary, c_{max} decreased from 0.58 in the first load increment to 0.22 at in the final increment.



(a) (b)

Figure 3.13: Young's modulus images computed using Equation 3.7 with stresses and strains estimated in FEA$^{10}_{NN}$ for (a) Phantom 2 and (b) Phantom 3.

Comparing Stress Response of Informational Models to a Constitutive Matrix

We previously discussed and illustrated in Fig. 3.5 how an informational model essentially replaces the constitutive matrix to relate stresses and strains. The linear-elastic, isotropic material properties and geometry of the phantoms allow us to use the plane-stress approximation for the 2-D models. This means the response of the informational models developed for Phantoms 1-3 should describe the same behavior as the constitutive matrix in the plane-stress model.

NN$_{soft}$ for Phantom 1, case (d) was interrogated to obtain the stress response of the background material to various strains. The isolated strain vector $[\varepsilon_{11}\ 0\ 0]$ was used as input to NN$_{soft}$. Values for ε_{11} spanned the ± 0.05 range. After each vector was input the ANN, the full stress response vector $[\sigma_{11}\ \sigma_{22}\ \sigma_{12}]$ was recorded. The same strain vectors were used in the plane-stress equation at the top of Fig. 3.14 to obtain the "ideal" stress response. In the plane-stress equation, the Young's modulus was set to the value measured from macro-indentation (9.16 kPa) and $\nu = 0.5$. This process was repeated for isolated strain vectors $[0\ \varepsilon_{22}\ 0]$ and $[0\ 0\ \varepsilon_{12}]$.

Stress responses computed from NN$_{soft}$ and the plane-stress model were plotted and arranged in Fig. 3.14 to mimic the 2-D constitutive matrix. In the plots, the dotted curves are the behavior described by NN$_{soft}$ and the solid curves are the response predicted by the plane-stress approximation. Perfect mechanical characterization by NN$_{soft}$ would result in the black curves falling on top of the red curves. We find agreement between the informational and constitutive models except for the shear plots of σ_{12} versus ε_{12}, also evident when we compare Figs. 3.9c and 3.9g to Figs. 3.10c and 3.10f. We will return to this point later to discover how the shear response can be better learned by the informational models.

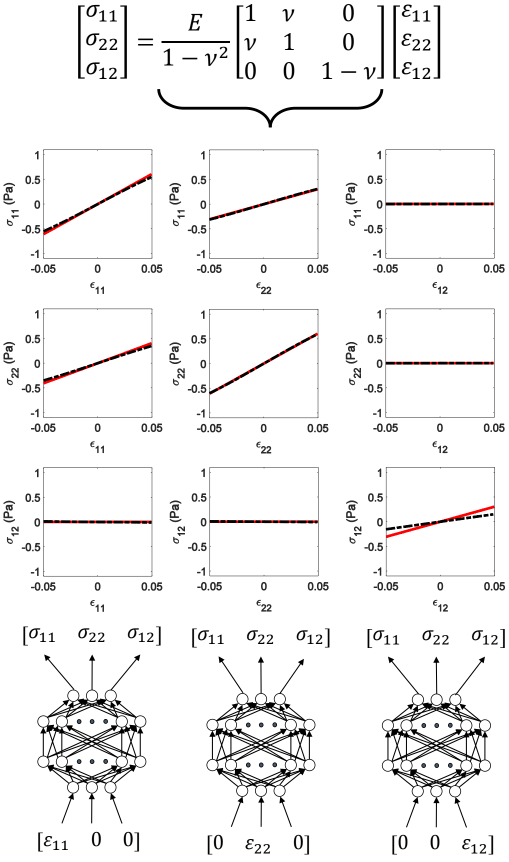
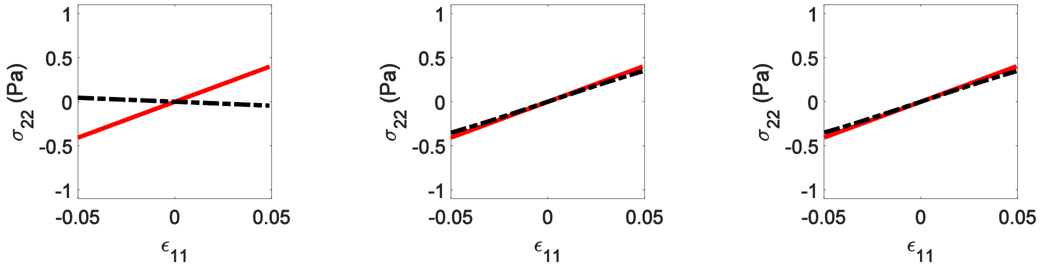


Figure 3.14: Stress response of a trained NN$_{soft}$ material model to the strain probes [ε_{11} 0 0], [0 ε_{22} 0], and [0 0 ε_{12}]. For each vector, the value of the strain component swept the ±0.05 range and the full stress vector was recorded. Slopes of the stress-strain curves should match the corresponding component of the constitutive matrix for a plane-stress model having a Young's modulus of 9.16 kPa (estimated from macro-indentation) and assumed Poisson's ratio of 0.5. Dotted curves correspond to the stress response of the ANN whereas the solid curves were computed by applying the strain vectors to the plane-stress model.

Varying the Amount and Uses of Training Data

Fig. 3.15 illustrates the influence of additional training data that convey material property information on ANN material model accuracy. All three plots show the axial-stress response σ_{22} to isolated lateral strains ε_{11} that are input to different versions of NN$_{soft}$ for Phantom 1. Plot (a) are the response of NN$_{soft}$ trained with data from single set, case (a) in Table 3.1. The large disagreement between the solid- and dotted-curve data in Fig. 3.15a suggests not enough information was supplied during AutoP for NN$_{soft}$ to learn the relationship between axial stress and lateral strain. Adding independent training data from another view (cases (c) and (d) in Table 3.1) greatly improve the ability of NN$_{soft}$ to characterize mechanical responses. Results of these AutoP trained models are shown in Figs. 3.15b and 3.15c.

We saw from the data in Figs. 3.9 and 3.14 that it is difficult to capture the shear stress response to shear strain by applying a uniaxial load and only measuring forces and displacements along the load axis. Although we do not apply a shearing stimulus, the phantom heterogeneity and external boundary conditions produce a small amount of shear, evidenced by the images of Figs. 3.10c and 3.10f.

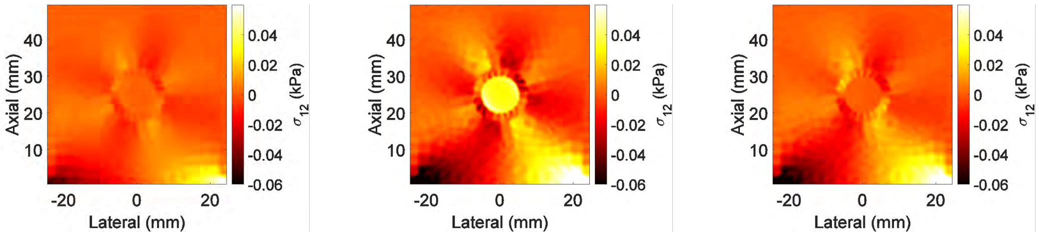


(a) (b) (c)

Figure 3.15: Comparison of the σ_{22} response to an isolated lateral strain (ε_{11}) input. The dotted curves are the recorded response of the ANN modeling the background gelatin material for Phantom 1 (a) case (a); (b) case(c); and (c) case (d). As with Figure 3.14, the solid curves were computed using a plane-stress approximation.

Several versions of AutoP training were performed to evaluate their effects on the ability of NN$_{soft}$ to learn the shear response. These were: 1) training using axial force and displacement information (case (a) in Table 3.1); 2) providing both axial and lateral displacements during training; 3) imposing an assumption that the Jacobian matrix, $[\frac{\partial\sigma}{\partial\varepsilon}]$, is symmetric during FE analysis; and 4) checking displacement errors at points not included in FEA$_{\varepsilon}$ during convergence testing. With the additional points in the displacement error calculation, a total of 65 points sampled uniformly across the imaged region were included in the convergence check for case (4).

While none of these techniques led to NN$_{soft}$ agreeing with the plane-stress model predictions, each of the adjusted training versions (2)-(4) improved the shear response compared to standard version (1). The greatest improvement occurred for version (4) when additional displacement data were included in the test for convergence. A comparison of the full shear stress field is shown in Fig. 3.16. Shown are the estimated shear stress distribution with no training adjustment (version(1)), enforcing stricter convergence testing (version (2)), and the behavior predicted by FEA$_{ind}$ (Fig. 3.16c). The increased accuracy in Fig. 3.16b over Fig. 3.16a suggests there is merit in including displacement errors from points not supplied displacement data during FEA$_{\varepsilon}$ when performing the convergence check.



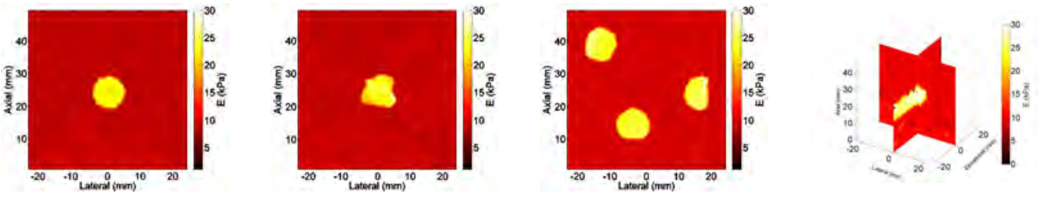
(a) (b) (c)

Figure 3.16: Shear stress fields (σ_{12}) computed in FEA$_{ANN}$ after training versions (1) and (4) for Phantom 1. (a) Training version (1) corresponds to Phantom 1, case (a) in Table 3.1. (b) Training version (4) where displacement convergence was required for additional nodes not provided in FEA$_{\varepsilon}$; (c) Shear stress computed for an ideal linear-elastic plane-stress model using a Poisson's ratio of 0.5 and Young's moduli estimated from macro-indentation.

3.3.2 2-D Phantoms with Coarsely Estimated Geometry

Images of the Young's modulus distribution for cases (e) and (f) of Phantom 1 and case (e) of Phantom 3 are shown in Figs. 3.17a-3.17c, respectively. These images were computed in the same manner described before: the stress and strain fields computed in FEA$^{10}_{NN}$ were used in Eq.(3.7) to calculate the Young's modulus. Estimates of the Young's modulus of the background and inclusion materials for these three cases are included in Table 3.2.

Errors are present in the Young's modulus distributions because of geometry errors resulting from segmentation; however, the errors did not seem to propagate through training. That is, the informational models were still able to accurately characterize the gelatin materials even though geometry errors were present, evidenced by the close match in Young's moduli computed from the ANN material models and compared to those estimated from macro-indentation.



(a) (b) (c) (d)

Figure 3.17: Young's modulus images computed after training on (a)-(c) 2-D models where segmentation was used to identify internal boundaries and (d) a 3-D model. (a) Good segmentation of the internal structure of Phantom 1. (b) Poor segmentation of the internal structure of Phantom 1. (c) Segmented version of Phantom 3. (d) Two planes of the Young's modulus distribution calculated from FEA$^{10}_{NN}$of Phantom 4 case (a).

3.3.3 3-D Phantoms

Returning to the case where the interior geometry is known, the Young's modulus distribution in Phantom 4 with 0° probe rotation is shown in Fig. 3.17d. The modulus was computed throughout the entire 50 x 50 x 50 mm^3 region; however, only two planes are shown for display purposes. All six components of stress and strain distributions were computed in FEA$^{10}_{NN}$and used in the following inversion based on the second row of the constitutive equation for a 3-D Hookean material:

$$E(x_1, x_2, x_3) = \frac{\sigma_{22}(1+\nu)(1-2\nu)}{\nu(\varepsilon_{11} - \varepsilon_{22} + \varepsilon_{33}) + \varepsilon_{22}} \tag{3.8}$$

where σ_{22} is the axial stress, $\nu = 0.495$, and $\varepsilon_{11}, \varepsilon_{22}$, and ε_{33} are the lateral, axial and elevational strains, respectively. For brevity in Eq.(3.8) the location (x_1, x_2, x_3) for the stress and strain components is implied.

Young's modulus values for Phantom 4 are reported in Table 3.2. Here, the reported Young's moduli were not computed as the mean of the regions in Fig. 3.17d. Instead, the ANN material models were probed directly to demonstrate alternative methods of estimating the mechanical parameters. Random values were chosen for all six strain components in the ±0.01 range and used as input to the ANN material models. Strain values were randomized in order to avoid probing the ANNs with data used during training and the ±0.01 range ensured all randomly generated strain vectors were within the bounds of the training data. The corresponding stress was computed and Eq.(3.8) solved to estimate the Young's modulus. A total of 300 random strain probes were input to each trained ANN to produce ten different estimates of the Young's modulus. The mean of these 300 values was taken as the estimate of the Young's modulus and included in Table 3.2.

3.3.4 Kidney Phantoms

Young's modulus images for the three tests corresponding to the kidney phantom are displayed in Fig. 3.18. Modulus estimates from these ANNs are compiled in Table 3.3. There are a couple of particularly interesting observations to note of these images. First, for the 2-D tests, allowing each region to be controlled by a unique ANN reveals what appears to be anisotropy in the kidney. Medullary regions whose fibers are oriented more orthogonal to the loading direction (regions 2 and 5 in Fig. 3.8c) appear stiffer than those whose fiber are more parallel to the applied load (regions 3 and 4). Comparison to Young's modulus values estimated via shear wave methods [124] shows good agreement.

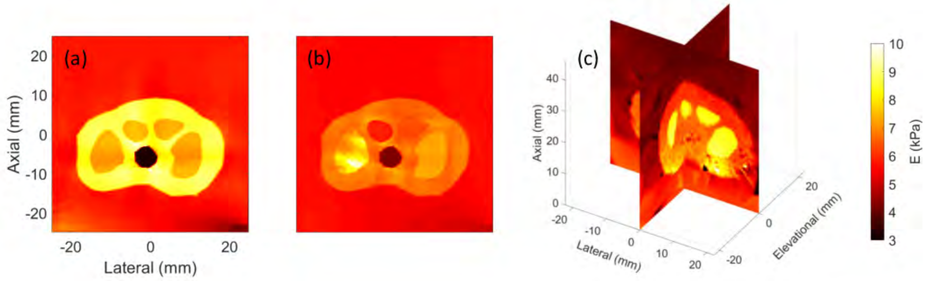


Figure 3.18: 2-D Young's modulus reconstructions after training (a) four or (b) seven NNCMs with the same force-displacement data. (c) 3-D Young's modulus estimation.

Comparing the modulus values estimated in 2-D and 3-D, there is a contrast inversion between the cortex and medullary regions. The estimated modulus of the background gelatin and cortex is fairly consistent across all three tests. However, in 3-D, the estimates increase significantly for both the medullae (region 2) and region 4. It is not yet clear if this issue is due to modeling errors caused during parameter estimation, limitations of the current architecture of the ANNs, or improper sampling in space and time.

Table 3.3: Young's modulus values estimated from the ANN trained with force-displacement measurements acquired on kidney phantom. Regions are identified in Fig. 3.8

4 ANN Kidney Model		7 ANN Kidney Model		(3-D) 4 ANN Kidney Model	
Region	Young's Modulus (kPa)	Region	Young's Modulus (kPa)	Region	Young's Modulus (kPa)
1	8.62 ± 0.32	1	6.77 ± 0.19	1	8.73 ± 2.85
2	7.32 ± 0.17	2	7.88 ± 0.74	2	12.20 ± 4.15
		3	5.86 ± 0.07		
		4	6.37 ± 0.06		
		5	7.27 ± 0.05		
3	3.2 ± 0.04	5	5.36 ± 0.22	3	9.61 ± 1.41
4	5.71 ± 0.26	7	5.58 ± 0.11	4	5.51 ± 0.99

3.4 Discussion

We have demonstrated the ability of informational models trained using the Autoprogressive method to learn the mechanical behavior of quasi-statically loaded, heterogeneous, linear-elastic, isotropic materials using a sparse sampling of force and displacement measurements. The results for the 2-D and 3-D phantom experiments show that all relevant stress and strain fields can be estimated with reasonable accuracy at a spatial resolution dictated by the FE mesh. The innovation in this approach is to replace the constitutive matrix normally used to solve the mathematical inverse problem with ANNs trained using the Autoprogressive method. The combination of finite-element methods and ANNs allows us to build models of mechanical properties without the need for simplifying assumptions.

Tests described in the Results section summarize our initial investigations at selecting measurement points for elasticity imaging. The procedure is to select as few points as possible to avoid the negative effects of noise in displacement estimation. We found that supplying displacement information at three points within each heterogeneity during FEA$_\varepsilon$ gave satisfactory results. Appendix C investigates the reason why only a few displacements are preferable when noise is present. These are combined with measurements of force and displacement at the surface where the ultrasound probe contacts the object. Such sparse sampling works well for quasi-static loading, where the applied force has time to propagate throughout the medium. In that case the displacement at each point carries information about all points in the medium.

One ANN was trained for each material with distinct mechanical properties in the object, resulting in two informational models being developed for each phantom: NN$_{stiff}$ that characterized the inclusion(s) and NN$_{soft}$ for the background gelatin material. We found that incremental loading and diversification of the scanning views while measuring displacements at the same object locations improves informational model accuracy. Also, diversifying the locations of displacements estimated in the object for convergence testing allowed the ANN material models to learn the full range of stress and strain information more accurately. Interestingly, the training technique that worked well for Phantoms 1 and 2 having a single inclusion and meshed assuming the interior geometry was known, worked equally well on Phantom 3 with three inclusions, the 2-D models where geometry was estimated by manual segmentation, and the 3-D model of Phantom 4, albeit with minor adjustments to ANN sizes. Furthermore, errors in estimating the inclusion boundaries did not inhibit the ability of the informational models to learn the correct mechanical properties. Therefore it appears that greater object heterogeneity and geometry errors do not significantly increase the training burden. We will learn more about these types of issues in future studies involving objects with complex spatial heterogeneities and materials behaving in non-linear, path- and time-dependent manners.

One particularly exciting aspect of our technique is the ability to update existing informational models when new measurements are made. Coupling the ability of ANNs to learn new information with measurements taken at different times under different loading geometries, we can update an existing ANN with the new data instead of building a new model. Case (c) of Phantom 1 was an example of “retraining” previously developed informational models after new data was gathered. We found that providing new information to an existing informational model increased the accuracy of stress, strain, and Young’s modulus estimates. This capability allows for resampling of a material after the initial measurements if it is found that the informational models are not converging or to initiate training of models for new materials under investigation. However, care must be taken to ensure that the ANNs do not “forget” previously learned behaviors. Case (c) for Phantom 1 supplied a second set of force-displacement data for ANN retraining that was very similar to the first set of data. As such, there was not much risk of NN$_{soft}$ and NN$_{stiff}$ discarding the mechanical properties already learned. Potential problems associated with ANN retraining are most likely to occur when the new information is acquired under different loading situations or when the mechanical behaviors become complex. Fortunately, such issues were circumvented by interspersing previously learned data with the new data so that the ANN retain old information while learning new properties.

3.4.1 Effects of Limited Information

Stress and strain vectors produced during uniaxial compression of cubic phantoms mostly span the axial and lateral spaces. Limited shearing occurs, resulting in less information available for training the ANNs on the shear behavior. However, we showed it was possible for the ANNs to better learn the shear response by making adjustments to the training process. The most successful adjustment was including more displacements in the error check for convergence. Forcing the ANNs to more accurately predict displacements not given during training creates a more challenging convergence criteria, which in turns increases the number of training iterations performed. It is true that this same scenario can be created by adjusting C^n_{max} and C^n_{μ}, but we then run the risk of overtraining the ANNs. Instead of forcing the ANNs to very accurately predict displacements at a few points, we relax the constraints and force the ANNs to approximate the displacements at many points.

3.4.2 Training Data Selection

There is much to be learned about what constitutes sufficient information to successfully build an informational model. The drawback of providing too much information when the measurement data are noisy was discussed previously. More interesting is that changing the nodes on which the model trains greatly affects the outcome of the informational modeling process. Three different cases illustrate this point, and in each case the only force measurements are those applied at the ultrasound probe-phantom interface.

Consider situations where displacements are given 1) at the probe only,

2) along a vertical line through the inclusion, and 3) along a horizontal line through the inclusion of Phantom 1. In all three cases, Young's modulus estimates of the background material were near the expected values; however, only in case (2) did we get accurate estimates for the Young's modulus of the inclusion. The result of case (1) is not completely unexpected since the background comprises the bulk of the phantom and will have the greatest effect on the force-displacement relationship measured by the US probe. Results of (2) and (3) start to reveal what type of information needs to be supplied for training.

It is evident that displacement information must be given for each region in the material with distinct mechanical properties. For the models in this study, probe displacements would be enough to characterize the axial stress-strain relationship of the gelatin comprising the background of the phantoms. Case (1) indicates that displacements must be provided for the inclusion. The same number of nodes were given displacement data in cases (2) and (3), the only difference lies in the orientation of the nodes compared to the loading direction. In (2), the given information lies along a line parallel to the load axis whereas in (3) the nodes given displacement data spanned a line perpendicular to the loading direction. Because only axial displacement data was given, the movement of the nodes in (3) reveals very little information about how the points move relative to each other after loading - indeed, if the given displacements contained no noise, all of the nodes would have very similar (or identical) axial motion. However, in (2), the nodes compress relative to each other, producing more information about the material properties of the inclusion.

We empirically find the best sampling strategy for these linear-elastic, isotropic materials is to measure displacements at nodes that vary significantly along the applied force gradient. For example, axial compression means the force gradient lies mostly along the x_2 axis. Therefore, we select nodes along the x_2 axis in which displacements vary to the greatest extent. It is important to note this sampling strategy may be insufficient for materials exhibiting more complex behavior, like a rabbit kidney. Future research of this method will include investigations into the minimum requirements of force-displacement measurements to build informational models of materials that have nonlinear, path-dependent, time-dependent, and/or anisotropic mechanical properties.

3.4.3 Choice of Convergence Criteria

The choice of C_{μ}^{n} and C_{max}^{n} largely affect the number of iterations performed in each training step of AutoP. Enforcing very strict convergence criteria increases the risk of overtraining the ANNs, especially if the displacement data involved in the displacement error calculation contains high levels of noise. Conversely, relaxing the convergence criteria too much may decrease the accuracy of the ANN material models. At this point in time, the choice of convergence criteria is largely heuristic. But, the same criteria were used for all phantoms when the geometry was known, which may imply the choice for convergence criteria can be based on the complexity of the object. Case (e) of Phantom 3 - where the internal geometry was estimated via manual segmentation - was the only case where the convergence criteria changed. At the start, C_{μ}^{n} and C_{max}^{n} were relaxed, allowing for larger displacement errors. After training the ANNs over a few load increments, the both values were decreased, making convergence more difficult to achieve.

Gradually restricting the convergence criteria follows the idea that the ANNs more accurately model the mechanical behavior as training progresses. More iterations of AutoP means the ANNs are provided with an increasing amount of information. And, because the stress-strain estimates in FEA$_{\sigma}$ and FEA$_{\varepsilon}$ become more accurate, it is natural to enforce stricter requirements for convergence as the training data becomes higher quality. In the case of Phantom 3, case (e), the estimated internal geometry produced more complex stress and strain distributions that required several iterations of AutoP for the ANNs to learn. But, as the ANNs better learned the mechanical properties, stronger requirements for convergence could be put in place.

Moving forward, the choice of C_{μ}^{n} and C_{max}^{n} likely depend on the geometric and material complexity of the object and hence will be determined empirically. Using AutoP on models of many different geometries and material property distributions may allow us to identify a trend.

3.4.4 Computational Load Considerations

With AutoP, most of the computation time is spent performing FEA$_{\sigma}$ and FEA$_{\varepsilon}$. For two passes of ten load steps, a minimum of 60 FEAs are performed, up to 220 in the worst-case where the maximum number of iterations is reached during training for each load step. This assumes a minimum of one and a maximum of five training iterations per load step. Our current implementation of AutoP in MATLAB 2013b required 51 ± 7 minutes to perform one pass of ten load steps in AutoP to create informational models of Phantoms 1-3 cases (a)-(d). However, after a few optimizations, AutoP was able to train 2-D models in ≈ 30 minutes. Conversely, developing 3-D models required ≈ 65 minutes. Note that these analyses were ran on a PC containing a quad-core processor operating at 3.4 GHz. To give an example of the time required to complete the FEAs, FEA$_{ind}$ for Phantoms 1-3 was solved in ≈ 4 seconds whereas the solution time for Phantom 4 increased to ≈ 30 seconds.

There are many ways to reduce the run time of AutoP, including the use of dedicated hardware and software and through optimized sampling and training strategies. Of particular interest is offloading computation to a GPU, which could provide up to 100x increase in speed. Other groups have been successful in developing real-time FEA solvers for biomedical applications ([125]) and structural analysis ([126]) with the use of a GPU. It is our belief that it will be possible to develop informational models in real-time during imaging. Methods to decrease the training time with AutoP will be investigated after we make significant progress in the basic research of this method.

3.5 Conclusions

Informational models based on AutoP have an ability to learn quasi-static mechanical behavior from measurements of surface forces and sparse surface and internal displacement information. This approach has distinct advantages when the mechanical properties of the medium are not well characterized, which is the situation in medical elasticity imaging. We trade the conventional parametric inverse problem approach involving numerous assumptions for a nonparametric machine-learning technique. This technique learns to model stress and strain fields from which imaging parameters are found.

The ability of informational models to accurately predict mechanical behavior is limited by the type and amount of force-displacement information provided. Model accuracy can be improved by adding data from different spatial views and carefully distributing measurement data for FEA training and convergence testing. An advantage that AutoP offers in studying similar objects is its ability to accumulate knowledge, which minimizes training time in ways we are just beginning to understand. We believe this novel computational approach has much to offer for imaging mechanical properties of biological media.

Chapter 4

Cartesian Neural Network Constitutive Models

4.1 Introduction

Elasticity imaging methods reconstruct a map of mechanical properties by observing tissue motion in response to a weak mechanical stimulus. For quasi-static ultrasonic elastography, measurement data are forces and displacements as an ultrasound (US) probe is slowly pressed into the tissue surface. Recorded displacements may include both probe motion and internal tissue deformation, the latter being estimated via speckle-tracking algorithms operating on pre- and post-deformation echo data (e.g.,[127, 38]). These time-varying force-displacement measurements are governed by the geometric and mechanical properties of the tissue and can provide diagnostic data relevant to cancer detection in the breast [128], liver [129, 130], and prostate [131], identifying atherosclerotic plaques [132], or treatment monitoring during high-intensity focused ultrasound or RF ablation [133, 134].

Estimating mechanical properties from measurement data constitutes the inverse problem in elastography. The goal for quasi-static elastography can be stated simply: given a set of force-displacement estimates and overall object shape, reconstruct the spatial distribution of mechanical properties. Current solutions take a model-based (or knowledge-driven) approach, where the mechanical properties of tissues are defined by parameters of a constitutive model relating stresses and strains. This problem is ill-posed in part due to the presence of measurement noise and limited force-displacement sampling from which stress-strain behavior is determined. Some strain information can be computed as spatial derivatives of the displacements, but stresses cannot be calculated from force data without knowing the object geometry, boundary conditions, and material properties.

Simplifying assumptions are adopted in model-based techniques to help overcome the ill-posed nature of the inverse problem. Most often the tissue is assumed to be linear-elastic, isotropic, (nearly) incompressible, and under small strain, limiting the parameter space to only the Young's modulus (or shear modulus). However, biological tissues are bi-phasic media exhibiting nonlinear and viscoelastic properties not summarized by a single linear parameter. Recent work in imaging nonlinear [96, 135] and viscoelastic [136, 54] material properties are promising, but still rely on initial constitutive model assumptions. And while a chosen model may be appropriate for certain tissue types, it may be incorrect for other tissues in the same field of view. The problem is that the constitutive model sets the parameters to be estimated and if an inappropriate model is chosen, parametric errors are made, corrupting the final elastogram. Also, the most diagnostic model parameters for each disease state have yet to be determined.

Our solution was to implement a *data-driven* approach using artificial neural networks (NNs) in place of a pre-defined constitutive model [108]. These neural network constitutive models (NNCMs) learn stress-strain behavior from force-displacement measurements without any initial assumptions of mechanical behavior. The benefit is that after training, NNCMs can be used to compute all relevant stresses and strains, from which material parameters from any constitutive model can be estimated. A block diagram of our method is shown in Fig. 4.1. After acquiring force-displacement measurements, we create a finite element (FE) mesh that conforms to both the internal and external geometries of the object from prior knowledge or manual segmentation of the US images. The mesh and measurement data are used in the Autoprogressive Method (AutoP) to train NNCMs for each region exhibiting unique material properties.

We demonstrated that our method could train NNCMs to accurately characterize the linear-elastic properties of gelatin phantoms in 2-D and 3-D, and obtain Young's modulus estimates of rabbit kidneys similar to those reported in the literature [137]. However, prior knowledge of object geometry will not be available in a clinical setting, nor can it be assumed that tissue boundaries observed in a US image correspond to actual material property boundaries, precluding the use of segmentation for defining internal structures. Furthermore, the NNCMs could only capture discrete material property distributions. A limitation of the NNCMs is their inability to account for geometric information; segmentation defined the spatial distribution of

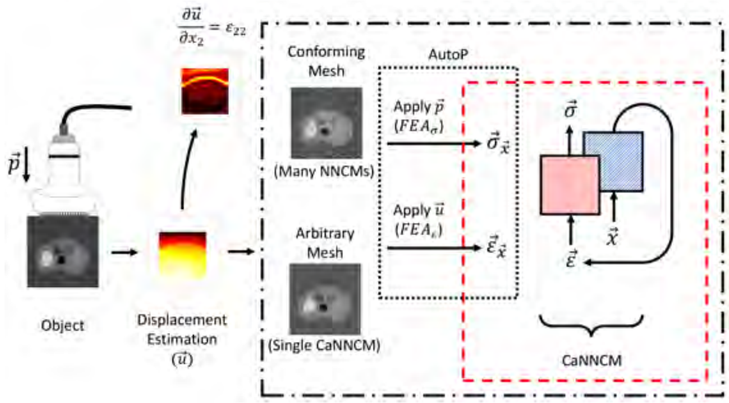

$\frac{\partial \vec{u}}{\partial x_2} = \varepsilon_{22}$
$\vec{p}$
Object
Displacement Estimation ($\vec{u}$)
Conforming Mesh
(Many NNCMs)
Arbitrary Mesh
(Single CaNNCM)
AutoP
Apply $\vec{p}$ (FEA_σ)
Apply $\vec{u}$ (FEA_ε)
$\vec{\sigma}_{\vec{x}}$
$\vec{\varepsilon}_{\vec{x}}$
$\vec{\sigma}$
$\vec{x}$
$\vec{\varepsilon}$
CaNNCM

Figure 4.1: An US probe is slowly pressed into an object with force $\boldsymbol{p}$ while pre- and post-deformation RF frames are acquired. Speckle-tracking methods applied to the RF frames estimate displacements $\boldsymbol{u}$ within the image. The axial strain computed along the direction of US beam propagation is often used to reconstruct a stiffness image, although current model-based methods are far more sophisticated and can provide more quantitative Young's modulus estimates. Our machine learning method (black dash-dot box) originally relied on a FE mesh conforming to the object geometry and a different NNCM for each region exhibiting unique material properties. The new CaNNCMs shown learn both material property and geometric information, allowing a single network structure to be used with an arbitrary mesh. AutoP is the method whereby force-displacement measurements are transformed into the stress-strain training data for NNCMs/CaNNCMs (black dotted box). In this report, we focus only on training CaNNCMs (red dashed box) and assume the spatially varying stresses $\boldsymbol{\sigma}(\boldsymbol{x})$ and strains $\boldsymbol{\varepsilon}(\boldsymbol{x})$ are known.

material properties at the time of FE mesh creation. We can greatly enhance the utility of NNCMs and AutoP for elastography by altering the network architecture to incorporate learning of spatial information.

Cartesian neural network constitutive models (CaNNCMs) address this issue. CaNNCMs accept Cartesian coordinate information as additional inputs to simultaneously learn material property and geometric information independent of the internal structure represented by the FE mesh. In this paper, we introduce CaNNCMs and describe how this novel architecture learns spatial distributions of material properties from stresses and strains. We leave the details of implementing CaNNCMs in AutoP to Chapter 5. Here we use stress-strain data acquired from simulated phantoms to demonstrate the ability of CaNNCMs to model material property distributions on an arbitrary FE mesh without affecting the learned mechanical behavior. Because CaNNCMs capture spatial information, we will show how an elastogram can be *reconstructed* directly from a trained model. Finally, we will also demonstrate that, in the presence of noise or changes to the FE mesh, CaNNCMs are able to accurately model linear-elastic mechanical behavior.

4.2 Methods

4.2.1 Overview of the Autoprogressive Method

AutoP has been primarily developed and used in civil and geotechnical applications to model the mechanical properties of various materials and structures [99, 100, 101, 102, 103, 104, 106, 107]. The AutoP approach to constitutive modeling is to apply force-displacement measurements to two finite element analyses associated with a mesh of the imaged object. Forces and displacements applied in separate finite element analyses (FEAs) are connected through NNCMs. Training is to develop NNCMs that consistently relate measured forces and displacements. Meaning, application of measured forces in a FEA results in the measured displacements and vice-versa.

All examples of AutoP prior to our initial report of its use in elasticity imaging have relied solely on surface measurements. The addition of ultrasonic imaging provides internal displacement estimates which are imposed in AutoP along with surface displacements. We exploited the extra displacement data to develop NNCMs that learned the linear-elastic behavior of gelatin phantoms; however, because the FE mesh matched both the internal and external phantom geometry, only a sparse sampling of the internal data was necessary.

Internal displacements under a quasi-static load provide an enormous amount of information regarding internal structure. Because the force stimulus has time to propagate throughout the entire object before force-displacement measurements are acquired, displacements at one location are affected by deformation at all other points in the object. Previously, the NNCMs ignored the spatial aspects of measurements when relating stress to strain. We are now changing the NNCM architecture to incorporate spatial information to make it possible to relax the geometric constraints on the FE mesh. We introduce Cartesian NNCMs that are capable of learning both material and internal geometric properties for the task of forming elasticity images.

4.2.2 Cartesian Neural Network Constitutive Models

The NN structure should be designed to represent the input/output relationship and the intermediate computations that must be performed. For

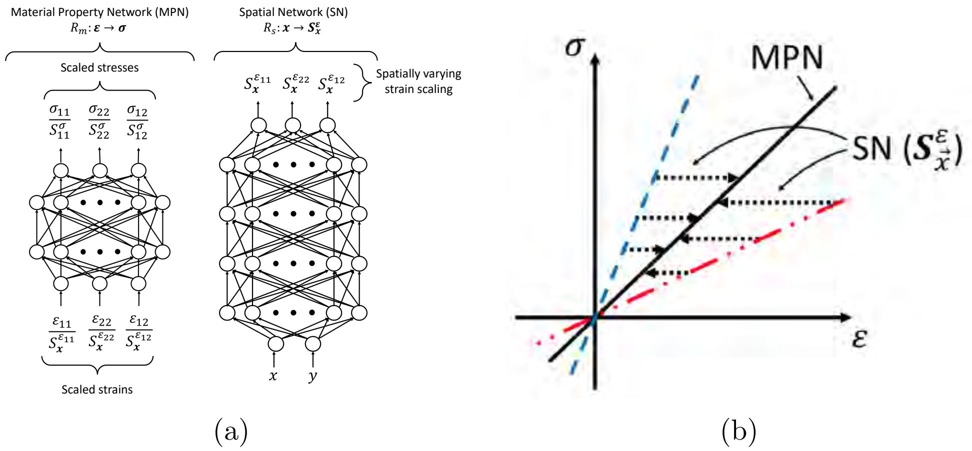

Material Property Network (MPN)
$R_m: \boldsymbol{\varepsilon} \rightarrow \boldsymbol{\sigma}$
Scaled stresses
$\frac{\sigma_{11}}{S_{11}^{\sigma}}$ $\frac{\sigma_{22}}{S_{22}^{\sigma}}$ $\frac{\sigma_{12}}{S_{12}^{\sigma}}$
$\frac{\varepsilon_{11}}{S_{\boldsymbol{x}}^{\varepsilon_{11}}}$ $\frac{\varepsilon_{22}}{S_{\boldsymbol{x}}^{\varepsilon_{22}}}$ $\frac{\varepsilon_{12}}{S_{\boldsymbol{x}}^{\varepsilon_{12}}}$
Scaled strains
Spatial Network (SN)
$R_s: \boldsymbol{x} \rightarrow \boldsymbol{S}_{x}^{\varepsilon}$
$S_{\boldsymbol{x}}^{\varepsilon_{11}}$ $S_{\boldsymbol{x}}^{\varepsilon_{22}}$ $S_{\boldsymbol{x}}^{\varepsilon_{12}}$
Spatially varying strain scaling
x y
(a)
σ
MPN
SN ($\boldsymbol{S}_{\vec{x}}^{\varepsilon}$)
ε
(b)

Figure 4.2: (a) Architecture of CaNNCMs. The material property network (left) accepts a scaled strain input vector and computes a scaled output stress vector. Previously, the strain scaling vectors were defined for the NNCM and were spatially invariant. Adding a spatial network (right) lets the strain scaling vary with position and allows the pair of networks to learn geometric information. For CaNNCMs, $S_{11}^{\sigma} = S_{22}^{\sigma} = S_{12}^{\sigma} = \boldsymbol{S}^{\sigma}$ and does not change with position. (b) Visualization of interaction between MPN and SN. The MPN learns a reference material property (solid black line). Different spatial locations in the object exhibit different material properties (blue-dashed and red-dashed-dot-dot lines). At each location $\boldsymbol{x}$, the scaling value $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ output by the SN transforms the spatially varying material properties to the reference material.

example, many previous implementations of AutoP constitutive modeling relied on fully-connected networks to represent a linear stress-strain relationship. Other NNCMs characterizing more complex behaviors, including nonlinearity and path-dependence, utilized a nested NN structure[110] to depict the influence of previous stress and strain states on the current mechanical behavior. In our prior work using AutoP for elastography, a simple feed-forward, fully-connected architecture was sufficient (left side of Fig. 4.2a). Here, NNCMs computed a stress vector output from a strain vector input. We accounted for the possible positive and negative input/output values — and the sign symmetry between stresses and strains — by using hyperbolic tangent activation functions. The choice of hyperbolic tangent also ensures that under zero strain, there is zero stress. To avoid saturating the input nodes and keep the NNCMs sensitive to changes in the input strains, a scaling vector $\boldsymbol{S}^{\varepsilon}$ was selected to scale each component of the input strains within the ±0.8 range. Similarly, the output is bounded to ±1.0 whereas stresses can extend well beyond that range. As such, a different scaling vector $\boldsymbol{S}^{\sigma}$ was selected to keep the output within ±0.8. The flow of data through the NNCM was therefore $\boldsymbol{\varepsilon} \rightarrow \boldsymbol{\varepsilon}/\boldsymbol{S}^{\varepsilon} \rightarrow$ NNCM $\rightarrow \boldsymbol{\sigma}/\boldsymbol{S}^{\sigma} \rightarrow \boldsymbol{\sigma}$, where the vector divisions are element-by-element operations. We refer to this type of

NNCM as a material property network (MPN).

Creating a NNCM to learn internal structure required spatial information be present somewhere in the flow of data through the network. Instead of increasing the number of inputs to include spatial coordinates, scaling factors are learned to alter the material properties characterized by the MPN. We consider the stress-strain behavior represented by the MPN with $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon} = \mathbf{1}$ to be the "reference" material property (i.e., $R_m : \boldsymbol{\varepsilon} \rightarrow \boldsymbol{\sigma}$). For the case of linear-elastic materials, the stress-strain diagram is a straight line with a slope that is the Young's modulus as determined by MPN. Stresses are computed by simply multiplying a strain vector by the stiffness matrix $\boldsymbol{C}$: $\boldsymbol{\sigma} = \boldsymbol{C}\boldsymbol{\varepsilon}$. Changing the strain scaling at a point (x, y) alters the slope of the line, effectively changing the Young's modulus at that location. CaNNCMs accomplish this change by introducing a spatial network (SN) that computes $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ based on a coordinate input (right side of Fig. 4.2a). A SN is the function $R_s : \boldsymbol{x} \rightarrow \boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$. We also reduce the stress scaling vector to a single value: $S_{11}^{\sigma} = S_{22}^{\sigma} = S_{12}^{\sigma} = \boldsymbol{S}^{\sigma}$.

Details of how $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ and $\boldsymbol{S}^{\sigma}$ are computed is covered in Sec. 4.2.3 where we show that the spatial scaling values produce a map of relative stiffnesses where larger $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ tend to correspond to softer regions. To understand why, consider the reference linear-elastic relationship learned by the MPN. Under uniaxial loading, a heterogeneous material exhibits large strains and small stresses in soft regions. In contrast, stiff regions produce large stresses for small strains. Therefore, larger $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ values in soft regions decrease the magnitude of the strain vector input to the MPN. Similarly, smaller scaling values in stiff regions result in larger strains after scaling. In broad terms, $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ acts as a function transformation for the relationship learned by the MPN: $\boldsymbol{\sigma} = f(\boldsymbol{\varepsilon}/\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon})$. In the more specific case of linear-elastic materials, the SN acts as a spatially-varying matrix that operates on the strains before multiplying by the stiffness matrix: $\boldsymbol{\sigma} = \boldsymbol{C}\boldsymbol{E}(x, y)\boldsymbol{\varepsilon}$. The interaction between the MPN and SN is illustrated in Fig. 4.2b. Together, the networks learn the mapping $R_m, R_s : \boldsymbol{\varepsilon}(\boldsymbol{x}) \rightarrow \boldsymbol{\sigma}(\boldsymbol{x})$

The SN also has a fully-connected, feed-forward architecture. Unlike the MPN counterpart, spatial networks use a mix of logistic and hyperbolic tangent activation functions. Considering that the output $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ is always positive, the logistic function is a natural choice for the output nodes. Conversely, the input (x, y) can span the positive and negative range, but a vector of zeros at the input does not imply the output should also be zero. We thus use a logistic activation function for the first layer as well. All intermediate layers use a hyperbolic tangent. As with the MPN, care must be taken to bound the spatial network inputs and outputs: input values outside the ± 1.0 can saturate the input nodes (and reduce sensitivity) while outputs not contained within $(0, 1)$ cannot be achieved by a logistic function. We therefore scale the input (x, y) values to within ± 1.0. Preliminary tests showed that setting the coordinate origin to the center of the FE mesh produced the best results. A similar shifting and scaling to the $0.1 - 0.8$ range is performed for the output $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ values before training the spatial network.

Both the material property and scaling networks learn from the same set of stress-strain data. These data would be estimated in AutoP after applying force and displacement measurements in FEAs. Each network extracts different information from the same set of data. Splitting the material property and geometry problems allows each network to learn a simpler input-output relationship. Combining the two networks results in a cooperative CaNNCM structure that captures both mechanical behavior and its geometric variation.

4.2.3 Calculating Spatial Values

Inputs to the SN — Cartesian coordinates $\boldsymbol{x}$ — are defined by the FE mesh and thus known *a priori*. Given a trained MPN, spatially varying stresses $\boldsymbol{\sigma}(\boldsymbol{x})$ strains $\boldsymbol{\varepsilon}(\boldsymbol{x})$ and coordinates, the task of determining the target output of the SN remains.

In preliminary studies, many methods of computing the spatial scaling values were evaluated. These methods all relied on calculating $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ using a predefined function of $\boldsymbol{x}$, $\boldsymbol{\varepsilon}(\boldsymbol{x})$, and/or the stiffness matrix. Each method was successful to some degree, with the caveat that chosen functions influenced the material properties or geometry learned by the CaNNCM. For example, a simple function we tested was computing the spatial scaling as the ratio of stress and strain magnitudes: $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon} = ||\boldsymbol{\sigma}_{\boldsymbol{x}}||/||\boldsymbol{\varepsilon}_{\boldsymbol{x}}||$. This method would likely work for 1-D linear-elastic materials, but 2-D stress-strain distributions are more complex and the magnitude of stress to strain changes, even for the same material, based on applied load and location in the material.

More importantly, a pre-defined function does not directly account for errors between the stress vector output from the MPN and the "target" stress. The MPN and SN are cooperative and therefore work together to minimize this error. Gradient-descent methods are utilized for computing $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ based on the difference between stress estimated by the MPN in response to a strain input and the target stress $\boldsymbol{\sigma}^t$ computed via FEA. Similar to the backpropagation algorithm for updating ANN connection weights, the error at the output of the material property network can be propagated back to the spatial scaling values. For simplicity, consider the stress $\boldsymbol{\sigma}^t(\boldsymbol{x})$ and strain $\boldsymbol{\varepsilon}(\boldsymbol{x})$ computed by FEA at a single location $\boldsymbol{x}$. The current value of $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ and $\boldsymbol{\varepsilon}(\boldsymbol{x})$ can be used to compute $\boldsymbol{\sigma}^{NN}(\boldsymbol{x})$, the value of stress predicted by the MPN: $R_m : \boldsymbol{\varepsilon}(\boldsymbol{x}) \rightarrow \boldsymbol{\sigma}^{NN}(\boldsymbol{x})$. The goal is to minimize the objective function

$$\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon} = \underset{\hat{\boldsymbol{S}}_{\boldsymbol{x}}^{\varepsilon} \in \mathbb{R}}{\operatorname{argmin}} \; \mathrm{f_m}(\boldsymbol{\sigma}^t(\boldsymbol{x}), \boldsymbol{\sigma}^{NN}(\boldsymbol{x})). \tag{4.1}$$

The function $\mathrm{f_m}(\cdot)$ is the L_2 norm:

$$\mathrm{f_m} = E = \frac{1}{2}\sum_{i=1}^{3}(\sigma_i^t(\boldsymbol{x}) - \sigma_i^{NN}(\boldsymbol{x}))^2 \tag{4.2}$$

$$= \frac{1}{2}\sum_{i=1}^{3} e_i^2, \tag{4.3}$$

where (dropping $(\boldsymbol{x})$ for brevity)

$$\sigma_i^{NN} = S_\sigma \sigma_i^{\prime,NN} \tag{4.4}$$

$$\varepsilon_j = S_{\boldsymbol{x}_i}^{\varepsilon_j} \varepsilon_j^{\prime} \tag{4.5}$$

$$e_i = \sigma_i^t - \sigma_i^{NN}. \tag{4.6}$$

We define σ_k^t as the k^{th} component[1] of "target" stress at $\boldsymbol{x}_i$, σ_k^{NN} is the corresponding stress component predicted by the MPN in response to $\boldsymbol{\varepsilon}$, and (i, j, k) have the range $(1, 2, 3)$. Scaled input and outputs of the MPN are denoted as $\varepsilon_j^{\prime}$ and $\sigma_i^{\prime,NN}$, respectively. Calculating the partial derivate of the error with respect to $S_{\boldsymbol{x}_i}^{\varepsilon_k}$ is straightforward:

[1] There are three components in the stress vector for 2-D models, ordered as $[\sigma_{11},\ \sigma_{22},\ \sigma_{12}]$. The same ordering is used for the strain vector.

$$
\begin{aligned}
\frac{\partial E}{\partial S^{\varepsilon_k}_{\boldsymbol{x}}} &= \frac{\partial}{\partial S^{\varepsilon_k}_{\boldsymbol{x}}} \frac{1}{2} \sum_{i=1}^{3} e_i^2 \\
&= \sum_{i=1}^{3} e_i \left[\frac{\partial}{\partial S^{\varepsilon_k}_{\boldsymbol{x}}} (\sigma_i^t - \sigma_i^{NN}) \right].
\end{aligned} \tag{4.7}
$$

The partial derivative is distributed to the stress terms, noting that σ_i^t was computed in a FEA and the partial derivative with respect to $S^{\varepsilon_k}_{\boldsymbol{x}_i}$ is zero. For σ_i^{NN} we invoke the chain rule:

$$
= \sum_{i=1}^{3} e_i \left[- \sum_{j=1}^{3} \frac{\partial \sigma_i^{NN}}{\partial \varepsilon_j} \frac{\partial \varepsilon_j}{\partial S^{\varepsilon_k}_{\boldsymbol{x}}} \right]. \tag{4.8}
$$

The term $\partial \sigma_i^{NN} / \partial \varepsilon_j = D_{ij}$ is the stiffness matrix relating stress to strain and can be calculated via the weights of the material property network[138]. Computing the last factor in the braces:

$$
\begin{aligned}
\frac{\partial \varepsilon_j}{\partial S^{\varepsilon_k}_{\boldsymbol{x}}} &= \frac{\partial}{\partial S^{\varepsilon_k}_{\boldsymbol{x}}} (S^{\varepsilon_j}_{\boldsymbol{x}} \varepsilon'_j) \\
&= \varepsilon'_j \delta_{kj},
\end{aligned} \tag{4.9}
$$

where δ_{kj} is the Kronecker delta function. Finally, we arrive at the final expression for the error gradient:

$$
\frac{\partial E}{\partial S^{\varepsilon_k}_{\boldsymbol{x}}} = - \sum_{i=1}^{3} e_i \sum_{j=1}^{3} D_{ij} \varepsilon'_j \delta_{jk}. \tag{4.10}
$$

The update increment for $\boldsymbol{S}^{\varepsilon}_{\boldsymbol{x}_i}$ is the negative of the gradient multiplied by the value η_ε to adjust the increment size:

$$
\Delta S^{\varepsilon_k}_{\boldsymbol{x}} = \eta_\varepsilon \sum_{i=1}^{3} e_i \sum_{j=k} D_{ij} \varepsilon'_j. \tag{4.11}
$$

In Eq. 4.11, the inner sum is approximately equal to the stress computed by the MPN with all $j \neq k$ components of the scaled input strain vector set to zero. We call this stress vector $\hat{\boldsymbol{\sigma}}_i^{\prime,NN}$. While it is possible to calculate D_{ij} directly, we can greatly reduce the computational load with this approximation, leading to a final equation for $\Delta S_{\boldsymbol{x}}^{\varepsilon_k}$:

$$\Delta S_{\boldsymbol{x}}^{\varepsilon_k} \approx \eta_\varepsilon \sum_{i=1}^{3} e_i \hat{\sigma}_i^{\prime,NN} \tag{4.12}$$

Computing $\Delta S_{\boldsymbol{x}_i}^{\varepsilon_k}$ using Eq. 4.12 is not significantly different from Eq. 4.11 and is nearly two orders of magnitude faster. Controlling the increment size with η_ε is equivalent to applying a learning rate in backpropagation.

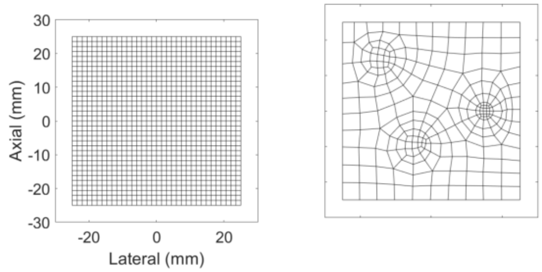
Eq. 4.1 attempts to minimize the stress error for a single stress-strain pair. However, many stress-strain pairs may exist at $\boldsymbol{x}$, meaning the stress error should be minimized in an average sense for all stress-strain pairs at $\boldsymbol{x}$. We do this by invoking Eq. 4.12 for each data pair at $\boldsymbol{x}$ and computing the mean of $\Delta \boldsymbol{S}_{\boldsymbol{x}_i}^{\varepsilon}$ before adding to $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$.

A single application of Eq. 4.12 is insufficient for updating $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$. Alg. 1 details the iterative process for computing a new $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ at location $\boldsymbol{x}$. N corresponds to the number of gradient-descent iterations and N_ε is the number of stress-strain pairs at $\boldsymbol{x}$.

Algorithm 1 Iterations for computing $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$

```
Given: current $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$, $\boldsymbol{\sigma}_i$, $\boldsymbol{\varepsilon}_i$ at $\boldsymbol{x}$
for $n = 1, 2, ..., N$ do
    for $k = 1, 2, 3$ do
        $\Delta S = 0$
        for $i = 1, 2, ..., N_\varepsilon$ do
            Compute $\boldsymbol{\sigma}_i^{NN}$ and (vector) $\hat{\boldsymbol{\sigma}}_{\boldsymbol{i}}^{\prime,NN}$ using $\boldsymbol{\varepsilon_i}$
            Compute $\mathbf{e}_i = \boldsymbol{\sigma}_i^t - \boldsymbol{\sigma}_i^{NN}$
            Compute $\Delta S = \Delta S + \sum_{p=1}^{3} e_p \hat{\sigma}_p^{\prime,NN}$
        end for
        $\Delta S_{\boldsymbol{x}}^{\varepsilon_k} = \eta_\varepsilon \Delta S / N_\varepsilon$
        $S_{\boldsymbol{x}}^{\varepsilon_k} = S_{\boldsymbol{x}}^{\varepsilon_k} + \Delta S_{\boldsymbol{x}}^{\varepsilon_k}$
    end for
end for
```

Computing the stress scaling value $\boldsymbol{S}^{\sigma}$ is far simpler. It is chosen to ensure all components of every stress vector falls within the ±0.8 range. Again, this follows from upper and lower bounds of the hyperbolic tangent activation function being ±1.0. In this study, setting $\boldsymbol{S}^{\sigma} = 1.0$ is sufficient because the magnitude of every computed stress falls below 0.8.



(a) (b)

Figure 4.3: (left) Rectilinear FE mesh (Mesh 1) with 35 nodes per edge. This mesh was used to compute stress and strain fields for all four models. (right) FE mesh that conforms to the geometry of the three inclusion model (Mesh 2).

A Faster Algorithm

Algorithm 1, developed around the approximation in (4.12), is much faster than the algorithm using the exact form in (4.11). Even though the decrease in computation time makes training 2-D CaNNCMs in AutoP feasible, the algorithm would take far too long in 3-D. Therefore, a much faster algorithm was developed and is described in Appendix F. For the current CaNNCM architecture, Algorithm 2 is used.

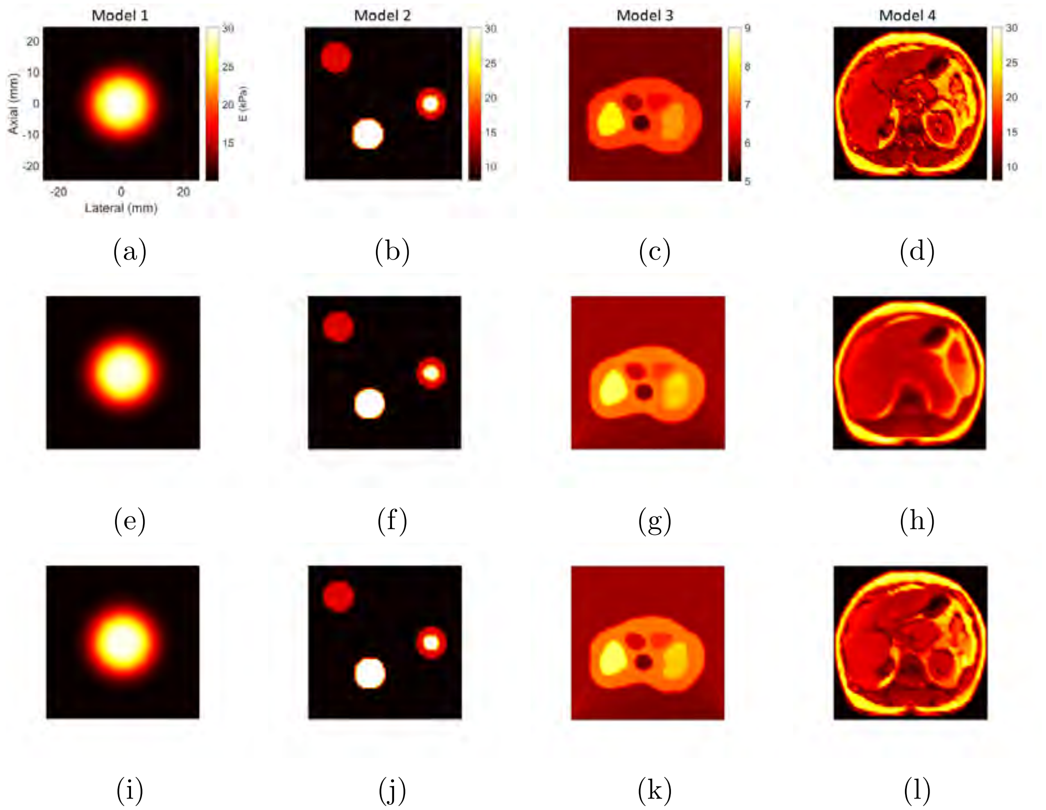
4.2.4 Simulated Phantoms

CaNNCMs learn material and geometric properties from stress-strain data. Here, we turn to simulated data to evaluate CaNNCMs under the ideal circumstance of stresses and strains being known exactly, and hence this study is confined to the dashed-line box in Fig. 4.1. Stress-strain data were generated by FEA (ABAQUS 6.13) with known Young's modulus distributions. A simple four-node, quadrilateral element rectilinear mesh with 35 nodes per 50 mm edge (Fig. 4.3a) was selected to demonstrate the independence of CaNNCMs to the FE mesh. In the FEA, we used a plane-stress, incompressible (Poisson's ratio $\nu = 0.5$) material model, the bottom surface of mesh was pinned to create a fixed boundary condition (BC), and a US probe was pressed into the top surface. The probe-phantom interface was modeled as frictionless to allow lateral motion of the top phantom surface during compression. Four equal compressive loads were applied by the probe up to 13.57 mN, leading to a minimum probe displacement of 0.98 mm and a maximum of 2.23 mm depending on the material property distribution.

Four different simulated phantom models, displayed in the top row of Fig. 4.4, were selected to test different aspects of CaNNCMs. Model 1 (Fig. 4.4a) is a stiff, Gaussian-shaped inclusion embedded in a soft background. The peak Young's modulus of the inclusion was 30 kPa and smoothly transitioned into the 10 kPa background. We chose this model to demonstrate the ability of CaNNCMs to capture smooth, continuous material property distributions, a feat not achievable in our prior work with NNCMs described in Chapter 3. Models 2 and 3 have abrupt transitions in the material property distributions. Model 2 contains three stiff inclusions (15 kPa and 30 kPa) in a 8 kPa background. Model 3 represents a rabbit kidney embedded in a block of gelatin. We previously performed this experiment and trained seven NNCMs in AutoP with the force-displacement measurements [137]. The Young's modulus values chosen for this simulated phantom correspond to the moduli estimated from those seven NNCMs. Model 4 was selected as an extreme case of complex spatial geometry. To generate this model, the gray-scale values of an abdominal MRI scan were scaled to the 8-30 kPa range of Young's modulus values. Model 4 does not represent a real case of elasticity imaging nor do we claim any translational use of CaNNCMs to MRI. The image was only chosen for its geometric complexity while also representing actual human physiological structure.

Two additional data sets were generated with Model 3 after adding noise to the target Young's modulus distribution. Uniformly distributed random values up to ±10% and ±30% the local Young's modulus value (≈ 27.2 dB and ≈ 17.6 dB peak signal-to-noise ratio, respectively) were added to the target distribution of Model 3. We generated two different corrupted target distributions for each test, performing the FEA with the loads and BCs previously specified, and compiled stresses from one analysis and strains from the other.

Figures 4.5a and 4.5b are provided to clarify the experimental procedure. When no noise was added to the target Young's modulus distribution, stresses and strains were both computed in a single FEA. Spatial scaling values were then computed using Algorithm 2 and the pre-trained MPN. Conversely, two



(a) (b) (c) (d)

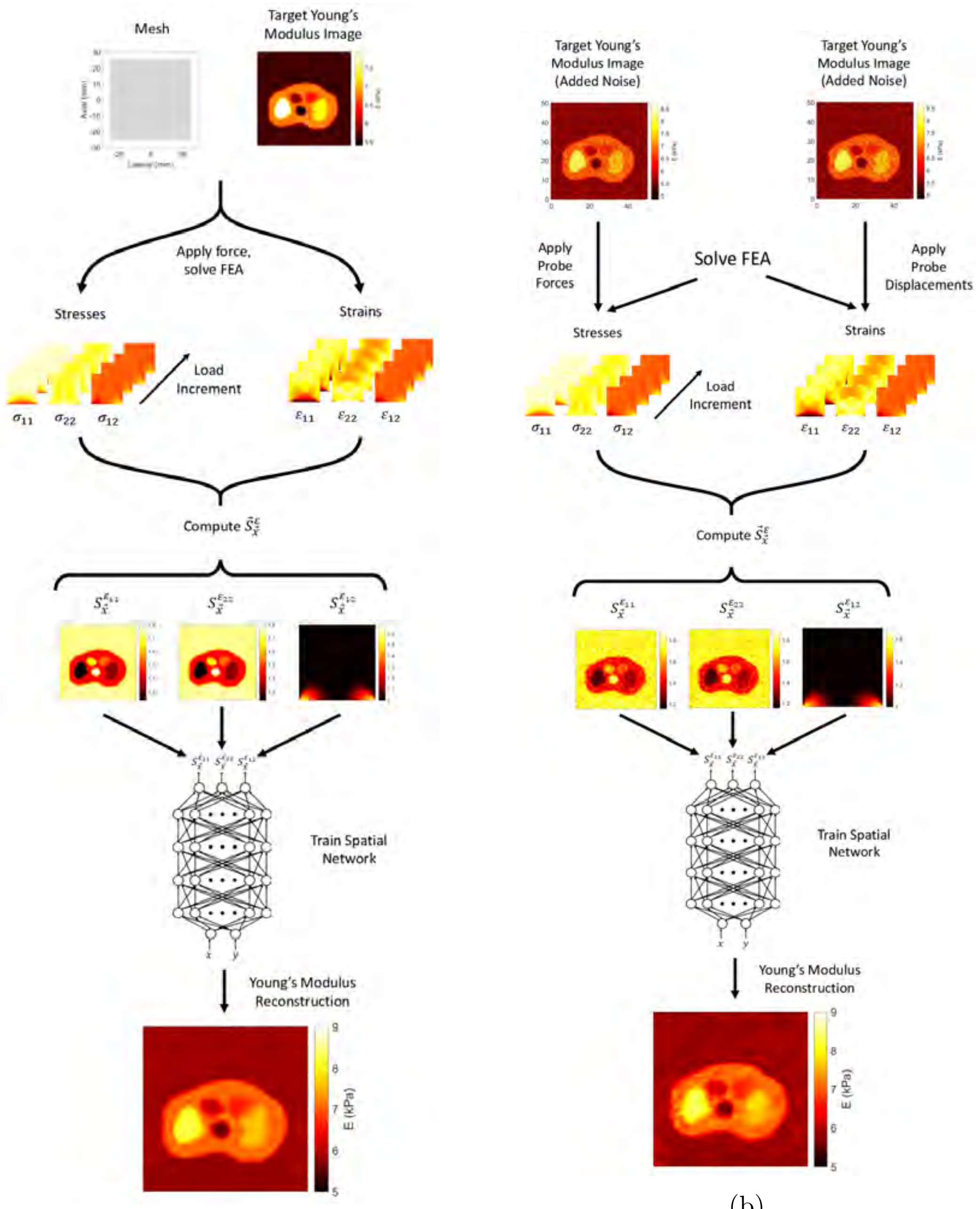
(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 4.4: (Top row) Target Young's modulus distributions for the four 50x50 mm models. Model 4 was created by converting the gray scale values from an abdominal MR image to Young's modulus values. It was included only as an extreme case of spatial complexity that represented actual human physiology and is not a real case of elasticity imaging. (Middle row) Young's modulus images reconstructed by CaNNCMs after spatial scaling update and training using the stress-strain data generated by a forward FEA. Spatial networks were trained using Test 1 parameters. (Bottomw row) Young's modulus images from after SNs trained with Test 2 parameters.

separate FEAs were solved to compute stresses and strains separately when noise was present. Noise was added by sampling from a zero-mean uniform distribution with maximum magnitude equal to either 10% or 30% of the Young's modulus at each $\boldsymbol{x}$. For example, when 10% noise was added, the target Young's modulus at each point was $E'(x, y) = E(x, y) + E(x, y) * p$, where $p \in [-0.1 * E(x, y),\ 0.1 * E(x, y)]$. Stresses and strains were computed in separate FEAs to avoid correlating the noise (i.e., the same value p was not used to compute $E'(x, y)$ at each location).

We created Mesh 2 (Fig. 4.3b) that conforms to the geometry of Model 2. The same force loads and BCs described above were applied in a FEA of this model to generate the stress-strain data. While CaNNCMs are independent of the internal FE mesh structure, the learned distribution of stresses and strains will be affected. For example, the inner inclusion of Model 2 fills


Mesh
Target Young's Modulus Image
Apply force, solve FEA
Stresses
Strains
Load Increment
σ_{11} σ_{22} σ_{12}
ε_{11} ε_{22} ε_{12}
Compute $\hat{S}_{\hat{x}}^{\varepsilon}$
$S_{\hat{x}}^{\varepsilon_{11}}$ $S_{\hat{x}}^{\varepsilon_{22}}$ $S_{\hat{x}}^{\varepsilon_{12}}$
Train Spatial Network
x y
Young's Modulus Reconstruction
9 8 7 6 5
E (kPa)
Target Young's Modulus Image (Added Noise)
Target Young's Modulus Image (Added Noise)
Apply Probe Forces
Solve FEA
Apply Probe Displacements
Stresses
Strains
Load Increment
σ_{11} σ_{22} σ_{12}
ε_{11} ε_{22} ε_{12}
Compute $\hat{S}_{\hat{x}}^{\varepsilon}$
$S_{\hat{x}}^{\varepsilon_{11}}$ $S_{\hat{x}}^{\varepsilon_{22}}$ $S_{\hat{x}}^{\varepsilon_{12}}$
Train Spatial Network
x y
Young's Modulus Reconstruction
9 8 7 6 5
E (kPa)

(a)

(b)

Figure 4.5: Diagram of experimental method. (a) No noise added to target Young's modulus distribution. (b) Uniformly distributed noise added to target distribution.

four whole elements in Mesh 1 and partially fills twelve adjacent elements. Conversely, 16 whole elements in Mesh 2 comprise the same nested inclusion. Furthermore, there are 1156 total elements in Mesh 1 and 235 is Mesh 2. The ratio of generated data from that one inclusion to all other points greatly increases and more accurately captures the geometry.

Before the spatial scaling values could be computed for each model, it was necessary to pretrain a MPN. Without a trained MPN there is no reference for updating $\boldsymbol{S}^{\varepsilon}_{\boldsymbol{x}}$. The material property network consisted of two hidden layers with six nodes per layer. Weights were initialized by drawing from a uniform distribution in the range [-0.2, 0.2]. We generated 5000 strain vectors, whose components were also randomly generated uniformly in the range [-0.2, 0.2], and computed the corresponding stress vectors using a plane-stress model with a Young's modulus value of 10 kPa and Poisson's ratio $\nu = 0.5$. Note that the initialization range for the weights and the strain vectors do not have to match. Previous results suggested this range performed well for the MPN. As for the strains, we chose a range that extends beyond the magnitude of the strain vectors generated for the aforementioned models. Before weight update via the resilient backpropagation (RPROP) algorithm [123], frame invariance of the stresses and strains was enforced by rotating the data 90° and appending the new rotated data to the original set, doubling the total number of training pairs. This rotation was done by simply swapping the axial and lateral components of the data and was implemented in our previous study with AutoP (and detailed in Appendix A. Finally, we trained the MPN over 50 epochs with $\boldsymbol{S}^{\varepsilon}_{\boldsymbol{x}} = \mathbf{1}$ and $\boldsymbol{S}^{\sigma} = 1$.

After generating all data sets and pretraining the MPN, we used the stress-strain data from all four load increments in each set to compute new spatial scaling values using Alg. 1 ($N = 150$, $NN_{\varepsilon} = 8$ due to frame invariance, $\eta_{\varepsilon} = 2.5$). Spatial networks were trained for each model using the newly computed $\boldsymbol{S}^{\varepsilon}_{\boldsymbol{x}}$. Each network was comprised of five hidden layers with 25 nodes per hidden layer. Training was split into iterations to mimic what occurs in AutoP. For example, instead of simply training the SN over 12000 epochs, we split training into iterations where fewer epochs were used. 10 iterations of 300 epochs were used in Test 1 whereas 30 iterations of 600 epochs were used in Test 2. These were equivalent to training for 3000 and 12000 epochs in a single iteration, respectively. Training for the SN was implemented in TensorFlow using He initialization of the weights [139], the Adam optimizer [140] with default settings, and a learning rate of 0.03.

The trained spatial network for each model was then paired with the pretrained material property network to form a CaNNCM and used to reconstruct the Young's modulus image. Reconstruction using only a CaNNCM is done by setting a constant strain vector $\boldsymbol{\varepsilon} = [0.003\ 0.005\ 0.0001]$ and varying

(x, y) over the domain of the mesh. At each (x, y), a corresponding stress $\boldsymbol{\sigma}$ was computed. Axial and lateral components of the input strain vector (ε_{11} and ε_{22}, respectively) and the axial component of the computed stress (σ_{22}) were used to compute the spatially varying Young's modulus $E(x, y)$ by inverting the plane-stress equation:

$$E(x, y) = \frac{\sigma_{22}(1 - \nu^2)}{\nu\varepsilon_{11} + \varepsilon_{22}} \tag{4.13}$$

where $\nu = 0.5$. The choice of strain vector is arbitrary so long as it resides within the range of training data. Selecting small values for each component ensured the strain was within said range.

4.3 Results

Fig. 4.6 contains the results of computing the spatial scaling values for all seven cases in this study. Plots in the left-most column are the mean, minimum, and maximum error in each iteration of Alg. 1. Errors are the RMS of the difference between $\boldsymbol{\sigma}^t$ and σ_i^{NN} over all stress-strain pairs for the model. Columns 2, 3, and 4 are the maps of $S_{\boldsymbol{x}}^{\varepsilon_1}$, $S_{\boldsymbol{x}}^{\varepsilon_2}$, and $S_{\boldsymbol{x}}^{\varepsilon_3}$, respectively.

The error curves provide insight on the number of iterations required in Alg. 1. When implemented in AutoP, the spatial scaling values will be recomputed many times, meaning there is a trade-off between computation speed and error. From these curves, 50 iterations appears to be sufficient. Images of the computed spatial scaling values, on the other hand, offer intuition on what information is contained with $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$. We observe the scaling values for the axial and lateral strains are inversely proportional to the Young's modulus. In the case of linear-elastic materials, the spatial scaling values, and thus the SN, contain information about distribution of the relative stiffness.

Young's modulus images reconstructed with CaNNCMs trained for the four models (no added noise and data generated on Mesh 1) are displayed in the middle and bottom rows of Fig. 4.4. A comparison of the target and CaNNCM-estimated Young's modulus along cross-sections of these two models are shown in Fig. 4.7. In the case of Model 1, there is no significant difference between the Young's modulus estimates between Test 1 and Test 2. Similarly, there is only a marginal difference in the results for Model 2. The

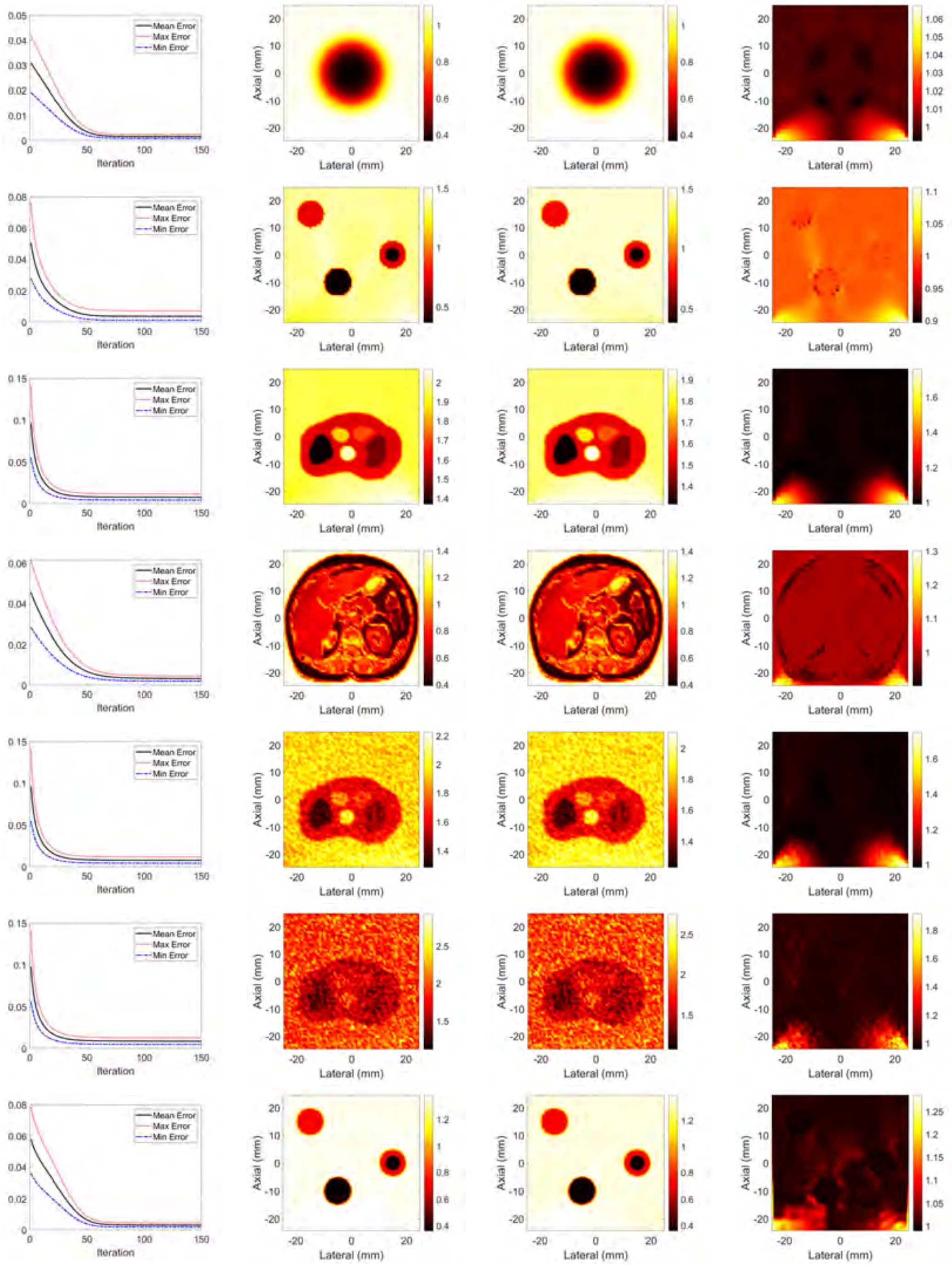
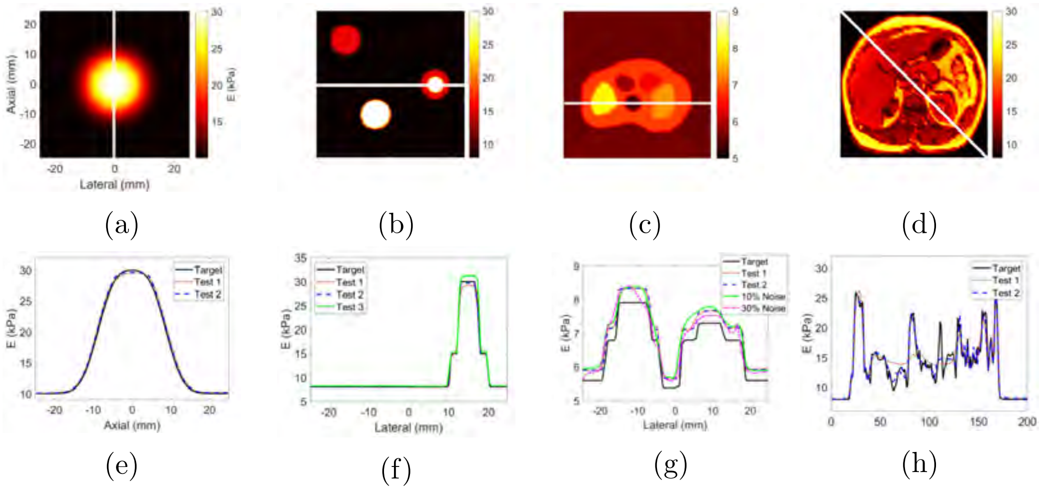


Figure 4.6: The left-most column contains the minimum, maximum, and mean RMS error curves from Alg. 1 while computing the values of $\boldsymbol{S}^{\varepsilon}_{\boldsymbol{x}}$ for each of the seven cases. Columns 2-4 are the resulting $S^{\varepsilon 1}_{\boldsymbol{x}}$, $S^{\varepsilon 2}_{\boldsymbol{x}}$, and $S^{\varepsilon 3}_{\boldsymbol{x}}$ distributions, respectively.

effect of increased training iterations/epochs is far more pronounced for Models 3 and 4. In the case of the former, the boundaries of the regions become sharper. For Model 4, the internal structures only become distinguishable under Test 2 training.

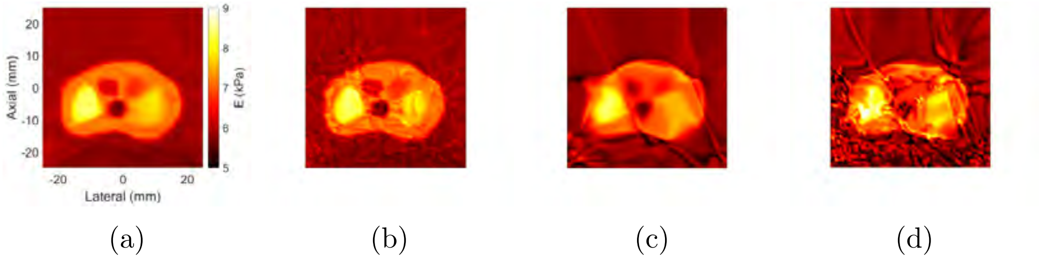
Images reconstructed by CaNNCMs are expected to improve when the number of training epochs increases. That Fig. 4.4l better matches the target distribution than Fig. 4.4h is no surprise. What these results do show is that 1) CaNNCMs are capable of learning fairly complex material property distributions and 2) in the absence of noise, the chosen training parameters do not result in over-training the SN.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.7: (a-d) The white lines indicate the locations where Young's modulus cross-section comparisons occur. (e) Target and CaNNCM Young's modulus along $x = 0$ for single inclusion model. (f) Young's modulus values along $y = 0$ for three inclusion model. Test 3 refers to the case where Mesh 2 and Test 1 parameters were used. (g) Target and CaNNCM modulus estimates for no added noise, 10% added noise, and 30% added noise along $y = -6.3$ of the kidney model. (h) Young's modulus value for abdominal MRI model along the diagonal.



(a) (b) (c) (d)

Figure 4.8: Young's modulus image after training CaNNCM with stress-strain data from kidney model with added noise. (a,b) Test 1 and Test 2 training parameters for model with 10% added noise, respectively. (c,d) Test 1 and Test 2 training parameters for model with 30% added noise, respectively.

Image reconstructions with CaNNCMs trained on data with 10% and 30% additive noise are displayed in Fig. 4.8. Modulus values estimated along the line indicated in Fig. 4.7c by these CaNNCMs (trained with Test 1 parameters) are included in the curve comparison in Fig. 4.7g. Finally, the Young's modulus values computed by the CaNNCM trained on Model 2, Mesh 2, and Test 1 parameters along the line $y = 0$ are included in Fig. 4.7f.

Table 4.1 contains quantitative comparisons between trained CaNNCMs and the target Young's modulus images. Training times for the SN are also included. For each point (x, y) in the mesh, the error between the target and CaNNCM Young's modulus estimate was computed using Eq. 4.14.

$$e^{E}_{(x,y)} = \frac{|E^{target}_{(x,y)} - E^{NN}_{(x,y)}|}{E^{target}_{(x,y)}} \tag{4.14}$$

where $E^{target}_{(x,y)}$ is the target Young's modulus value at (x, y) and $E^{NN}_{(x,y)}$ is the modulus estimated by the corresponding trained CaNNCM.

We observe that as the complexity of the model geometry increases and training parameters remain the same, the mean error in the CaNNCM-reconstructed Young's modulus image also increases. As expected, increasing the number training iterations and epochs in Test 2 generally reduced the error between target and CaNNCM-estimated values. Our expectation was not met for Model 2 and Model 3 with 30% added noise. The increase in error for Model 2, Test 2 is likely a statistical artifact. Comparing the curves in Fig. 4.7f reveals the CaNNCM trained with Test 2 parameters better approximates the target curve. Furthermore, the images were interpolated to a new rectilinear grid before computing the error. It is possible that the interpolation procedure led the the slightly increased error.

Over-training led to the increased error for Model 3 with 30% added noise. Comparing Figs. 4.8c and 4.8d reveals the corruption caused by the SN fitting the noise. Similarly, over-training may be also responsible for the increased error in Model 2 when Mesh 2, Test 1 parameters were used. The "Test 3" curve in Fig. 4.7f refers to this case and shows a slight bias in the Young's modulus estimate. A particularly large over-estimate occurs for the small, stiff inclusion embedded in the softer inclusion.

Biased errors also occur for Model 3 where all the CaNNCM-estimated Young's modulus values in Fig. 4.7g lie above the line denoting target modulus values. The same bias appears in all training cases for this model. While not displayed, this bias can be removed by pretraining the MPN as a 5 kPa linear-elastic material instead of 10 kPa. We do not expect this to be an issue when CaNNCMs are implemented in AutoP because the MPN may be retrained multiple times. Retraining the MPN with updated stress-strain data should alleviate these types of issues by adjusting the "reference" material to a form more suitable for the generated data.

Table 4.1: Young's modulus errors and SN training times for the four models shown in Fig. 4.4. No asterisk indicates Mesh 1, Test 1 parameters whereas single asterisk (*) indicates Test 2 parameters. The double asterisk (**) specifies the use of Mesh 2, Test 1 parameters .

Model	Modulus Error	Training
	Mean ± STD	Time (s)
1	0.0166 ± 0.0099	71
1^{*}	0.0146 ± 0.0076	271
2	0.0131 ± 0.0172	71
2^{*}	0.0140 ± 0.0190	271
$2.^{**}$	0.0258 ± 0.0334	56
3	0.0504 ± 0.0142	72
3^{*}	0.0493 ± 0.0131	272
3 (10% noise)	0.0534 ± 0.0198	71
3^{*} (10% noise)	0.0487 ± 0.0230	272
3 (30% noise)	0.0418 ± 0.0252	71
3^{*} (30% noise)	0.0549 ± 0.0399	271
4	0.0658 ± 0.0755	71
4^{*}	0.0485 ± 0.0518	270

4.3.1 Effect of Incorrect Poisson's Ratio

All of the results thus far have been achieved with a MPN which describes a linear-elastic material with the same Poisson's ratio as the stress-strain data generated in forward FEAs of the four Models. We previously stated the MPN was pretrained to characterize a linear-elastic material with a Young's modulus of 10 kPa and $\nu = 0.5$. Models 1–4 had more complex Young's modulus distributions, but were also defined to be incompressible. What happens if there is a mismatch between the Poisson's ratio of the MPN and the generated stress-strain data?

To test this, we pretrained a MPN with $E = 10$ kPa and $\nu = 0.3$. The same stress-strain data generated for the incompressible Models 1–4 were again used to compute spatial values following the same procedure described above. Results of this test are shown in Fig. 4.9. Plots in column 1 are the mean and maximum error curves computed in Algorithm 2. Columns 2 and

3 are maps of the axial strain scaling $S_{\boldsymbol{x}}^{\varepsilon_{22}}$ and Young's modulus, respectively. The last column are curves of the Young's modulus along the lines specified in Fig. 4.7a–d for each respective Model.

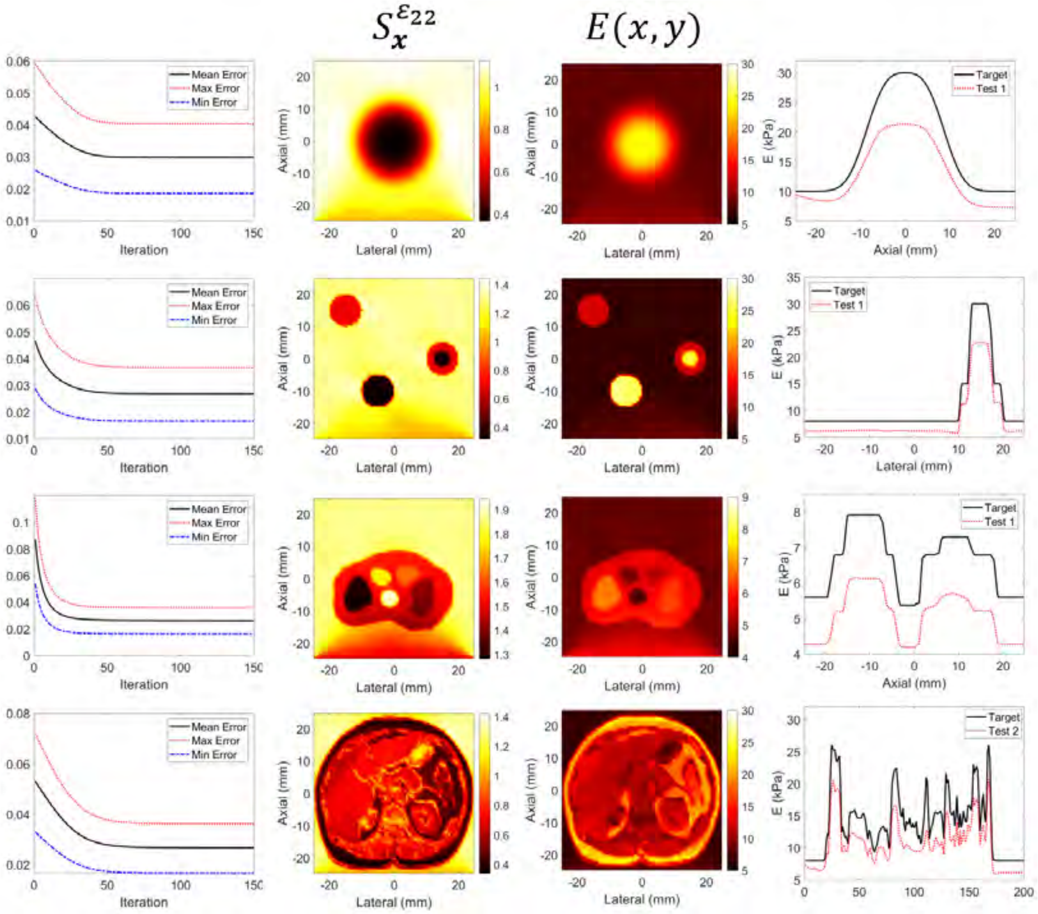


Figure 4.9: Effect of the MPN pretrained with Poisson's ratio not matching the stress-strain data. The left-most column contains the minimum, maximum, and mean RMS error curves from Alg. 2 while computing the values of $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ for each of the seven cases. Columns 2-3 are the resulting $S_{\boldsymbol{x}}^{\varepsilon_{22}}$ and Young's modulus distributions, respectively. Curves in column 4 are the target and reconstructed Young's modulus values along the white lines in Figs. 4.7a–d.

Comparing the error curves to Fig. 4.6, we observe that error values appears converge when the Poisson's ratio of the MPN and stress-strain data do not match to a larger value. The maps in columns 2 and 3, as well as the Young's modulus curves in column 4, reveal the reason: the resulting Young's modulus estimates are too low. It appears the full CaNNCM underestimates the Young's modulus everywhere. Furthermore, a large artifact appears toward the bottom boundary of the phantom, where stresses tend to increase because of the pinned boundary.

Fortunately, this should not be a problem in AutoP, regardless of whether the Poisson's ratio of the pretrained MPN matches the data. The MPN is repeatedly retrained with new stress-strain data throughout the course of AutoP. We expect the MPN to learn the correct Poisson's ratio as training progresses.

4.4 Discussion

We have demonstrated the ability of CaNNCMs to learn both the material properties and geometry of 2-D linear-elastic, isotropic materials under plane-stress loading when the stress-strain data are known. CaNNCMs utilize two cooperating NNs to minimize the error between known and predicted stresses after a forward propagation of a strain vector and a (x, y) coordinate. The material property network learns a general stress-strain relation — here, the linear-elastic response of a 10 kPa homogeneous material — whereas the spatial network learns a map of relative stiffness. Developing the two-network cooperating architecture followed from attempts to simply add two extra inputs to the material property network for (scaled) Cartesian x- and y-coordinates. These early CaNNCMs performed poorly and were unable to accurately learn geometric information regardless of the number of hidden layers, nodes per layer, and/or training parameters. We believe the failure was caused in large part by an incompatibility between the hyperbolic tangent activation function and the geometry-material property relationship. In short, a hyperbolic tangent is symmetric in sign so that a negative valued input produces a negative valued output. Use of this function for the MPN works because stresses and strains exhibit the same relationship. Furthermore, a zero-valued input vector should produce a zero-valued output vector (i.e., zero strain means zero stress). But, the same does not hold for spatial location and material property. Reformulating the problem as adjusting a reference material property learned by the MPN with some auxiliary function led to incorporating the spatial network.

Adding the spatial network required developing a method of computing its outputs. We use a gradient-descent based approach by backpropagating the output error all the way to the MPN inputs/spatial network outputs. The spatial scaling values effectively encode stiffness relative to the reference material property. For example, $S_{\boldsymbol{x}}^{\varepsilon_2}$ calculated for Model 1 is ≈ 0.3657 at the

30 kPa inclusion peak and ≈ 1.1 for the 10 kPa background, maintaining the ratio $30/10 = 1/3 \approx 1.1/0.3657$. Similarly, Model 4 values span 8-30 kPa and the resulting $S^{\varepsilon_2}_{\boldsymbol{x}}$ are in the range $[0.3891,\ 1.4014]$. We must emphasize that the spatial scaling values were not specifically computed to behave in this manner; this property emerged from minimizing the error function defined in Eq. 4.2.

From our previous study, we found only two hidden layers were necessary for the material property network to learn a linear-elastic relationship for 2-D and 3-D materials. Spatial networks, though, require a larger network. We chose to increase the number of hidden layers instead of vastly increasing the number of nodes per layer. The spatial network must be larger because the mapping from (x, y) to $\boldsymbol{S}^{\varepsilon}_{\boldsymbol{x}}$ is more complicated than a linear stress-strain relationship and thus requires a network with larger capacity. While there is no strict rule for determining NN size, five hidden layers comprised of 25 nodes each was sufficient for all four models in this study. However, the larger size of the SN increases the risk of over-training, as observed in Model 2 with 30% added noise.

Keeping the spatial network size constant, we could improve the Young's modulus reconstruction by changing the training parameters or the mesh. In the former case, increasing the number of epochs and training iterations produced better results by simply allowing more weight updates to occur. But increasing training epochs is not the best choice if training time is to be minimized. For the cases where Test 1 parameters were used, training time was $\approx$ 71s on a quad-core CPU operating at 2.7 GHz. Test 2 parameters increased the time to $\approx$ 271s on the same computer.

Conversely, changing from Mesh 1 to Mesh 2 and resorting back to Test 1 parameters only required $\approx$ 56s of training time. Changing the mesh reduced the total number of data points from 4624 to 940, in turn reducing the training time. However, maintaining the same number of training epochs for the reduced amount of training data did not improve the resulting Young's modulus estimates. There was arguably a qualitative improvement due to changes in the data sampling distribution. With the rectilinear mesh, the edges of the inclusions are coarsely sampled and the number of data points pulled from said inclusions are small compared to the soft background material. Changing the mesh altered this sampling distribution so that data are better sampled around edges.

Finally, while the addition of noise does not appear to significantly affect the ability of CaNNCMs to learn material properties, the geometry is corrupted. Unfortunately, it is difficult to extrapolate these results to AutoP because the noise will appear in force-displacement measurements and then propagate through the stress-strain calculations in non-straightforward ways. Implementation in AutoP paper will reveal how robust CaNNCMs are to the noise encountered in real measurement data.

4.5 Conclusions

Cartesian neural network constitutive models can simultaneously learn material property and geometric information. Unlike previous machine-learning methods, CaNNCMs are able to capture continuous material property distributions. Furthermore, these networks can resolve fine structures with minor adjustments to training or the finite element mesh, the latter of which changes the distribution of available training data. CaNNCMs are fairly robust to noise and can still produce accurate estimates of Young's modulus for linear-elastic materials at the cost geometric distortion. CaNNCMs are a novel NN architecture that, once incorporated into the Autoprogressive Method, will offer a machine-learning alternative to elasticity imaging.

Chapter 5

Autoprogressive Training of CaNNCMs for Linear-elastic Materials

5.1 Introduction

Quasi-static ultrasonic elastography (QUSE) is generally an ill-posed inverse problem because we cannot normally acquire all of the data necessary to solve for material properties exactly. Instead we impose assumptions by selecting, for example, a plane-strain linear-elastic constitutive model that constrains the problem to closely match the data that can be acquired. The data typically are acquired by slowly pressing the US probe into the tissue surface while capturing RF echo frames. Speckle-tracking algorithms applied to RF data estimate internal axial displacements (along the direction of US beam propagation) resulting from the quasi-static loading. At most, each measured data set contains force applied by the US probe, motion of the probe, and internal deformation of a tissue volume sampled within a finite plane. Model-based inverse methods can be expressed as an objective function that is minimized by seeking a pre-defined set of model parameters at position $\boldsymbol{x}$, $\hat{\boldsymbol{\theta}}(\boldsymbol{x})$,

$$\boldsymbol{\theta}(\boldsymbol{x}) = \underset{\hat{\boldsymbol{\theta}}(\boldsymbol{x})\, \in\, \mathbb{R}}{\operatorname{argmin}} \sum_{n=1}^{N_p} \sum_{k=1}^{N_d} \mathrm{f_u}(\boldsymbol{u}_k^n\{\hat{\boldsymbol{\theta}}(\boldsymbol{x})\}, \hat{\boldsymbol{u}}_k^n), \tag{5.1}$$

where N_p refers to the number of measured data sets, N_d is the number of measured displacements in each of the N_p sets, $\hat{\boldsymbol{u}}_k^n$ are the measured displacement vectors, $\boldsymbol{u}_k^n$ are displacements estimated via a forward problem (e.g., via finite element analysis (FEA)), and $\mathrm{f_u}$ is often the L_2 norm of their dif-

The majority of this chapter is reproduced from [141]. ©2018 IEEE. Reprinted, with permission, from Cameron Hoerig, Jamshid Ghaboussi, and Michael F. Insana, "Data-driven Elasticity Imaging Using Cartesian Neural Network Constitutive Models and the Autoprogressive Method", IEEE Transactions on Medical Imaging, November 2018 ference. If the total deformation is small and applied slowly, soft tissues are often assumed to be linear-elastic, isotropic, and incompressible, leaving the spatial distribution of the Young's modulus to be estimated ($\boldsymbol{\theta}(\boldsymbol{x}) = E(\boldsymbol{x})$). That is, if $\boldsymbol{\theta}(\boldsymbol{x})$ is a vector composed of nonzero elements from the constitutive matrix, then this model assumes Young's modulus $E(\boldsymbol{x})$ is the only unknown. With larger deformations applied quickly, tissues exhibit non-linear [27, 52] and viscoelastic [142] material properties that require models with more parameters and force-displacement measurements acquired over time.

Accurately estimating material properties requires acquisition of more force-displacement data than can be obtained using pulse-echo US imaging. Barbone and Bamber proved that a single displacement measurement is insufficient to uniquely estimate Young's modulus [143]. Barbone *et al.* later showed that the Young's modulus distribution can be determined up to a multiplicative constant if multiple displacement measurements are available [90]. Instead of relying on multiple displacement measurements, other approaches to the inverse problem include the use of *a priori* information [144] and/or regularization [92, 145, 146]. More recently, Tyagi *et al.* [89] demonstrated how measurements of the surface force applied by the US probe [24, 34] can provide the additional information necessary to estimate the magnitude of the Young's modulus distribution, not just the relative values. Other investigators have proposed a method using only surface information to reconstruct the interior modulus distribution [147, 148]. QUSE has also been extended to the estimation of non-linear and viscoelastic properties of tissues [96, 149, 150, 136, 151, 54, 152, 153], which provides more diagnostic information than the Young's modulus [154, 52]. For a comprehensive review of model-based methods, see [86].

Model-based methods provide no means for discovering new diagnostically-relevant mechanical properties or for exploring ranges of known model parameters for relevance in a given situation. Consequently, we adopt a data-driven approach that first provides a nonparametric method for estimating stresses and strains from force and displacement measurements. Then, from stresses and strains, any and all models can be tested to find those parameters with the most diagnostic potential.

We previously described the first steps in the development of our approach using neural network constitutive models (NNCMs) and the Autoprogressive

Method (AutoP) [108]. AutoP combines FEA and artificial neural networks (NNs) to build data-driven constitutive models from force-displacement data [117]. AutoP has been used in many civil and geotechnical engineering applications to model linear, nonlinear, path-dependent, and time-dependent material properties [99, 100, 101, 102, 103, 104, 106, 107]. Force-displacement measurement data are iteratively applied in two separate FEAs operating on one mesh to estimate increasingly accurate distributions of stresses and strains. From these data, NNCMs gradually learn material properties. This is possible as the equilibrium and compatibility conditions imposed by the FEAs are able to consistently convert force and displacements into stresses and strains, which in turn are made consistent with the data through the NNCM. The goal of AutoP is to reconcile the stress-strain distributions estimated when force or displacement measurements are applied to a FE model by training the NNCM. Unlike model-based methods, NNCMs are theoretically flexible enough to approximate any physically realizable stress-strain relationship without a prior assumption of the underlying material behavior.

5.2 Methods

All prior implementations of AutoP for mechanical modeling have used a form of material property networks (MPN, left side of Fig. 5.1) that accept a strain vector at the input and return a stress vector at the output (i.e., $R_m : \boldsymbol{\varepsilon} \rightarrow \boldsymbol{\sigma}$). MPNs characterizing viscoelastic or non-linear material properties must also include stress and strain history points at the input. Strain values applied to the input of the MPN were scaled by a vector $\boldsymbol{S}^{\varepsilon}$ whereas the stresses at the output were scaled by the vector $\boldsymbol{S}^{\sigma}$. However, MPNs are mapped to specific homogeneous regions of the FE mesh and thus are only effective when the internal geometry of the object is known. Without spatial information at the input of the network, a MPN is unable to learn spatially-varying material property distributions. Therefore, we invented Cartesian NNCMs (CaNNCMs) that simultaneously learn material property and geometric information.

The architecture of a CaNNCM is depicted in Fig. 5.1 and its core theory of operation was described in Chapter 4. It is comprised of both a MPN and spatial network (SN). The structure and function of the MPN is un-

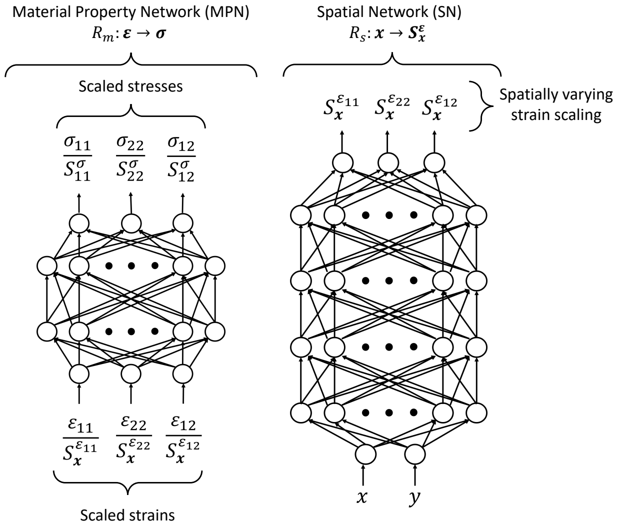

Material Property Network (MPN)
$R_m\colon \boldsymbol{\varepsilon} \to \boldsymbol{\sigma}$
Scaled stresses
$\frac{\sigma_{11}}{S_{11}^{\sigma}}$ $\frac{\sigma_{22}}{S_{22}^{\sigma}}$ $\frac{\sigma_{12}}{S_{12}^{\sigma}}$
$\frac{\varepsilon_{11}}{S_{\boldsymbol{x}}^{\varepsilon_{11}}}$ $\frac{\varepsilon_{22}}{S_{\boldsymbol{x}}^{\varepsilon_{22}}}$ $\frac{\varepsilon_{12}}{S_{\boldsymbol{x}}^{\varepsilon_{12}}}$
Scaled strains
Spatial Network (SN)
$R_s\colon \boldsymbol{x} \to \boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$
$S_{\boldsymbol{x}}^{\varepsilon_{11}}$ $S_{\boldsymbol{x}}^{\varepsilon_{22}}$ $S_{\boldsymbol{x}}^{\varepsilon_{12}}$
Spatially varying strain scaling
x y

Figure 5.1: Structure of the CaNNCM composed of a MPN and SN. The MPN learns a "reference" material property whereas the SN learns spatial variation of the reference.

changed, but the addition of the SN allows this pair of cooperating networks to learn spatially varying material properties. The MPN learns a "reference" stress-strain relationship whereas the SN learns how the object deviates from the reference as a function of position. Outputs from the SN are spatially varying strain scaling vectors, $\boldsymbol{S}^{\varepsilon} \to \boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$, meaning the SN transforms the strain vectors input to the MPN. The SN can be represented by the function $R_s : \boldsymbol{x} \to \boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$.

While previous work with NNCMs and AutoP demonstrate the ability to learn complex material properties [102, 103, 104, 106, 155], the initial network architecture introduced in this report is limited to 2-D, linear-elastic, isotropic materials. In this case, the MPN effectively learns a plane-stress relationship with a constant Young's modulus whereas the SN learns relative stiffness. Results of prior work with AutoP leads us to believe that CaNNCMs will be capable of capturing non-linear and viscoelastic behaviors in 3-D as we further develop this method.

A CaNNCM replaces the constitutive model in (5.1) and the objective function is minimized when the network learns the spatially-varying stress-strain relationship described by the measured data:

$$R_m, R_s = \underset{R_{\theta_m}, R_{\theta_s} \in \mathbb{R}}{\operatorname{argmin}} \sum_{n=1}^{N_p} \sum_{k=1}^{N_d} \mathrm{f_u}(\boldsymbol{u}_k^n\{R_m, R_s\}, \hat{\boldsymbol{u}}_k^n), \tag{5.2}$$

where R_{θ_m} refers to the weights of the MPN and R_{θ_s} are the weights of the SN. Here, $\mathrm{f_u}(\cdot)$ is the L_1 norm of the difference between measured $\hat{\boldsymbol{u}}_k^n$ and computed displacements. We use an L_1 norm in this case rather than an L_2 norm to reduce the effects of extensive outliers. We describe in the next section how the computed displacements arise in AutoP.

Equation (5.2) is useful as a description of the inverse problem but misses the nuances of our data-driven approach. First, in model-based methods the error computed by $\mathrm{f_u}(\cdot)$ would directly affect the choice of parameter values. Gradient-descent based schemes are typically employed to gradually adjust parameters values based on error gradients. However, we will describe in Sec. 5.2.1 how (5.2) is used to determine if the CaNNCM has learned the material properties consistent with the data, but has no bearing on the weight update procedure. That is, the error computed by $\mathrm{f_u}(\cdot)$ only indirectly affect R_{θ_m} or R_{θ_s}. Second, weights of the MPN and SN do not map directly to material parameters. There is no weight in either network that represents Young's modulus. The MPN and SN together define the function $R_m, R_s : \boldsymbol{\varepsilon}(\boldsymbol{x}) \to \boldsymbol{\sigma}(\boldsymbol{x})$. Image reconstruction occurs by relating the mechanical behavior learned by CaNNCMs to a chosen constitutive model *after* training. Even though modeling errors are still possible, the benefit lies in the potential for using a CaNNCM to estimate the parameters that might apply to any constitutive model.

We will show in the following section how AutoP is used to minimize (5.2) by reconciling stresses and strains estimated by the CaNNCM being trained. Then, using force-displacement measurements acquired through simulation and experimentally, we will 1) demonstrate the ability of CaNNCMs to learn stress and strain maps and ultimately reconstruct accurate Young's modulus images and 2) explore how sampling affects the ability of CaNNCMs to learn these properties. This last objective is part of our ongoing investigation to determine how data should be sampled in time and space to accurately estimate all stresses and strains.

5.2.1 The Autoprogressive Method

Training CaNNCMs requires an adjustment to the AutoP procedure described in prior reports [117, 108]. AutoP uses FEA to compute stresses and strains in response to force and displacement load increments. Internal $\boldsymbol{I}^n$ and external $\boldsymbol{P}^n$ forces are balanced for boundary conditions (BCs) applied in the nth load increment in the solution of a FEA [84]:

$$\boldsymbol{P}^n = \boldsymbol{I}^n \tag{5.3}$$

$$\boldsymbol{I}^n = \boldsymbol{I}^{n-1} + \boldsymbol{K}^n \Delta \boldsymbol{U}^n \tag{5.4}$$

$$\boldsymbol{K}^n \Delta \boldsymbol{U}^n = \boldsymbol{P}^n - \boldsymbol{I}^{n-1} \tag{5.5}$$

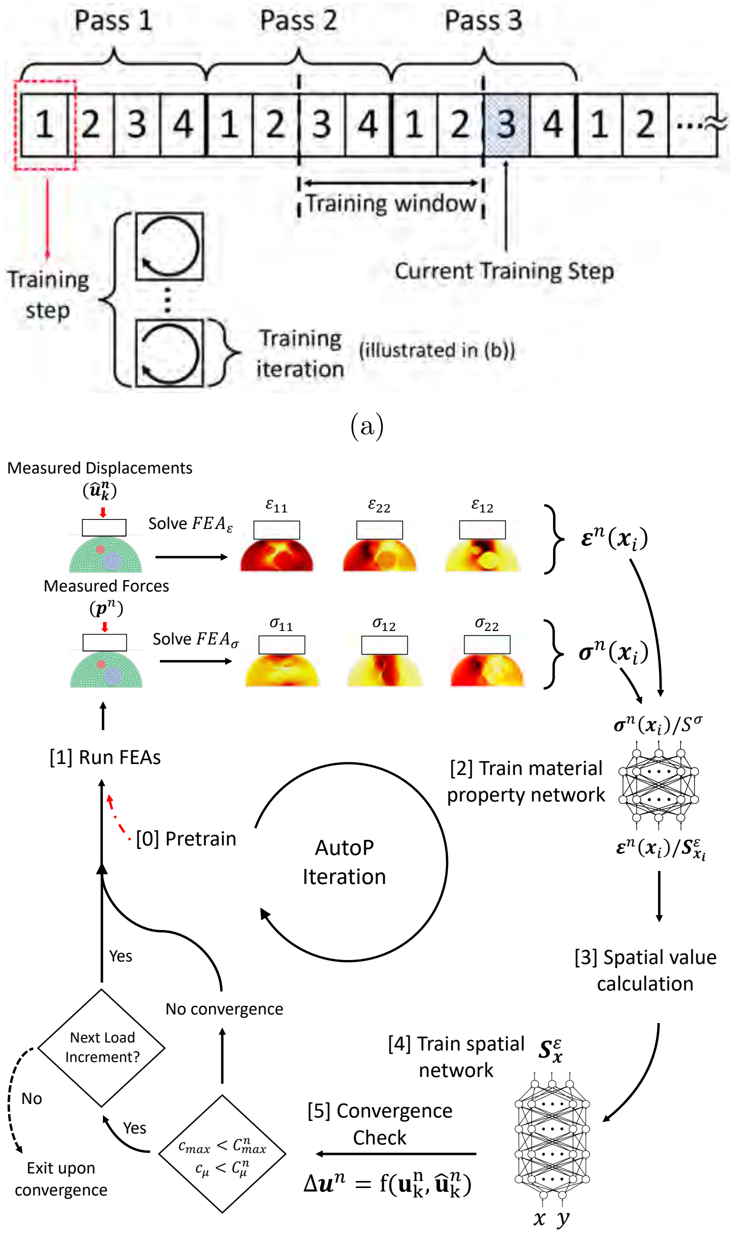
$$\boldsymbol{K}^n = \sum_{e=1}^{N_e} \int_{\Omega_e} \boldsymbol{B}_e^T \overbrace{\boldsymbol{D}^n(\boldsymbol{x})}^{CaNNCM} \boldsymbol{B}_e \mathrm{d}\Omega_e \tag{5.6}$$

$$\boldsymbol{I}^{n-1} = \sum_{e=1}^{N_e} \int_{\Omega_e} \boldsymbol{B}_e^T \overbrace{\boldsymbol{\sigma}^{n-1}(\boldsymbol{x})}^{CaNNCM} \mathrm{d}\Omega_e \tag{5.7}$$

where $\boldsymbol{K}^n$ is the tangent stiffness matrix computed in the nth load increment, $\Delta\boldsymbol{U}^n$ is the vector of displacement increments, $\boldsymbol{P}^n$ is the vector of applied surface forces, and $\boldsymbol{I}^n$ is the vector of internal resisting forces. In Eq. 5.5, $\boldsymbol{I}^n$ is expressed as the sum over all N_e elements by multiplying the strain-displacement matrix $\boldsymbol{B}_e$ with the stress vector $\boldsymbol{\sigma}^{n-1}(\boldsymbol{x})$ and integrating over the element domains Ω_e. Force BCs reside in $\boldsymbol{P}^n$ and displacement BCs populate components of $\Delta\boldsymbol{U}^n$.

A forward analysis consists of applying force and/or displacement BCs to the FE model and solving the system of equations (5.5) for the unknown displacement increments that that satisfy (5.3). During the analysis, the stiffness matrix $\boldsymbol{D}^n(\boldsymbol{x})$ and stress vector are computed using a constitutive model or, in our method, the CaNNCM being trained. To be clear, both the stiffness matrix and stress vector in (5.6) are computed from the CaNNCM, not a pre-selected constitutive model as would be done in model-based methods. The analytical expression for $\boldsymbol{D}^n(\boldsymbol{x})$ is provided in Appendix D. FEA techniques are thus used to solve the forward problem for $\boldsymbol{u}_k^n\{R_m, R_s\}$ in (5.2).

AutoP is organized in a hierarchy of training passes, steps, and iterations as shown in Fig. 5.2a. A single training iteration comprises several stages utilizing force-displacement measurements from a single load increment. A set of iterations performed with the same input measurement data is one training step. Completing a series of training steps over all load increments constitutes a pass. The following paragraphs track the six stages of AutoP



(a)

(b)

Figure 5.2: (a) Hierarchy of training passes, steps, and iterations in AutoP. The training window determines the number of preceding training steps from which stress-strain data are included during stages 2–4 of the current step. (b) One AutoP training iteration using a CaNNCM is illustrated. Adding the spatial network necessitates two extra stages where S^ε_x are updated and the SN is trained. Training of the MPN and test for convergence follow the same procedure we describe in a previous report [108].

processing illustrated in Fig. 5.2b. Many AutoP iterations are performed throughout training, during which the MPN and SN are repeatedly retrained. We will consider the jth training iteration and introduce the superscripts j and $j+1$ to R_m and R_s to clarify which version of each network is active during each stage.

[0] Pretraining Before the first set of measurement data is input, the CaNNCM is initialized using linear-elastic equations (stage [0]). For the 2-D problems we describe, a Young's modulus value and Poisson's ratio are chosen, a set of strain vectors are generated with randomly selected values over a small range. The corresponding stress vectors are computed using the plane-stress equation. Theoretically, any value of Young's modulus and Poisson's ratio can be selected for pretraining, although, as one might expect, accurate initializations result in faster convergence and avoids non-physical behavior in early FEA iterations. The stress scaling value and all spatial scaling vectors are set to one ($\boldsymbol{S}^{\sigma} = 1, \boldsymbol{S}^{\varepsilon}_{\boldsymbol{x}} = \mathbf{1}$).

[1] Estimation of stresses and strains Stage [1] consists of estimating stresses and strains from measurement data. Forces for the nth load increment are applied to the FE model in FEA$_{\sigma}$. Referring back to (5.5), force measurements are imposed as BCs in $\boldsymbol{P}^n$ and total mesh deformation is computed. Due to equilibrium conditions relating stresses to forces, all stresses $\boldsymbol{\sigma}^n(\boldsymbol{x})$ computed throughout the model in FEA$_{\sigma}$ are assumed to be physically consistent estimates of the true stress. Similarly, in FEA$_{\varepsilon}$ US probe and internal displacement measurements from the nth load increment are input as components of $\Delta\boldsymbol{U}^n$ to compute displacements of the remaining nodes. Due to compatibility requirements relating node displacements to strains, the strains $\boldsymbol{\varepsilon}^n(\boldsymbol{x})$ computed in FEA$_{\varepsilon}$ are assumed to be physically consistent estimates of the true strains. Recall that R^j_m and R^j_s are invoked by (5.6)–(5.7) to solve each FEA.

After estimating all stresses and strains, the stress scaling value $\boldsymbol{S}^{\sigma}$ is checked to ensure all scaled stresses fall within ±0.8. That is, we check for $\max(\boldsymbol{\sigma}^n(\boldsymbol{x})/\boldsymbol{S}^{\sigma}) > 0.8$, where the division is performed element-wise. If any component of $\boldsymbol{\sigma}^n(\boldsymbol{x})$ falls outside this range, $\boldsymbol{S}^{\sigma}$ is increased so that $\max(\boldsymbol{\sigma}^n(\boldsymbol{x})/\boldsymbol{S}^{\sigma}) = 0.8$.

[2] Training MPN A total of N_x stresses $\boldsymbol{\sigma}^n(\boldsymbol{x}_i)$ and strains $\boldsymbol{\varepsilon}^n(\boldsymbol{x}_i)$ are computed in FEA$_{\sigma}$ and FEA$_{\varepsilon}$, respectively. Each strain can by scaled by the corresponding $\boldsymbol{S}^{\varepsilon}_{\boldsymbol{x}_i}$ computed by R^j_s and input to R^j_m to compute a new stress $\hat{\boldsymbol{\sigma}}^n(\boldsymbol{x}_i)$. The goal of stage [2] is to adjust the weights of the MPN to minimize the error between $\boldsymbol{\sigma}^n(\boldsymbol{x})$ and $\hat{\boldsymbol{\sigma}}^n(\boldsymbol{x})$:

$$R_m^{j+1} = \tag{5.8}$$
$$\underset{R_{\theta_m}\in\mathbb{R}}{\operatorname{argmin}} \sum_{i=1}^{N_x}\sum_{n=1}^{N_p}\sum_{l=1}^{N_\sigma} \mathrm{f_m}(\boldsymbol{\sigma}_l^n(\boldsymbol{x}_i), \overbrace{R_m^j\{\boldsymbol{\varepsilon}_l^n(\boldsymbol{x}_i), R_s^j(\boldsymbol{x}_i)\}}^{\hat{\boldsymbol{\sigma}}_l^n(\boldsymbol{x}_i)}).$$

$R_s(\boldsymbol{x}_i)$ is the output of the SN at $\boldsymbol{x}_i$ and N_σ is the number of stress-strains pairs at $\boldsymbol{x}_i$ in the nth load increment. This value is greater than one when frame-invariance is enforced or a training window is implemented, both of which are described in Appendix 5.2.7. $\mathrm{f_m}(\cdot)$ is the L_2 norm of the difference between $\boldsymbol{\sigma}^n(\boldsymbol{x})$ and $\hat{\boldsymbol{\sigma}}^n(\boldsymbol{x})$; i.e., the MPN is trained via backpropagation.

Eq. (5.8) can not be minimized to zero for a heterogeneous material given the current MPN architecture. As previously stated, the MPN accepts a single strain vector as input and responds with a single stress vector as output. There is a many-to-many mapping from $\boldsymbol{\varepsilon}^n(\boldsymbol{x})$ to $\boldsymbol{\sigma}^n(\boldsymbol{x})$ in heterogeneous materials. For example, in the case where $\boldsymbol{\varepsilon}^n(\boldsymbol{x}_i) = \boldsymbol{\varepsilon}^n(\boldsymbol{x}_j)$, it is not necessarily true that $\boldsymbol{\sigma}^n(\boldsymbol{x}_i) = \boldsymbol{\sigma}^n(\boldsymbol{x}_j)$ (i.e., $R_m : \boldsymbol{\varepsilon} \to \boldsymbol{\sigma}$ is not bijective). Thus, the SN must supply additional information in the form of $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ so that the MPN can determine which stress should be returned for a given strain.

[3] Spatial scaling calculation The spatial values $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ are computed in stage [3]. While the coordinates input to the SN are given by the FE mesh, the target spatial values must be determined based on R_m^{j+1}, $\boldsymbol{\sigma}^n(\boldsymbol{x})$, and $\boldsymbol{\varepsilon}^n(\boldsymbol{x})$. The goal is to further minimize the error between $\boldsymbol{\sigma}^n(\boldsymbol{x})$ and $\hat{\boldsymbol{\sigma}}^n(\boldsymbol{x})$ by altering the spatial values instead of the weights of the MPN:

$$\boldsymbol{S}_{\boldsymbol{x}_i}^{\varepsilon} = \underset{\hat{\boldsymbol{S}}_{\boldsymbol{x}_i}^{\varepsilon}\in\mathbb{R}}{\operatorname{argmin}} \sum_{n=1}^{N_p}\sum_{l=1}^{N_\sigma} \mathrm{f_m}(\boldsymbol{\sigma}_l^n(\boldsymbol{x}_i), R_m^{j+1}\{\boldsymbol{\varepsilon}_l^n(\boldsymbol{x}_i), \hat{\boldsymbol{S}}_{\boldsymbol{x}_i}^{\varepsilon}\}), \tag{5.9}$$

where R_m^{j+1} is the output of the MPN retrained in (5.8).

[4] Training SN With a complete set of training data, the SN is trained in stage [4] via backpropagation:

$$R_s^{j+1} = \underset{R_{\theta_s}\in\mathbb{R}}{\operatorname{argmin}} \sum_{n=1}^{N_p}\sum_{i=1}^{N_x} \mathrm{f_s}(\boldsymbol{S}_{\boldsymbol{x}_i}^{\varepsilon}, R_s^j\{\boldsymbol{x}_i\}). \tag{5.10}$$

Details of solving (5.9) and (5.10) are covered in Appendix F.

[5] Convergence check Finally, a convergence check is performed in stage [5] to determine if training iterations for the current step should continue. Node displacements $\boldsymbol{u}_k^n$ computed in FEA$_\sigma$ are compared to the measured displacements $\hat{\boldsymbol{u}}_k^n$ using the L_1 norm of their difference:

$$\begin{aligned}\Delta \boldsymbol{u}_k^n &= |\boldsymbol{u}_k^n - \hat{\boldsymbol{u}}_k^n| \\ &= \mathrm{f_u}(\boldsymbol{u}_k^n\{R_{\theta_m}, R_{\theta_s}\}, \hat{\boldsymbol{u}}_k^n)\end{aligned} \tag{5.11}$$

which is the objective function defined in (5.2). We only use axial displacements in this study, although lateral and/or elevational displacements can also be used if available.

Following previous implementations of AutoP to determine NNCM convergence, displacement errors are used to compute two new values:

$$c_{\max}^n = \frac{\max(\Delta \boldsymbol{u}_k^n)}{|\max(\boldsymbol{u}_k^n)|} \tag{5.12}$$

$$c_\mu^n = \frac{\mathrm{mean}(\Delta \boldsymbol{u}_k^n)}{|\max(\boldsymbol{u}_k^n)|}. \tag{5.13}$$

We define convergence criteria $C_{\max}^n$ and C_μ^n for the nth training step. If both $c_{\max} < C_{\max}^n$ and $c_\mu < C_\mu^n$, convergence has been achieved and AutoP training iterations stop for the current training step. Otherwise, iterations consisting of stages [1]–[5] continue until the convergence criteria are met or a maximum number of iterations are reached. Training steps continue for each increment of force-displacement data in the set to complete a pass. Multiple passes are typically needed to ensure the CaNNCM has fully learned the material properties. A CaNNCM is not expected to accurately model the material properties in the first few passes. We therefore begin AutoP with relaxed convergence criteria (larger values of $C_{\max}^n$ and C_μ^n) that gradually become more restrictive.

5.2.2 Measurements from Linear-Elastic Phantoms

An imaging phantom was constructed from a mixture of deionized water, gelatin powder, and cornstarch acting as a scattering agent. The phantom was comprised of a $50 \times 50 \times 50\text{mm}^3$ cube of a soft background gel ($\approx 7.15 \pm 0.18$ kPa, 8% gelatin by mass) with three stiff, cylindrical inclusions. Each inclusion was 10 mm in diameter and 50 mm long. Mechanical contrast was controlled by the ratio of gelatin to water and each inclusion was a different stiffness ($\approx 10.93\pm0.57$, 14.15 ± 0.71, or 20.51 ± 0.84 kPa, 10%, 12%, and 14% gelatin by mass, respectively). The phantom was manufactured in the same manner described in a previous report [108] and separate samples of each gelatin mixture were stored to independently estimate Young's modulus values via macro-indentation methods [121].

We used the same experimental setup described in [108]. Compressive loads were applied to the phantom over four equal axial displacement increments of the US probe. Total probe motion was 1.44 mm, 3% of the pre-loaded phantom height. After applying each load increment, axial force and probe position were measured and an RF echo frame was acquired. The speckle-tracking algorithm GLUE [38] was applied to the echo data to estimate axial displacements in the field of view. Axial forces, probe displacements, and internal displacements over all four load increments constitute one set of force-displacement data. Fig. 5.3 shows six different data sets collected. Sets 1–3 were all acquired by compressing downward onto the top surface but with different lateral placements of the US probe. Sets 4–6 were acquired by keeping the probe centered laterally, but rotating the phantom 90°, 180°, and 270° around x_3, respectively. Two additional data sets were acquired where the imaged regions was centered on the x_2 axis like Data Set 1, but the probe moved in elevation, along x_3, by ±4mm. We refer to these data sets as Data Set $1'$ and $1''$, respectively. Total forces applied by the US probe ranged from 17.95 to 21.22 mN.

5.2.3 Finite Element Model

A simple FE model was created for use in AutoP and to generate the noise-free data sets described in the next section. The FE model FEM-1 is a 2-D, rectilinear FE mesh with 50 nodes per edge (Fig. 5.4) to represent the phantom. The ultrasound probe was modeled as a rigid body in frictionless contact with the top surface of the phantom model, approximating the condition created by the US gel. The bottom nodes of the phantom mesh were

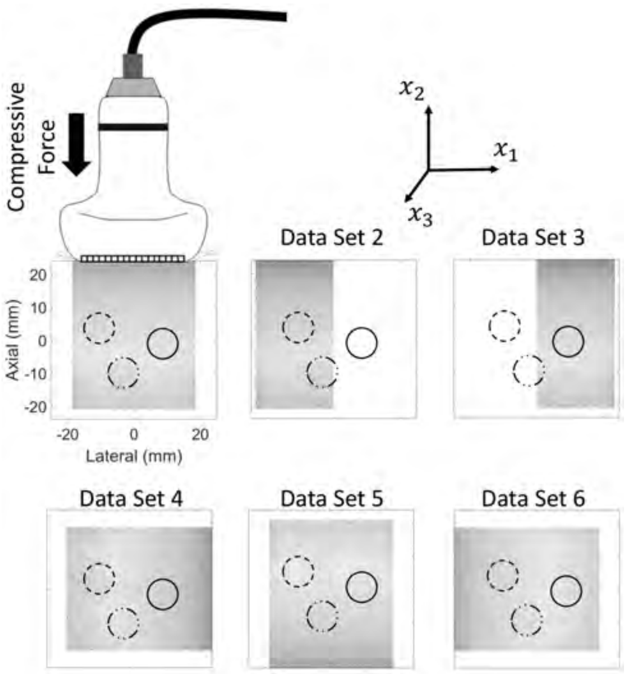


Figure 5.3: The cubic phantom was imaged from four sides. Data sets 1–3 were obtained with the US probe along the same surface but at different lateral positions. For sets 4, 5, and 6, the phantom was rotated 90°, 180° and 270° about x_3, respectively, while keeping the probe laterally centered.

fixed to mimic contact between the gelatin phantom and rubber pad. The full mesh was composed of 2516 nodes (5032 DOF) and 2401 plane-stress elements (CPS4 in Abaqus 6.13). Given that 4-node quadrilateral elements contain four integration points, a total of 9604 stress-strain pairs are computed in each of two FEAs, labeled FEA$_\sigma$ and FEA$_\varepsilon$, which are described in Sect. 5.2.1. Force loads, when applied as boundary conditions (BCs) in FEA$_\sigma$ or in a forward problem, were defined as concentrated forces to the top of the probe model. Similarly, probe displacements in FEA$_\varepsilon$ were defined as BCs for the entire probe model. Note that FEM-1 refers to the mesh and method of applying BCs. All FEAs were solved with ABAQUS 6.13 commercial finite element software.

5.2.4 Simulated Force-displacement Measurements

First, we tested AutoP employing CaNNCMs and noise-free force and displacement data. Three different material property distributions (Figs. 5.5a-c) were created to generate simulated measurements. Model 1 consists of a stiff Gaussian-shaped inclusion with a peak stiffness of 30 kPa embedded in

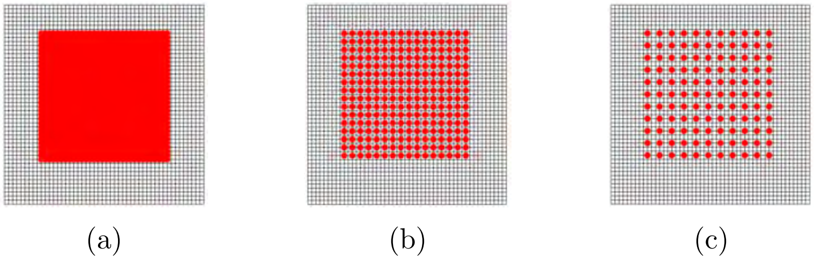

(a) (b) (c)

Figure 5.4: Rectilinear meshed used in FEM-1. Highlighted nodes indicate locations where displacement data was provided in FEA$_\varepsilon$. (a) Displacements are given at every node in the ROI, (b) at nodes separated by a minimum of 1.5mm, or (c) at nodes with a minimum 3mm separation.

the center of a 10 kPa background. Model 2 was a replicate of the gelatin phantom described in Sect. 5.2.2. Young's modulus values for the background and three inclusions were selected to be the same as those estimated via macro-indentation for each gelatin material. Model 3 corresponded to a rabbit kidney embedded in a block of gelatin with background Young's modulus 5.61 kPa. Modulus values for the kidney were based on previous results using AutoP and linear-elastic MPNs [137]. Models 1 and 3 were chosen to enable comparison with results reported Chapter 4 where stress-strain data were known.

Force-displacement data were generated by solving a forward FEA using FEM-1 and the target Young's modulus distributions of Models 1–3. The same forces and loading geometry of Data Set 1 were applied to the model and displacements were computed at all nodes. Each simulated data set contained noise-free data over four load increments.

5.2.5 Simulated RF Images

Simulated RF echo frames were generated to test the capabilities CaNNCMs in the presence of noise. A detailed description of the method used to generate these images is provided in Appendix E. The two simulated sets were intended to emulate Data Set 1 with different SNR in the RF echo data: one with 30dB SNR and one with 15dB SNR. We refer to the former as Data Set $1^{\dagger}$ and the latter as Set $1^{\dagger\dagger}$.

5.2.6 AutoP Analyses

CaNNCMs were trained in AutoP using force-displacement data obtained in three ways. First, forward FE modeling simulated noise-free displacements at each node in the FE mesh. Second, the same simulated displacements were entered into an RF echo simulator to simulate noisy experimental data. Both data sets have exactly-known displacements and material properties. Third, RF echo signals were recorded experimentally from phantoms. All tests used the same training parameters described in the next section. Any differences in how CaNNCMs were trained lie in how the force-displacement data were sampled. We will show in Sect. 3.3 that changes to the input data do not imply a need to alter training parameters.

Several different training cases were used that differed in the number of axial displacements applied in FEA$_\varepsilon$. Table 5.1 summarizes these four cases. For case 1, displacements are given at every node in the mesh. Cases 2–4 only used displacements in the ROI. The ROI is the region corresponding to the size of the displacement image after speckle-tracking was applied to the RF frames. Case 2 (Fig. 5.4a) indicates all nodes in the ROI were assigned displacements. For Cases 3 and 4 (Figs. 5.4b and c, respectively), axial displacements were only given at nodes with a minimum separation of 1.5 mm and 3 mm, respectively. These are the variable sampling settings selected to explore the role of displacement sampling in AutoP convergence.

Table 5.1: Four cases for applying axial displacements in AutoP. Cases 2–4 are illustrated in Fig. 5.4.

case 1	All nodes in mesh
case 2	All nodes in ROI < mesh size
case 3	Nodes in ROI, 1.5mm minimum separation
case 4	Nodes in ROI, 3mm minimum separation

Upon completion of AutoP, each CaNNCM was used to reconstruct a map of the Young's modulus distribution. Image reconstruction was performed by setting a constant strain vector $\boldsymbol{\varepsilon} = [0.003\ 0.005\ 0.0001]$ and computing the stiffness matrix $\hat{D}_{ij}$ using (D.2) in Appendix D. The Young's modulus distribution $E(\boldsymbol{x})$ was then estimated by varying $\boldsymbol{x}$ in the domain of the mesh and evaluating the function

$$E(\boldsymbol{x}) = \frac{S^{\sigma}}{S_x^{\varepsilon_2}} \hat{D}_{22}(1 - \nu^2), \tag{5.14}$$

where $\nu = 0.5$ and $S_x^{\varepsilon_2}$ is the axial component of the spatial scaling vector at $\boldsymbol{x}_i$. The choice of constant strain vector is not important so long as it resides within the range of training data. We chose small values for each component to ensure the strain was within range, and we emphasized the axial strain and used D_{22} in the modulus estimate because the models were axially compressed.

Young's modulus distributions estimated by the CaNNCMs were compared to the target maps shown in the top row of Fig. 5.5. Errors were computed as

$$e_{\boldsymbol{x}}^{E} = \frac{|E_{\boldsymbol{x}}^{target} - E_{\boldsymbol{x}}^{NN}|}{E_{\boldsymbol{x}}^{target}} \tag{5.15}$$

where $E_{\boldsymbol{x}}^{target}$ is the target Young's modulus distribution and $E_{\boldsymbol{x}}^{NN}$ is the CaNNCM estimate. Because displacements are only provided in the field of view for Cases 2–3, we do not expect the CaNNCM to accurately estimate Young's modulus values outside of the ROI where no displacement measurements are acquired. We therefore compute $e_{\boldsymbol{x}}^{E}$ only within the ROI for all cases.

The following describes each of the CaNNCMs trained.

Simulated Force-Displacement Data A total of six CaNNCMs were trained in AutoP using noise-free force-displacement data generated from the three simulated models. One network was trained for each model using Case 1 and Case 2 displacement sampling. The results from training these networks demonstrate the ability of CaNNCMs to learn material properties when the sampling space is reduced.

Simulated RF Echo Data Another six CaNNCMs were trained using force-displacement data gathered from the simulated RF echo frames with varying amounts of echo noise. Three CaNNCMs for Data Set 1† and three for Set 1†† using Cases 2–4 sampling distributions. Results from these analyses demonstrate how reducing the number of sampling points affects the ability of CaNNCMs to learn material properties and geometry in the presence of noise.

Gelatin Phantom We trained 12 CaNNCMs with experimentally measurement force-displacement data. The first three were trained with Data Set 1 and sampling Cases 2–4. Results obtained from these CaNNCMs and those trained with data acquired via the simulated RF data guided the choice of sampling for the remaining CaNNCMs. Using Case 3 sampling, one CaNNCM was trained with each of Data Sets 2–6, $1'$, and $1''$, one with Sets 1–3 simultaneously, and one with Sets 1, $1'$, and $1''$ simultaneously. Results from these CaNNCMs demonstrate the ability of CaNNCMs to learn material properties under different loading scenarios, how the inclusion of multiple independent data sets affects the Young's modulus estimates, and how sparser sampling influences the learned material and geometric properties.

5.2.7 AutoP Training Parameters

Each CaNNCM was trained over 10 passes for each data set based on experience. A Young's modulus value of 5 kPa and a strain range of ±0.01 was selected for linear-elastic pre-training. FEA_σ and FEA_ε were computed by applying loads to FEM-1 as described in Sect. II-C.

A four-load training "window" was also incorporated which includes stress-strain data from prior training steps in stages [2]–[4] of the current AutoP iteration. Fig. 2a illustrates the example where training step 3 of pass 3 would also include stresses and strains from training steps 1 and 2 of pass 3 and training steps 3 and 4 of pass 2.

Furthermore, *frame invariance* was enforced by rotating the stress-strain data 90° and appending the rotated pairs to the original set, effectively doubling the number of stress-strain pairs. Any rotation angle could be used, but we chose 90° because it is easily implemented by swapping the normal components of the data, as we described in our initial report of AutoP [108]. Building frame invariance into the training data means the material properties learned by the CaNNCM are independent of the chosen coordinate system. With the given training window and enforcement of frame invariance, a total of 19208 stress-strain pairs are used to train the MPN ($N_x \times N_p \times N_\sigma = 19208$ in (6) starting in the second pass.

Spatial scaling values were computed using Algorithm 2 described in Appendix F ($mnalg = 50$, $N_\varepsilon = 8$ due to frame invariance and training window, spatial scaling update rate $\eta_\varepsilon = 0.5$). The MPN had two hidden layers of six nodes each, whereas the SN had five hidden layers with ten nodes each. The MPN was trained using the resilient propagation (RPROP) algorithm [123] over 15 epochs. Conversely, the SN training was implemented in TensorFlow using the ADAM optimizer [140] (with default parameter settings) and a learning rate of 0.03.

Convergence criteria changed as training progressed. Convergence criteria were initialized as $(0.65, 0.5)$, using the notation $(C^n_{\max}, C^n_\mu)$. These reduced to $(0.4, 0.3)$, $(0.3, 0.2)$, and $(0.2, 0.01)$ at the beginning of passes 2, 3, and 4, respectively. The last set of criteria were also used in passes 5–10. We chose to set a limit of two AutoP iterations per training step, regardless of whether convergence criteria were satisfied. An upper limit ensures that iterations do not continue indefinitely. We chose a maximum of two based on preliminary results.

5.3 Results

Young's modulus images reconstructed by CaNNCMs trained with noise-free force-displacement data are displayed in Fig. 5.5. Errors in the modulus estimates computed with (5.15) are compiled in Table 5.2 for all CaNNCMs. Also included in the table are the processing times and total number of AutoP training iterations performed. For all three Models, reconstruction error increased slightly when the displacement sampling was reduced to the ROI only (i.e., from Case 1 to Case 2). CaNNCMs trained for Model 1 required the most number of AutoP iterations and were unable to capture the peak stiffness of the inclusion. We observed this type of behavior in Chapter 4 for the same Model. It could be corrected by increasing the number of iterations performed when solving (5.9). For example, increasing the number of spatial scaling update iterations from 50 to 150 for Model 1, Case 2, the peak stiffness estimated by the CaNNCM is ≈ 25 kPa (actual is 30 kPa) and the modulus reconstruction error decreases to 0.1291 ± 0.0753. However, preliminary results showed that in the presence of noise, using a large number of iterations could result in overfitting thus magnifying the influence of noise in the Young's modulus reconstruction. It is possible that the addition of a regularization term in (5.9) could reduce the sensitivity to the number of iterations and noise at the cost of increased computational complexity.

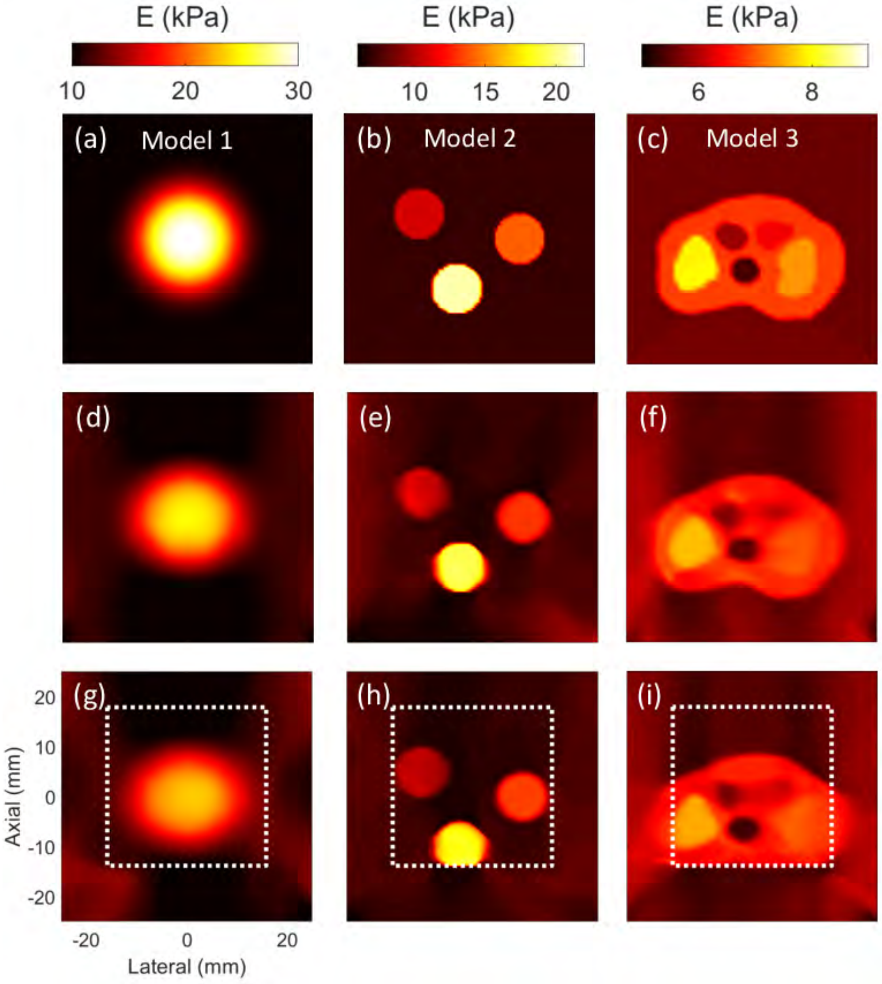


Figure 5.5: (Top Row) Target Young's modulus distributions for Models 1–3. Young's modulus reconstructions from CaNNCMs trained with noise-free simulated displacements for Case 1 (Middle row) and Case 2 (Bottom row). The white box indicates the boundaries of the ROI. Color scales at the top of the figure apply to all images within the column.

Fig. 5.6 displays the Young's modulus images reconstructed by CaNNCMs trained with force-displacement data from Data Sets 1, $1^{\dagger}$, and $1^{\dagger\dagger}$. Across the columns left to right, the images correspond to Cases 2, 3, and 4, respectively. We observe that the smallest error occurs for Set $1^{\dagger\dagger}$, Case 3, albeit said CaNNCM required the largest number of AutoP training iterations. We also note that, for the CaNNCMs trained with Set 1, there is a trade-off between reconstruction error and artifacts in the images. For example, Fig. 5.6g displays the Young's modulus image with the smallest error for the row, corresponding to Case 2. The error slightly increases for Case 3 (Fig. 5.6h), but fewer noise artifacts are present. Generally, the trend appears to be that

Table 5.2: Young's modulus reconstruction errors and AutoP run time. Models are illustrated in Figs. 5.5a-c. Sets are illustrated in Fig. 5.3. Cases are described in Table 5.1. A superscript indicates subfigure in Fig. 5.8. The last column indicates the figure number of the corresponding Young's modulus image.

(Model #){Set}[case]	Modulus Error Mean ± STD	Time (min.)	Iters.	Fig.
Simulated Force-Displacement				
(1){1}[1]	0.1055 ± 0.0545	151	74	5d
(1){1}[2]	0.1349 ± 0.0670	153	75	5g
(2){1}[1]	0.0621 ± 0.0730	88	42	5e
(2){1}[2]	0.0643 ± 0.0690	82	46	5h
(3){1}[1]	0.0306 ± 0.0188	83	40	5f
(3){1}[2]	0.0370 ± 0.0211	80	40	5i
Simulated RF				
(2){1†}[2]	0.0961 ± 0.1240	80	40	6a
(2){1†}[3]	0.1325 ± 0.1294	79	40	6b
(2){1†}[4]	0.0970 ± 0.1126	82	42	6c
(2){1††}[2]	0.0914 ± 0.1336	80	40	6d
(2){1††}[3]$^{(a)}$	0.0899 ± 0.1111	110	56	6e
(2){1††}[4]	0.1386 ± 0.1313	90	46	6f
Gelatin Phantom				
(2){1}[2]	0.2136 ± 0.1264	139	69	6g
(2){1}[3]$^{(b)}$	0.2736 ± 0.1563	136	69	6h
(2){1}[4]	0.3168 ± 0.1784	136	69	6i
(2){$1'$}[3]	0.2604 ± 0.1576	139	69	7a
(2){$1''$}[3]	0.2522 ± 0.1480	138	69	7b
(2){2}[3]	0.1828 ± 0.1175	138	69	7d
(2){3}[3]	0.3415 ± 0.1606	137	69	7e
(2){4}[3]$^{(d)}$	0.2549 ± 0.1645	138	69	7g
(2){5}[3]	0.3208 ± 0.1905	127	64	7h
(2){6}[3]	0.2887 ± 0.1767	137	69	7i
(2){1,$1'$,$1''$}[3]	0.4380 ± 0.2163	393	69	7c
(2){1,2,3}[3]$^{(c)}$	0.3228 ± 0.1713	393	69	7f
Gelatin Phantom, $\nu = 0.3$ Pretraining				
(2){1}[3]	0.2546 ± 0.1486	147	40	
(2){4}[3]	0.1975 ± 0.1479	146	70	
(2){5}[3]	0.2862 ± 0.1810	131	64	
(2){6}[3]	0.2388 ± 0.1582	140	69	

the effect of noise increases as the displacement sampling density increases. The influence of noise can be decreased by reducing sampling density at the cost of resolution and reconstruction accuracy. Note that this applies to the case where only a single data set is used during training. For these reasons, we chose Case 3 sampling for training the remainder of the CaNNCMs with the experimental measurement data.

Interestingly, a stiffening artifact appears between the top of the ROI and phantom surface in the images reconstructed by CaNNCMs trained with experimentally measured force-displacement data (Figs.5.6g–i), but not when trained with Data Sets $1^{\dagger}$ or $1^{\dagger\dagger}$ (Figs. 5.6a–f). It is likely due to noise occurring in both the force and displacement measurements. Displacements estimated from the simulated RF frames contain noise, but the force measurements are exact. Furthermore, displacements imposed when generating the simulated RF data were obtained from a 2-D FEA whereas the experimental measurements are a 2-D approximation of a 3-D object.

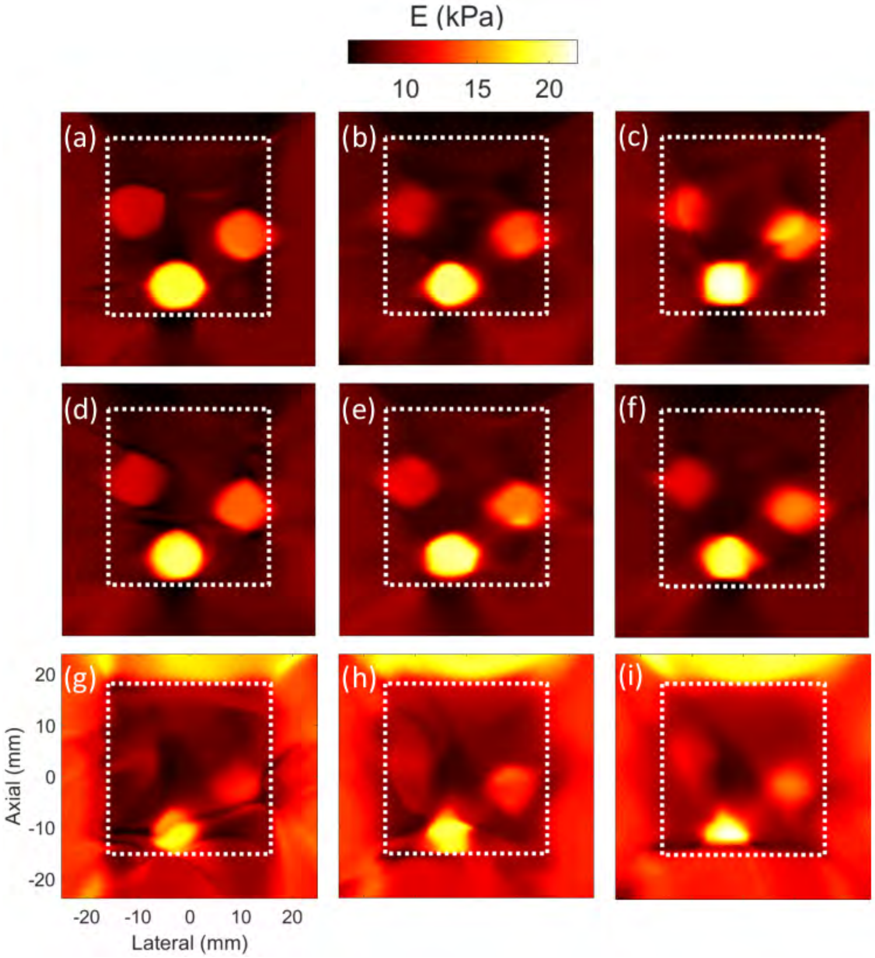


Figure 5.6: Comparisons of Young's modulus image reconstructions from RF echo signals for three levels of noise and different displacement sampling densities. Force-displacement measurements from rows 1 and 2 are from simulated echo data at 30 and 15 dB SNR. Row 3 is from experimental phantom measurements with echo SNR in the same range using Data Set 1.

Young's modulus reconstructions by CaNNCMs trained with experimental measurement data are shown in Fig. 5.7. We indicate in the bottom-right of each image the Data Set(s) used for training in AutoP. As expected, the Young's modulus estimates are most accurate within the ROI. Reducing the size of the ROI (Figs. 5.7d,e) did not inhibit the ability of the CaNNCM to learn the correct material properties. The exception is the inclusion at the bottom of the ROI in Fig. 5.7d, where said inclusion is only partially within view. Material properties estimated by each CaNNCM are consistent, barring Figs. 5.7g and 5.7i. These correspond to Data Sets 4 and 6, where the phantom was rotated by 90° and 270°, respectively, before data acquisition. Both CaNNCMs learned the correct locations of all three inclusions, but the estimated Young's modulus of the two stiffest inclusions are inaccurate.

It is difficult to identify the source of the error. To determine if the issue was caused by the relative locations of the inclusions within the ROI, we created a simulated RF data set (using the same methods described previously) to mimic Data Set 4. A CaNNCM trained with these data (not shown) was able to accurately estimate modulus values for all three inclusions. Furthermore, if we compare displacement errors computed by (5.12) and (5.13) for CaNNCMs corresponding to Figs. 5.7a and 5.7g, there is no significant difference ($c^n_{\max} = 0.0796$, $c^n_{\mu} = 0.0694$ compared to 0.0812 and 0.0692, respectively). Meaning, the CaNNCMs are estimating material properties consistent with the data.

From a qualitative standpoint, including multiple data sets during training (Figs. 5.7c, f) improves the appearance of the reconstructed image. Contrary to our expectation, the reconstruction error increases when multiple data sets are used. To explore why this occurred, we generated images of the Young's modulus error by computing the difference between the target and reconstructed Young's modulus images. Error maps are displayed in Fig. 5.8 and the CaNNCMs are indicated in Table 4.1 by a superscript. The largest errors occur at the boundaries of the inclusions; most notably, for the stiffest inclusion located at the bottom of the ROI. These maps suggest the largest errors are due to CaNNCMs learning the incorrect geometry. Specifically, the stiffest inclusion appears too small, particularly for CaNNCMs trained with experimental data. Neglecting geometric errors, we observe that when multiple experimentally measured data sets are used in training (Fig. 5.8d), the CaNNCM more accurately estimates the Young's modulus of the inclusions. The increased error reported in Table 4.1 is mostly attributed to over-estimating the Young's modulus of the background gelatin, by up to

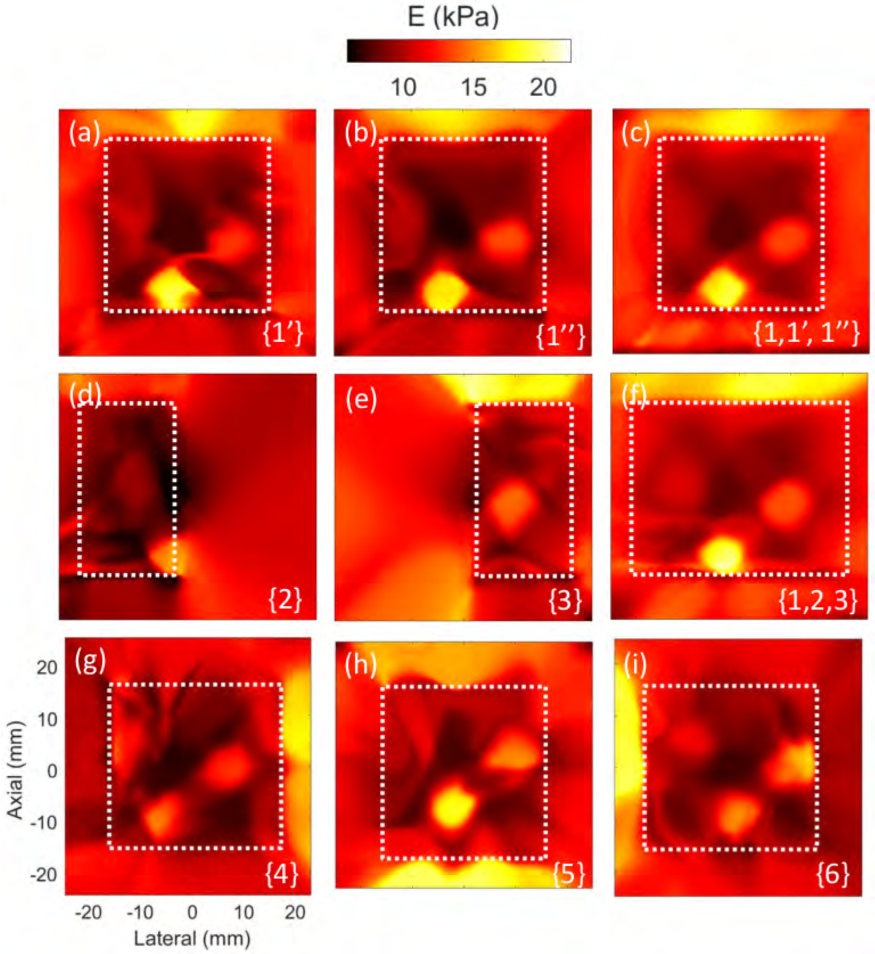


Figure 5.7: Young's modulus image reconstruction by CaNNCMs trained from experimental measurements. Bracketed numbers in the lower-right corner of each image indicates the Data Set(s) used to train the CaNNCM. The dotted frame indicates the region over which displacements were estimated.

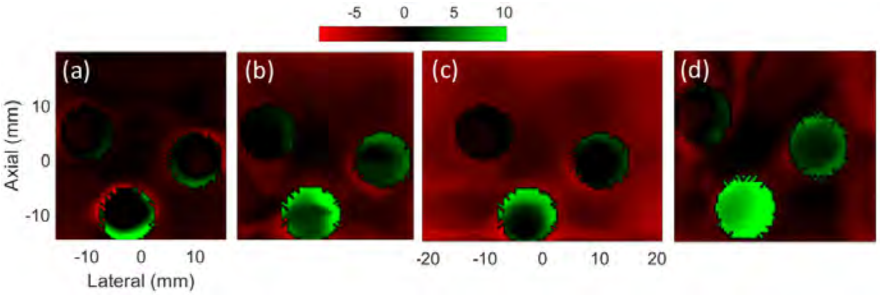
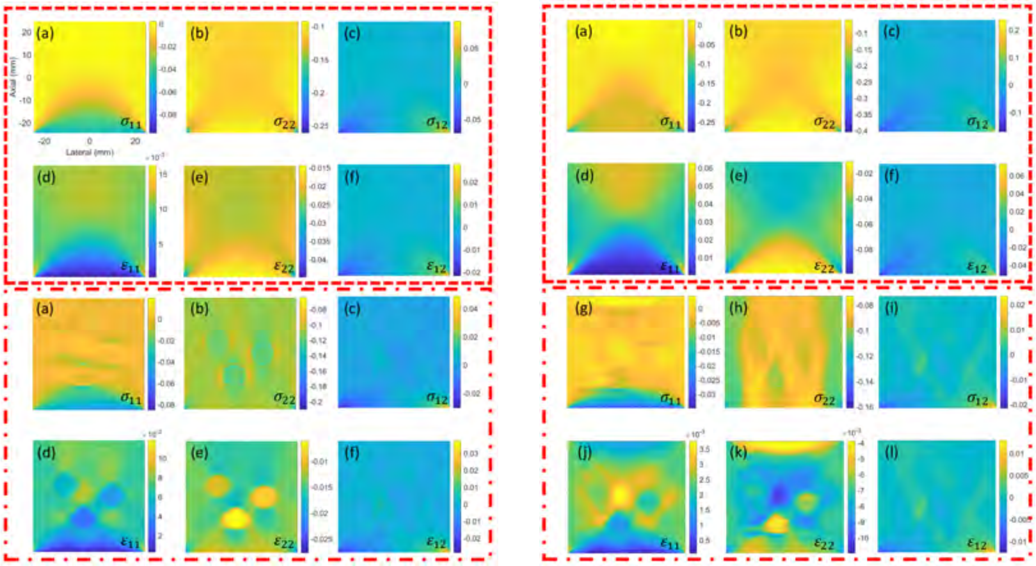


Figure 5.8: Maps of Young's modulus error for the four tests indicated in Table 4.1. The error is the difference between the target distribution and that computed by the CaNNCM. Red/green indicates the CaNNCM-estimated Young's modulus was too large/small.

5 kPa in the most extreme cases and mostly toward the boundary of the ROI. It is unclear at this time if errors are larger near the boundary because of artifacts introduced through speckle-tracking, the distance between the ROI

and phantom boundaries, or because we are limited to a 2-D approximation of a 3-D problem.

A point hinted at in the preceding results is the ability of CaNNCMs to not only estimate material parameters, but also estimate the spatial distribution of *all* stresses and strains without assuming a constitutive model. In Fig. 5.9, maps of the lateral, axial, and shear stresses and strains estimated by CaNNCMs after pretraining (top box) and training in AutoP (bottom box) are shown. The maps were estimated by applying solving appling a 4.6 mN force load to the US probe in FEM-1 and solving the FEA using the corresponding CaNNCM as the constitutive model in (5.6) and (5.7). To the best of our knowledge, this is the first time the spatial distribution of all stresses and strains have been estimated from force-displacement data without assuming the underlying constitutive model.



(a)

(b)

Figure 5.9: The top box in each figure are maps of the stresses and strains estimated by a CaNNCM pretrained as a 5 kPa, incompressible, linear-elastic material. Conversely, the bottom boxes are the stress and strain maps after CaNNCMs were trained in AutoP using (a) noise-free force-displacement data from Model 2 and (b) Data Set 1.

5.3.1 Pretraining with Incorrect Poisson's Ratio

Back in Section 4.3.1 of Chapter 4, the question was raised of what happens when the Poisson's ratio used for pretraining the MPN does not match the Poisson's ratio expressed by the stress-strain data. It was not surprising that the spatial values computed via Algorithm 2 contained artefacts and the resulting Young's modulus images were biased toward underestimating the true value. This is important because the MPN is pretrained at the beginning of AutoP, during which time a value for the Poisson's ratio must be chosen. I made the claim that it should not matter because the MPN is retrained in every AutoP iterations and thus should learn the Poisson's ratio encoded in the force-displacement data. That hypothesis is tested here.

Four CaNNCMs were trained with Data Sets 1, 4, 5, and 6. All AutoP training parameters were the same as described in Sec. 5.2.7, except a Poisson's ratio of 0.3 was selected for pretraining and the number of training epochs for the MPN was increased from 15 to 25. The rationale for the increase in training epochs being that the MPN has to "forget" the pretraining Poisson's ratio. Young's modulus images reconstructed by these CaNNCMs (using $\nu = 0.5$ in Eq. 5.14) are displayed in the top row of Fig. 5.10 and the corresponding error images are in the bottom row. Modulus estimation errors are also included in Table 5.2.

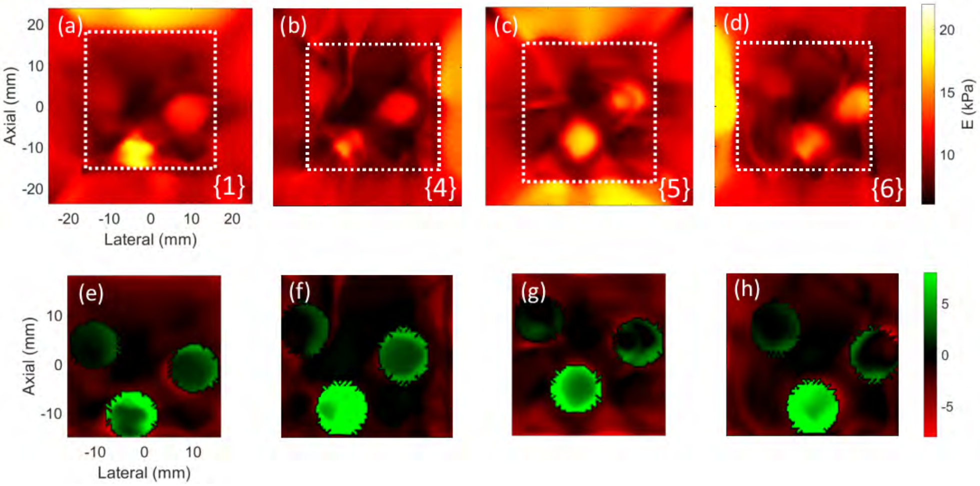


Figure 5.10: (a–d) Young's modulus images reconstructed by CaNNCMs pre-trained with $\nu = 0.3$. (e–h) Maps of the difference between target and estimated Young's modulus. Green indicates the CaNNCM underestimated the Young's modulus whereas red indicates overestimation.

From a qualitative standpoint, the Young's modulus images in Fig. 5.10 are comparable to their counterparts in Fig. 5.7. Similarly, the error images in Fig. 5.10 show similar characteristics to those in Fig. 5.8: the CaNNCMs most accurately estimate the Young's moduli of the background gelatin material and two softer inclusions. The largest errors occur for the stiffest inclusion at the bottom of th ROI. Interestingly, the modulus estimates are, on average, more accurate for the CaNNCMs pretrained as a compressible material. Increased accuracy in the modulus estimates is more likely due to the increased training epochs rather than the Poisson's ratio used for pretraining. Even so, these results suggest the CaNNCMs are not sensitive to this particular pretraining parameter.

5.3.2 Rabbit Kidney Embedded in Gelatin Cube

Following the above tests, we created another rabbit kidney phantom similar to that in Sec. 3.2.8. To recap, a rabbit kidney was suspended by sutures in the cubic mold while gelatin material was poured to fill the remaining space, resulting in the kidney embedded in a $50 \times 50 \times 50$ mm^3 block of gelatin. Force-displacement measurements were acquired on this phantom following the same procedure described in Sec. 5.2.2. We obtained three different data sets in the same manner as Sets 1, $1'$, and $1''$ for the three-inclusion phantom. That is, the US probe was laterally centered, but was at $x_3 = 0$, $x_3 = -4$, and $x_3 = +4$ mm for each of the data sets, respectively. B-mode images are displayed in Figs. 5.11a–c. Corresponding axial strain maps computed from GLUE are shown in Figs. 5.11d–f. Even though strain is only a qualitative measure of stiffness, these images imply the renal cortex is stiffer than the medulla, which in turn is stiffer than the background gelatin material.

Three different CaNNCMs were trained with the aforementioned Data Sets using Case 3 displacement sampling the AutoP parameters specified in Sec. 5.2.7. Young's modulus images reconstructed by these CaNNCMs are exhibited in Figs. 5.11g–i. Macro-indentation was performed on the background gelatin material only and suggests the Young's modulus to be 7.4931 ± 0.1234 kPa. Each Young's modulus image was manually segmented to obtain modulus estimates for the background gelatin, renal cortex, and medulla, all of which are compiled in Table 5.3. Included in the table are Young's modulus values estimated for the cortex and medulla via shear-wave elasticity imaging of healthy human volunteers [156].

Results in Table 5.3 indicate the CaNNCMs accurately learned the material properties of the background gelatin and produced Young's modulus estimates of the renal cortex and medulla in good agreement with those reported by other investigators, albeit we acquired data on rabbit kidney tissue,

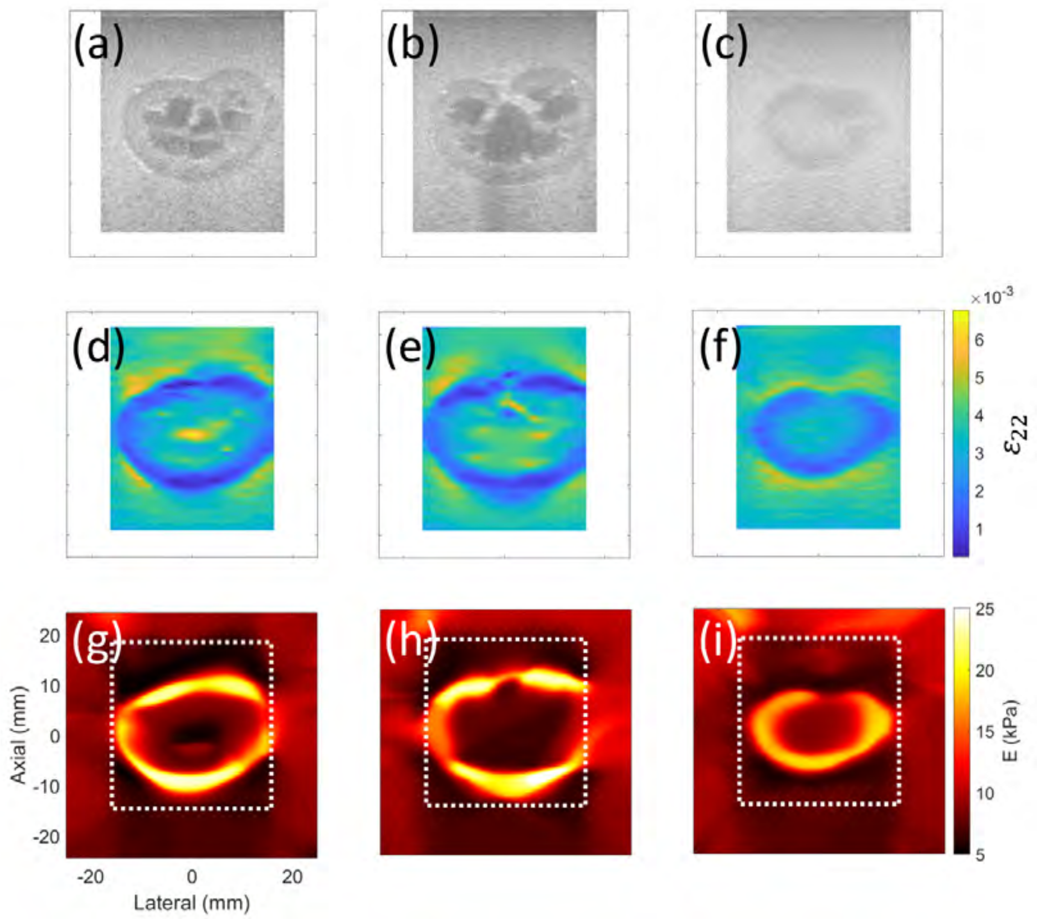


Figure 5.11: (a–c) B-mode images from Data Sets 1, 1′, and 1″ acquired on the rabbit kidney phantom, respectively. (d–f) Axial strain maps computed by the GLUE speckle-tracking algorithm. (g–i) Young's modulus images reconstructed by CaNNCMs.

Table 5.3: Young's modulus values estimated from the CaNNCMs trained with force-displacement measurements acquired on the kidney phantom. Also included are the modulus estimates of the renal cortex and medulla of humans *in vivo* reported by [156]. All values are in units of kPa.

Region	Data Set			Reported by [156]
	1	1′	1″	
Gelatin	7.1555 ± 1.3035	7.2778 ± 1.3560	7.6918 ± 0.9818	N/A
Cortex	19.7188 ± 4.0332	18.8844 ± 3.9255	16.3383 ± 2.1846	15.4 ± 2.5
Medulla	9.5340 ± 2.0592	8.3723 ± 1.8858	10.1893 ± 1.2774	10.8 ± 2.7

not humans. The images in Figs. 5.11g–i reveal much more interesting information about the Young's modulus distribution. As was expected based on the strain images, the renal cortex is stiffer than the medulla which is stiffer than the background gelatin. What is more, the top and bottom edges of the cortex appear stiffer than the rest, a feature present in both Figs. 5.11g–h. Such spatial variations in the stiffness could be an indicator of anisotropy. Similarly, the renal cortex in Fig. 5.11i exhibits a Young's modulus similar to that observed in Figs. 5.11g–i on the left and right edges. Once again, given the orientation of the kidney and where the image planes reside, this could be another indicator of anisotropy. Nevertheless, we observe consistent Young's modulus estimates for all three regions amongst the three CaNNCMs.

5.4 Discussion

We have implemented CaNNCMs in AutoP to build data-driven constitutive models that learn stress and strain profiles of linear-elastic materials from force-displacement data. Adjusting the NN architecture to learn both material property and geometric information expands the abilities of AutoP by removing any prior assumptions of material property distributions. Additionally, a single CaNNCM can model heterogeneous materials where previously multiple NNCMs would be necessary. Unlike networks used in prior work, CaNNCMs are able to model both discrete and continuous material property distributions regardless of the chosen mesh geometry. This is a critical step toward the use of AutoP in clinical imaging where accurately segmenting images is not always feasible nor possible.

QUSE acquisitions are well suited to data-driven modeling due to the enormous information content in each force-displacement data sample. Quasi-static loading gives the force stimulus time to propagate throughout the entire object before measurements are acquired. Each displacement therefore carries information of not only local material properties, but of the whole contiguous object. AutoP exploits this fact by using FEA to propagate a sparse sampling of force-displacement measurements throughout an entire object model for estimating stresses and strains. Several model-based inverse approaches also rely on FEA to compare computed and measured displacements while estimating the material parameter distribution that best fits the data (e.g., [86, 151, 157]). However, CaNNCMs trained with AutoP learn stress-strain behavior consistent with the measurement data without prior assumptions of material properties.

A trained CaNNCM can be related back to a known constitutive model to estimate material parameters *after* learning stress and strain profiles. We chose to estimate Young's modulus distribution via the stiffness matrix to demonstrate a capability of CaNNCMs not possible with model-based methods. When computing D_{22} in (5.14), there is still no assumption of the constitutive model. We effectively recovered the stiffness matrix *from the data*. Further development of CaNNCMs for non-linear and viscoelasticity imaging can make use of this ability to uncover the fundamental mechanical behavior governing the data, which may allow for discovery of the most relevant material parameters.

The additional task of learning the geometric shape of the medium requires a higher displacement sampling density compared to our previous report [108]. Our choice of simulated and experimental measurement data combined with the four displacement sampling Cases was intended to provide insight on the trade-off between sampling density, resolution, and modulus estimation accuracy. Results from CaNNCMs trained with noise-free force-displacement data suggest that restricting sampling to the ROI has a slight negative affect on the accuracy of reconstructed Young's modulus images. Data acquired from simulated RF frames and Data Set 1 of the experimental measurements better illustrate the trade-off in Fig. 5.6. In the presence of noise, dense sampling resulted in more accurate Young's modulus estimates, but artifacts due to noise become more apparent. Conversely, as sampling becomes increasingly sparse, noise artifacts are reduced at the cost of decreased resolution and accuracy of material parameter estimates. We observed this same behavior in several cases, although we provide only one example of Case 2 sampling with experimental measurements (Fig. 5.6g). That said, a cubic phantom with three parallel cylindrical inclusions limits the conclusions that can be drawn. Comparing Figs. 5.6g–h, the noise artifacts do not obstruct any of the inclusions and are thus not detrimental to the final image. However, it is possible that fine structures could be hidden in more complex media. Further investigation into the best sampling strategy will require data acquisition on an object with more complex geometry.

We expected CaNNCMs trained with multiple data sets to provide more accurate Young's modulus estimates. This was not the case. The argument can be made that qualitatively, Figs. 5.7c and 5.7f are improvements over training with any one data set: the inclusions are better resolved and fluctuations due to noise are reduced. It is unlikely that increasing the number of passes in AutoP would improve the results considering displacement errors ($c^n_{\max}$ and c^n_μ) computed for the convergence check do not continue to decrease by the end of pass 10. Some of the error can be attributed to the 2-D approximation of 3-D problem, which helps explain why Figs. 5.6a–f are much more accurate than Figs. 5.6g–i. Extending CaNNCMs to learn volumetric material properties will help us better understand how noise affects the material properties and geometry learned by the networks. We find the quality of Young's modulus images depends on the coupled effects of spatial sampling and noise, which are not the same as those seen in other applications of QUSE.

AutoP required a minimum of 79 minutes to complete on a quad-core CPU operating at 3.4 GHz. As we expand this method to model more complex behaviors in 2-D and 3-D, the run time will only increase. The two major bottle necks are the use of Abaqus as the FEA solver and training the SN. Abaqus is able to solve the FEAs in seconds, but requires much longer to perform a license check and pre-processing step. We can greatly increase the speed in this regard by creating a custom finite element solver that natively supports CaNNCMs. Other groups have shown the potential for enormous increases in FEA solution times using GPUs [126, 158, 159, 160]. We believe AutoP run time can be significantly decreased using custom software running on a single GPU, but that work is ongoing.

AutoP and CaNNCMs are a major departure from current model-based approaches to elasticity imaging. The latter assume the underlying constitutive model is constant but the associated mechanical parameters vary with position. In contrast, CaNNCMs trained in AutoP make no *a priori* assumptions of constitutive model. They are data-driven and therefore *learn the spatial distribution of material behaviors that best fit the data.* The current CaNNCM architecture is limited to linear-elastic materials, but prior implementations of AutoP suggest CaNNCMs can be developed that are able to model the heterogeneous distribution of linear, nonlinear, and viscoelastic behaviors, thus allowing for the exploration of diagnostically relevant imaging parameters.

5.5 Conclusion

Cartesian neural network constitutive models trained in the Autoprogressive method can learn spatially-varying linear-elastic material properties from force-displacement measurements. Young's modulus images can be reconstructed by relating the stress-strain behavior learned by CaNNCMs to a chosen constitutive model after training. CaNNCMs are robust to measurement noise and can model the internal structure of both discrete and continuous material property distributions. Further development of this method will offer a data-driven approach to imaging nonlinear and viscoelastic properties of soft tissues in 3-D.

Chapter 6

3-D CaNNCMs

6.1 Introduction

Chapters 4 and 5 demonstrated the capabilities of CaNNCMs for imaging the linear-elastic properties of tissues in two dimensions. The clinical utility of these networks may be limited because they do not reveal the full 3-D geometry of internal structures. For example, the circular inclusions in Model 2 are known to be cylindrical, but the cross-section does not preclude the possibility that one or more of the inclusions are spherical (or any geometry that has a circular cross-section). Extending elasticity imaging to three spatial dimensions will allow for estimating the material properties throughout a volume, eliminating the need for a sonographer to mentally "stitch" together elasticity images to form a 3-D representation as well as allowing for more accurate estimates of lesion volume [161]. Even if the goal is to only estimate material properties within a plane, the force-displacement measurements are acquired on a 3-D object. Imposing a 2-D plane-stress approximation to solve FEA_σ and FEA_ε may introduce errors in the computed stress and strain fields. Eliminating the plane-stress assumption and solving full 3-D analyses may reduce some of the errors observed in Chapter 5.

Lindop *et al.* proposed a 3-D QUSE method using a linear US array [162]. RF echo frames were recorded as the US probe was swept over the tissue surface by a human operator. The frame-rate was set so that there was significant overlap between successive echo frames, reducing the decorrelation due to the elevational motion. Speckle-tracking methods operating on successive RF frames provided estimates of axial displacements, from which axial strain is easily computed. Axial strains through the volume are thus computed by estimating strain in multiple parallel planes and then stacking the 2-D strain images to generate a 3-D data set.

Speckle-tracking algorithms operating on data sets produced by more sophisticated 3-D QUSE methods — which acquire volumetric RF echo frames — are used to estimate three-dimensional displacements. Treece *et al.* [163] and Housden *et al.* [164] used a mechanically swept US probe to acquire pre- and post-compression RF volumes. Others have investigated the use of automated breast volume scanners (ABVS) for data collection [165, 166]. Richards *et al.* [97] gathered RF volumes in a similar manner by scanning a robotically controlled linear array across a sample compressed by two plates (first discussed in Sec. 1.2). Others used the 3-D imaging capabilities of 2-D US arrays to volumetric RF data [167]. Regardless of the data acquisition method, the results are full 3-D displacement estimates within a section of a body. These data can be used to compute strains or input to an inverse problem to estimate material parameters.

In this chapter, we introduce 3-D CaNNCMs and present the first implementation of these networks in AutoP. Our primary goal is to investigate the spatial sampling requirements to accurately reconstruct the Young's modulus within a volume. We present results of CaNNCMs trained with both simulated and experimentally acquired force-displacement measurements on several linear-elastic gelatin phantoms. Furthermore, be begin investigations into the feasibility of free-hand QUSE with CaNNCMs and AutoP by reformulating the problem to account for large deforming forces and unknown surface geometry.

6.2 Methods

The CaNNCM architecture must be modified to incorporate 3-D information. First, the number of components in the stress and strain vectors increases from three (in plane-stress) to six: $\boldsymbol{\sigma} = [\sigma_{11}\ \sigma_{22}\ \sigma_{33}\ \sigma_{12}\ \sigma_{13}\ \sigma_{23}]^{\mathrm{T}}$ (similarly for $\boldsymbol{\varepsilon}$ and $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$). Therefore, the MPN must have six nodes at the input and output, and the SN must have six outputs. Second, the SN must have three input nodes to accommodate the three spatial dimensions. Fig. 6.1 illustrates the 3-D CaNNCM architecture.

Although the number of spatial dimensions has increased and the CaNNCM has been modified, no changes need be made to the stages of AutoP as described in Sec. 5.2.1. Nor do any modifications need to be made to

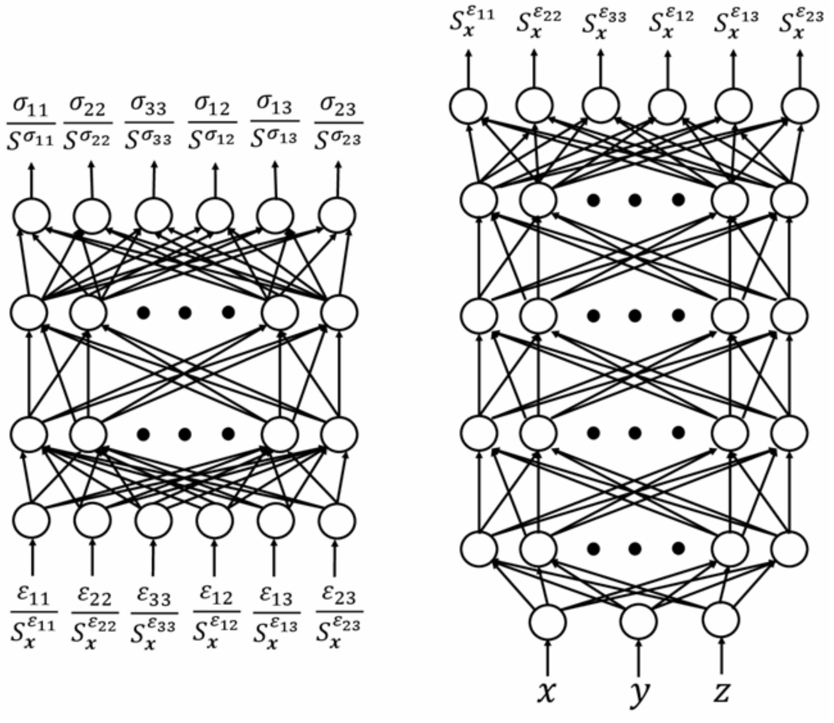

$\frac{\sigma_{11}}{S^{\sigma_{11}}}$ $\frac{\sigma_{22}}{S^{\sigma_{22}}}$ $\frac{\sigma_{33}}{S^{\sigma_{33}}}$ $\frac{\sigma_{12}}{S^{\sigma_{12}}}$ $\frac{\sigma_{13}}{S^{\sigma_{13}}}$ $\frac{\sigma_{23}}{S^{\sigma_{23}}}$
$\frac{\varepsilon_{11}}{S_x^{\varepsilon_{11}}}$ $\frac{\varepsilon_{22}}{S_x^{\varepsilon_{22}}}$ $\frac{\varepsilon_{33}}{S_x^{\varepsilon_{33}}}$ $\frac{\varepsilon_{12}}{S_x^{\varepsilon_{12}}}$ $\frac{\varepsilon_{13}}{S_x^{\varepsilon_{13}}}$ $\frac{\varepsilon_{23}}{S_x^{\varepsilon_{23}}}$
$S_x^{\varepsilon_{11}}$ $S_x^{\varepsilon_{22}}$ $S_x^{\varepsilon_{33}}$ $S_x^{\varepsilon_{12}}$ $S_x^{\varepsilon_{13}}$ $S_x^{\varepsilon_{23}}$
x y z

Figure 6.1: Architecture of 3-D CaNNCM.

Algorithm 2 for computing new $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$. A relatively minor change is covered in Sec. 6.2.5 to account for objects under large deformation, but that change affects the stresses and strains, not the stages of AutoP.

The nature of US imaging with a linear-array probe limits internal displacement measurements to a finite plane. Due to this limitation, several force-displacement measurements must be acquired throughout a volume in order for a CaNNCM to learn the 3-D Young's modulus distribution. Rather than the sampling issue we encountered in the last chapter, where the concern was reducing the effect of noise while preserving the ability of a CaNNCM to learn geometric information, we must investigate spatial sampling strategies to accurately and efficiently reconstruct the distribution of material properties throughout a volume.

Typically, 3-D QUSE techniques using a linear-array (like the examples provided in Sec. 6.1) solve a series of 2-D problems and stack the result to estimate volumetric material properties or sweep the probe (or use a 2-D array) to acquire a 3-D data set. Our approach does neither; instead, we use a combination of planar data and 3-D FE models. We discussed in Sec. 5.4 that a quasi-statically applied load has time to propagate through a volume before tissue motion is measured. Displacements at each point therefore contain some information about the material properties of the whole contiguous region. As a consequence, displacements applied in FEA$_\varepsilon$, even when confined to nodes only within the measurement plane, propagate through the mesh. We can exploit this fact to develop sampling strategies that minimizes redundancy in the measurement data — there is no need to scan the tissue in 1 mm increments when 4 mm increments will suffice. What is unknown at this time is the distance effects of locally applied displacements are distributed. As we will later show, simply including more force-displacement data from many image planes does not necessarily lead to CaNNCMs that reconstruct Young's modulus distributions with the smallest error, but will always result in increased computation time. Moreover, the best sampling strategy may not be one where displacements are measured in multiple parallel planes. Rotating the US probe to acquire 3-D data sets may be more efficient and produce comparable results.

6.2.1 Phantom Models

We began our investigation using the three Models shown in Fig. 6.2a–c. Model 2 (Fig. 6.2a) is the same from Chapter 5. It is a $50 \times 50 \times 50$ mm^3 cube comprised of a soft background material (7.15 kPa) and three inclusions, each with a different Young's modulus (10.93, 14.1, and 20.51 kPa). These modulus values are the same measured via macro-indentation methods for the gelatin phantom described in Sec. 5.2.2. Model 4 (Fig. 6.2b) is another cubic phantom with the same outer dimensions as Model 2. It contains two 20 kPa inclusions, one spherical centered at $(0, -10, 0)$ and one cylindrical centered at $(0, 10, x_3)$ (both having a 5 mm radius), embedded in a 7 kPa background. Model 5 (Fig. 6.2c) is a truncated hemisphere. It was created by generating a hemisphere with 59 mm radius and performing a plane-cut to reduce its height (along x_2-axis) to 50 mm. We chose these dimensions to approximate the geometry of preliminary gelatin phantoms with this shape. The background material is 7 kPa and the two spherical inclusions are centered at $(0, 0, 0)$ and $(0, 0, -20)$ with Young's modulus values 20 kPa and 15 kPa, respectively.

To visualize the 3-D Young's modulus distribution, we chose to display

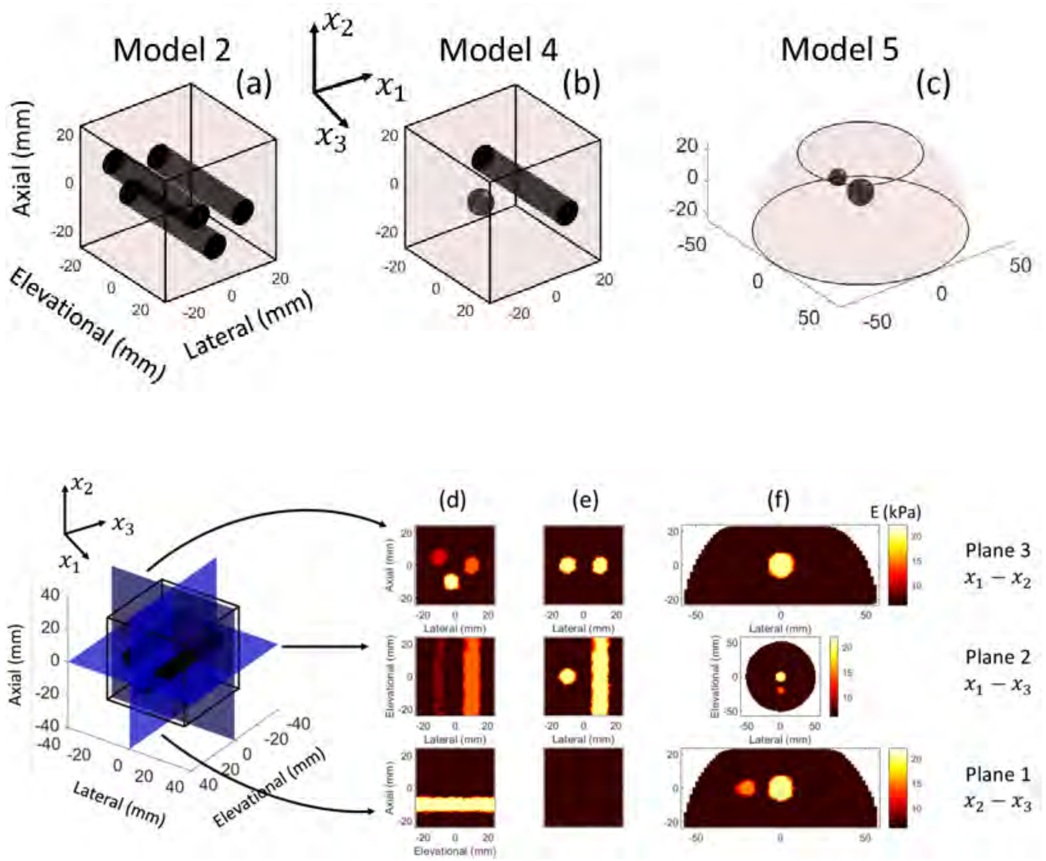


Figure 6.2: (a-c) Simulated phantom models. (a) Model 2 is the same three-inclusion phantom from Chapters 4 and 5. (b) Model 3 is a soft cubic phantom with two inclusions, one spherical and the other cylindrical. (c) Model 5 is a truncated hemisphere with two stiff spherical inclusions. (d–f) Young's modulus distributions for the three Models. Planes 1, 2, and 3 show three slices through the volume and are indicated by the blue planes to the left.

the modulus within the three planes illustrated to the left of Figs. 6.2d–f. Plane 3 is the $x_1 - x_2$ plane at $x_3 = 0$. Plane 2 is the $x_1 - x_3$ plane at $x_2 = 0$ and Plane 1 resides at $x_1 = 0$. Numbers denoting the Planes correspond to the axis in which the normal vector to the plane points; e.g., the normal vector to Plane 3 points in the x_3 direction. Figs. 6.2d–f show the Young's modulus distribution of Models 1–3 within these planes.

6.2.2 Finite Element Meshes

The FE meshes are comprised of 10-node tetrahedral elements (C3D10 in Abaqus 6.13). These elements are not as "stiff" as their 4-node counterparts, but do increase the computational load when solving a FEA. We chose tetrahedral over hexahedral (or "brick") elements because the former can be used to mesh any arbitrary geometry. It is possible that brick elements could mesh Model 3 (and Model 6 that will be introduced in a following section), but doing so would require smaller and thus many more elements, leading to increased FEA computation time. For these reasons, we chose 10-node tetrahedral elements.

Furthermore, unlike the rectilinear FE meshes used in AutoP for 2-D CaNNCMs, we do not use a consistent mesh for all models. Nor are all meshes necessarily the same for each model across all AutoP analyses. Several aspects do remain consistent amongst all FE models, though, and are the same as specified in Sec. 5.2.3. The US probe was modeled as a rigid body in frictionless contact with the top surface of the phantom model, approximating the condition created by the US gel at the contact surface. Nodes on the bottom surface of the phantom mesh were fixed (or "pinned") to represent the no-slip contact between the phantom and rubber pad during experimental data acquisition. Force loads applied in FEA_σ were imposed as a concentrated force to top of the US probe, whereas the corresponding displacements applied in FEA_ε were defined as BCs for all probe nodes. All FEAs were solved with ABAQUS 6.13 commercial finite element software.

6.2.3 Simulated Force-displacement Data

3-D CaNNCMs were first tested by training in AutoP with noise-free force-displacement data. FE meshes were generated for the three Models in Fig. 6.2a–c. Note that the meshes only conformed to the surface geometry, not the internal structure. Using defined Young's modulus distributions, a force load up to 848 mN was applied by the probe over six load increments to compute the mesh deformation, resulting in six force-displacement data sets for each FEA.

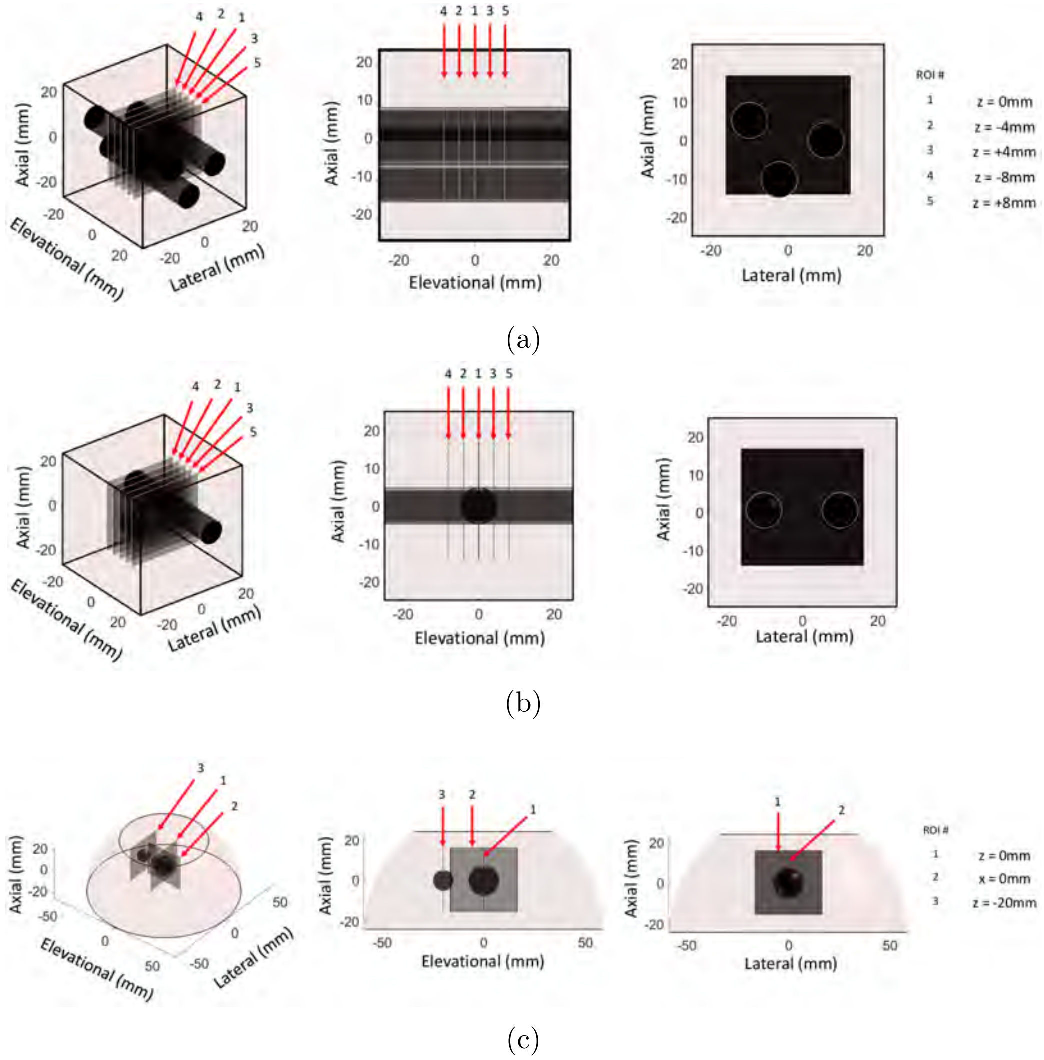
For a 3-D, linear-elastic material, the constitutive relationship can be expressed as

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{bmatrix} = \tfrac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & 1-2\nu & 0 & 0 \\ 0 & 0 & 0 & 0 & 1-2\nu & 0 \\ 0 & 0 & 0 & 0 & 0 & 1-2\nu \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \varepsilon_{12} \\ \varepsilon_{13} \\ \varepsilon_{23} \end{bmatrix}. \tag{6.1}$$

Note the $(1 - 2\nu)$ term in the denominator. Clearly, setting $\nu = 0.5$ would result in a singularity. Even a Poisson's ratio close to incompressible would result in exceptionally large stresses for very small strains. An incompressible material can be modeled in FEA using additional constraint equations, but (6.1) is the equation used to generate stress-strain pairs for CaNNCM pretraining in AutoP. Therefore, most of the simulated data sets we generated used $\nu = 0.3$ so that a matching Poisson's ratio could be used for pretraining. We will later show that the Poisson's ratio does not have to match.

Several force-displacement data sets were generated for each model by moving the location of the US probe. Fig. 6.3 depicts the image region of the probe for each set, referred to as the ROI. The probe was positioned at five different locations for Models 2 and 4, as shown in Figs. 6.3a–b. ROI 1 arises from the probe centered both laterally and elevationally along the top surface of the phantom. Moving the probe +4 mm and -4 mm in the x_3 direction gives ROIs 2 and 3, respectively. Similarly, ROIs 4 and 5 are the result of advancing the probe +8 mm and -8 mm along the x_3 axis.

ROIs for Model 5 were chosen to investigate a more efficient sampling strategy. Relative placement of the ROIs for Models 2 and 4 may be effective, yet not the most efficient because we require many data sets to learn 3-D information. Instead of acquiring data in parallel planes, we may be able to obtain volumetric information more efficiently by altering the relative position of the data planes. We tested this hypothesis with Model 5 by defining only three ROIs. The first two are orthogonal and bisect the central spherical inclusion. ROI 3 is parallel to ROI 1 and passes through the smaller inclusion. We expect ROIs 1 and 2 to provide enough information for the CaNNCM to learn the spherical shape of the inclusion. Given the size of the smaller inclusion sampled by ROI 3, a single plane my be adequate for a CaNNCM to learn its 3-D structure.


(a)

(b)

(c)

Figure 6.3: Planes indicating the ROI of the US probe where internal displacement measurements are acquired. (a–b) Displacements are sampled in five parallel planes/ROIs for Models 2 and 4. (c) Data are acquired in only three planes for Model 5.

We tested the ability of 3-D CaNNCMs to learn the Young's modulus distribution of the three Models before turning to AutoP. A MPN was pre-trained as a 5 kPa homogeneous material with $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon} = \mathbf{1}$ and $\nu = 0.3$. A total of 10000 strain vectors were generated whose components were randomly distributed in the ±0.3 range. Corresponding stress vectors were computed with (6.1) along with the aforementioned Young's modulus and Poisson's ratio. The MPN was trained using the RPROP algorithm. Stresses and strains were then compiled for all three Models from the forward FEAs used to generated force-displacement data for ROI 1. Stress-strain data sets and the pre-trained MPN and entered into Algorithm 2 ($N_{\varepsilon} = 300, \eta_{\varepsilon} = 1.5$) to compute $\boldsymbol{S}^{\varepsilon}_{\boldsymbol{x}}$ for the three Models.

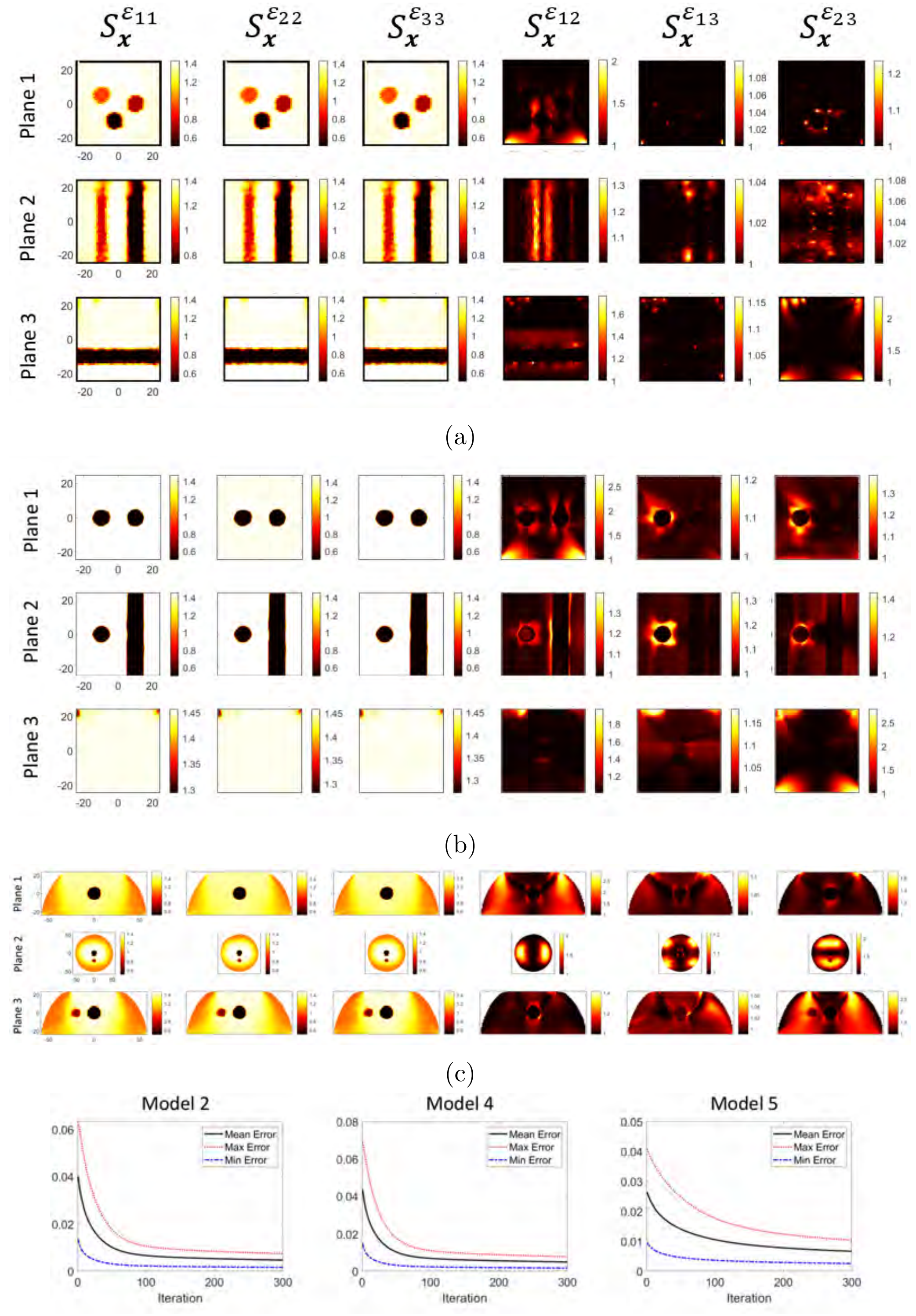
Fig. 6.4a–c are maps of each component of $\boldsymbol{S}^{\varepsilon}_{\boldsymbol{x}}$ computed for the three Models. Scaling values of the normal strain components ($S^{\varepsilon_{11}}_{\boldsymbol{x}}, S^{\varepsilon_{22}}_{\boldsymbol{x}}, S^{\varepsilon_{33}}_{\boldsymbol{x}}$) for Models 2 and 4 show the background material to be uniform. Interestingly, the same cannot be said of the corresponding maps for Model 5. Recall that softer regions are indicated by larger scaling values. Based on the maps in Fig. 6.4c, the regions of the background material not directly under the probe are stiffer. This result is particularly intriguing because it is the first model tested with CaNNCMs that is not under uniaxial compression. Sections of the phantom not under the probe are under increased lateral and shear strains (the latter evidenced by maps of the shear strain scaling values) rather than compressive strain.

Three different SNs were trained with the computed spatial values over ten iterations each consisting of 800 epochs (the same procedure used in Sec. 4.2.4). Each SN was comprised of five hidden layers each with 20 nodes. Training was implemented in TensorFlow using the Adam optimizer (with default parameter settings) and a learning rate of 0.03. Pairing each SN with the MPN, the Young's modulus distribution $E(\boldsymbol{x})$ was computed from the resulting CaNNCM using $\nu = 0.3$, the strain vector $\boldsymbol{\varepsilon} = [0.0035\ -0.01\ 0.0035\ 0.001\ 0.001\ 0.001]^{\mathrm{T}}$, and stiffness matrix $\hat{\boldsymbol{D}}$ (see Appendix D):

$$E(\boldsymbol{x}) = (\frac{\hat{D}_{11}}{S^{\varepsilon_{11}}_{\boldsymbol{x}}} - \frac{\hat{D}_{12}}{S^{\varepsilon_{12}}_{\boldsymbol{x}}})(1+\nu). \tag{6.2}$$

Fig. 6.5 are the Young's modulus distributions resulting from the CaNNCMs trained with the spatial scaling values in Fig. 6.4. The networks trained with data from Models 2 and 4 (Figs. 6.5a–b) are nearly identical to their counterparts in Fig. 6.2, as expected. However, the CaNNCM trained for Model 5 is only accurate in the region directly under the US probe. As we indicated for the spatial values in Fig. 6.4c, the regions of the phantom not under direct compression appear stiffer than expected. Changing the US probe position to apply compressive loads to these regions may reduce these stiffening artifacts.[1]

[1] We will return to this point in Sec. 6.4. It may be be due in part to the constant value of $\boldsymbol{S}^{\sigma}$.



(a)

(b)

(c)

(d)

Figure 6.4: Spatial values for the three models computed with Algorithm 2. (a) Model 2 (b) Model 4 (c) Model 5. (d) Maximum, minimum, and mean error curves over the 300 iterations used to compute $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$.

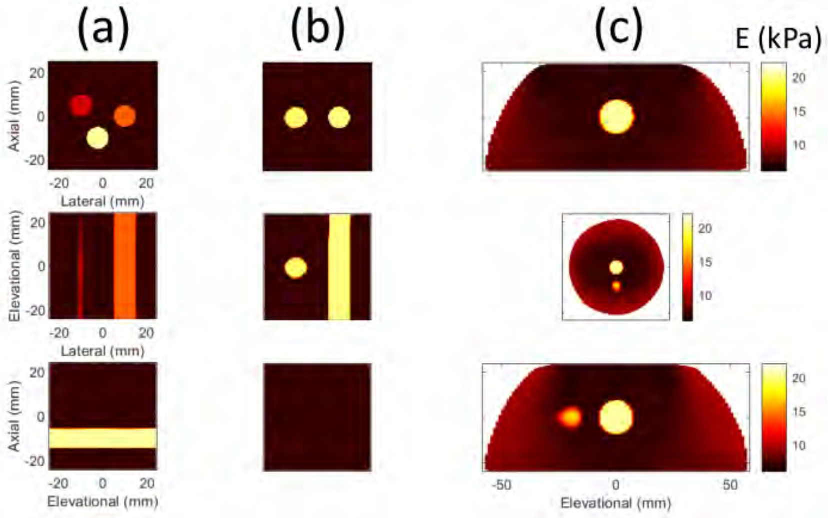


Figure 6.5: Young's modulus reconstructions after training a 3-D CaNNCM with the spatial values displayed in Fig. 6.4. (a-c) Models 2, 4, and 5, respectively.

6.2.4 Measurements on Gelatin Phantoms

The same imaging phantom and data acquisition methods detailed in Sec. 5.2.2 were used to test 3-D CaNNCMs. In fact, several of the same Data Sets were re-used, albeit we utilized six load increment instead of four and denote them differently here. Data Sets 1–5 correspond to ROIs 1–5 (Sets 1–3 are the same as $1, 1'$, and $1''$ defined in Sec. 5.2.2, respectively). Three additional data sets were acquired by rotating the phantom in 90° increments around the x_3 axis. Aligning the probe to ROI 1, Data Sets 6–8 were obtained by rotating the phantom 90°, 180°, and 270°, respectively (these are the same as Data Sets 4-6 depicted in Fig. 5.3). One primary difference is the magnitude of the force measured during compression. Only the in-plane force was applied in FEA_σ for 2-D CaNNCMs, calculated by dividing the total measured force by the contact width in the elevational direction (43 mm in this case). Conversely, the full contact surface is accounted for in 3-D, meaning the total applied force ranged from 779.68–853.13 mN as opposed to 17.95–21.22 mN. An additional set of force-displacement measurements was acquired in the same manner as Data Set 1, but 1330.06 mN of compressive force was applied to produce 5% strain in the phantom, not 3%. We refer to this as Data Set $1'$.

Phantom with Estimated External Geometry

Data acquisition in a clinical setting will preclude prior knowledge of the tissue geometry. Unlike the gelatin phantom, real biological tissues are not well-defined cubes and manufactured in a tightly controlled setting. Therefore, we will require a method to measure the external shape of the object. A detailed explanation of using a Microsoft Kinect v2 for this purpose is provided in Appendix G. We tested this method on a different gelatin phantom whose external geometry is similar to Model 5 and contains only one spherical inclusion with a radius of 7.5 mm. We refer to this phantom as Model 7.

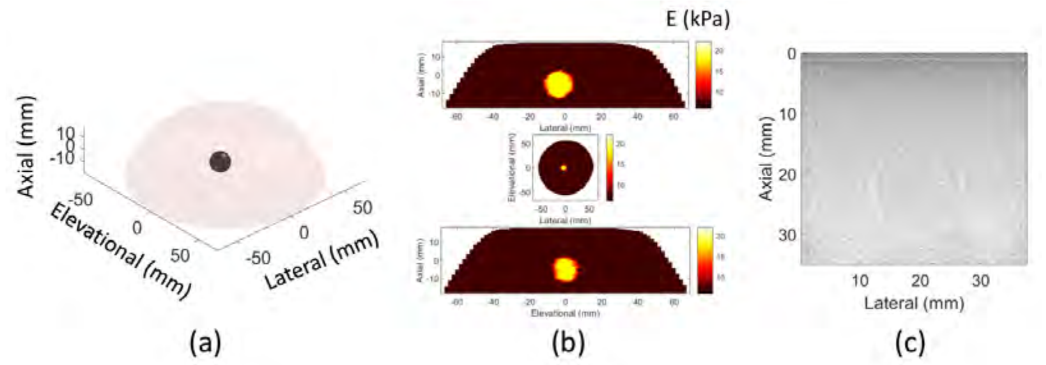


Figure 6.6: (a) Geometry of Model 6. (b) Young's modulus distribution estimated from B-mode images and macro-indentation. (c) Example B-mode image of the phantom.

To make this phantom, we first manufactured the spherical inclusion (≈ 17.91 ± 0.45 kPa, 12% gelatin by mass). Then, using a glass bowl as the phantom mold, the spherical inclusion was suspended by thread while the background gelatin material (≈ 7.39 ± 0.14, 8% gelatin by mass) was poured. The intent was to center the inclusion along all three axes. Given the imprecision involved, the actual location of the spherical inclusion was estimated from US B-mode images. Figs. 6.6a–b display the approximate phantom geometry and Young's modulus distribution, where the modulus values were estimated via macro-indentation methods. An example B-mode image of the phantom is provided in Fig. 6.6c.

Two data sets were measured on this phantom using the same experimental technique in Sec. 5.2.2. Data Set 9 corresponds to ROI 1 for Model 5 (Fig. 6.3c) and Data Set 10 matches ROI 2. To be clear, the image planes for Data Sets 9 and 10 are orthogonal. The US probe compressed the phantom

1.87 mm, resulting in 5% strain and total applied force ranging from 2937.5–3167.19 mN.

6.2.5 Linear-Elastic Materials Under Large Deformation

We have thus far only considered quasi-static loads that produce very little deformation and approximate the strain as infinitesimal. For a 48 mm tall phantom, the probe is only displacing ≈ 1.5 mm which results in $\approx 3\%$ strain. Characterizing the hyperelastic and/or non-linear properties of soft tissues will require larger loads resulting in finite deformation. At this point non-linear geometric effects are no longer negligible and the previous definition of strain in (1.3) no longer applies.

We provide here a brief review of some fundamental concepts of the kinematics of non-linear deformation. Index notation will be used in several equations where summation is assumed over repeated indices, as was done back in (1.4). A material point $\boldsymbol{X}$ in the *reference* configuration displaces to a new spatial location $\boldsymbol{x}$ in the *current* (or updated) configuration.The deformation gradient tensor $\boldsymbol{F}$ can then defined as

$$\boldsymbol{F} = F_{iJ} = \frac{\partial x_i}{\partial X_J} \tag{6.3}$$

$$\tag{6.4}$$

where lower-case subscripts are used for tensors in the updated configuration and upper-case subscripts are for tensors in the reference configuration. Infinitesimal deformation, or a small-strain assumption, means the local displacement is negligible. More precisely, $\lim \boldsymbol{F} \to \boldsymbol{I}$ and strain is defined by (1.3). Objects undergoing finite deformation can have stresses and strains defined for the reference or current configuration (or a mix of both). Here, we consider Green-Lagrange strain $\boldsymbol{E}$ and Almansi strain $\boldsymbol{e}$:

$$\boldsymbol{E} = \frac{1}{2}(\boldsymbol{F}^T\boldsymbol{F} - \boldsymbol{I}), \tag{6.5}$$

$$\boldsymbol{e} = \frac{1}{2}(\boldsymbol{I} - \boldsymbol{F}^{-T}\boldsymbol{F}^{-1}). \tag{6.6}$$

$\boldsymbol{E}$ is a description of strain in the reference configuration whereas $\boldsymbol{e}$ describes strain in the deformed state. To more directly compare $\boldsymbol{\varepsilon}$ and $\boldsymbol{E}$, both definitions of strain can be rewritten as gradients of displacement $\boldsymbol{u}$:

$$\varepsilon_{ij} = \frac{1}{2}(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i})$$
$$E_{ij} = \frac{1}{2}(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} + \frac{\partial u_k}{\partial X_i}\frac{\partial u_k}{\partial X_j}). \tag{6.7}$$

The last term in (6.7) is the non-linear geometric effect of large deformations.

Like strain, many different definitions of stress exist. We will only consider the 2nd Piola-Kirchoff (P-K) stress $\boldsymbol{S}$ and Cauchy (true) stress $\boldsymbol{\sigma}$. $\boldsymbol{S}$ is related to $\boldsymbol{E}$ through the stiffness tensor $\boldsymbol{C}$ and $\boldsymbol{\sigma}$ is related to $\boldsymbol{e}$ through $\boldsymbol{c}$:

$$\boldsymbol{S} = S_{IJ} = C_{IJKL}E_{KL} \tag{6.8}$$

$$\boldsymbol{\sigma} = \sigma_{ij} = c_{ijkl}e_{kl} \tag{6.9}$$

The definition of stress and strain used will affect the material properties learned by a NNCM because generally $\boldsymbol{C} \neq \boldsymbol{c}$. Even more important is the fact that certain stress-strain pairs are energetic conjugates. Green-Lagrange strain is the energetic conjugate of 2nd P-K stress while Almansi strain pairs with Cauchy stress. One cannot simply pair Almansi strain with 2nd P-K stress or Green-Lagrange strain with Cauchy stress. However, one pair can be transformed into the other through the deformation gradient. First, expressing Cauchy stress in terms of 2nd P-K stress:

$$\boldsymbol{\sigma} = \frac{1}{|\boldsymbol{F}|}\boldsymbol{F}\boldsymbol{S}\boldsymbol{F}^T = \frac{1}{|\boldsymbol{F}|}F_{iI}S_{IJ}F_{jJ}. \tag{6.10}$$

where $|\boldsymbol{F}|$ is the determinant of the deformation gradient. Putting (6.8) into (6.10),

$$\sigma_{ij} = \frac{1}{|\boldsymbol{F}|}F_{iI}C_{IJKL}E_{KL}F_{jJ}. \tag{6.11}$$

Green-Lagrange strain can be transformed to Almansi strain and then substituted into (6.11):

$$\boldsymbol{E} = \boldsymbol{F}^T\boldsymbol{e}\boldsymbol{F} = F_{kK}e_{kl}F_{lL} \tag{6.12}$$

$$\boldsymbol{\sigma} = \frac{1}{|\boldsymbol{F}|}F_{iI}C_{IJKL}F_{kK}e_{kl}F_{lL}F_{jJ}, \tag{6.13}$$

and rearranging to end with an expression defining Cauchy stress in terms of the deformation gradient and elasticity tensor $\boldsymbol{C}$,

$$\sigma_{ij} = \frac{1}{|\boldsymbol{F}|} F_{iI} F_{jJ} F_{kK} F_{lL} C_{IJKL} e_{kl}. \tag{6.14}$$

This equation can be further simplified by defining $\boldsymbol{c}$ in terms of $\boldsymbol{C}$,

$$c_{ijkl} = \frac{1}{|\boldsymbol{F}|} F_{iI} F_{jJ} F_{kK} F_{lL} C_{IJKL}. \tag{6.15}$$

The expression for $\boldsymbol{c}$ in (6.15) can be substituted in (6.14) to return to the original expression for Cauchy stress in (6.9).

The FEA formulation for a system under finite deformation is beyond the scope of this thesis. What is important is the fact that Abaqus uses an Updated Lagrangian formulation, which means $\boldsymbol{X}$ is updated to $\boldsymbol{x}$ after each load increment (the updated configuration becomes the reference for the next load increment) and the stresses and strains are expressed as $\boldsymbol{\sigma}$ and $\boldsymbol{e}$. Yet the current CaNNCM architecture is unable to account for a changing reference configuration. CaNNCMs must therefore learn the relationship between $\boldsymbol{E}$ and $\boldsymbol{S}$.

Implementing this in AutoP is straight-forward. Cauchy stress and Almansi strain are computed in FEA$_\sigma$ and FEA$_\varepsilon$, respectively. Strains $\boldsymbol{e}$ are converted to $\boldsymbol{E}$ via (6.12) and $\boldsymbol{\sigma}$ to $\boldsymbol{S}$ through the relation

$$\boldsymbol{S} = |\boldsymbol{F}| \boldsymbol{F}^{-1} \boldsymbol{\sigma} \boldsymbol{F}^{-T}. \tag{6.16}$$

Green-Lagrange strain and 2nd P-K stress are then used to compute the spatial values and train the MPN. When computing $\boldsymbol{D}(\boldsymbol{x})^n$ (which is can be computed from $\boldsymbol{C}$ [84]) and $\boldsymbol{\sigma}^{n-1}$ in (5.6) and (5.7), the conversions must be made in (6.15) and (6.10).

Another simulated phantom was created to test 3-D CaNNCMs with objects under finite deformation. Model 6 is a hemispherical phantom comprised of a soft background (7 kPa) with two stiff, spherical inclusions. One inclusion is centered at $(0, 0, 0)$, has a radius of 7.5 mm, and has a Young's modulus of 20 kPa. The second inclusion is centered at $(-20, 0, 0)$ with a 5 mm radius Young's modulus of 15 kPa.

Force-displacement data were acquired on Model 6 for the ROIs in Fig. 6.7.

ROIs 1 and 2 are the same as those for Model 5. Given the phantom shape, simply translating the probe is not effective for sampling the volume. ROI 3 is then obtained by first rotating the probe 90° around the x_2-axis, followed by a rotation of atan(4/5) around x_3. The lower diagram in the figure depicts the angle between ROIs 1 and 3. This final ROI was chosen to bisect the smaller inclusion because it does not reside within ROI 1 or 2. A compressive force of 1377.19 mN was applied over six load increments resulting in $\approx 14\%$ maximum strain. We arrived at this force load from experimental measurements on a gelatin phantom with geometry and material properties similar to Model 6. Data from said phantom are not included in this dissertation because the US probe was not in full contact with the phantom surface when compressive loads were first applied, meaning many of the elements in the linear array recorded strong reverberations. Thus, the speckle-tracking algorithm was unable to reliably estimate internal displacements.

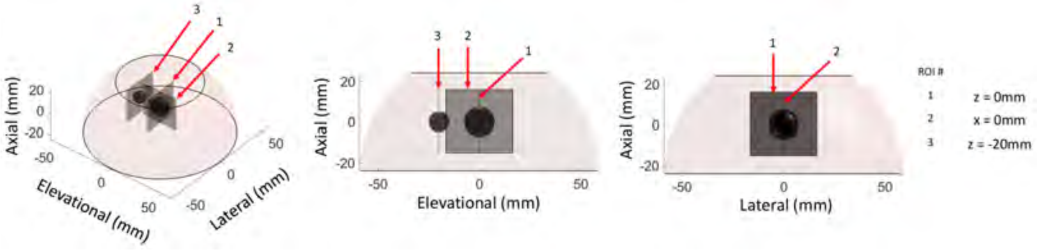


Figure 6.7: ROIs where displacement data are acquired for Model 6. The lower diagram illustrates the angle between ROIs 2 and 3.

Fig. 6.8 displays the geometry of Model 6, its Young's modulus distribution within Planes 1–3 (red-dotted box), and the spatial values computed with simulated stress-strain data. Curves of the minimum, maximum, and mean error calculated in Algorithm 2 are plotted left of the $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ maps. Algorithm 2 was tested with the simulated stress-strain data to ensure compatibility with geometrically non-linear analyses. Similar to preliminary results in Sec. 6.2.3, the error curves provide a guideline for implementation in AutoP.

6.2.6 AutoP Parameters

Most of the training parameters are identical for the AutoP analyses described in this chapter. Parameters that differed for each test are listed in Table 6.1. Each CaNNCM was trained over at least five passes, with each

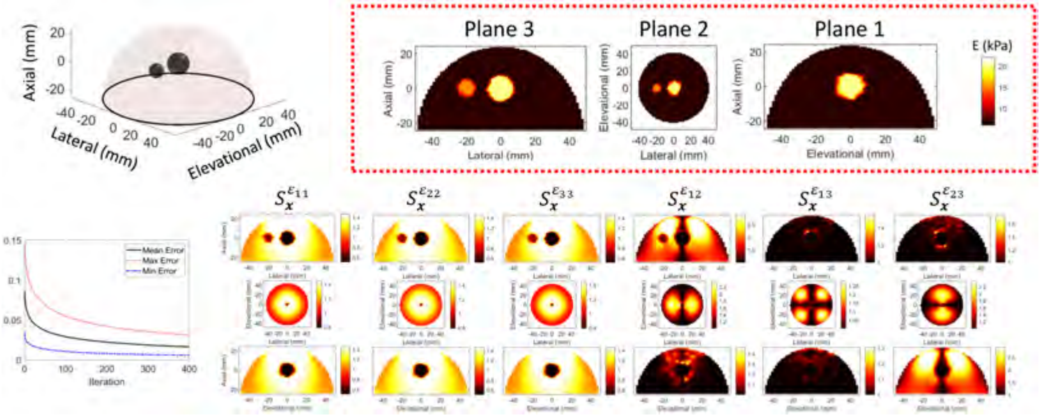


Figure 6.8: Model 6 is a hemispherical object with two spherical inclusions. The red-dotted box displays the Young's modulus distribution. Also included are the spatial scaling values calculated after computing stresses and strains from a forward FEA. The plot on the left are the mean, minimum, and maximum error from Algorithm 2.

pass comprised of six load increments. We also implemented a four-load "training window" and retained the convergence criteria $C^n_{\max}$, C^n_{μ} defined for 2-D CaNNCMs (both described in Sec. 5.2.7). To recapitulate, convergence criteria were initialized as $(0.65, 0.5)$, using the notation $(C^n_{\max}, C^n_{\mu})$, and reduced to $(0.4, 0.3)$, $(0.3, 0.2)$, and $(0.2, 0.01)$ at the beginning of passes 2, 3, and 4, respectively. The last set of criteria were carried through the remaining passes. The upper limit of AutoP iterations per training step was set to three — which was shown to be sufficient in preliminary tests — in order to avoid iterations do that continue indefinitely.

Case 2 sampling (Table 5.1) was used for all CaNNCMs trained with simulated, noise-free force-displacement data. When experimental measurements were entered into AutoP, Case 3 sampling was employed. Recall that Case 2 means displacements were given at all nodes in the ROI for FEA$_\varepsilon$. On the other hand, displacements are only imposed for nodes with a minimum 1.5 mm separation in Case 3 sampling.

A Young's modulus value of 5 kPa and strain range of ±0.3 was selected for linear-elastic pretraining. Following the same reasoning provided in Sec. 6.2.3, $\nu = 0.5$ is incompatible with (6.1), so we selected $\nu = 0.3$. We demonstrated in Chapter 5 that the choice of Poisson's ratio for pretraining did not have a significant impact on the material properties learned by the CaNNCM within the field of view. The same will be shown for 3-D.

The MPN had two hidden layers of ten nodes each whereas the SN had five hidden layers with 15 nodes each. Backpropagation-based training of the MPN was implemented with the RPROP algorithm [123] over 50 epochs. Training of the SN was again implemented in TensorFlow using the Adam optimizer (with default parameter settings) and a learning rate of 0.03 over 700 epochs. Frame invariance was also enforced, although the data had to be rotated through three axes instead of one which quadruples the amount of data. However, note that only the original stress-strain data computed in FEA_σ and FEA_ε are rotated. That is, after rotating 90° around the x_1 axis, the newly generated data is not then rotated around the x_2 and x_3 axes. Spatial scaling values $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ were computed in Algorithm 2. Depending on the AutoP analysis, the number of iterations N_ε and update value η_ε varied, but were within the ranges $150 \leq N_\varepsilon \leq 500$ and $0.25 \leq \eta_\varepsilon \leq 1.5$ (listed in Table 6.1).

One of the major differences between 2-D and 3-D analyses is the amount of data generated in FEA_σ and FEA_ε. For example, one of the 3-D meshes contains 7747 10-node, tetrahedral elements. A total of four stress-strain pairs are computed within each element. Coupled with the four-load training window and frame invariance, $7747 \times 4 \times 4 \times 4 = 495808$ training pairs comprise the data set in each AutoP iteration. We have found this is too much for training the MPN, although all pairs are used to compute $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$. The MPN is trained with a subset of the generated data by first randomly selecting a number of elements. Then, all of the training data from this small set of elements is used for training. We keep all the data from the selected elements to ensure the original stress-strain pairs and the corresponding frame-invariant pairs are all included during training. If we had instead randomly selected a subset of training pairs, it is likely all frame-invariant versions of a data-pair would be included. The number of elements chosen varies based on the AutoP analysis, but we used a maximum of 400 ($400 \times 4 \times 4 \times 4 = 25600$ training pairs).

Once training completed, Young's modulus distributions $E(\boldsymbol{x})$ were computed by each CaNNCM in the same manner described in Sec. 6.2.3. We defined the strain vector $\boldsymbol{\varepsilon} = [0.0035\ -0.01\ 0.0035\ 0.001\ 0.001\ 0.001]^{\mathrm{T}}$ and computed the stiffness matrix $\hat{\boldsymbol{D}}$ from the MPN (Appendix D). $\boldsymbol{x}$ was varied over the domain of the mesh and input to the SN to compute $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$. Both $\hat{\boldsymbol{D}}$ and $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ were then input to (6.2).

Young's modulus distributions estimated by the CaNNCMs were compared to target maps, shown in Figs. 6.2,6.6, and 6.8. Errors were computed as

$$e^E_{\boldsymbol{x}} = \frac{|E^{target}_{\boldsymbol{x}} - E^{NN}_{\boldsymbol{x}}|}{E^{target}_{\boldsymbol{x}}}, \tag{6.17}$$

where $E^{target}_{\boldsymbol{x}}$ is the target Young's modulus distribution and $E^{NN}_{\boldsymbol{x}}$ is computed by the CaNNCM. However, the error was not calculated over the entire domain of the mesh. Displacements are only provided within the ROIs during AutoP training. Consequently, we expect the CaNNCMs to learn accurate material properties in the regions where displacement data was supplied. For this reason the error in (6.17) is only computed within the ROIs used for training.

6.3 Results

Young's modulus distributions reconstruction by CaNNCMs-1–3 trained with simulated force-displacement from Model 2 are shown in Fig. 6.9. Modulus errors computed via (6.17) are provided in Table 6.2. The number of AutoP iterations performed during training and computation time are also included in the table. When only data from ROI 1 was input during training (Fig. 6.9a), CaNNCM-1 was able to learn the material properties within the plane, but was unable to accurately reconstruct the geometry along the x_3 axis. We do observe, though, that CaNNCM-1 has learned the inclusions extend $\approx \pm 2$ mm in the elevational direction. Hence our decision to separate the ROIs by 4 mm. Using data from all five ROIs provided enough information for CaNNCM-2 to learn the 3-D geometry, albeit only within the region spanned by the ROIs (Fig. 6.9b). We show in Fig. 6.9c the result of pre-training with $\nu = 0.3$ when the force-displacement data were generated with $\nu = 0.495$. Similar to the 2-D results in Sec. 5.3.1, CaNNCM-3 learns the correct material properties in the ROI even though a stiffening artifact appears below image region.

Figs. 6.9d–e are 3-D renders of the inclusions from Figs. 6.9a–b. These were created in ParaView by computing the Young's modulus throughout the volume and setting a minimum threshold for display. Therefore, only modulus values above a given stiffness — here, set to 10 kPa — are shown. Parts of the inclusions captured by the CaNNCMs are superimposed on the

Table 6.1: AutoP training parameters specific to each trained CaNNCM. # Elem. refers to the number of elements from which stress-strain data are collected for training the MPN. *mnalg* and η_ε are the parameters for Algorithm 2.

			Alg. 2	
CaNNCM-#	Passes	# Elem.	N_ε	η_ε
CaNNCM-1	5	400	150	0.5
CaNNCM-2	10	400	300	1.0
CaNNCM-3	5	400	300	1.0
CaNNCM-4	5	400	150	0.5
CaNNCM-5	7	400	300	1.0
CaNNCM-6	7	400	300	1.0
CaNNCM-7	10	400	300	1.0
CaNNCM-8	10	400	300	1.0
CaNNCM-9	10	400	300	1.0
CaNNCM-10	5	400	200	0.5
CaNNCM-11	7	400	300	1.0
CaNNCM-12	10	400	300	1.0
CaNNCM-13	5	400	300	0.5
CaNNCM-14	5	400	300	0.5
CaNNCM-15	5	400	300	0.5
CaNNCM-16	5	400	300	0.5
CaNNCM-17	13	75	500	1.5
CaNNCM-18	5	400	200	0.5
CaNNCM-19	5	400	200	0.5
CaNNCM-20	5	400	200	0.5
CaNNCM-21	5	100	200	0.25
CaNNCM-22	5	100	200	0.25
CaNNCM-23	7	100	200	0.25
CaNNCM-24	5	100	200	0.25
CaNNCM-25	5	400	200	0.5
CaNNCM-26	5	200	200	0.5
CaNNCM-27	15	100	400	1.0

target geometry to better visualize the effect additional ROIs have on reconstructing material properties within a volume.

Fig. 6.10 contains the Young's modulus reconstructions by CaNNCMs trained with noise-free data from Model 4. Once again, including data from only a single ROI (CaNNCM-4, 6.10a) does not provide the network with enough information to learn volumetric properties. Adding data from ROIs 2 and 3, as was done for CaNNCM-5 (Fig. 6.10b), reveals the shape of the spherical inclusion and the cylindrical shape of the other. Conversely, training CaNNCM-6 with data from ROIs 1, 4, and 5 (Fig. 6.10c) more clearly

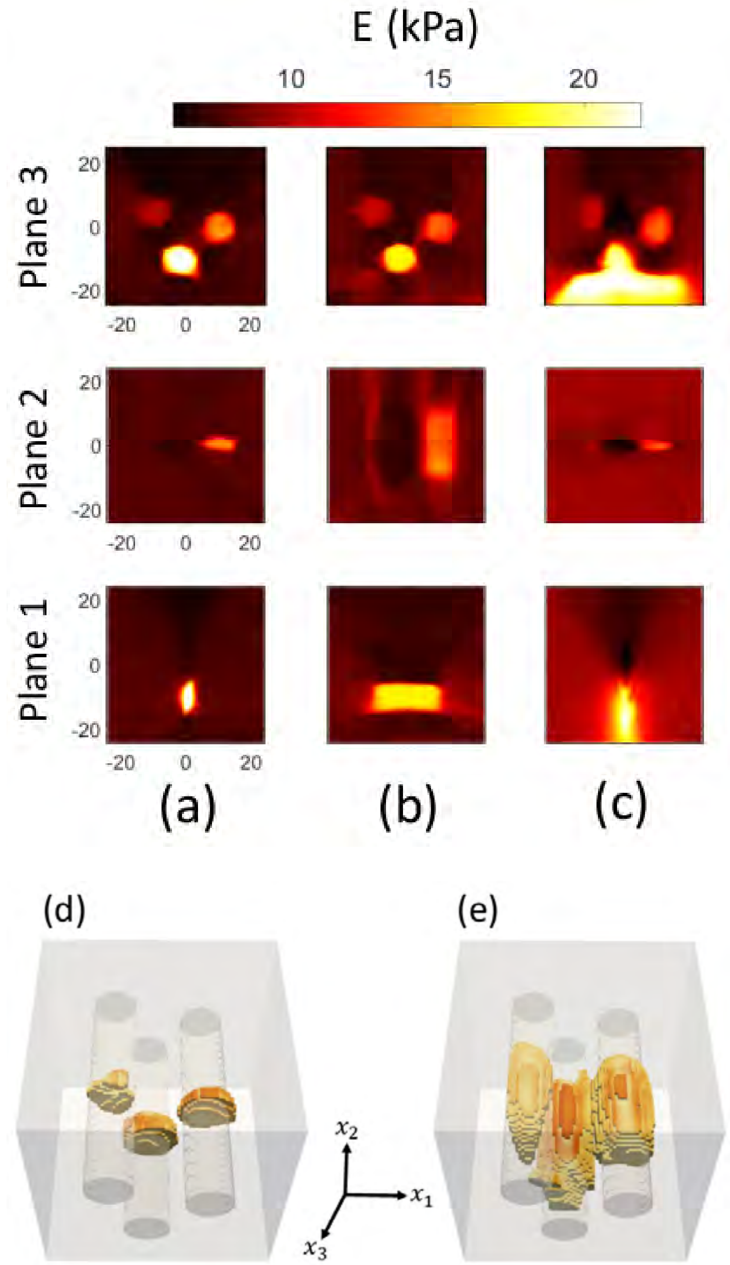


Figure 6.9: Young's modulus reconstructions for Model 2 by CaNNCMs trained with simulated data from ROIs (a) 1, (b) 1–5, and (c) 1 (data generated with $\nu = 0.495$). The Planes are those specified in Fig. 6.2. (d–e) Volumetric reconstruction of the inclusions from (a) and (b).

shows the size of the cylindrical inclusion while missing the 3-D structure of the spherical inclusion. This is likely due to the fact that ROIs 4 and 5 do not intersect the spherical inclusion and thus provide no directly meaningful information about its material properties or shape. However, training with all five ROIs (Fig. 6.10d) was sufficient for CaNNCM-7 to learn the geometry

Table 6.2: Young's modulus reconstruction errors and AutoP run time. Models are illustrated in Figs. 6.2 and 6.8. A * indicates an increased force load to induce 5% strain and † denotes a geometrically non-linear analysis. The last column indicates the figure number of the corresponding Young's modulus image.

Case 2 sampling was used for all AutoP analyses with simulated data as inpute. A ‡ indicates data generated with $\nu = 0.495$ in the forward FEA.

Simulated Models					
CaNNCM-#	(Model #)[ROI]	Modulus Error Mean ± STD	Iters.	Time (min.)	Fig.
CaNNCM-1	(1)[1]	0.0872 ± 0.1105	54	474	6.9a
CaNNCM-18	(1)[1]*†	0.0864 ± 0.0895	36	396	6.9c
CaNNCM-2	(1)[1,2,3,4,5]	0.0913 ± 0.1062	96	3956	6.9b
CaNNCM-3	(1)[1]‡	0.1326 ± 0.1664	36	318	6.15a
CaNNCM-4	(4)[1]	0.0782 ± 0.0743	54	471	6.10a
CaNNCM-5	(4)[1,2,3]	0.0844 ± 0.0793	60	1454	6.10b
CaNNCM-6	(4)[1,4,5]	0.0942 ± 0.0721	60	1481	6.10c
CaNNCM-7	(4)[1,2,3,4,5]	0.0886 ± 0.0699	96	3922	6.10d
CaNNCM-8	(4)[1,6,8]	0.1198 ± 0.0938	96	1428	6.11b
CaNNCM-9	(4)[1,6,7,8]	0.0974 ± 0.0650	42	824	6.11c
CaNNCM-10	(5)[1]	0.0955 ± 0.1030	36	179	6.12a
CaNNCM-11	(5)[1,2]	0.0890 ± 0.0896	60	606	6.12b
CaNNCM-12	(5)[1,2,3]	0.0762 ± 0.0787	96	1729	6.12b
CaNNCM-21	(6)[1]	0.2203 ± 0.1480	36	324	6.12a
CaNNCM-22	(6)[1]†	0.1631 ± 0.1280	36	470	6.12b
CaNNCM-23	(6)[1,2]†	0.1368 ± 0.0956	36	877	6.12c
CaNNCM-24	(6)[1,2,3]†	0.1316 ± 0.1375	30	992	6.12d

Data Sets are described in Sec. 6.2.4 and correspond to those defined in Chapter 4, Fig. 5.3, ableit with different labels. Case 3 sampling (Table 5.1) was selected for all analyses.

Gelatin Phantom						
CaNNCM-#	(Model #)[Set]	Modulus Error, Mean ± STD		Iters.	Time (min.)	Fig.
		3-D	2-D			
CaNNCM-13	(2)[1]	0.2805 ± 0.2049	0.2136 ± 0.1264	45	260	6.13a
CaNNCM-19	(2)[1′]*	0.2406 ± 0.1702	0.2136 ± 0.1264	40	282	6.15b
CaNNCM-20	(2)[1′]*†	0.2036 ± 0.1412	0.2136 ± 0.1264	40	141	6.15c
CaNNCM-14	(2)[6]	0.2811 ± 0.2182	0.2549 ± 0.1645	40	229	6.13b
CaNNCM-15	(2)[7]	0.3181 ± 0.2535	0.3208 ± 0.1905	40	229	6.13c
CaNNCM-16	(2)[8]	0.2963 ± 0.2268	0.2887 ± 0.1767	40	323	6.13d
CaNNCM-17	(2)[1,2,3,4,5]	0.3670 ± 0.2785	N/A	108	2313	6.13e
CaNNCM-25	(7)[9]	0.3045 ± 0.3263	N/A	37	286	6.17a
CaNNCM-26	(7)[9,10]	0.2270 ± 0.1964	N/A	43	1032	6.17b

of both inclusions. Volumetric renderings for Figs. 6.10a,d are displayed in Figs. 6.10e–f.

Notice that the CaNNCMs are slightly underestimating the Young's modulus of the inclusions as more data sets are input to AutoP. The difference

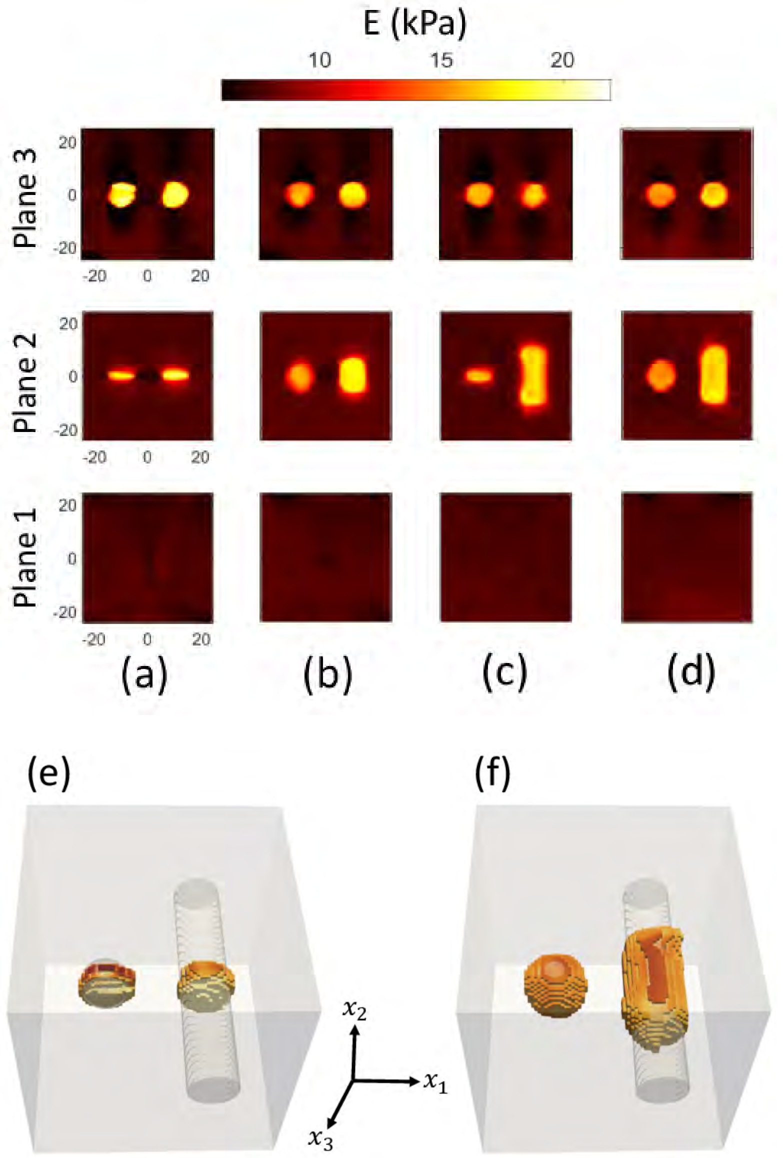


Figure 6.10: Young's modulus reconstructions by CaNNCMs trained on simulated data from Model 4. Using ROIs (a) 1, (b) 1–3, (c) 1, 4, 5, and (c) 1–5. (e–f) Volumetric reconstruction of the inclusions from (a) and (d).

is slight and the modulus errors reported in Table 6.2 increase marginally. Interestingly, the spherical inclusion appears to be more affected. We hypothesized it was because all five ROIs passed through the cylindrical inclusion, but only ROIs 1–3 intersected the spherical one. To test this, we defined ROIs 6–8 shown in Fig. 6.11a. ROI 7 is just ROI 1 rotated 90° around the x_2 axis. Moving this plane +10 mm and $-10\ mm$ along the x_1 axis produces ROIs 6 and 8, respectively.

We first trained CaNNCM-8 with simulated data from ROIs 1, 6, and

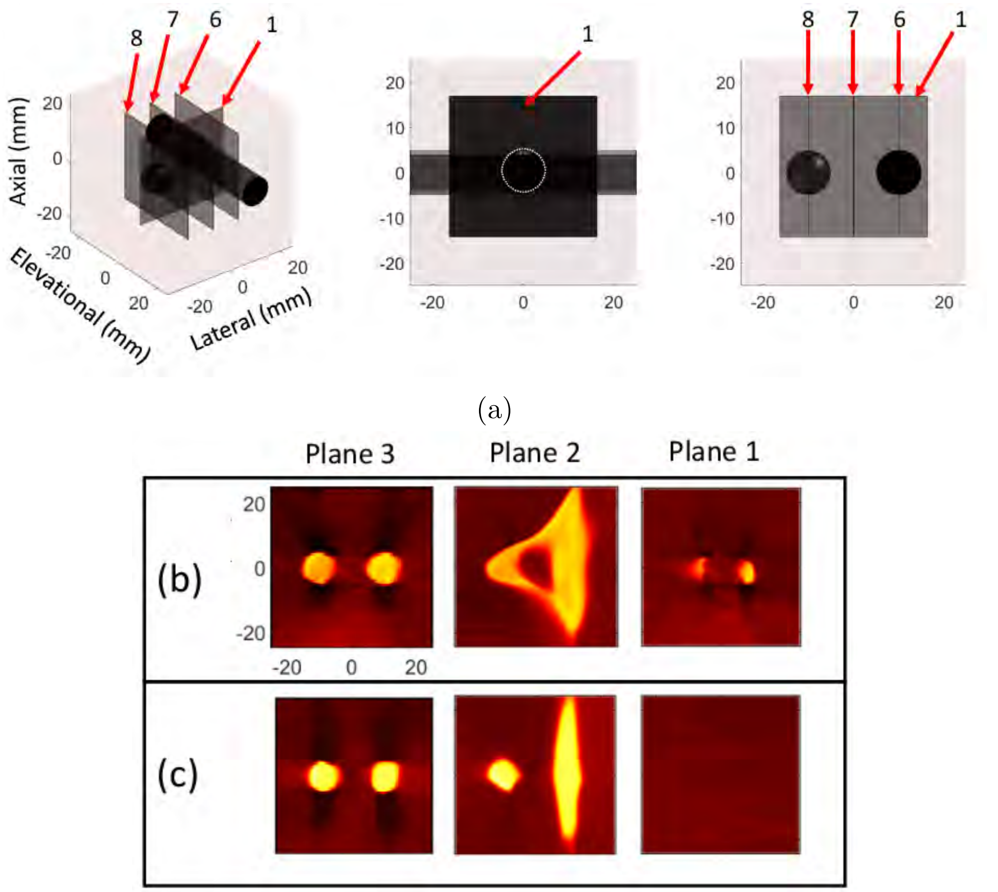

(a)

Figure 6.11: (a) Additional ROIs for training CaNNCMs. Young's modulus reconstructions by CaNNCMs trained with data from ROIs (b) 1, 6, 8 and (c) 1, 6, 7, 8.

8. Two of the three ROIs pass through each inclusion. Fig. 6.11b is the Young's modulus reconstruction by CaNNCM-8. We note that with only three planes, the shape of both inclusions is apparent and they exhibit approximately the same Young's modulus. It is also clear the CaNNCM did not quite correctly learn the mechanical properties of the background material between the inclusions as they appear connected, hence the significant error increase in Table 6.2. The inclusions are clearly separated within the plane $x_3 = 0$, though, the same plane in which ROI 1 resides. However, including ROI 7 during training (ROIs 1, 6, 7 and 8 total) adds the extra information necessary for CaNNCM-9 to learn the two inclusions are indeed separate. Also note that both inclusions exhibit approximately the same material properties, as expected.

Continuing the pattern observed with Models 2 and 4, CaNNCM-10 trained with only ROI 1 for Model 5 learns the material properties only within

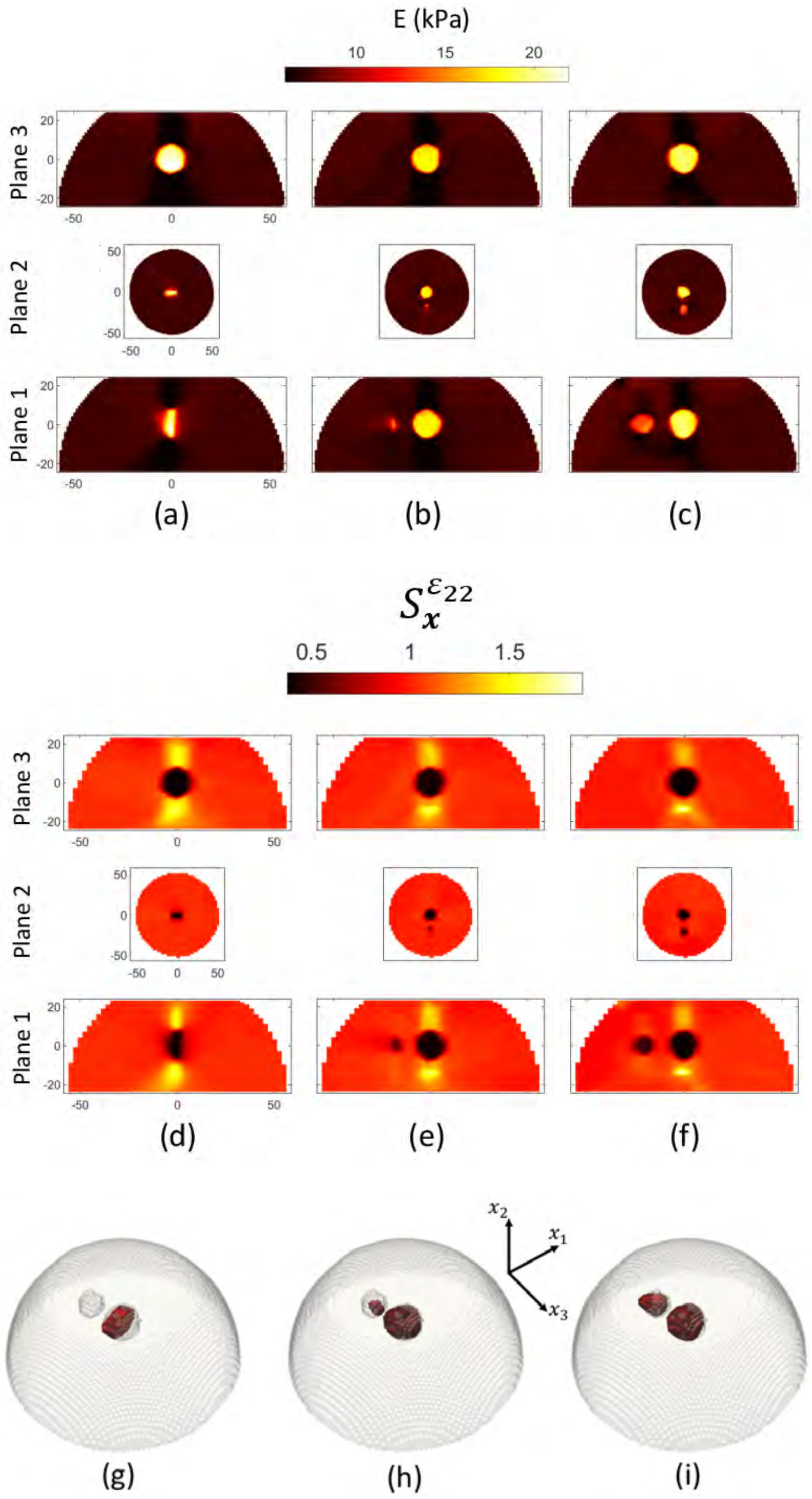


Figure 6.12: Young's modulus reconstructions by CaNNCMs trained with simulated data from Model 5, ROIs (a) 1, (b) 1–2, and (c) 1–3. (d–e) Axial strain scaling values $S_{\boldsymbol{x}}^{\varepsilon_{22}}$ from the corresponding SN in (a–c). (g–i) Volumetric modulus reconstructions of the inclusions.

the plane, as shown in Fig. 6.12a. Including data from ROI 2 (CaNNCM-11, Fig. 6.12b) or training with data from all three ROIs (CaNNCM-12, Fig. 6.12) provides the CaNNCM additional information necessary to learn the 3-D Young's modulus distribution of the inclusions, albeit the smaller spherical inclusion becomes properly visible only when ROI 3 is included. Given the previous results, this is expected because ROI 3 is the only data set that fully intersects said inclusion. Figs. 6.12g–i better illustrate the effect of ROIs on learning the inclusion shapes.

Figs. 6.12d–e show the axial strain scaling value $S_{\boldsymbol{x}}^{\varepsilon_{22}}$ output from the SN once AutoP training completed. Increased strain appears above and below the large central inclusion and propagates through to the scaling values. Even though it is not as apparent in Figs. 6.12a–c, the Young's modulus estimate in these regions is less than the expected value. Not only do the inclusion shapes become more visible as additional ROIs are input to AutoP, the "softening" artifact is reduced. These results provide some evidence that proper sampling is not only important for estimating internal structure, but for learning the correct material properties.

CaNNCMs-13–17 were trained with experimentally acquired force-displacement measurements on Model 2 and the corresponding Young's modulus reconstructions are shown in Fig. 6.13. The first four were each trained with a single data set (ROI 1) measured from the four sides of the phantom. As such, they only capture the material properties within a single plane. CaNNCM-17 was trained with Data Sets -1–5 (i.e., ROIs 1–5) and was therefore able to characterize more of the phantom volume at the cost of a significant increase in computation time, according to Table 6.2.

In general, the modulus error was larger for the 3-D CaNNCMs compared to their 2-D counterparts, the latter are included in Table 6.2 for direct comparison. It is not clear why this occurred. From a qualitative perspective, the 3-D Young's modulus reconstructions are similar to the 2-D images for CaNNCMs trained with the same Data Sets. In fact, for some cases the 3-D version appears to better characterize the materials. This is best exemplified by the results of CaNNCMs trained with Data Sets 6 and 8 (Data Sets 4 and 6 in chapter 5). The 2-D CaNNCMs did not correctly characterize the lower inclusion as being the stiffest of the three. CaNNCMs-14 and 16 did accurately represent the relative modulus values. Nevertheless, the 3-D CaNNCMs were able to estimate the Young's modulus of the phantom within

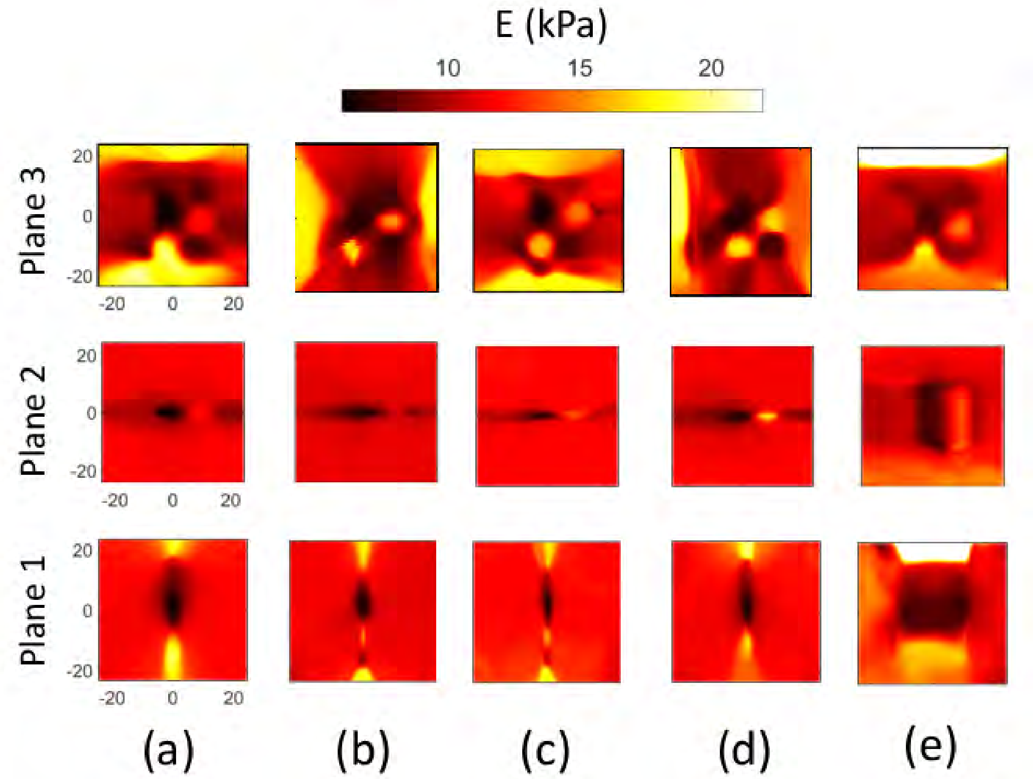


Figure 6.13: Maps of Young's modulus by CaNNCMs trained with experimental force-displacement measurements acquired on the gelatin phantom. Data Sets entered into AutoP: (a) 1, (b) 6, (c) 7, (d) 8, and (e) 1–5.

the ROI with reasonable accuracy.

The caveat to the above is that the Young's modulus distributions displayed in Fig. 6.13 were estimated with $\nu = 0.3$ — the Poisson's ratio used for pretraining — instead of $\nu = 0.5$. Adopting a Poisson's ratio for an incompressible material resulted in significantly over-estimated Young's modulus values. However, the modulus distributions in Fig. 6.9c were estimated with $\nu 0.495$, which matches the Poisson's ratio of the forward model. We do not yet understand why the CaNNCMs trained with experimentally acquired force-displacement data did not correctly capture the incompressible nature of the gelatin phantom, but the CaNNCM trained with simulated data did. If we compute the modulus error values in Table 6.2 by comparing the *shear* modulus μ instead of Young's modulus, we obtain the same values. Meaning, the CaNNCMs appear to be learning the shear modulus, but data available during training is insufficient for learning the Poisson's ratio and thus the Young's modulus.

CaNNCM-27 was trained with the force-displacement measurements acquired on the rabbit kidney phantom described in Sec. 5.3.2. For the 2-D case, three different networks were trained with data from ROIs 1–3 as defined for Model 2. Here, a single CaNNCM was trained with the same three data sets, plus two more corresponding to ROIs 4 and 5, although the former was positioned at $x_3 = 6$ mm.[2]

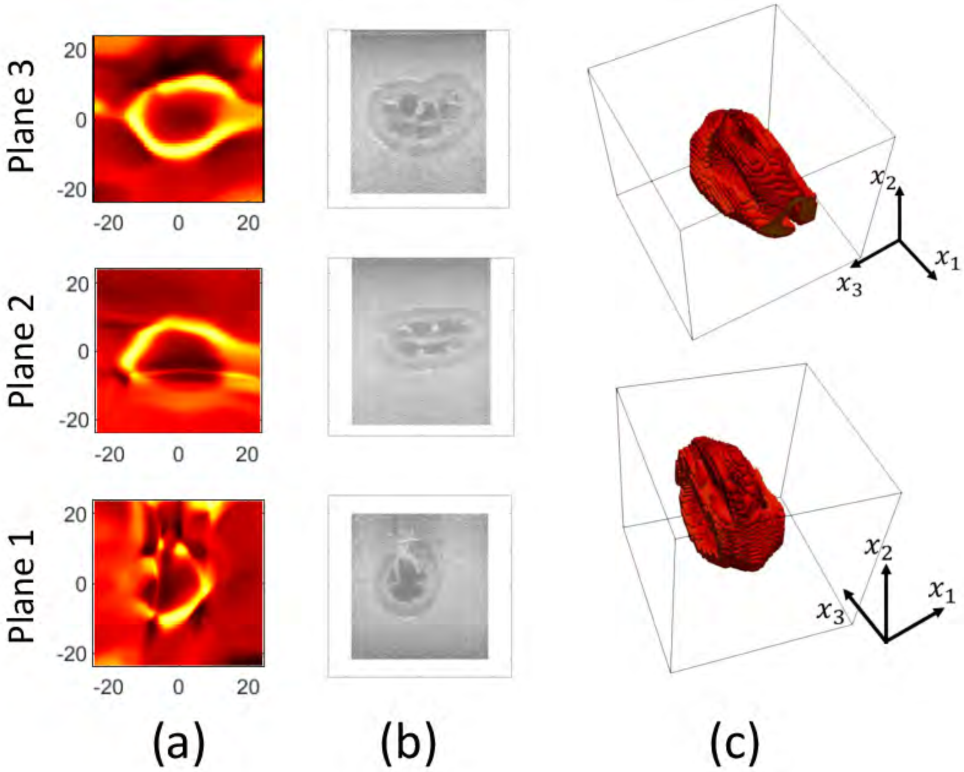


Figure 6.14: (a) Young's modulus reconstruction by CaNNCM trained with data acquired on rabbit kidney phantom. (b) US B-mode images in the corresponding planes. (c) 3-D renderings of the renal cortex.

Fig. 6.14a displays the Young's modulus reconstruction by CaNNCM-27. The B-mode images in Fig. 6.14b were acquired in approximately the same planes as the Young's modulus images. Unlike the 2-D CaNNCMs, we do not observe significant stiffening of the renal cortex in Plane 3 at the proximal and distal edges. Nor is there evidence of the cortex appearing softer as the probe moves in the elevational direction. In short, what appeared as possible anisotropy of the renal cortex in 2-D does not appear for the 3-D CaNNCM.

[2] To be clear, the ROIs were positioned at $x_3 = 0$, ± 4 mm, $+6$ mm, and -8 mm. The image plane resulting from aligning the probe at $x_3 = 8$ mm did not intersect the kidney. We did not know sampling requirements at the time of data acquisition and thus moved the probe so the kidney was in view. If this experiment was performed again, we would stick with the ROIs defined for Model 2.

While this is insufficient to make a definitive conclusion, these results do suggest 3-D modeling does reduce some of the artifacts and errors that occur in a 2-D approximation.

Materials Under Large Deformation

Three different CaNNCMs were trained with data from Model 2 under large deformation. All were trained with data from ROI 1 only. Fig. 6.15 is the Young's modulus distributed estimated by CaNNCM-18 trained with simulated data and accounting for geometrically non-linear effects. It is nearly indistinguishable from its linear counterpart (Fig. 6.9a). Two different CaNNCMs were trained with experimental measured data and the results are shown in Figs. 6.15b–c. The first was trained under small-strain assumption (CaNNCM-19, Fig. 6.15b) whereas the second accounted for geometric non-linearity (CaNNCM-20, Fig. 6.15c). To be clear, both networks were trained with the same force-displacement data, the difference resides in the FEA formulation (small or large deformation) and the corresponding definitions of stress and strain.

There are two particularly interesting aspects of these results. First, the modulus errors listed in Table 6.2 are smaller compared to their geometrically linear counterparts. Second, in regards to the CaNNCMs trained with experimental measurement data, CaNNCM-20 (trained under a large deformation assumption) resulted in a smaller modulus error compared CaNNCM-19, even smaller than the results obtained with a 2-D CaNNCM. While certainly inconclusive, these results suggest the magnitude of the applied force load may effect the ability of CaNNCMs to learn material properties. We will return to this idea in later work when CaNNCMs are further developed to characterize the non-linear and viscoelastic properties of materials.

CaNNCMs-21–24 were trained with noise-free data from Model 6. Fig. 6.16 contains the modulus estimates by these networks. All AutoP analyses used a geometrically non-linear FEA formulation. CaNNCM-21 and 22 were both trained with data from ROI 1 only, but the MPN of the former learned the relationship between Cauchy stress and Almansi strain whereas the latter mapped Green strain to 2nd P-K stress. We performed these tests to illustrate the importance of choosing an appropriate definition for stress and strain. This is particularly important for a situation like Model 6 where the

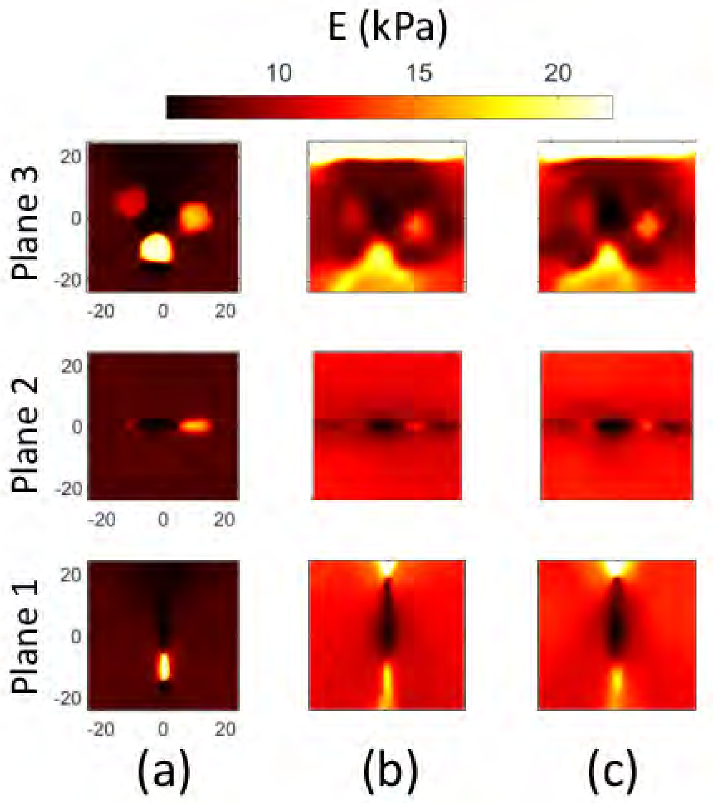


Figure 6.15: Young's modulus reconstructions by CaNNCMs on objects under large deformation. (a) Force-displacement data simulated for Model 2. (b–c) Experimental measurements on gelatin phantom, but (b) not accounting for large deformation and (c) incorporating geometric non-linearity in AutoP.

displacements become fairly large. Figs. 6.16 are the modulus estimates by CaNNCMs-21 and 22, respectively. There are larger artifacts produced by CaNNCM-21 compared to CaNNCM-22 and the latter resulted in a lower modulus error (Table 6.2).

Figs. 6.16 display the Young's modulus distributions reconstructed by CaNNCM-23 (trained with ROIs 1–2) and CaNNCM-24 (trained with ROIs 1–3). Volumetric representations of the inclusions for Figs. 6.16b–d are shown in Figs. 6.16e–g. Following the pattern observed in the previous results, multiple ROIs are necessary to reveal the 3-D structure of the inclusions.

Young's modulus distributions reconstructed by CaNNCMs-25–26, trained with force-displacement data measured experimentally on Model 7, are presented in Fig. 6.17. The same observations hold here as for previous results: the spherical shape of the inclusion only becomes apparent when multiple data sets are used for training. More importantly, the external shape of the phantom was estimated using a Microsoft Kinect. We did not expect the

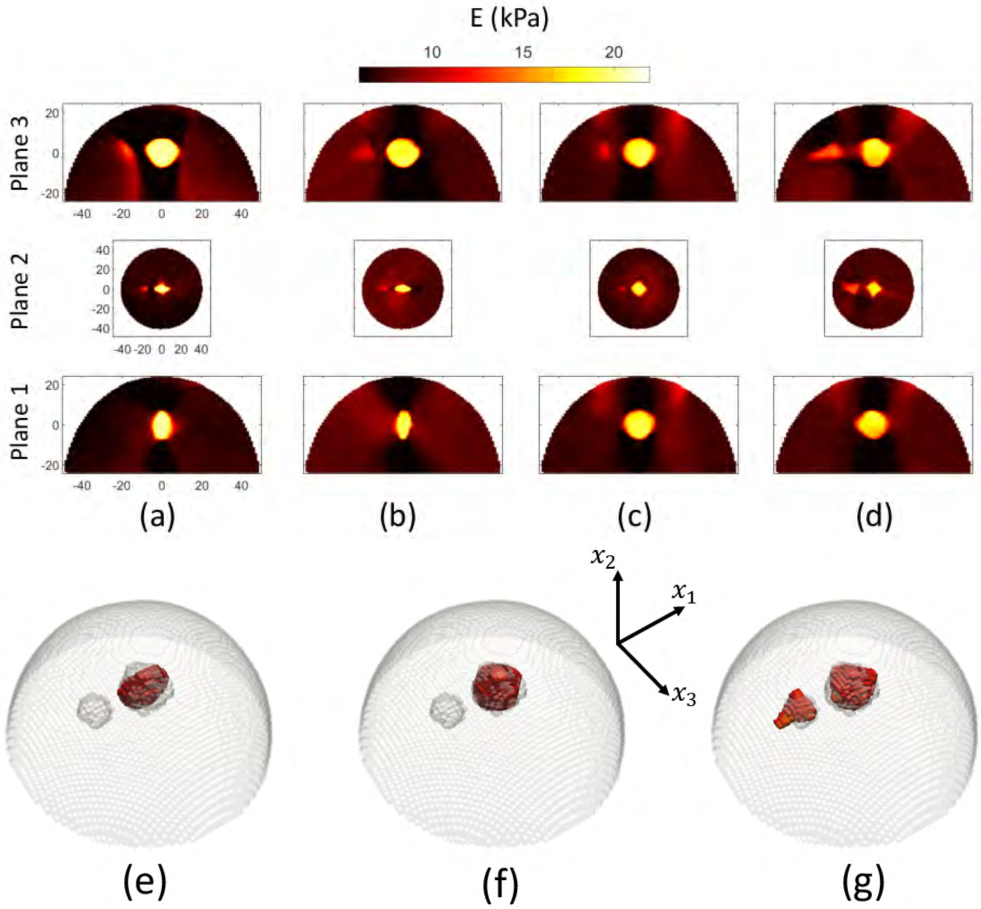


Figure 6.16: Young's modulus reconstructions by CaNNCMs trained with simulated force-displacement data from Model 6. Displacements input to AutoP from ROIs (a) 1, but not accounting for geometric non-linearity, (b) 1, (c) 1–2, and (d) 1–3. (e–f) Volumetric reconstruction of the inclusions, corresponding to (b–d).

results of these two tests to reveal new insights into the sampling requirements for 3-D CaNNCM training. Rather, they suggest the networks are robust to geometric modeling errors, providing evidence for the feasibility of translating this technology to a clinical setting.

6.4 Discussion

This chapter introduced the development and first implementation of 3-D CaNNCMs in AutoP. Using only measurements of axial force applied by the probe and axial displacements within finite planes, 3-D CaNNCMs are able to learn the spatial distribution of linear-elastic material throughout a volume and reveal the structure of internal objects. Unlike prior approaches to

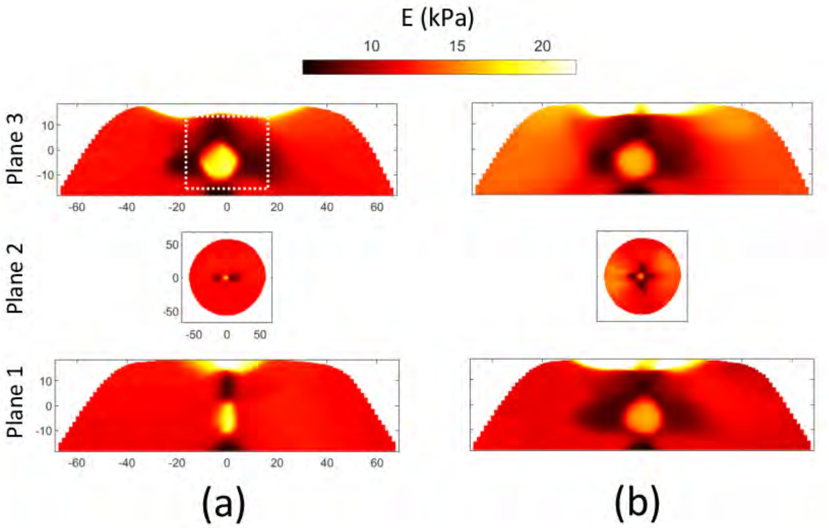


Figure 6.17: Young's modulus reconstruction by CaNNCMs trained with experimental force-displacement measurements on Model 7, using Data Sets (a) 9 and (b) 9–10.

3-D QUSE, our method does not "stack" a sequence of planar Young's modulus reconstructions nor require full 3-D displacement data within a volume. AutoP exploits to information content of quasi-statically applied loads and finite element analysis to recover material properties from a small number of strategically selected force-displacement measurements.

Including displacements within a single plane during training provides enough information for the CaNNCM to learn the material properties within the plane, but very little is learned about the 3-D structure. For Model 2, we demonstrated that training with five parallel planes separated by 4 mm is sufficient to reconstruct the Young's modulus within the 16 mm elevational span of the data. Separation of the planes was determined through preliminary testing. In Fig. 6.9a, where CaNNCM-1 was trained with data from ROI 1, the network learns some of the 3-D shape of the inclusions. More specifically, the Young's modulus extends $\approx \pm 2$ mm in the x_3 direction. Assuming the same holds for a network trained with any one data set, the reconstructed volumes "connect" if the data planes are separated by 4 mm. Hence, ROIs 1–5 for Models 2 and 4 are separated by 4 mm.

Results from Model 4 revealed some of the more interesting effects of spatial sampling. Training with ROIs 1–3 was sufficient for CaNNCM-5 to learn the full shape of the spherical inclusion and ≈ 10 mm length of the cylindrical one. Increasing the distance between the planes of data to 8 mm and training CaNNCM-6 with ROIs 1, 6, and 8 resulted in the network reconstructing more of the cylindrical inclusion volume while capturing very little of spherical inclusion structure. These results suggest that an effective sampling strategy is one where planes of data 1) intersect objects of interest and 2) are sufficiently close to interact with each other. With this hypothesis, we defined ROIs 6–8 for Model 4. Training CaNNCM-8 with data from ROIs 1, 6, and 8, resulted in a Young's modulus reconstruction where the shapes of both inclusions are mostly apparent, but the network learned incorrect mechanical properties of the background material between the inclusions. Notice, though, that CaNNCM-8 did capture the correct Young's modulus between the inclusions at $x_3 = 0$, the same plane of ROI 1. The inclusions connect at $\approx x_3 = \pm 8$ mm, suggesting the distributed effects of displacements applied at ROIs 1, 6, and 8 in FEA$_\varepsilon$ do not reach these regions. Following our prior reasoning, including displacement data within a plane between the inclusions should be sufficient for the network to learn the correct material property distribution. This hypothesis was corroborated by training CaNNCM-9 with ROIs 1, 6–8. Displacements within ROI 7 provided the additional information necessary for the network to more accurately learn the Young's modulus distribution of the model.

The choice of ROIs for Model 5 was informed by the results from Models 2 and 4. With only three well-placed data planes for training, CaNNCM-12 was able to learn the shape and material properties of both spherical inclusions. ROIs 1 and 2 for this Model were orthogonal and thus maximize the information content while minimizing redundancy. For example, if ROI 2 was only rotated 10° around the x_2 axis instead of 90°, the amount of new information about the shape of the central inclusion in the x_3 direction compared to ROI 1 would be minimal. Given the symmetry of spheres, two orthogonal ROIs appear to be sufficient for a CaNNCM to learn their shape. More complex structures would likely require several more data sets, either acquired in parallel planes,through several rotation angles, or a mix of both. It is reasonable to believe that CaNNCM-12 could have achieved the same results if multiple parallel ROIs were input to AutoP in lieu of orthogonal planes of data, but several more data sets would have been required to fully sample both inclusions. This is not an issue from a sampling perspective; rather, more data sets means more FEAs must be solved, causing a significant increase in training time. The best sampling strategy will likely depend on the task. If the goal is to accurately reconstruct the 3-D material property distribution of a particular structure, several ROIs should be acquired by rotating the probe, effectively acquiring a cylinder of force-displacement measurements encompassing said structure. More exploratory imaging where the goal is to reconstruct the material properties of a large volume would benefit more from acquiring data in several parallel ROIs.

CaNNCMs trained with data from Models 4 and 5 in particular begin to touch on the idea of a sufficient training data set. As we discussed, CaNNCM-8 (Fig. 6.11b) was unable to learn that the two inclusions were discrete structures. Only after ROI 7 was included to train CaNNCM-9 (Fig. 6.11c) did the network learn the correct inclusion geometries. This implies that data planes which do not directly intersect structures of interest can be important to clearly delineate internal boundaries. On the other hand, maps of $S_{\boldsymbol{x}}^{\varepsilon_{22}}$ in Figs. 6.12d–f illustrate the importance of sampling for learning the correct material properties. However, the artifacts surrounding the central inclusion that propagate through to the Young's modulus reconstruction could potentially be mitigated by allowing $\boldsymbol{S}^{\sigma}$ to vary spatially. The source of this artifact is particularly difficult to trace because it does not appear when $\boldsymbol{S}^{\varepsilon}$ is computed with known stresses and strains (e.g., Figs. 6.4a–c). Future work will involve investigations into reformulating Algorithm 2 to simultaneously compute $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ and a spatially varying $\boldsymbol{S}^{\sigma}$.

Young's modulus reconstructions by CaNNCMs-1, 6, 7, 8, and 17 were comparable to their 2-D counterparts, but raise the issue of how to properly estimate material parameters from a trained network. Recall that the modulus distributions were estimated with $\nu = 0.3$, the same Poisson's ratio selected for pre-training, instead of the expected $\nu = 0.5$. Conversely, the Young's modulus was estimated by CaNNCM-3 using $\nu = 0.495$, matching the Poisson's ratio used for generating the simulated data, not the pre-training value. It appears noise in the force-displacement measurements affected the ability of the CaNNCMs to learn the correct Poisson's ratio. However, the networks learn accurate values of the shear modulus, which does not depend on Poisson's ratio.

The same issue occurs for CaNNCMs-19 and 20 that were trained with experimentally acquired force-displacement data where the latter accounted for non-linear geometric effects in the FEA formulation. Interestingly, this CaNNCM provided the most accurate Young's modulus estimate, even better than the 2-D CaNNCMs. We have trained other CaNNCMs on simulated data from Model 2 under very small deforming forces where the total strain was ≤ 0.005. The results are not shown, but the networks were unable to learn the correct material properties of the inclusions. While inconclusive, the results of CaNNCMs trained with data under large and very small deforming forces suggest the importance of applying a sufficiently large load to properly probe the material properties. Increasing the force load effectively increases the SNR of the force-displacement measurements.

Model 6 was intended as a more extreme case of a material undergoing large deformation. As we expected, a CaNNCM trained using the Cauchy stress - Almansi strain relationship introduced more errors into the Young's modulus reconstruction than a CaNNCM learning the relationship between 2nd P-K stress and Green strain. Even though both relationships describe the same material, the current CaNNCM architecture is unable to account for a changing reference configuration. That said, more complex materials may be better described by relating stress and strain *rates*, in which case Cauchy stress is the better choice. Previous investigation with NNCMs and AutoP have described non-linear and time-dependent materials [102, 103, 104, 106, 155], some in terms of stress-strain rates, hence we believe the CaNNCM architecture can be adapted to accommodate different types of stress-strain relationships.

We introduced a method to estimate external geometry with Model 7. To the best of our knowledge, all previous investigations involving AutoP assumed the surface shape was known. Imaging of an unconfined breast precludes knowledge of the tissue shape and therefore requires a method to estimate surface geometry. A Microsoft Kinect v2 was chosen due to its low cost and the available software development tools. The shape of Model 7 (and Model 5) was chosen because it loosely resembles the shape a of breast under gravity loading. At this time, we have not evaluated the ability of the Kinect to capture sharp edges and corners (e.g., Model 2) because it is unlikely for breast tissue to conform to such shapes.

Finally, the values listed in Table 6.1 provide some insight into how the AutoP training parameters should be selected. Increasing the number of passes as the number of data sets increases is fairly intuitive: more AutoP iterations are required to reconcile the stresses and strains generated in FEA$_\sigma$ and FEA$_\varepsilon$ of a larger force-displacement data set. The number of elements from which stress-strain data were used for training the MPN is generally inversely proportional to the number of data sets. We selected the of this subset to keep the number of stress-strain pairs relatively constant. For example, only one data set was used to train CaNNCM-13, so the number of stress-strain pairs used for training the MPN was $400 \times 4 \times 4 \times 4 = 25600$. We decreased the number of elements for CaNNCM-17 to 75, reducing the training data set to $75 \times 5 \times 4 \times 4 \times 4 = 24000$. N_ε and η_ε were similarly adjusted based on the number of force-displacement data sets. With more data, Algorithm 2 requires more iterations to converge, but more iterations increases the computation time. Increasing η_ε can also increase the rate of convergence. The caveat is that selecting a large value of η_ε can cause the algorithm to become unstable, which is why we chose $\eta_\varepsilon = 0.25$ for Model 6. Therefore, both N_ε and η_ε are slightly increased as more force-displacement data sets are input to AutoP to decrease the computation required for Algorithm 2.

6.5 Conclusion

3-D Cartesian neural network constitutive models trained in the Autoprogressive method can learn linear-elastic material properties throughout a volume from force-displacement measurements acquired in planes. The ability of CaNNCMs to learn full 3-D structure is dependent upon the number and location of measured data sets. Noise in the force-displacement measurements not only causes geometric errors, but appears also to reduce the ability of a 3-D CaNNCM to learn the correct Poisson's ratio even though it correctly identifies the shear modulus. Current limitations of the method as applied to linear-elastic media suggest a need for more efficient FEA computation to further develop this method to non-linear and viscoelasticity imaging.

Chapter 7

Concluding Remarks

Elasticity imaging provides a means to probe and map the mechanical properties of tissues to visualize a source of contrast not captured by other medical imaging modalities. Quasi-static ultrasonic elasticity imaging is a well-defined method to reconstruct images of axial strain, but quantifying material properties remains a challenge. Much effort over the last couple of decades has been spent on model-based approaches to the inverse problem in QUSE. These methods take advantage of mechanical principles and physical laws to extract information from measurement data and estimate material parameters associated with a pre-selected constitutive model. However, model-based methods may discard important information about mechanical properties encoded in the data. The mismatch between assumed constitutive model and true material properties governing the measurements results in modeling errors. Not only can such errors introduce artifacts in to the final elastograms and make their interpretation more difficult, it is not always obvious when a modeling error occurs. Model-based inverse methods are limited in their ability to explore and discover material properties most relevant to clinical imaging.

Our solution is to use a data-driven approach to the inverse problem. Replacing the classic constitutive model with a neural network constitutive model provides the flexibility to characterize virtually any material behavior expressed by measured data. In QUSE, these measurements are in the form of compressive forces applied by the US probe along with a sparse sampling of surface and internal displacements. Measured data are input to the Autoprogresssive method to train NNCMs on the stress-strain behavior governing the force-displacement measurements. Only after training in AutoP are NNCMs interrogated to estimated material parameters. At this time, we are limited to estimating Young's modulus (or shear modulus) due to the architecture of NNCMs used for QUSE.

We first adapted NNCMs and AutoP for elasticity imaging in Chapter 3. When the internal structure of the object is known, we demonstrated that displacement estimates at a few well-chosen locations are sufficient for training NNCMs on the linear-elastic stress-strain behavior. The fully trained networks were then used to reconstruct maps of all stresses and strains. To the best of or knowledge, this is the first time accurate estimates of all stresses and strains were measured from force-displacement measurements without a constitutive model assumption. Maps of these fields were then used reconstruct accurate estimate of the Young's modulus for gelatin phantoms. We also presented results of imaging a rabbit kidney embedded in a gelatin cube in both 2-D and 3-D. Young's modulus estimates of the kidney structures were in agreement with reports by other investigators.

Imaging in a clinical setting precludes prior knowledge of internal structures. Interfaces between regions exhibiting unique material properties will not necessarily match boundaries observed in B-mode imaging. Nor are the transitions between regions always discrete. The NNCMs in Chapter 3 are thus not suitable for clinical breast imaging because they 1) rely on a FE mesh conforming to both internal and external geometry and 2) only capture discrete material property distributions. We created Cartesian NNCMs to overcome these limitations (Chapter 4). Splitting the CaNNCM architecture into a material property and spatial network was not an obvious solution. We tested a multitude of network structures before discovering a pair of *cooperating* networks was an appropriate choice for data-driven QUSE. We emphasize the cooperative nature of the MPN and SN because generative adversarial networks are currently of major interest and operate on very different principles. Using a pair of ANNs allows the overall problem to be split into two smaller tasks: the MPN learns a "reference" material property while the SN learns spatial variations of this reference. In case of linear-elastic materials, the SN effectively maps the relative stiffness.

Given that CaNNCMs are tasked with learning both material property and geometric information, it was necessary to increase the sampling density of displacements input to AutoP. In Chapter 5, we investigated the effects of sampling density on Young's modulus reconstruction. The negative effective of noise in the displacement measurements can be reduced by using fewer samples, with the caveat that decreasing the sampling density reduces the resolution of the final Young's modulus image. For the phantom geometries explored, we concluded that imposing displacements at points separated by 1.5 mm maintained accurate Young's modulus estimates without significantly losing resolution. This separation may have some dependency on the FE mesh. A more refined mesh (i.e., one comprised of smaller elements) may permit a higher sampling density to better resolve fine structures. Conversely, a coarser mesh may require even sparser sampling to mitigate the effects of noise.

Extending CaNNCMs to 3-D imaging (Chapter 6) created a new challenge in terms of sampling. We investigated strategies to effectively and efficiently acquire force-displacement measurements for accurately reconstructing the Young's modulus distribution within a volume. The most straight-forward method is to scan the US probe in the elevational direction to collect a series of measurements in parallel planes, thus filling a volume. But, given the way in which quasi-static loads propagate through a body, the distance between data planes need not be on the order of elevational width of the ultrasound beam. And while this method of scanning might be effective, it may not be the most efficient. If the task is to estimate the material properties and geometry of a particular structure, a more efficient sampling strategy may be to rotate the probe to acquire force-displacement measurements within a cylindrical volume. These insights could be particularly important for imaging in a clinical setting. First, a raster scan to coarsely estimate material properties and geometry, followed by a finer sampling of suspicious lesions.

7.1 Future Directions

7.1.1 Human Imaging

Training CaNNCMs requires knowledge of external object shape along with measurements of the force applied by the US probe and its displacement as compressive loads are applied. We described a method to estimate surface shape using a Microsoft Kinect v2 in Appendix G. Measuring forces applied by the US probe has been done by other researchers (e.g., [168, 24, 34]). The more difficult problem is measuring probe displacement and orientation in relation to the tissue. At this time, we do not know how errors in probe displacement estimates will affect the ability of CaNNCMs to learn material property distributions. For more precise free-hand QUSE, we can adapt a co-robotic system (e.g., [169]). We have also developed a preliminary method to track the US probe using RGB imaged captured by the Kinect, although it introduces more uncertainty than a robotic arm. Future investigation will involve evaluating the trade-off between equipment cost and imprecision.

7.1.2 Estimating Non-linear and Viscoelastic Parameters

CaNNCMs are currently only capable of characterizing linear-elastic material properties. We introduced in Sec. 6.3 a method to accommodate materials under large deformation which may be sufficient for estimating hyperelastic material parameters, although this is untested. Given prior studies involving AutoP to characterize non-linear, time-dependent, and rate-dependent materials [99, 100, 101, 102, 103, 104, 106, 107], we are reasonably certain the CaNNCM architecture can be modified to learn more complex mechanical behaviors. Investigators in the previous studies typically introduced history points at the input of the NNCM to provide the additional information necessary for the network to learn such properties. If a similar approach is to be used for CaNNCMs we must determine if history points are input to the MPN, SN, or both, and the merits of each option.

In addition to determining an appropriate CaNNCM architecture, characterizing non-linear and viscoelastic material properties will likely require new sampling strategies. Increased force load to cause larger deformation will presumably be necessary to probe non-linear mechanical behaviors. We provided an example in Chapter 1 of other investigators estimating the linear and non-linear parameters of the Veronda-Westmann model constitutive model. The objective function defined by Oberai *et al.* [50] in (1.12) required two displacement fields: one under small deformation the other induced by a larger deforming force. We expect similar principles to apply for CaNNCMs.

Viscoelasticity imaging will necessitate sampling through time as well as space. Current model-based approaches to quasi-static viscoelasticity imaging rely on tightly controlled force-loading (usually a step or ramp force) and observation of tissue creep or relaxation [51, 24, 53, 54]. Strict adherence to load application and measurements of tissue motion is required by the chosen viscoelastic model. Given the flexibility of CaNNCMs it is unclear how these sampling requirements will translate to CaNNCMs.

7.1.3 Decreasing AutoP Training Time

The most important drawback to using CaNNCMs and AutoP for QUSE is arguably the amount of time required for training. Clinical use of this method will require a dramatic reduction in computation time. Ideally, CaNNCMs will be trained in real-time to allow feedback during the imaging exam. To achieve this goal, we have identified the processes in AutoP that must accelerated: 1) FEA$_\sigma$ and FEA$_\varepsilon$, 2) computing $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$, and 3) training the SN.

We first brought attention to the computational load for solving FEAs in Sec. 3.4.4. As we discussed, there is effort by other groups to implement FEA methods on GPUs, reducing the solve time to seconds (e.g., [125, 126]). We hope to achieve similar speeds with a custom FEA solver specifically suited for CaNNCMs and AutoP. Similarly, computing $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ for 3-D CaNNCMs can take minutes, depending on the size of the FE mesh, number of force-displacement measurements input to AutoP, the size of the training window, and N_ε. Algorithm 2 increases the computation speed by orders of magnitude on a CPU compared to Algorithm 1 and made 3-D imaging feasible. This speed was obtained by re-casting the algorithm as a series of matrix-matrix and matrix-vector multiplications, operations that are well-suited for GPUs. Therefore, it is reasonable to believe we can further increase the speed of Algorithm 2 to real-time.

Reducing training time for the SN is trickier. The TensorFlow implementation of training used for the SN is GPU-enabled but is not fast enough for real-time imaging. The primary cause of the sheer number of training pairs. For example, a mesh comprised of 10000 10-node tetrahedral elements will generate 40000 values of $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ distributed throughout the volume. That is an average distance of 0.25 mm between points in the body for a $50 \times 50 \times 50$ mm^3 cube. Unless the goal is to resolve structures with very small features, there is no need to training with all available data. A straight-forward way to achieve a speed increase is to reduce the number of training pairs by selecting a subset of all $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$. The difficulty arises in determining an appropriate subset to simultaneously reduce training time and preserve resolution. One simple idea is to sparsely sample $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ in homogeneous regions and increase sampling density within and around finer structures. Of course, such a heterogeneous sampling density cannot be defined before training because the internal geometry is unknown. An adaptive strategy could be implemented based on the spatial distribution of displacement errors computed in (5.11).

7.1.4 Heterogeneous Constitutive Modeling

Model-based inverse methods for QUSE assume the same constitutive model for all tissues in the image region. As we further develop CaNNCMs, it may be possible to estimate different material parameters based on the mechanical properties learned by the network. For example, a lesion could exhibit strain-stiffening behavior whereas the surrounding tissue expresses strain-softening properties. A model-based approach would assume all tissues either stiffen or soften under an increasing compressive load and therefore discard important information about the distribution of material properties. Conversely, the CaNNCM architecture could potentially be adapted to learn such a heterogeneous distribution of mechanical behaviors, further adding to their capabilities for discovering material parameters most relevant to clinical imaging.

Appendix A

Enforcing Frame Invariance

Stresses and strains generated in FEA$_\sigma$ and FEA$_\varepsilon$ are with respect to one coordinate system. Much information about the material properties are embedded within these data, but the concept of frame invariance (or objectivity) is not captured. Stated simply, frame invariance means the description of the material properties is unchanged for different reference frames (e.g., for different coordinate systems) [170, 84]. The structure of the NNCMs does not inherently encode this behavior. Therefore, the training data can altered to express frame invariance, allowing the network to learn this fundamental property. One simple method is to express the stress and strain tensors with respect to a new coordinate system and add the rotated data back to the original set.

Consider the stress tensor $\boldsymbol{\sigma}$. It can be expressed with respect to a new coordinate system by through a rotation defined by $\boldsymbol{R}$ (where $\boldsymbol{R}^{\mathrm{T}}\boldsymbol{R} = \boldsymbol{I}$). Given that $\boldsymbol{\sigma}$ is a 2nd order tensor, the rotation is performed as a similarity transform:

$$\boldsymbol{\sigma'} = \boldsymbol{R}\boldsymbol{\sigma}\boldsymbol{R}^{T}. \tag{A.1}$$

Matrices $\boldsymbol{R}_{x_1}$, $\boldsymbol{R}_{x_2}$, and $\boldsymbol{R}_{x_3}$ rotate the tensors about x_1, x_2, and x_3, respectively. NNCMs do not require the stress-strain data to be rotated through many angles. The computational burden can be significantly reduced by rotating the data $\frac{\pi}{2}$ around each axis. Each rotation matrix is thus defined as

$$\boldsymbol{R}_{x_1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \xrightarrow{\theta=\frac{\pi}{2}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \tag{A.2}$$

$$\boldsymbol{R}_{x_2} = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \xrightarrow{\theta=\frac{\pi}{2}} \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{A.3}$$

$$\boldsymbol{R}_{x_3} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \xrightarrow{\theta=\frac{\pi}{2}} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{A.4}$$

Carrying out the similarity transform in (A.1) using the rotation matrix $\boldsymbol{R}_{x_1}$, the stress tensor becomes

$$\begin{aligned} \boldsymbol{\sigma}' &= \boldsymbol{R}_{x_1}\boldsymbol{\sigma}\boldsymbol{R}_{x_1}^T \\ &= \begin{bmatrix} \sigma_{11} & -\sigma_{13} & \sigma_{12} \\ -\sigma_{13} & \sigma_{33} & -\sigma_{23} \\ \sigma_{12} & -\sigma_{23} & \sigma_{22} \end{bmatrix} \rightarrow \begin{bmatrix} \sigma_{11} \\ \sigma_{33} \\ \sigma_{22} \\ -2\sigma_{13} \\ 2\sigma_{12} \\ -2\sigma_{23} \end{bmatrix}. \end{aligned} \tag{A.5}$$

The vector form of the second order stress tensor is used for FEA and as the NNCM output. By choosing an angle of $\frac{\pi}{2}$, rotation is performed by simply swapping components of the stress vector (and multiplying by -1 for some shear components). Following the same procedure, the rotated stress tensors found by application of $\boldsymbol{R}_{x_2}$ and $\boldsymbol{R}_{x_3}$ are

$$\boldsymbol{R}_{x_2} : \begin{bmatrix} \sigma_{33} \\ \sigma_{22} \\ \sigma_{11} \\ -2\sigma_{23} \\ -2\sigma_{13} \\ 2\sigma_{12} \end{bmatrix}, \quad \text{(A.6)} \qquad \boldsymbol{R}_{x_3} : \begin{bmatrix} \sigma_{22} \\ \sigma_{11} \\ \sigma_{33} \\ -2\sigma_{12} \\ -2\sigma_{23} \\ 2\sigma_{13} \end{bmatrix}. \quad \text{(A.7)}$$

The above apply to 3-D NNCMs. Stresses and strains in 2-D plane-stress analyses — the approximation for 2-D NNCMs — are only rotated about the

x_3 axis:

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ 2\sigma_{12} \end{bmatrix} \xrightarrow{\boldsymbol{R}_{x_3}} \begin{bmatrix} \sigma_{22} \\ \sigma_{11} \\ -2\sigma_{12} \end{bmatrix} \tag{A.8}$$

Rotations have only been shown for stresses, but the same methods are also applied to strain $\boldsymbol{\varepsilon}$. Thus, for 2-D analyses, the number of stress-strain pairs are doubled about enforcing frame invariance to the data. For 3-D, the number quadruples.

Appendix B

Accuracy of Displacement Estimates

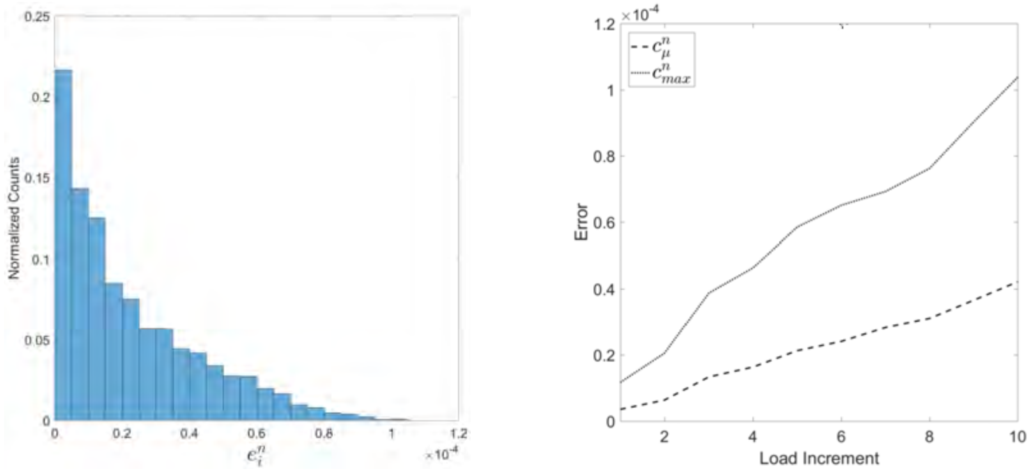
Building accurate informational models depends on acquiring accurate force and displacement measurements. We were interested in evaluating the displacement errors introduced through speckle-tracking and how those errors propagated through the load increments. Because displacement estimates were made between consecutive RF frames, estimates for the later load increments are the sum of many displacement maps. Quantifying displacement errors and tracking how they changed with successive load increments allowed us to ensure the effects of displacement errors on ANN training with AutoP would be minimal.

A 50x50x50mm^3 homogeneous phantom was constructed and imaged in the same manner described in Section 3.2.1. After creating the FE model (here, a rectilinear mesh was used for the phantom), a version of FEA$_\varepsilon$ was solved for the model where only probe displacements were applied. A total of 10 load increments were applied to compress the phantom 1.5mm. Because the phantom was homogeneous and the FEA was displacement controlled, the displacements computed at the nodes are the same regardless of the stiffness of the phantom.

Axial node displacements estimated from the US images were compared to those computed in the FEA using the equation

$$e_i^n = \frac{|u_i^n - \hat{u}_i^n|}{\max(|u_i^n|)} \tag{B.1}$$

where i indexes the nodes, n indexes the load increment, e_i^n is the displacement error, u_i^n is the displacement computed in the FEA, and $\hat{u}_i^n$ is the displacement estimated via speckle tracking. This is the same error computation for convergence described in Section 3.2.3. A histogram of the axial displacement errors for all 399 nodes in the imaged region across the ten load increments is plotted in Fig. B.1a. To compare these errors to the con-



(a)

(b)

Figure B.1: (a) Histogram of displacement errors. (b) Both c_μ^n and c_{max}^n, where n is the load increment, increase approximately linearly, suggesting the speckle-tracking produces a consistent, biased error.

vergence criteria specified in Sections 3.2.5 and 3.2.7, $c_\mu = 0.000022$ and $c_{max} = 0.0001$. These values are several orders of magnitude smaller than the chosen convergence criteria, suggesting that the test for convergence in each iteration of AutoP will be mostly affected by the material properties learned by the ANNs, not errors caused by speckle-tracking.

Tracking c_μ and c_{max} through the load increments, it was found that both increased monotonically and approximately linearly, as shown in Fig. B.1b. A monotonically increasing error implies that the displacement errors are biased and the linearity suggests the error in displacement estimates between RF frames is consistent. But, as just mentioned, the errors are well below the set convergence criteria and should not have a significant impact on the displacement error calculation. It is a point to consider, though, if applied loads produce large deformations over many increments.

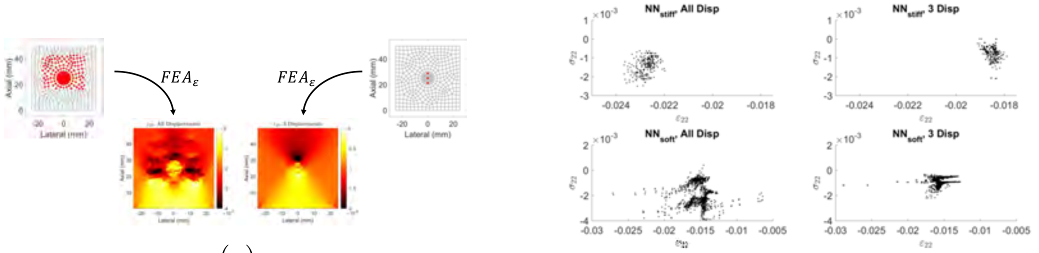
Appendix C

The Effect of Noise on Stress-strain Training Data

Classic mathematical inverse methods typically require many samples of data to reduce the effect of noise on the solution. It seems counter-intuitive, then, when we claim that using a smaller number of displacements in FEA$_\varepsilon$ enables the ANNs to more reliably learn the underlying mechanical properties. Our new method of elasticity imaging represents a paradigm shift that appears to not follow the same rules as classical inverse methods.

Current elastographic modalities formulate inverse problems that involve first or second order derivatives that are highly sensitive to noise. To combat noise amplification, those methods use sums or averages over many samples. We, however, are not directly solving the inverse problem. The sparse measurements of force and displacements are distributed across the entire mesh of the two FEA elements in AutoP and iteratively reconciled through the ANNs. Since AutoP model building occurs after several load steps and passes, the influence of noise introduced by sparse measurements at each load step is diminished through the process of FEA reconciliation. The effect of noise in the pipeline can best be demonstrated with a small example.

Using measurements from Phantom 1, consider two situations: 1) axial



(a)

(b)

Figure C.1: Comparing the effects of using many noisy displacements to only a few. Using many noisy displacements (a) causes large variance in the stress-strain training data (b) that have a negative impact on the ability of the ANNs to learn the underlying mechanical properties.

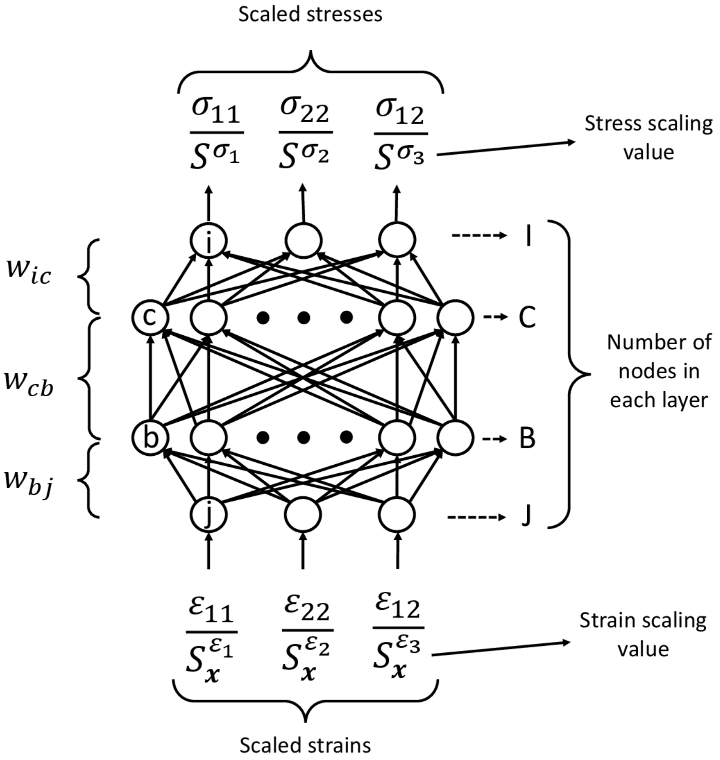
displacements are provided at every node in the imaged region (Fig. C.1a, left) and 2) axial displacements are provided at three locations within the inclusion (Fig. C.1a, right) in FEA$_\varepsilon$. The two axial strain images in the center of Fig. C.1a visualize the effect of noise on the resulting strain. Note that the color scales are different to help visualize the axial strain distribution in both cases. Going one step further, the axial stress-strain data is plotted in Fig. C.1b for NN$_{soft}$ and NN$_{stiff}$. These plot were obtained by pairing stresses from FEA$_\sigma$ with strains from FEA$_\varepsilon$ and pulling out only the axial components.

It is the job of the ANNs to generalize the data in Fig. C.1b. Note that the stress and strain vectors each have three components in 2-D (six in 3-D), meaning the Fig. C.1b only provides a partial picture of the function being approximated by the ANNs. However, the plots do provide a useful comparison. In the case of many displacements, the increased variance present in the stress-strain data may cause the ANNs to learn mechanical behaviors that are non-physical or incorrect. Using fewer displacements causes *some* variance that influences changes in the ANN connection weights during training. As AutoP iterations progress, the smaller variations caused by using few displacements in FEA$_\varepsilon$ allow the ANNs to gradually learn the correct material properties instead of attempting to force large changes in the learned behavior at once.

Appendix D

Calculating Stiffness Matrix from Material Property Network

Constructing the global tangent stiffness matrix in the solution of a FEA requires the stiffness matrix $\boldsymbol{D}$. Typically, $\boldsymbol{D}$ is known or computed from a pre-selected material model (e.g., linear-elasticity). Hashash *et al.* [138] derived an analytical expression for the stiffness matrix based on the connection weights and node activations of a MPN for a given strain input.



Scaled stresses
$\frac{\sigma_{11}}{S^{\sigma_1}}$
$\frac{\sigma_{22}}{S^{\sigma_2}}$
$\frac{\sigma_{12}}{S^{\sigma_3}}$
Stress scaling value
w_{ic}
w_{cb}
w_{bj}
i
c
b
j
I
C
B
J
Number of nodes in each layer
$\frac{\varepsilon_{11}}{S_{\boldsymbol{x}}^{\varepsilon_1}}$
$\frac{\varepsilon_{22}}{S_{\boldsymbol{x}}^{\varepsilon_2}}$
$\frac{\varepsilon_{12}}{S_{\boldsymbol{x}}^{\varepsilon_3}}$
Strain scaling value
Scaled strains

Figure D.1

Fig. D.1 is an annotated diagram of a MPN with two hidden layers. This notation is used in the following derivation of the expression to compute $\boldsymbol{D}$. First, the stresses $\hat{\boldsymbol{\sigma}}$ and strains $\hat{\boldsymbol{\varepsilon}}$ at the output and input of the MPN, respectively, are scaled versions of the actual stresses and strains:

$$\begin{aligned}\sigma_i &= S^{\sigma_i}\hat{\sigma}_i \\ \varepsilon_j &= S_{\boldsymbol{x}}^{\varepsilon_j}\hat{\varepsilon}_j,\end{aligned}$$

where the subscripts i, j index the components of the vectors.

The stiffness matrix is then found by computing the partial derivative of each component of stress with respect to each component of strain:

$$D_{ij} = \frac{\partial \sigma_i}{\partial \varepsilon_j} = \frac{\partial S^{\sigma_i}\hat{\sigma}_i}{\partial S_{\boldsymbol{x}}^{\varepsilon_j}\hat{\varepsilon}_j} \tag{D.1}$$

$$= \frac{S^{\sigma_i}}{S_{\boldsymbol{x}}^{\varepsilon_j}} \overbrace{\frac{\partial \hat{\sigma}_i}{\partial \hat{\varepsilon}_j}}^{\hat{D}_{ij}}. \tag{D.2}$$

Define the activation f_n and output ψ_n of node n in a layer as

$$\begin{aligned}f_n &= \sum_{m=1}^{M} w_{nm}\psi_m \\ \psi_n &= \tanh(\beta f_n).\end{aligned}$$

Note that β is included for generality but is set to 1 for all applications described in this thesis.

The stiffness matrix $\hat{\boldsymbol{D}}$ in equation (D.2) is then found through repeated application of the chain-rule:

$$\hat{D}_{ij} = \overbrace{\frac{\partial \hat{\sigma}_i}{\partial f_i}}^{1} \cdot \overbrace{\frac{\partial f_i}{\partial \psi_c}}^{2} \cdot \overbrace{\frac{\partial \psi_c}{\partial f_c}}^{3} \cdot \overbrace{\frac{\partial f_c}{\partial \psi_b}}^{4} \cdot \overbrace{\frac{\partial \psi_b}{\partial f_b}}^{5} \cdot \overbrace{\frac{\partial f_b}{\partial \psi_j}}^{6} \cdot \overbrace{\frac{\partial \psi_j}{\partial \hat{\varepsilon}_j}}^{7}. \tag{D.3}$$

Each term in (D.3) is easily found from the definition of the node activation and output:

$$
\begin{aligned}
&1)\ \frac{\partial \hat{\sigma}_i}{\partial f_i} = \frac{\partial \psi_i}{\partial f_i} = \beta(1 - \tanh(\beta f_i)^2 \\
&2)\ \frac{\partial f_i}{\partial \psi_c} = \frac{\partial}{\partial \psi_c} \sum_{c=1}^{C} w_{ic} \psi_c = \sum_{c=1}^{C} w_{ic} \\
&3)\ \frac{\partial \psi_c}{\partial f_c} = \beta(1 - \tanh(\beta f_c)^2) \\
&4)\ \frac{\partial f_c}{\partial \psi_b} = \sum_{b=1}^{B} w_{cb} \\
&5)\ \frac{\partial \psi_b}{\partial f_b} = \beta(1 - \tanh(\beta f_b)^2) \\
&6)\ \frac{\partial f_b}{\partial \psi_j} = \sum_{k=1}^{J} w_{bk} \\
&7)\ \frac{\partial \psi_j}{\partial \hat{\varepsilon}_j} = \beta(1 - \tanh(\beta \hat{\varepsilon}_j)^2).
\end{aligned}
$$

Combining all these terms and placing back into (D.3), the full expression for $\hat{\boldsymbol{D}}$ becomes

$$
\begin{aligned}
\hat{D}_{ij} = (1 - \tanh(f_i)^2) \sum_{c=1}^{C} \Bigg[& w_{ic}(1 - \tanh(f_c)^2) \times \\
& \sum_{b=1}^{B} \Bigg\{ w_{cb}(1 - \tanh(f_b)^2) \sum_{k=1}^{J} \Bigg(w_{bk}(1 - \tanh(\hat{\varepsilon}_j)^2) \Bigg) \Bigg\} \Bigg].
\end{aligned}
\tag{D.4}
$$

Appendix E

Simulating RF Echo Frames of Phantom Under Compression

We generated simulated RF echo data sets to investigate the effect of noise on CaNNCMs trained in AutoP. The goal was to emulate the experimental force-displacement data acquisition process. RF frames were generated using a linear systems approach, where each frame $g(x, y)$ was computed by convolving an US pulse $H(x, y)$ with field of scatterers $f(x, y)$ and adding Gaussian white noise: $g(x, y) = [H * f](x, y) + n(x, y)$. Two sets of RF frames were generated with echo SNR values of 30 and 15 dB. Each data set contained a total of five RF frames: a pre-deformation frame and four post-deformation frames with increasing compression.

Creating the pre-deformation RF frame required four steps: 1) generate the US pulse, 2) create the field of scatterers, 3) perform the convolution and add Gaussian white noise to obtain a specified SNR. Simulated post-compression frames followed steps (2)–(3), albeit in step (2) the existing field of scatterers was deformed, not recreated. We will describe in the last section how we implement the deformation.

[1] Creating US Pulse *H* Our goal was to roughly approximate images acquired by a VF10-5 linear array probe and a Siemens Antares scanner. The sampling frequency for constructing $H(x, y)$ and $f(x, y)$ was set as $F_s =$ 120 MHz. However, we will later describe how we modified the convolution to obtain an echo sampling rate $F_s^e = 40$ MHz. We chose a high sampling frequency to avoid aliasing effects in the modified convolution.

The US pulse was modeled as a Gabor pulse:

$$H(x, y) = e^{\frac{(y-y_0)^2}{2\sigma_a^2} \frac{(x-x_0)^2}{w_b^2}} \sin(\frac{2\pi y}{\lambda}) \tag{E.1}$$

$$w_b = \lambda f_N, \tag{E.2}$$

where $\sigma_a = 0.14$mm was chosen to create a two-cycle pulse for $\approx 50\%$

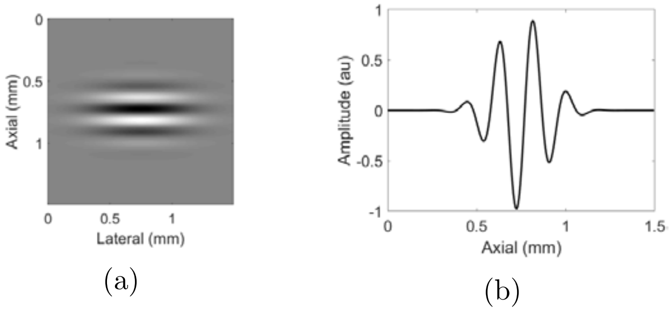


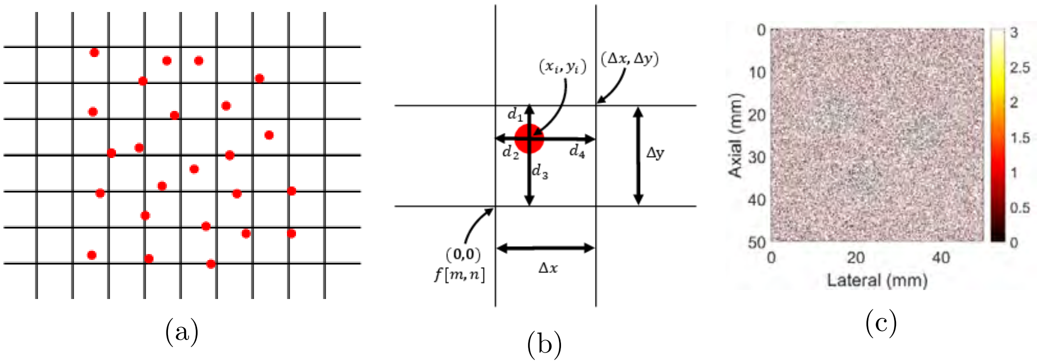
Figure E.1: (a) US pulse with a center frequency of 8 MHz created in a 1.5 × 1.5 mm field. (b) Cross-section of the US pulse along the line $x = 0.75$.

fractional pulse bandwidth, $f_N = 2$ is the f-number, and w_b is the beam width parameter. The pulse was created over a 1.5×1.5 mm field, therefore $x_0 = y_0 = 0.75$. Fig. E.1a shows the generated pulse and Fig. E.1b is the cross-section of the pulse along the vertical line $x = 0.75$ mm.

[2] **Scattering Field $\boldsymbol{f}$** The object $f(x, y)$ was created to imitate Model 2 (i.e., the gelatin phantom). Five million point scatterers were generated and randomly placed in the field. The coordinate of each was selected from a uniform random distribution in the range $(x, y) \in [-50, 50]$, which matches the domain of Model 2. Scatterers residing within the inclusions had their reflectivity reduced by half to produce hypoechoic regions.

Before the convolution with the US pulse, the field of scatterers had to be discretized. One simple method is to simply round the coordinate of each scatterer to the nearest corresponding location in the matrix f. However, we found that this method produced significant discretization errors in the final RF frames. Instead, we distributed the reflectivity of each scatterer to the four nearest elements in the matrix.

Fig. E.2a illustrates a set of scatterers overlayed on a discrete grid. Intersections of the grid correspond to locations in $\boldsymbol{f}$, the discretized form of $f(x, y)$. For each scatterer, we identify the four nearest locations in $\boldsymbol{f}$ and define a local coordinate system as illustrated in Fig. E.2b. Note that the lower-left corner is the local origin and corresponds to matrix location $f[m, n]$. If the scatterer is at (x_i, y_i) (in the local coordinate system) and the lateral and axial resolution of $\boldsymbol{f}$ is Δx and Δy, respectively, we can define



(a)

(b)

(c)

Figure E.2: (a) The grid represents the discrete field of scatterers f. Intersections of the grid lines correspond to locations in the matrix. Scatterers (red points) are placed continuously in the domain of the phantom. (b) The amplitude of each scatter is divided among the four nearest entries of f to help reduce discretization errors. (c) The matrix f after the reflectivity of the scatterers have been distributed.

the distances d_1–d_4 as

$$\begin{aligned} d_1 &= 1 - d_3 = \frac{y_i}{\Delta y} \\ d_2 &= 1 - \frac{x_i}{\Delta x} \\ d_3 &= 1 - \frac{y}{\Delta y} \\ d_4 &= 1 - d_2 = \frac{x}{\Delta x}. \end{aligned}$$

The reflectivity ϕ_i of the scatterer is divided to the four locations in $\boldsymbol{f}$ based on distance:

$$f[m,n] \overset{+}{=} d_2 \cdot d_3 \cdot \phi_i \tag{E.3}$$

$$f[m+1,n] \overset{+}{=} d_3 \cdot d_4 \cdot \phi_i \tag{E.4}$$

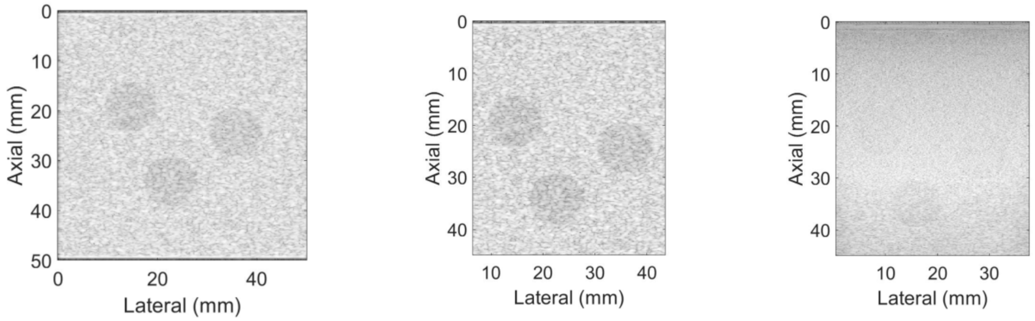
$$f[m,n+1] \overset{+}{=} d_1 \cdot d_2 \cdot \phi_i \tag{E.5}$$

$$f[m+1,n+1] \overset{+}{=} d_1 \cdot d_4 \cdot \phi_i. \tag{E.6}$$

As a check, $d_2 d_3 + d_3 d_y + d_1 d_2 + d_1 d_4 = 1$, ensuring ϕ_i is completely distributed. Fig. E.2c shows $\boldsymbol{f}$ after the reflectivity of all scatterers has been distributed.

[3] Modified Convolution The simplest way to compute $\boldsymbol{g}$ is via a straightforward discrete convolution:

$$g[m,n] = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} H[m-i, n-j] \cdot f[i,j]. \tag{E.7}$$



(a) (b) (c)

Figure E.3: (a)B-mode image from simulated RF echo frame of whole phantom. (b) A section of the whole echo frame is saved whose dimensions match the image region acquired experimentally. (c) Example B-mode image acquired on a gelatin phantom.

The resulting RF frame would be sampled at 120 MHz both axially and laterally whereas we want the vertical-axis sampling rate to be 40 MHz (0.02 mm) and the lateral pitch of ≈ 0.15 mm. Furthermore, $\boldsymbol{g}$ measured experimentally would not be square because of the difference in axial and lateral resolution. We modified the convolution by introducing "skips" to achieve the desired sampling both axially and laterally.

Define $\delta_x = \lfloor \frac{N_D}{N_{RF}} \rfloor$ and $\delta_y = \lfloor \frac{F_s}{F_s^e} \rfloor$. N_D is the number of columns in f whereas N_{RF} is the number of RF lines we want the final image to have. With the echo sampling intervals δ_x and δ_y, the convolution becomes

$$g[m,n] = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} H[m - i + (i-1) \cdot \delta_y, n - j + (j-1) \cdot \delta_x] \cdot f[i,j] \tag{E.8}$$

After performing the convolution, additive acquisition noise was introduced by sampling from standard Gaussian distribution and adjusting the variance to achieve the desired echo SNR.

The dimensions of $\boldsymbol{g}$ resulting from (E.8) are reduced from $N_D \times N_D$ — the result from (E.7) — to $\frac{N_D}{\Delta_y} \times \frac{N_D}{\Delta_x}$. An example B-mode imaged computed from a RF echo frame generated from (E.8 is shown in Fig. E.3a. However, RF images acquired experimentally do not encompass the entire object. We thus cut out the portion of the simulated RF image to match what is acquired experimentally. Fig. E.3b shows the B-mode of the reduced simulated RF frame and Fig. E.3c is an example of a B-mode image acquired for the gelatin phantom.

Post-Compression RF Frames Creating post-deformation RF frames required the scatterers to be moved before applying (E.8). Displacements computed in the FEA of Model 2 — from which noise-free displacement measurements were acquired for use in AutoP — were used to determine how the scatterers should move. A total of four load increments were applied in the FEA, meaning an additional four RF frames were generated (for a total of five, including the pre-deformation frame). Displacements from each load increment were used to move the scatterers. Steps (2)–(3) were repeated for each increment. Note that $\boldsymbol{f}$ was re-computed in (E.3)–(E.6) for each of the deformation images. In other words, $\boldsymbol{f}$ was initialized as a matrix of zeros before the field of scatterers was discretized.

Appendix F

Modified Algorithm to Compute $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$

Algorithm 1 is computationally expensive. Even though the modified version of the algorithm using the stress vector $\hat{\boldsymbol{\sigma}}_i^{\prime,NN}$ computed by the MPN instead of the stiffness matrix $\boldsymbol{D}$ is orders of magnitude faster than Eq. 4.11, calculating $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ for every point in a fairly coarse 2-D mesh still requires several minutes. The number of points increases exponentially in a 3-D mesh. Training 3-D CaNNCMs is infeasible if Algorithm 1 requires hours to complete. Note that $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ are recalculated in every iteration of AutoP. Therefore, development of 3-D CaNNCMs that are trained in a reasonable amount of time in AutoP required a faster algorithm to be created.

The architecture of the MPN lends itself to learning only a linear-elastic relationship. Meaning, the stress-strain behavior defined by the stiffness matrix $\boldsymbol{D}$ calculated from the connection weights of the MPN (Appendix D) is constant regardless of the input strain (within the training range). Furthermore, the scaling values at each of the N_x points in the mesh are independent and can be computed concurrently. Thus, the new algorithm should take advantage of the constant $\boldsymbol{D}$ and spatial independence of $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ to calculate updated spatial values at every $\boldsymbol{x}$ simultaneously.

I will describe the new algorithm in two steps. Recall that there are N_σ stress-strain pairs at every location $\boldsymbol{x}$. An increment value $\Delta\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ must be computed for each pair. The mean of these increments is calculated and then added to $\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ to compute an updated value (lines 10-11 of Algorithm 1). Therefore, I will first describe how the mean $\Delta\boldsymbol{S}_{\boldsymbol{x}}^{\varepsilon}$ can be computed over all N_σ pairs simultaneously (i.e., compute lines 3-10 of Algorithm 1 through a set of matrix multiplications) and then expand the process to solve for all points at once.

First, the stiffness matrix $\hat{\boldsymbol{D}}$ is computed using the average of all N_σ strain vectors at $\boldsymbol{x}_i$, $\varepsilon_i^{\mu} = (1/N_\sigma)\sum_{k=1}^{d} \varepsilon_i^k$, where d is the number of stress/strain components. Recall for 2-D plane-stress problems, $d = 3$. For 3-D problems,

$d = 6$.

Next, scale each of the strain vectors by the corresponding $\boldsymbol{S}^{\varepsilon}_{\boldsymbol{x}_i}$, $\boldsymbol{\varepsilon}(\boldsymbol{x}_i)' = \boldsymbol{\varepsilon}(\boldsymbol{x}_i)/\boldsymbol{S}^{\varepsilon}_{\boldsymbol{x}_i}$. The last sum in Eq. 4.11 can be replaced by a matrix-matrix multiplication to compute each $\boldsymbol{\sigma}^{NN}_{\boldsymbol{x}_i}$ (dropping $\boldsymbol{x}_i$ for brevity):

$$\begin{bmatrix} \boldsymbol{\sigma}^{NN}_1 \\ \boldsymbol{\sigma}^{NN}_2 \\ \vdots \\ \boldsymbol{\sigma}^{NN}_{N_\sigma} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\varepsilon}'_1 \\ \boldsymbol{\varepsilon}'_2 \\ \vdots \\ \boldsymbol{\varepsilon}'_{N_\sigma} \end{bmatrix} \hat{\boldsymbol{D}}^{\mathrm{T}} \rightarrow \bar{\boldsymbol{\sigma}}^{NN} = \boldsymbol{\varepsilon}' \hat{\boldsymbol{D}}^{\mathrm{T}} \tag{F.1}$$

The resulting $\bar{\boldsymbol{\sigma}}^{NN}$ is a $N_\sigma \times d$ matrix, where each row corresponds to a different stress vector.

Error vectors $\boldsymbol{e}_i$ are computed as the difference between the "true" stress $\boldsymbol{\sigma}^t_i$ (computed in FEA$_\sigma$ or resulting from enforcing frame invariance) and the stress vector $\boldsymbol{\sigma}^{NN}_i$:

$$\boldsymbol{e}_i = \boldsymbol{\sigma}^t_k - \boldsymbol{\sigma}^{NN}_k, \tag{F.2}$$

where $\boldsymbol{e}_k$ is also a $1 \times d$ vector.

A few more matrices need to be defined. The error matrix e_σ is a block matrix of all the error vectors and D^B is comprised of the stiffness matrix $\hat{\boldsymbol{D}}$:

$$\boldsymbol{e}_\sigma = \begin{bmatrix} \boldsymbol{e}_1 & \boldsymbol{0} & \dots & \boldsymbol{0} \\ & \boldsymbol{e}_2 & \dots & \boldsymbol{0} \\ & \text{symm.} & \ddots & \vdots \\ & & & \boldsymbol{e}_{N_\sigma} \end{bmatrix} \tag{F.3}$$

$$\boldsymbol{D}^B = \begin{bmatrix} \hat{\boldsymbol{D}} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ & \hat{\boldsymbol{D}} & \dots & \boldsymbol{0} \\ & \text{symm.} & \ddots & \vdots \\ & & & \hat{\boldsymbol{D}} \end{bmatrix}. \tag{F.4}$$

The block matrix $\boldsymbol{e}_\sigma$ is $N_\sigma \times d \cdot N_\sigma$ and $\hat{\boldsymbol{D}}^B$ is $d \cdot N_\sigma \times d \cdot N_\sigma$

Next, the matrix $\boldsymbol{\varepsilon}^{iso}_i$ whose rows are the "isolated" strain vectors. For

2-D, it is a 3×3 matrix:

$$\boldsymbol{\varepsilon}_i^{iso} = \begin{bmatrix} \varepsilon_{11}^i & 0 & 0 \\ 0 & \varepsilon_{22}^i & 0 \\ 0 & 0 & \varepsilon_{12}^i \end{bmatrix}. \tag{F.5}$$

Alternatively, in 3-D,

$$\boldsymbol{\varepsilon}_i^{iso} = \begin{bmatrix} \varepsilon_{11}^i & 0 & 0 & 0 & 0 & 0 \\ 0 & \varepsilon_{22}^i & 0 & 0 & 0 & 0 \\ 0 & 0 & \varepsilon_{33}^i & 0 & 0 & 0 \\ 0 & 0 & 0 & \varepsilon_{12}^i & 0 & 0 \\ 0 & 0 & 0 & 0 & \varepsilon_{13}^i & 0 \\ 0 & 0 & 0 & 0 & 0 & \varepsilon_{23}^i \end{bmatrix} \tag{F.6}$$

Create a new block matrix of all the isolated strain matrices for each of the N stress-strain pairs at $\boldsymbol{x}$

$$\boldsymbol{\varepsilon}^{iso} = \begin{bmatrix} \boldsymbol{\varepsilon}_1^{iso} \\ \boldsymbol{\varepsilon}_2^{iso} \\ \vdots \\ \boldsymbol{\varepsilon}_{N_\sigma}^{iso} \end{bmatrix} \tag{F.7}$$

Finally, to take the average over all N_σ pairs, take the dot-product with a vector

$$\boldsymbol{v}_\mu = \mathbf{1}/N_\sigma \tag{F.8}$$

where $\mathbf{1}$ is $1 \times d$.

With these newly defined matrices and vectors, the spatial scaling vector for all d components computed as the mean error over all N_σ stress-strain pairs can then be computed as

$$\overbrace{\Delta \boldsymbol{S}^{\varepsilon}_{\boldsymbol{x}_i}}^{1\times d} = \overbrace{v_\mu}^{1\times d} \cdot \overbrace{\boldsymbol{e}_\sigma}^{d\times dN} \cdot \overbrace{\boldsymbol{D}^B}^{dN\times dN} \cdot \overbrace{\boldsymbol{\varepsilon}^{iso}}^{dN\times d} . \tag{F.9}$$

Expanding (F.9) is straight-forward, but must take advantage of some programmatic tricks to reduce memory requirements. For example, D^B as defined above would be of size $N_\sigma \cdot N_x \cdot d \times N_\sigma \cdot N_x \cdot d$ if constructed for all stress-strain pairs for all N_x points. If there were 2000 points in a 2-D, each with four stress-strain pairs, then D^B would require ≈ 72 megabytes for storage. A more extreme example is a 3-D problem with $N_x = 64000$ (which occurs for a $20 \times 20 \times 20$ element mesh) and $N_\sigma = 16$ (a four-load AutoP training window with frame-invariance enforced). D^B in this case requires 471 GB of storage. Clearly, (F.9) cannot be used as-is to compute all $\Delta \boldsymbol{S}^{\varepsilon}_{\boldsymbol{x}}$ simultaneously.

I will begin by describing the new data structures and reshape$(\cdot)$ function, followed by the new algorithm. MATLAB syntax will be used to aid the explanation of some arrays and computations. Finally, a sample MATLAB implementation of the new algorithm follows the description.

Let $\hat{\boldsymbol{E}}$ be the $N_x \times d \times N_\sigma$ array containing the complete set of strain data. Define $\hat{\boldsymbol{E}}_i$ as the $N_x \times d$ matrix at $N_\sigma = i$ ($\hat{\boldsymbol{E}}(:,:,i)$ in MATLAB syntax). The same notation applies for $\hat{\boldsymbol{S}}$ and $\hat{\boldsymbol{S}}_i$ which contain the corresponding stress data. Furthermore, let $\boldsymbol{S}^{\varepsilon}$ be the $N_x \times d$ matrix comprised of all the scaling vectors:

$$\boldsymbol{S}^{\varepsilon} = \begin{bmatrix} \boldsymbol{S}^{\varepsilon}_{\boldsymbol{x}_1} \\ \boldsymbol{S}^{\varepsilon}_{\boldsymbol{x}_2} \\ \vdots \\ \boldsymbol{S}^{\varepsilon}_{\boldsymbol{x}_{N_x}} \end{bmatrix} . \tag{F.10}$$

The matrix $\hat{\boldsymbol{D}}$ is now computed from the MPN using the average of all $N_\sigma \cdot N_x$ strain vectors. Similar adjustments are made for some of the other block matrices defined for (F.9). Consider the N_x stress-strain pairs in $\hat{\boldsymbol{E}}_i$:

$$\boldsymbol{e}_\sigma = \begin{bmatrix} \boldsymbol{e}_1 & \mathbf{0} & \dots & \mathbf{0} \\ & \boldsymbol{e}_2 & \dots & \mathbf{0} \\ & \text{symm.} & \ddots & \vdots \\ & & & \boldsymbol{e}_{N_x} \end{bmatrix} \tag{F.11}$$

$$\boldsymbol{\varepsilon}^{iso} = \begin{bmatrix} \boldsymbol{\varepsilon}_1^{iso} \\ \boldsymbol{\varepsilon}_2^{iso} \\ \vdots \\ \boldsymbol{\varepsilon}_{N_x}^{iso} \end{bmatrix} \tag{F.12}$$

Finally, it is necessary to define matrix reshaping. The function uses column-major reordering. More information on this function can be found in the MATLAB documentation for the function `reshape`. Per the documentation, the syntax is `reshape(A,sz1,...,szN)`, where A is the matrix to be reshaped and each of the `sz1,...,szN` specify the size of each dimension. As an example, suppose the matrix $A_{2\times2\times2}$ is to be reshaped into a single column vector or a 2×4 matrix. The `reshape` syntax would be

$$A_{ijk} \xrightarrow{\texttt{reshape(A,[ ],1)}} \begin{bmatrix} A_{111} \\ A_{211} \\ A_{121} \\ A_{221} \\ A_{112} \\ A_{212} \\ A_{122} \\ A_{222} \end{bmatrix}, \text{or } A_{ijk} \xrightarrow{\texttt{reshape(A,2,4,1)}} \begin{bmatrix} A_{111} & A_{121} & A_{112} & A_{122} \\ A_{211} & A_{221} & A_{212} & A_{222} \end{bmatrix}. \tag{F.13}$$

Also note that the syntax `A(:)` is equivalent to `reshape(A,[],1)`. Element-by-element multiplication and division will follow MATLAB syntax: $.*$ and $./$, respectively. With these definitions, Algorithm 2 details the new, faster method of computing updated values of $\boldsymbol{S}^{\varepsilon}_{\boldsymbol{x}}$.

The following MATLAB code is an implementation of Algorithm 2. $\hat{\boldsymbol{E}}$ and $\hat{\boldsymbol{S}}$ are represented by `strain_data` and `stress_data`, respectively, whereas

Algorithm 2 Faster algorithm for computing $\boldsymbol{S}^{\varepsilon}_{\boldsymbol{x}}$

```
Given: current $\boldsymbol{S}^{\varepsilon}, \hat{\boldsymbol{S}}, \hat{\boldsymbol{E}}$
for $i = 1, 2, ..., N_{\varepsilon}$ do
    $\Delta S = \mathbf{0}_{N \times D}$
    for $j = 1, 2, ..., N_{\sigma}$ do
        $\hat{\boldsymbol{E}}' = \hat{\boldsymbol{E}}_j./\boldsymbol{S}^{\varepsilon}$
        $\boldsymbol{\sigma}^{NN} = \hat{\boldsymbol{E}}' * (\hat{\boldsymbol{D}}^{\mathrm{T}})$
        $\boldsymbol{e}_{\sigma} = \boldsymbol{S}_j - \boldsymbol{\sigma}^{NN}$
        $\boldsymbol{U} = \boldsymbol{e}_{\sigma} * \boldsymbol{D}_{\mu}$
        $\boldsymbol{U} = \texttt{reshape}(\boldsymbol{U}^{\mathrm{T}}, 1, [\,])$
        $\boldsymbol{\varepsilon}^{iso} = \boldsymbol{E}_j^{\mathrm{T}}$
        $\boldsymbol{\varepsilon}^{iso} = \texttt{reshape}(\boldsymbol{\varepsilon}^{iso}, [\,], 1)^{\mathrm{T}}$
        $\boldsymbol{U} = -\boldsymbol{U}. * \boldsymbol{\varepsilon}^{iso}$
        $\Delta S = \Delta S + \texttt{reshape}(\boldsymbol{U}, D, [\,])^{\mathrm{T}}$
    end for
    $\Delta S = \Delta S./P$
    $\boldsymbol{S}^{\varepsilon} = \boldsymbol{S}^{\varepsilon} + \eta_{\varepsilon} * \Delta S$
end for
```

$\boldsymbol{S}^{\sigma}$ is referred to as `strain_scale`. $\hat{\boldsymbol{D}}$ is computed before this section of code and is stored in `D_hat`.

A comparison of Algorithms 1 and 2 is illustrated in Fig. F.1 (rows 2 and 3, respectively). Also included in row 1 are the results of computing $\boldsymbol{S}^{\varepsilon}_{\boldsymbol{x}}$ with (4.11). For all three, the maps of the spatial values are indistinguishable. The most notable difference lies in the error curves. Equation 4.11 led to the fastest convergence in terms of number of iterations, but that does not account for time. Total running times for (4.11), Algorithm 1, and Algorithm 2 were 1734 s, 759 s, and 0.4133 s. Even though the error curves do not converge as fast with Algorithm 2, it is over 1000× faster to compute. It is because of this drastic increase in speed that training 3-D CaNNCMs in AutoP is feasible.

```
        for ii=1:grad_iter
          delta_s = zeros(N,D);
          for jj=1:P
            s_prime = strain_data(:,:,mm)./strain_scale;
            stress_nn = s_prime*(D_hat');
            error_mat = stress_data(:,:,mm) - stress_nn;
            U = error_mat*this_jac;
            U = reshape(T',1,[]);
            eps_iso = strain_data(:,:,mm)';
            eps_iso = eps_iso(:)';
            U = -U.*eps_iso;
            U = (reshape(U,num_comp,[]))';
            delta_s = delta_s + U;
          end
          delta_s = delta_s./P;
          strain_scale = strain_scale + eta_val.*delta_s;
        end
```

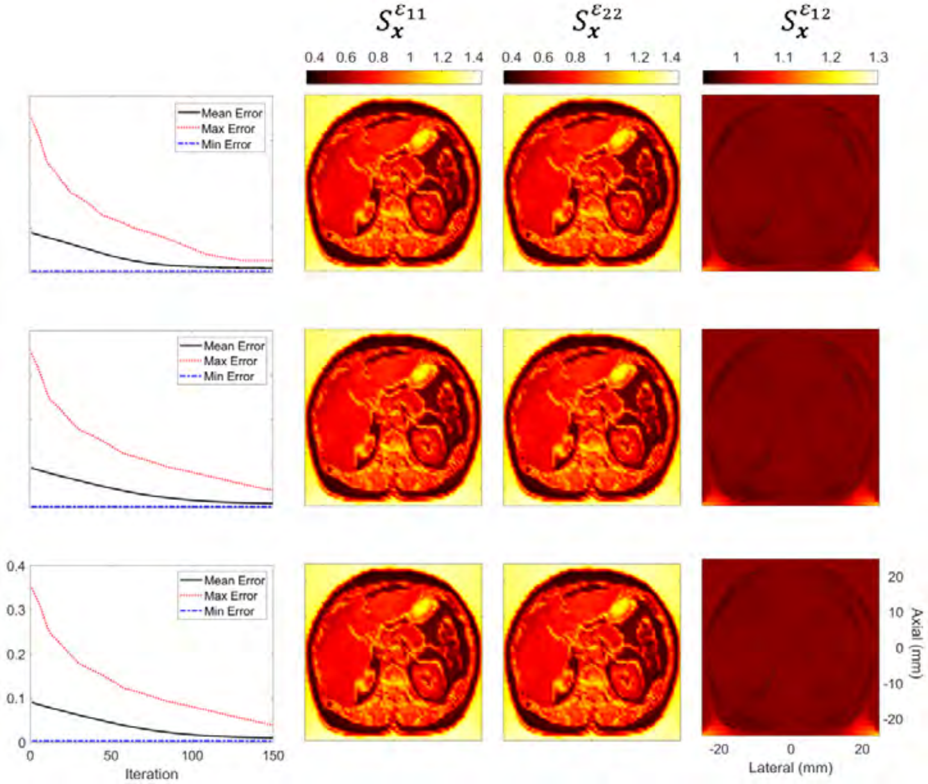


Figure F.1: Comparison of spatial values computed with three different methods. Rows 1–3 are the results using Eq. 4.11, Algorithm 1, and Algorithm 2, respectively.

Appendix G

Estimating Phantom Geometry with Microsoft Kinect

CaNNCMs are able to learn internal geometry, but the external shape must be known in order to create the finite element mesh used in AutoP. Unfortunately, the human body is not a cube. We must devise a method to estimate external geometry in order to translate our data-driven QUSE approach to a clinical setting. Our current solution for estimating surface geometry makes use of a Microsoft Kinect V2.

Esimating Surface Geometry

The Kinect is equipped with an IR camera capable of estimating object depth. Furthermore, the Kinect SDK is packaged with a "Fusion" library to reconstruct surfaces maps from depth images acquired by scanning a scene. We utilize the Fusion functionality to estimate phantom geometry and create a FE mesh for use in AutoP. Fig. G.1 illustrates the process.

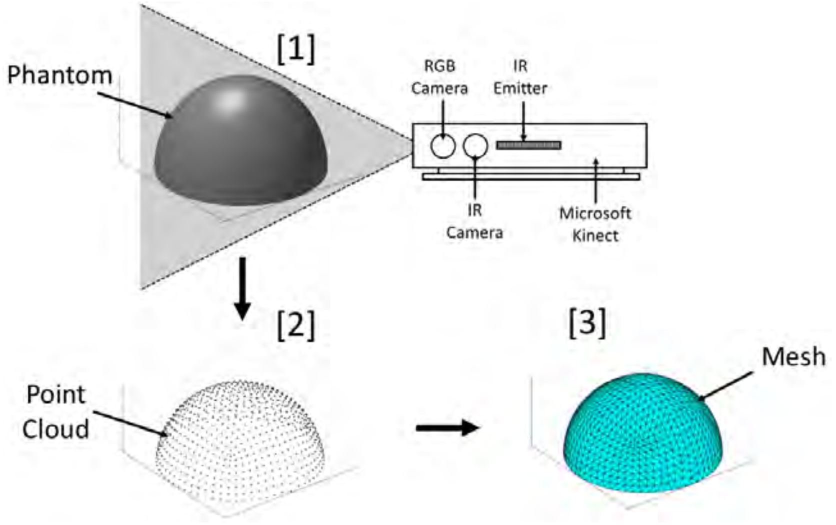
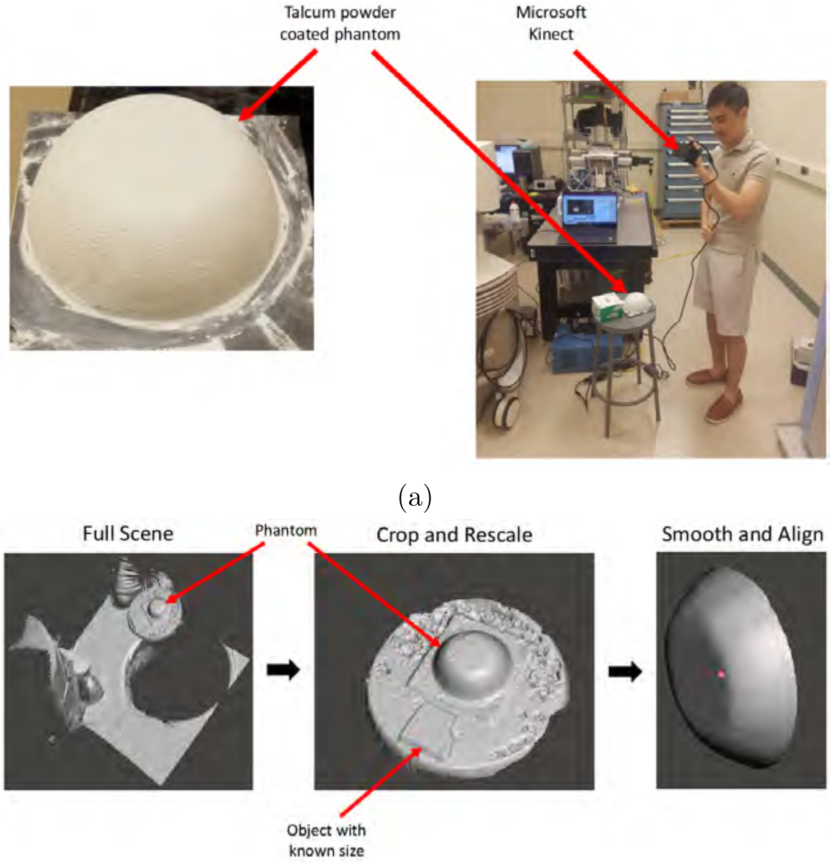


Figure G.1: Process by which phantom geometry is estimated via Microsoft Kinect. [1] The depth camera and Fusion library capture the surface geometry. [2] A point-cloud (or stereolithography file) is exported and [3] used to create a FE mesh.

Unfortunately, gelatin is nearly invisible to IR light, meaning the depth images of the phantom cannot be captured. We remedy this issue by coating the phantom in talcum powder prior to imaging (left, Fig. G.2a). We have found that the phantom must be thoroughly coated (at least three powder coats) in order to effectively image it with the IR camera. After applying the powder coating, the phantom is placed on a raised platform next to an object of known size (right, Fig. G.2a). The Kinect is pointed toward the phantom (along the x_2 axis) from an elevated location before data acquisition begins. After starting depth image capture, the Kinect is slowly panned a few centimeters in the x_1 and x_3 directions to ensure the full surface of the phantom is imaged.



(a)

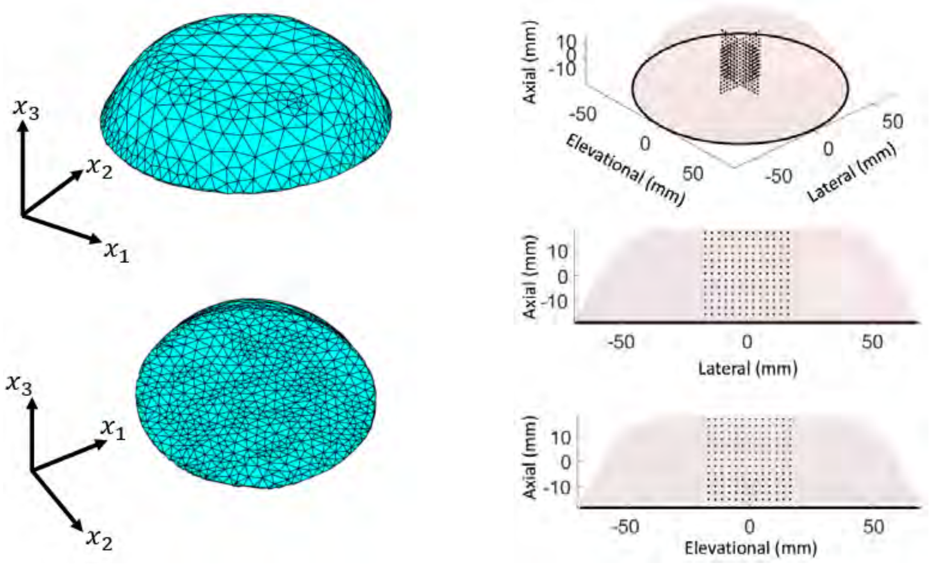
(b)

Figure G.2: (a) The phantom is coated in talcum powder and imaged with the Kinect's depth camera. (b) The phantom is isolated from the full scene, smoothed, and properly oriented with the world axis.

An example of the full scene capture in this manner is shown in Fig. G.2b. Using MeshMixer (Autodesk, USA, San Rafael, CA), we first reduce the scene to encompass the phantom and the object with known dimensions.

The surface mesh is not set to the proper scale, therefore this object allows one to correct the scale of objects in the scene. Furthermore, because the additional object is flat and has uniform width, it can be used to adjust the global coordinate system and correctly orient the phantom with the world axis. After making these adjustments, the phantom is isolated and some smoothing operations are performed to reduce fluctuations due to noise.

Once adjustments to the phantom geometry in MeshMixer are complete, the model is exported as a STL file, which is then used in MATLAB with the PDE Solver toolbox. With said toolbox, we create an initial FE mesh with 4-node tetrahedral elements (Fig. G.3a). However, at this stage there is no way to precisely control the location of nodes in the mesh. In our case, this is important because we require nodes in the US image planes in order to apply displacement data in FEA$_{\varepsilon}$. Therefore, we use TetGen [171] to add nodes to the mesh, as shown in Fig. G.3b. TetGen adds the new nodes, remeshes as necessary to accommodate the additional points, and converts the elements to 10-node tetrahedrons.



(a)

(b)

Figure G.3: (a) Initial mesh created with PDE Solver toolbox in MATLAB from STL created in Fig. G.2b. (b) Additional points are added to the US image planes and the mesh is reprocessed with TetGen.

References

[1] D. B. Plewes and W. Kucharczyk, "Physics of MRI: a primer," *Journal of Magnetic Resonance Imaging*, vol. 35, no. 5, pp. 1038–1054, 2012.

[2] S. R. Cherry and M. Dahlbom, *PET: physics, instrumentation, and scanners*, 2006.

[3] A. Sarvazyan, A. Skovoroda, S. Emelianov, J. Fowlkes, J. Pipe, R. Adler, R. Buxton, and P. Carson, "Biophysical bases of elasticity imaging," in *Acoustical Imaging.* Springer, 1995, pp. 223–240.

[4] A. P. Sarvazyan, O. V. Rudenko, S. D. Swanson, J. B. Fowlkes, and S. Y. Emelianov, "Shear wave elasticity imaging: a new ultrasonic technology of medical diagnostics," *Ultrasound in Medicine & Biology*, vol. 24, no. 9, pp. 1419–1435, 1998.

[5] K. L. Mathis, T. L. Hoskin, J. C. Boughey, B. S. Crownhart, K. R. Brandt, C. M. Vachon, C. S. Grant, and A. C. Degnim, "Palpable presentation of breast cancer persists in the era of screening mammography," *Journal of the American College of Surgeons*, vol. 210, no. 3, pp. 314–318, 2010.

[6] D. Saslow, J. Hannan, J. Osuch, M. H. Alciati, C. Baines, M. Barton, J. K. Bobo, C. Coleman, M. Dolan, G. Gaumer *et al.*, "Clinical breast examination: practical recommendations for optimizing performance and reporting," *CA: a Cancer Journal for Clinicians*, vol. 54, no. 6, pp. 327–344, 2004.

[7] K. C. Hilty and D. H. Steinberg, "Vulnerable plaque imagingcurrent techniques," *Journal of Cardiovascular Translational Research*, vol. 2, no. 1, pp. 9–18, 2009.

[8] C. Pasterkamp and E. Falk, "Atherosclerotic plaque rupture: an overview," *Journal of Clinical and Basic Cardiology*, vol. 3, no. 2, pp. 81–86, 2000.

[9] A. H. Chau, R. C. Chan, M. Shishkov, B. MacNeill, N. Iftimia, G. J. Tearney, R. D. Kamm, B. E. Bouma, and M. R. Kaazempur-Mofrad, "Mechanical analysis of atherosclerotic plaques based on optical coherence tomography," *Annals of Biomedical Engineering*, vol. 32, no. 11, pp. 1494–1503, 2004.

[10] L. Sandrin, B. Fourquet, J.-M. Hasquenoph, S. Yon, C. Fournier, F. Mal, C. Christidis, M. Ziol, B. Poulet, F. Kazemi *et al.*, "Transient elastography: a new noninvasive method for assessment of hepatic fibrosis," *Ultrasound in Medicine & Biology*, vol. 29, no. 12, pp. 1705–1713, 2003.

[11] J. Stebbing, L. Farouk, G. Panos, M. Anderson, L. R. Jiao, S. Mandalia, M. Bower, B. Gazzard, and M. Nelson, "A meta-analysis of transient elastography for the detection of hepatic fibrosis," *Journal of Clinical Gastroenterology*, vol. 44, no. 3, pp. 214–219, 2010.

[12] V. W.-S. Wong, J. Vergniol, G. L.-H. Wong, J. Foucher, H. L.-Y. Chan, B. Le Bail, P. C.-L. Choi, M. Kowo, A. W.-H. Chan, W. Merrouche *et al.*, "Diagnosis of fibrosis and cirrhosis using liver stiffness measurement in nonalcoholic fatty liver disease," *Hepatology*, vol. 51, no. 2, pp. 454–462, 2010.

[13] N. Salameh, F. Peeters, R. Sinkus, J. Abarca-Quinones, L. Annet, L. C. Ter Beek, I. Leclercq, and B. E. Van Beers, "Hepatic viscoelastic parameters measured with MR elastography: correlations with quantitative analysis of liver fibrosis in the rat," *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine*, vol. 26, no. 4, pp. 956–962, 2007.

[14] R. Masuzaki, R. Tateishi, H. Yoshida, E. Goto, T. Sato, T. Ohki, J. Imamura, T. Goto, F. Kanai, N. Kato *et al.*, "Prospective risk assessment for hepatocellular carcinoma development in patients with chronic hepatitis C by transient elastography," *Hepatology*, vol. 49, no. 6, pp. 1954–1961, 2009.

[15] K. S. Jung, S. U. Kim, S. H. Ahn, Y. N. Park, D. Y. Kim, J. Y. Park, C. Y. Chon, E. H. Choi, and K.-H. Han, "Risk assessment of hepatitis B virus–related hepatocellular carcinoma development using liver stiffness measurement (FibroScan)," *Hepatology*, vol. 53, no. 3, pp. 885–894, 2011.

[16] S. K. Venkatesh, M. Yin, J. F. Glockner, N. Takahashi, P. A. Araoz, J. A. Talwalkar, and R. L. Ehman, "MR elastography of liver tumors: preliminary results," *American Journal of Roentgenology*, vol. 190, no. 6, pp. 1534–1540, 2008.

[17] S. Doblas, P. Garteiser, N. Haddad, J. Daire, M. Wagner, H. Leitao, V. Vilgrain, R. Sinkus, and B. Van Beers, "Magnetic resonance elastography measurements of viscosity: a novel biomarker for human hepatic tumor malignancy," in *Proc. Intl. Soc. Mag. Reson. Med*, vol. 19, 2011, p. 389.

[18] S. Chen, W. Sanchez, M. R. Callstrom, B. Gorman, J. T. Lewis, S. O. Sanderson, J. F. Greenleaf, H. Xie, Y. Shi, M. Pashley *et al.*, "Assessment of liver viscoelasticity by using shear waves induced by ultrasound radiation force," *Radiology*, vol. 266, no. 3, pp. 964–970, 2013.

[19] W. A. Berg, D. O. Cosgrove, C. J. Doré, F. K. Schäfer, W. E. Svensson, R. J. Hooley, R. Ohlinger, E. B. Mendelson, C. Balu-Maestro, M. Locatelli *et al.*, "Shear-wave elastography improves the specificity of breast US: the BE1 multinational study of 939 masses," *Radiology*, vol. 262, no. 2, pp. 435–449, 2012.

[20] A. Evans, P. Whelehan, K. Thomson, D. McLean, K. Brauer, C. Purdie, L. Jordan, L. Baker, and A. Thompson, "Quantitative shear wave ultra-

sound elastography: initial experience in solid breast masses," *Breast Cancer Research*, vol. 12, no. 6, p. R104, 2010.

[21] K. C. Siegmann, T. Xydeas, R. Sinkus, B. Kraemer, U. Vogel, and C. D. Claussen, "Diagnostic value of MR elastography in addition to contrast-enhanced MR imaging of the breastinitial clinical results," *European Radiology*, vol. 20, no. 2, pp. 318–325, 2010.

[22] R. Sinkus, M. Tanter, T. Xydeas, S. Catheline, J. Bercoff, and M. Fink, "Viscoelastic shear properties of in vivo breast lesions measured by MR elastography," *Magnetic Resonance Imaging*, vol. 23, no. 2, pp. 159–165, 2005.

[23] M. Tanter, J. Bercoff, A. Athanasiou, T. Deffieux, J.-L. Gennisson, G. Montaldo, M. Muller, A. Tardivon, and M. Fink, "Quantitative assessment of breast lesion viscoelasticity: initial clinical results using supersonic shear imaging," *Ultrasound in Medicine & Biology*, vol. 34, no. 9, pp. 1373–1386, 2008.

[24] Y. Qiu, M. Sridhar, J. K. Tsou, K. K. Lindfors, and M. F. Insana, "Ultrasonic viscoelasticity imaging of nonpalpable breast tumors: preliminary results," *Academic Radiology*, vol. 15, no. 12, pp. 1526–1533, 2008.

[25] T. Xydeas, K. Siegmann, R. Sinkus, U. Krainick-Strobel, S. Miller, and C. D. Claussen, "Magnetic resonance elastography of the breast: correlation of signal intensity data with viscoelastic properties," *Investigative Radiology*, vol. 40, no. 7, pp. 412–420, 2005.

[26] A. Sayed, G. Layne, J. Abraham, and O. M. Mukdadi, "3-d visualization and non-linear tissue classification of breast tumors using ultrasound elastography in vivo," *Ultrasound in Medicine & Biology*, vol. 40, no. 7, pp. 1490–1502, 2014.

[27] A. Samani and D. Plewes, "A method to measure the hyperelastic parameters of ex vivo breast tissue samples," *Physics in Medicine & Biology*, vol. 49, no. 18, p. 4395, 2004.

[28] A. Samani, J. Zubovits, and D. Plewes, "Elastic moduli of normal and pathological human breast tissues: an inversion-technique-based investigation of 169 samples," *Physics in Medicine & Biology*, vol. 52, no. 6, p. 1565, 2007.

[29] H. Mehrabian, G. Campbell, and A. Samani, "A constrained reconstruction technique of hyperelasticity parameters for breast cancer assessment," *Physics in Medicine & Biology*, vol. 55, no. 24, p. 7489, 2010.

[30] A. Itoh, E. Ueno, E. Tohno, H. Kamma, H. Takahashi, T. Shiina, M. Yamakawa, and T. Matsumura, "Breast disease: clinical application of US elastography for diagnosis," *Radiology*, vol. 239, no. 2, pp. 341–350, 2006.

[31] R. Isermann, S. Grunwald, G. Hatzung, D. Könsgen-Mustea, P.-O. Behrndt, A.-A. Geaid, B. Jäger, and R. Ohlinger, "Breast lesion sizing by b-mode

imaging and sonoelastography in comparison to histopathological sizing–a prospective study," *Ultraschall in der Medizin-European Journal of Ultrasound*, vol. 32, no. S 01, pp. 21–26, 2011.

[32] R. G. Barr, S. Destounis, L. B. Lackey, W. E. Svensson, C. Balleyguier, and C. Smith, "Evaluation of breast lesions using sonographic elasticity imaging: a multicenter trial," *Journal of Ultrasound in Medicine*, vol. 31, no. 2, pp. 281–287, 2012.

[33] D. T. Butcher, T. Alliston, and V. M. Weaver, "A tense situation: forcing tumour progression," *Nature Reviews Cancer*, vol. 9, no. 2, p. 108, 2009.

[34] C. Pellot-Barakat, M. Sridhar, K. K. Lindfors, and M. F. Insana, "Ultrasonic elasticity imaging as a tool for breast cancer diagnosis and research," *Current Medical Imaging Reviews*, vol. 2, no. 1, pp. 157–164, 2006.

[35] D. Cosgrove, F. Piscaglia, J. Bamber, J. Bojunga, J.-M. Correas, O. Gilja, A. Klauser, I. Sporea, F. Calliada, V. Cantisani *et al.*, "EFSUMB guidelines and recommendations on the clinical use of ultrasound elastography. part 2: Clinical applications," *Ultraschall in der Medizin-European Journal of Ultrasound*, vol. 34, no. 03, pp. 238–253, 2013.

[36] J. Ophir, I. Cespedes, H. Ponnekanti, Y. Yazdi, and X. Li, "Elastography: a quantitative method for imaging the elasticity of biological tissues," *Ultrasonic Imaging*, vol. 13, no. 2, pp. 111–134, 1991.

[37] E. Konofagou and J. Ophir, "A new elastographic method for estimation and imaging of lateral displacements, lateral strains, corrected axial strains and Poissons ratios in tissues," *Ultrasound in Medicine & Biology*, vol. 24, no. 8, pp. 1183–1199, 1998.

[38] H. S. Hashemi and H. Rivaz, "Global time-delay estimation in ultrasound elastography," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 64, no. 10, pp. 1625–1636, 2017.

[39] E. Konofagou and J. Ophir, "A new elastographic method for estimation and imaging of lateral displacements, lateral strains, corrected axial strains and Poisson's ratios in tissues," *Ultrasound in Medicine and Biology*, vol. 24, no. 8, pp. 1183–1199, 1998.

[40] X. Chen, M. J. Zohdy, S. Y. Emelianov, and M. O'Donnell, "Lateral speckle tracking using synthetic lateral phase," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 51, no. 5, pp. 540–550, 2004.

[41] H. Rivaz, E. M. Boctor, M. A. Choti, and G. D. Hager, "Real-time regularized ultrasound elastography." *IEEE Transactions on Medical Imaging*, vol. 30, no. 4, pp. 928–45, Apr. 2011.

[42] J. Ophir, S. K. Alam, B. Garra, F. Kallel, E. Konofagou, T. Krouskop, and T. Varghese, "Elastography: ultrasonic estimation and imaging of the elastic properties of tissues," *Proceedings of the Institution of Mechanical

Engineers, Part H: Journal of Engineering in Medicine, vol. 213, no. 3, pp. 203–233, 1999.

[43] M. Sridhar, J. Liu, and M. F. Insana, "Elasticity imaging of polymeric media," *Journal of Biomechanical Engineering*, vol. 129, no. 2, pp. 259–272, 2007.

[44] J. M. Chang, J.-K. Won, K.-B. Lee, I. A. Park, A. Yi, and W. K. Moon, "Comparison of shear-wave and strain ultrasound elastography in the differentiation of benign and malignant breast lesions," *American Journal of Roentgenology*, vol. 201, no. 2, pp. W347–W356, 2013.

[45] J. F. Carlsen, C. Ewertsen, L. Lönn, and M. B. Nielsen, "Strain elastography ultrasound: an overview with emphasis on breast cancer diagnosis," *Diagnostics*, vol. 3, no. 1, pp. 117–125, 2013.

[46] J. R. Grajo and R. G. Barr, "Strain elastography for prediction of breast cancer tumor grades," *Journal of Ultrasound in Medicine*, vol. 33, no. 1, pp. 129–134, 2014.

[47] M. R. Onur, A. K. Poyraz, E. E. Ucak, Z. Bozgeyik, I. H. Özercan, and E. Ogur, "Semiquantitative strain elastography of liver masses," *Journal of Ultrasound in Medicine*, vol. 31, no. 7, pp. 1061–1067, 2012.

[48] Z. Wang, Y. Liu, G. Wang, and L. Sun, "Elastography method for reconstruction of nonlinear breast tissue properties," *Journal of Biomedical Imaging*, vol. 2009, p. 6, 2009.

[49] T. Z. Pavan, E. L. Madsen, G. R. Frank, A. A. O. Carneiro, and T. J. Hall, "Nonlinear elastic behavior of phantom materials for elastography," *Physics in Medicine & Biology*, vol. 55, no. 9, p. 2679, 2010.

[50] A. A. Oberai, N. H. Gokhale, S. Goenezen, P. E. Barbone, T. J. Hall, A. M. Sommer, and J. Jiang, "Linear and nonlinear elasticity imaging of soft tissue in vivo: demonstration of feasibility," *Physics in Medicine & Biology*, vol. 54, no. 5, p. 1191, 2009.

[51] M. F. Insana, C. Pellot-Barakat, M. Sridhar, and K. K. Lindfors, "Viscoelastic imaging of breast tumor microenvironment with ultrasound," *Journal of Mammary Gland Biology and Neoplasia*, vol. 9, no. 4, pp. 393–404, 2004.

[52] J. J. O'Hagan and A. Samani, "Measurement of the hyperelastic properties of 44 pathological ex vivo breast tissue samples," *Physics in Medicine & Biology*, vol. 54, no. 8, p. 2557, 2009.

[53] H. Zhang, Y. Wang, M. Fatemi, and M. F. Insana, "Assessing composition and structure of soft biphasic media from kelvin–voigt fractional derivative model parameters," *Measurement Science and Technology*, vol. 28, no. 3, p. 035703, 2017.

[54] M. Bayat, A. Nabavizadeh, V. Kumar, A. Gregory, M. Insana, A. Alizad, and M. Fatemi, "Automated in vivo sub-hertz analysis of viscoelasticity (SAVE) for evaluation of breast lesions," *IEEE Transactions on Biomedical Engineering*, 2017.

[55] G. Torr, "The acoustic radiation force," *American Journal of Physics*, vol. 52, no. 5, pp. 402–408, 1984.

[56] T. Sugimoto, S. Ueha, and K. Itoh, "Tissue hardness measurement using the radiation force of focused ultrasound," in *Ultrasonics Symposium, 1990. Proceedings., IEEE 1990.* IEEE, 1990, pp. 1377–1380.

[57] K. R. Nightingale, M. L. Palmeri, R. W. Nightingale, and G. E. Trahey, "On the feasibility of remote palpation using acoustic radiation force," *The Journal of the Acoustical Society of America*, vol. 110, no. 1, pp. 625–634, 2001.

[58] M. L. Palmeri and K. R. Nightingale, "On the thermal effects associated with radiation force imaging of soft tissue," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 51, no. 5, pp. 551–565, 2004.

[59] K. Nightingale, "Acoustic radiation force impulse (ARFI) imaging: a review," *Current Medical Imaging Reviews*, vol. 7, no. 4, pp. 328–339, 2011.

[60] S. Bota, H. Herkner, I. Sporea, P. Salzl, R. Sirli, A. M. Neghina, and M. Peck-Radosavljevic, "Meta-analysis: ARFI elastography versus transient elastography for the evaluation of liver fibrosis," *Liver International*, vol. 33, no. 8, pp. 1138–1147, 2013.

[61] C. Efe, M. S. Gungoren, E. Ozaslan, F. Akbiyik, and T. Kav, "Acoustic radiation force impulse (ARFI) for fibrosis staging in patients with autoimmune hepatitis." *Hepato-Gastroenterology*, vol. 62, no. 139, pp. 670–672, 2015.

[62] W. Meng, G. Zhang, C. Wu, G. Wu, Y. Song, and Z. Lu, "Preliminary results of acoustic radiation force impulse (ARFI) ultrasound imaging of breast lesions," *Ultrasound in Medicine & Biology*, vol. 37, no. 9, pp. 1436–1443, 2011.

[63] M. Tozaki, S. Isobe, and E. Fukuma, "Preliminary study of ultrasonographic tissue quantification of the breast using the acoustic radiation force impulse (ARFI) technology," *European Journal of Radiology*, vol. 80, no. 2, pp. e182–e187, 2011.

[64] M. Friedrich-Rust, O. Romenski, G. Meyer, N. Dauth, K. Holzer, F. Grünwald, S. Kriener, E. Herrmann, S. Zeuzem, and J. Bojunga, "Acoustic radiation force impulse-imaging for the evaluation of the thyroid gland: a limited patient feasibility study," *Ultrasonics*, vol. 52, no. 1, pp. 69–74, 2012.

[65] Y.-F. Zhang, H.-X. Xu, Y. He, C. Liu, L.-H. Guo, L.-N. Liu, and J.-M. Xu, "Virtual touch tissue quantification of acoustic radiation force impulse: a

new ultrasound elastic imaging in the diagnosis of thyroid nodules," *PloS One*, vol. 7, no. 11, p. e49094, 2012.

[66] M. R. Selzo and C. M. Gallippi, "Viscoelastic response (VisR) imaging for assessment of viscoelasticity in voigt materials," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 60, no. 12, p. 2488, 2013.

[67] C. J. Moore, M. R. Selzo, M. C. Caughey, D. O. Meyer, R. Emmett, J. F. Howard, M. Chopra, and C. M. Gallippi, "Viscoelastic response (VisR) assessment of longitudinal dystrophic degeneration in clinical duchenne muscular dystrophy," in *Ultrasonics Symposium (IUS), 2015 IEEE International.* IEEE, 2015, pp. 1–4.

[68] M. R. Selzo, C. J. Moore, M. M. Hossain, M. L. Palmeri, and C. M. Gallippi, "On the quantitative potential of viscoelastic response (visr) ultrasound using the one-dimensional mass-spring-damper model," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 63, no. 9, pp. 1276–1287, 2016.

[69] R. M. Lerner, K. J. Parker, J. Holen, R. Gramiak, and R. C. Waag, "Sonoelasticity: medical elasticity images derived from ultrasound signals in mechanically vibrated targets," in *Acoustical Imaging.* Springer, 1988, pp. 317–327.

[70] R. M. Lerner, S. Huang, and K. J. Parker, "Sonoelasticity images derived from ultrasound signals in mechanically vibrated tissues," *Ultrasound in Medicine and Biology*, vol. 16, no. 3, pp. 231–239, 1990.

[71] T. Krouskop, D. Dougherty, F. Vinson *et al.*, "A pulsed doppler ultrasonic system for making noninvasive measurements of the mechanical properties of soft tissue," *J Rehabil Res Dev*, vol. 24, no. 2, pp. 1–8, 1987.

[72] Y. Yamakoshi, J. Sato, and T. Sato, "Ultrasonic imaging of the internal vibration of soft tissue under forced vibration," *The Journal of the Acoustical Society of America*, vol. 84, no. S1, pp. S139–S139, 1988.

[73] ——, "Ultrasonic imaging of internal vibration of soft tissue under forced vibration," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 37, no. 2, pp. 45–53, 1990.

[74] M. Fatemi and J. F. Greenleaf, "Ultrasound-stimulated vibro-acoustic spectrography," *Science*, vol. 280, no. 5360, pp. 82–85, 1998.

[75] A. Alizad, M. Fatemi, L. E. Wold, and J. F. Greenleaf, "Performance of vibro-acoustography in detecting microcalcifications in excised human breast tissue: A study of 74 tissue samples," *IEEE Transactions on Medical Imaging*, vol. 23, no. 3, pp. 307–312, 2004.

[76] A. Alizad, M. Fatemi, D. H. Whaley, and J. F. Greenleaf, "Application of vibro-acoustography for detection of calcified arteries in breast tissue," *Journal of Ultrasound in Medicine*, vol. 23, no. 2, pp. 267–273, 2004.

[77] Z. Wu, L. S. Taylor, D. J. Rubens, and K. J. Parker, "Sonoelastographic imaging of interference patterns for estimation of the shear velocity of homogeneous biomaterials," *Physics in Medicine & Biology*, vol. 49, no. 6, p. 911, 2004.

[78] K. Hoyt, K. J. Parker, and D. J. Rubens, "Real-time shear velocity imaging using sonoelastographic techniques," *Ultrasound in Medicine & Biology*, vol. 33, no. 7, pp. 1086–1097, 2007.

[79] K. Hoyt, T. Kneezel, B. Castaneda, and K. J. Parker, "Quantitative sonoelastography for the in vivo assessment of skeletal muscle viscoelasticity," *Physics in Medicine & Biology*, vol. 53, no. 15, p. 4063, 2008.

[80] J. E. Brandenburg, S. F. Eby, P. Song, H. Zhao, J. S. Brault, S. Chen, and K.-N. An, "Ultrasound elastography: the new frontier in direct measurement of muscle stiffness," *Archives of Physical Medicine and Rehabilitation*, vol. 95, no. 11, pp. 2207–2219, 2014.

[81] J. Bercoff, M. Tanter, and M. Fink, "Supersonic shear imaging: a new technique for soft tissue elasticity mapping," *IEEE transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 51, no. 4, pp. 396–409, 2004.

[82] S. Chen, M. Fatemi, and J. F. Greenleaf, "Quantifying elasticity and viscosity from measurement of shear wave speed dispersion," *The Journal of the Acoustical Society of America*, vol. 115, no. 6, pp. 2781–2785, 2004.

[83] S. Chen, M. W. Urban, C. Pislaru, R. Kinnick, Y. Zheng, A. Yao, and J. F. Greenleaf, "Shearwave dispersion ultrasound vibrometry (SDUV) for measuring tissue elasticity and viscosity," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 56, no. 1, pp. 55–62, 2009.

[84] J. Ghaboussi, D. A. Pecknold, and X. S. Wu, *Nonlinear computational solid mechanics.* CRC Press, 2017.

[85] Y.-c. Fung, *Biomechanics: mechanical properties of living tissues.* Springer Science & Business Media, 2013.

[86] M. Doyley, "Model-based elastography: a survey of approaches to the inverse elasticity problem," *Physics in Medicine & Biology*, vol. 57, no. 3, p. R35, 2012.

[87] A. Skovoroda, S. Emelianov, and M. o'Donnell, "Tissue elasticity reconstruction based on ultrasonic displacement and strain images," *IEEE Transactions on Ultrasonics Ferroelectrics and Frequency Control*, vol. 42, no. 4, pp. 747–765, 1995.

[88] P. E. Barbone and J. C. Bamber, "Quantitative elasticity imaging: what can and cannot be inferred from strain images," *Physics in Medicine and Biology*, vol. 47, no. 12, p. 2147, 2002.

[89] M. Tyagi, S. Goenezen, P. E. Barbone, and A. A. Oberai, “Algorithms for quantitative quasi-static elasticity imaging using force data,” *International Journal for Numerical Methods in Biomedical Engineering*, vol. 30, no. 12, pp. 1421–1436, Dec. 2014.

[90] P. E. Barbone and N. H. Gokhale, “Elastic modulus imaging: on the uniqueness and nonuniqueness of the elastography inverse problem in two dimensions,” *Inverse Problems*, vol. 20, no. 1, p. 283, 2004.

[91] Y. Zhu, T. J. Hall, and J. Jiang, “A finite-element approach for Young’s modulus reconstruction,” *IEEE Transactions on Medical Imaging*, vol. 22, no. 7, pp. 890–901, 2003.

[92] F. Kallel and M. Bertrand, “Tissue elasticity reconstruction using linear perturbation method,” *IEEE Transactions on Medical Imaging*, vol. 15, no. 3, pp. 299–313, Jun. 1996.

[93] M. Doyley, P. Meaney, and J. Bamber, “Evaluation of an iterative reconstruction method for quantitative elastography,” *Physics in Medicine & Biology*, vol. 45, no. 6, p. 1521, 2000.

[94] A. Samani, J. Bishop, and D. B. Plewes, “A constrained modulus reconstruction technique for breast cancer assessment,” *IEEE Transactions on Medical Imaging*, vol. 20, no. 9, pp. 877–885, 2001.

[95] S. R. Mousavi, A. Sadeghi-Naini, G. J. Czarnota, and A. Samani, “Towards clinical prostate ultrasound elastography using full inversion approach,” *Medical Physics*, vol. 41, no. 3, 2014.

[96] S. Goenezen, J.-F. Dord, Z. Sink, P. E. Barbone, J. Jiang, T. J. Hall, and A. A. Oberai, “Linear and nonlinear elastic modulus imaging: An application to breast cancer diagnosis,” *IEEE Transactions on Medical Imaging*, vol. 31, no. 8, pp. 1628–1637, 2012.

[97] M. S. Richards, P. E. Barbone, and A. A. Oberai, “Quantitative three-dimensional elasticity imaging from quasi-static deformation: a phantom study,” *Physics in Medicine & Biology*, vol. 54, no. 3, p. 757, 2009.

[98] C. Lanczos, *Linear differential operators.* SIAM, 1997, vol. 18.

[99] D. Sidarta and J. Ghaboussi, “Constitutive modeling of geomaterials from non-uniform material tests,” *Computers and Geotechnics*, vol. 22, no. 1, pp. 53 – 71, 1998.

[100] Y. M. Hashash, J. Ghaboussi, Q. Fu, and C. Marulanda, “Constitutive soil behavior representation via artificial neural networks: a shift from soil models to soil behavior data,” in *GeoCongress 2006: Geotechnical Engineering in the Information Technology Age*, 2006, pp. 1–6.

[101] Y. M. Hashash, J. Ghaboussi, and S. Jung, “Characterizing granular material constitutive behavior using SelfSim with boundary load-displacement

measurements," in *Earth & Space 2006: Engineering, Construction, and Operations in Challenging Environment*, 2006, pp. 1–8.

[102] S. Jung and J. Ghaboussi, "Characterizing rate-dependent material behaviors in self-learning simulation," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 1, pp. 608 – 619, 2006.

[103] ——, "Inverse identification of creep of concrete from in situ load-displacement monitoring," *Engineering Structures*, vol. 32, no. 5, pp. 1437 – 1445, 2010.

[104] J. Kim, J. Ghaboussi, and A. S. Elnashai, "Hysteretic mechanical-informational modeling of bolted steel frame connections," *Engineering Structures*, vol. 45, pp. 1 – 11, 2012.

[105] G. J. Yun, J. Ghaboussi, and A. S. Elnashai, "Development of neural network based hysteretic models for steel beam-column connections through self-learning simulation," *Journal of Earthquake Engineering*, vol. 11, no. 3, pp. 453–467, 2007.

[106] G. J. Yun, A. Saleeb, S. Shang, W. Binienda, and C. Menzemer, "Improved SelfSim for inverse extraction of nonuniform, nonlinear, and inelastic material behavior under cyclic loadings," *Journal of Aerospace Engineering*, vol. 25, no. 2, pp. 256–272, 2012.

[107] J. Ghaboussi, D. Pecknold, Y. Hashash, and T. Kwon, "Accurate determination of intraocular pressure and characterization of mechanical properties of the cornea," Dec. 6 2011, US Patent 8,070,679. [Online]. Available: https://www.google.com/patents/US8070679

[108] C. Hoerig, J. Ghaboussi, and M. F. Insana, "An information-based machine learning approach to elasticity imaging," *Biomechanics and Modeling in Mechanobiology*, vol. 16, no. 3, pp. 805–822, 2017.

[109] C. Hoerig, J. Ghaboussi, and M. F. Insana, "Cartesian neural network constitutive models for data-driven elasticity imaging," *ArXiv e-prints*, Sep. 2018.

[110] J. Ghaboussi and D. Sidarta, "New nested adaptive neural networks (NANN) for constitutive modeling," *Computers and Geotechnics*, vol. 22, no. 1, pp. 29 – 52, 1998.

[111] A. Jain, J. Mao, and K. Mohiuddin, "Artificial neural networks: a tutorial," *Computer*, vol. 29, no. 3, pp. 31–44, mar 1996.

[112] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251 – 257, 1991.

[113] M. T. Hagan, H. B. Demuth, M. H. Beale, and O. De Jesús, *Neural network design*. Pws Pub. Boston, 1996, vol. 20.

[114] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural Networks for Perception.* Elsevier, 1992, pp. 65–93.

[115] K.-J. Bathe, *Finite Element Procedures.* Klaus-Jurgen Bathe, 2006.

[116] R. G. Barr, "Elastography in clinical practice," *Radiologic Clinics of North America*, vol. 52, no. 6, pp. 1145 – 1162, 2014.

[117] J. Ghaboussi, D. A. Pecknold, M. Zhang, and R. M. Haj-ali, "Autoprogressive Training of Neural Network Constitutive Models," *International Journal for Numerical Methods in Engineering*, vol. 42, no. 1, pp. 105–126, 1998.

[118] C. Hoerig, J. Ghaboussi, and M. F. Insana, "Informational modeling of tissue-like materials using ultrasound," in *12th IEEE International Symposium on Biomedical Imaging, ISBI 2015, Brooklyn, NY, USA, April 16-19, 2015*, 2015, pp. 239–242.

[119] C. Hoerig, J. Ghaboussi, M. Fatemi, and M. F. Insana, "A new approach to ultrasonic elasticity imaging," in *Proc SPIE Medical Imaging: Ultrasonic Imaging and Tomography*, vol. 9790, 2016, pp. G1–9.

[120] M. Orescanin and M. Insana, "Shear modulus estimation with vibrating needle stimulation," *Ultrasonics, Ferroelectrics, and Frequency Control, IEEE Transactions on*, vol. 57, no. 6, pp. 1358–1367, June 2010.

[121] K. N. Altahhan, Y. Wang, N. Sobh, and M. F. Insana, "Indentation measurements to validate dynamic elasticity imaging methods," *Ultrasonic Imaging*, vol. 38, no. 5, pp. 332–345, 2016.

[122] J. Ghaboussi, "Advances in neural networks in computational mechanics and engineering," in *Advances of Soft Computing in Engineering.* Springer, 2010, pp. 191–236.

[123] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: the RPROP algorithm," in *IEEE International Conference on Neural Networks*, vol. 1. IEEE, 1993, pp. 586–591.

[124] J.-L. Gennisson, N. Grenier, C. Combe, and M. Tanter, "Supersonic shear wave elastography of in vivo pig kidney: influence of blood pressure, urinary pressure and tissue anisotropy," *Ultrasound in Medicine & Biology*, vol. 38, no. 9, pp. 1559 – 1567, 2012.

[125] Z. A. Taylor, M. Cheng, and S. Ourselin, "Real-time nonlinear finite element analysis for surgical simulation using graphics processing units," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2007: 10th International Conference, Brisbane, Australia, October 29 - November 2, 2007, Proceedings, Part I.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 701–708.

[126] J. Huang, S. Ong, and A. Nee, "Real-time finite element structural analysis in augmented reality," *Advances in Engineering Software*, vol. 87, pp. 43–56, 2015.

[127] M. A. Lubinski, S. Y. Emelianov, and M. O'Donnell, "Speckle tracking methods for ultrasonic elasticity imaging using short-time correlation," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 46, no. 1, pp. 82–96, 1999.

[128] A. Evans, P. Whelehan, K. Thomson, D. McLean, K. Brauer, C. Purdie, L. Jordan, L. Baker, and A. Thompson, "Quantitative shear wave ultrasound elastography: initial experience in solid breast masses," *Breast Cancer Research*, vol. 12, no. 6, p. R104, 2010.

[129] K. Fujimoto, M. Kato, M. Kudo, N. Yada, T. Shiina, K. Ueshima, Y. Yamada, T. Ishida, M. Azuma, M. Yamasaki *et al.*, "Novel image analysis method using ultrasound elastography for noninvasive evaluation of hepatic fibrosis in patients with chronic hepatitis c," *Oncology*, vol. 84, no. Suppl. 1, pp. 3–12, 2013.

[130] L. Castéra, J. Vergniol, J. Foucher, B. Le Bail, E. Chanteloup, M. Haaser, M. Darriet, P. Couzigou, and V. de Lédinghen, "Prospective comparison of transient elastography, Fibrotest, APRI, and liver biopsy for the assessment of fibrosis in chronic hepatitis C," *Gastroenterology*, vol. 128, no. 2, pp. 343–350, 2005.

[131] S. Ahmad, R. Cao, T. Varghese, L. Bidaut, and G. Nabi, "Transrectal quantitative shear wave elastography in the detection and characterisation of prostate cancer," *Surgical Endoscopy*, vol. 27, no. 9, pp. 3280–3287, 2013.

[132] E. Widman, E. Maksuti, D. Larsson, M. W. Urban, A. Bjällmark, and M. Larsson, "Shear wave elastography plaque characterization with mechanical testing validation: a phantom study," *Physics in Medicine & Biology*, vol. 60, no. 8, p. 3151, 2015.

[133] M. G. Van Vledder, E. M. Boctor, L. R. Assumpcao, H. Rivaz, P. Foroughi, G. D. Hager, U. M. Hamper, T. M. Pawlik, and M. A. Choti, "Intra-operative ultrasound elasticity imaging for monitoring of hepatic tumour thermal ablation," *Hpb*, vol. 12, no. 10, pp. 717–723, 2010.

[134] R. Souchon, O. Rouvière, A. Gelet, V. Detti, S. Srinivasan, J. Ophir, and J. Y. Chapelon, "Visualisation of HIFU lesions using elastography of the human prostate in vivo: Preliminary results," *Ultrasound in Medicine and Biology*, vol. 29, no. 7, pp. 1007–1015, 2003.

[135] S. L. Sridhar, Y. Mei, and S. Goenezen, "Improving the sensitivity to map nonlinear parameters for hyperelastic problems," *Computer Methods in Applied Mechanics and Engineering*, 2017.

[136] M. F. Insana, C. Pellot-Barakat, M. Sridhar, and K. K. Lindfors, "Viscoelastic imaging of breast tumor microenvironment with ultrasound," *Journal of Mammary Gland Biology and Neoplasia*, vol. 9, no. 4, pp. 393–404, 2004.

[137] C. Hoerig, W. Reyes, L. Fabre, J. Ghaboussi, and M. F. Insana, "Imaging spatially varying biomechanical properties with neural networks," in *Proc*

SPIE Medical Imaging: Ultrasonic Imaging and Tomography, vol. 1013905, 2017, pp. 1–7.

[138] Y. M. A. Hashash, S. Jung, and J. Ghaboussi, "Numerical implementation of a neural network based material model in finite element analysis," *International Journal for Numerical Methods in Engineering*, vol. 59, no. 7, pp. 989–1005, 2004.

[139] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[140] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[141] C. Hoerig, J. Ghaboussi, and M. F. Insana, "Data-driven elasticity imaging using cartesian neural network constitutive models and the autoprogressive method," *IEEE Transactions on Medical Imaging*, 2018.

[142] M. Sridhar, J. Liu, and M. Insana, "Viscoelasticity imaging using ultrasound: parameters and error analysis," *Physics in Medicine & Biology*, vol. 52, no. 9, p. 2425, 2007.

[143] P. E. Barbone and J. C. Bamber, "Quantitative elasticity imaging: What can and cannot be inferred from strain images," *Physics in Medicine and Biology*, vol. 47, no. 12, pp. 2147–2164, 2002.

[144] M. M. Doyley, S. Srinivasan, E. Dimidenko, N. Soni, and J. Ophir, "Enhancing the performance of model-based elastography by incorporating additional a priori information in the modulus image reconstruction process," *Physics in Medicine and Biology*, vol. 51, no. 1, pp. 95–112, 2006.

[145] U. Albocher, P. Barbone, M. Richards, A. Oberai, and I. Harari, "Approaches to accommodate noisy data in the direct solution of inverse problems in incompressible plane strain elasticity," *Inverse Problems in Science and Engineering*, vol. 22, no. 8, pp. 1307–1328, 2014.

[146] Y. Mei, S. Kuznetsov, and S. Goenezen, "Reduced Boundary Sensitivity and Improved Contrast of the Regularized Inverse Problem Solution in Elasticity," *Journal of Applied Mechanics*, vol. 83, no. 3, p. 031001, 2015.

[147] Y. Mei, R. Fulmer, V. Raja, S. Wang, and S. Goenezen, "Estimating the non-homogeneous elastic modulus distribution from surface deformations," *International Journal of Solids and Structures*, vol. 83, pp. 73–80, 2015.

[148] V. Egorov and A. Sarvazyan, "Mechanical imaging of the breast," *IEEE Transactions on Medical Imaging*, vol. 27, no. 9, pp. 1275–1287, 2008.

[149] J. Palacio-Torralba, S. Hammer, D. W. Good, S. Alan McNeill, G. D. Stewart, R. L. Reuben, and Y. Chen, "Quantitative diagnostics of soft tissue

through viscoelastic characterization using time-based instrumented palpation." *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 41, pp. 149–60, 2015.

[150] A. Sayed, G. Layne, J. Abraham, and O. Mukdadi, "Nonlinear characterization of breast cancer using multi-compression 3D ultrasound elastography in vivo," *Ultrasonics*, vol. 53, no. 5, pp. 979–991, 2013.

[151] S. L. Sridhar, Y. Mei, and S. Goenezen, "Improving the sensitivity to map nonlinear parameters for hyperelastic problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 331, pp. 474–491, 2018.

[152] N. H. Gokhale, P. E. Barbone, and A. A. Oberai, "Solution of the nonlinear elasticity imaging inverse problem: the compressible case," *Inverse Problems*, vol. 24, no. 4, p. 045010, 2008.

[153] S. Goenezen, P. Barbone, and A. A. Oberai, "Solution of the nonlinear elasticity imaging inverse problem: The incompressible case," *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 13-16, pp. 1406–1420, 2011.

[154] T. A. Krouskop, T. M. Wheeler, F. Kallel, B. S. Garra, and T. Hall, "Elastic moduli of breast and prostate tissues under compression," *Ultrasonic Imaging*, vol. 20, no. 4, pp. 260–274, 1998.

[155] A. H. Gandomi and G. J. Yun, "Coupled SelfSim and genetic programming for non-linear material constitutive modelling," *Inverse Problems in Science and Engineering*, vol. 23, no. 7, pp. 1101–1119, 2015.

[156] N. Grenier, J.-L. Gennisson, F. Cornelis, Y. L. Bras, and L. Couzi, "Renal ultrasound elastography," *Diagnostic and Interventional Imaging*, vol. 94, no. 5, pp. 545 – 550, 2013, ultrasound elastography.

[157] Y. Fu, C. Chui, C. Teo, and E. Kobayashi, "Elasticity imaging of biological soft tissue using a combined finite element and non-linear optimization method," *Inverse Problems in Science and Engineering*, vol. 23, no. 2, pp. 179–196, 2015.

[158] F. Martínez-Martínez, M. Rupérez-Moreno, M. Martínez-Sober, J. Solves-Llorens, D. Lorente, A. Serrano-López, S. Martínez-Sanchis, C. Monserrat, and J. Martín-Guerrero, "A finite element-based machine learning approach for modeling the mechanical behavior of the breast tissues under compression in real-time," *Computers in Biology and Medicine*, vol. 90, pp. 116–124, 2017.

[159] V. Strbac, J. Vander Sloten, and N. Famaey, "Analyzing the potential of GPGPUs for real-time explicit finite element analysis of soft tissue deformation using CUDA," *Finite Elements in Analysis and Design*, vol. 105, pp. 79–89, 2015.

[160] S. F. Johnsen, Z. A. Taylor, M. J. Clarkson, J. Hipwell, M. Modat, B. Eiben, L. Han, Y. Hu, T. Mertzanidou, D. J. Hawkes *et al.*, "Niftysim: A GPU-based nonlinear finite element package for simulation of soft tissue biomechanics," *International Journal of Computer Assisted Radiology and Surgery*, vol. 10, no. 7, pp. 1077–1095, 2015.

[161] A. Fenster, D. B. Downey, and H. N. Cardinal, "Three-dimensional ultrasound imaging," *Physics in Medicine & Biology*, vol. 46, no. 5, p. R67, 2001.

[162] J. E. Lindop, G. M. Treece, A. H. Gee, and R. W. Prager, "3d elastography using freehand ultrasound," *Ultrasound in Medicine & Biology*, vol. 32, no. 4, pp. 529–545, 2006.

[163] G. M. Treece, J. E. Lindop, A. H. Gee, and R. W. Prager, "Freehand ultrasound elastography with a 3-d probe," *Ultrasound in Medicine & Biology*, vol. 34, no. 3, pp. 463–474, 2008.

[164] R. J. Housden, A. H. Gee, G. M. Treece, and R. W. Prager, "3-d ultrasonic strain imaging using freehand scanning and a mechanically-swept probe-correspondence," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 57, no. 2, 2010.

[165] G. A. Hendriks, B. Holländer, J. Menssen, A. Milkowski, H. H. Hansen, and C. L. de Korte, "Automated 3d ultrasound elastography of the breast: a phantom validation study," *Physics in Medicine & Biology*, vol. 61, no. 7, p. 2665, 2016.

[166] Y. Wang, H. G. Nasief, S. Kohn, A. Milkowski, T. Clary, S. Barnes, P. E. Barbone, and T. J. Hall, "Three-dimensional ultrasound elasticity imaging on an automated breast volume scanning system," *Ultrasonic Imaging*, vol. 39, no. 6, pp. 369–392, 2017.

[167] C. Papadacci, E. A. Bunting, and E. E. Konofagou, "3d quasi-static ultrasound elastography with plane wave in vivo," *IEEE Transactions on Medical Imaging*, vol. 36, no. 2, pp. 357–365, 2017.

[168] M. W. Gilbertson and B. W. Anthony, "Force and position control system for freehand ultrasound," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 835–849, 2015.

[169] T.-Y. Fang, H. K. Zhang, R. Finocchi, R. H. Taylor, and E. M. Boctor, "Force-assisted ultrasound imaging system through dual force sensing and admittance robot control," *International Journal of Computer Assisted Radiology and Surgery*, vol. 12, no. 6, pp. 983–991, 2017.

[170] M. Frewer, "More clarity on the concept of material frame-indifference in classical continuum mechanics," *Acta Mechanica*, vol. 202, no. 1-4, p. 213, 2009.

[171] H. Si, "TetGen, a delaunay-based quality tetrahedral mesh generator," *ACM Transactions on Mathematical Software (TOMS)*, vol. 41, no. 2, p. 11, 2015.