

Data-Driven Elasticity Imaging Using Cartesian Neural Network Constitutive Models and the Autoprogressive Method

Cameron Hoerig¹, Student Member, IEEE, Jamshid Ghaboussi, and Michael F. Insana, Fellow, IEEE

Abstract—Quasi-static elasticity imaging techniques rely on model-based mathematical inverse methods to estimate mechanical parameters from force–displacement measurements. These techniques introduce simplifying assumptions that preclude exploration of unknown mechanical properties with potential diagnostic value. We previously reported a data-driven approach to elasticity imaging using artificial neural networks (NNs) that circumvents limitations associated with model-based inverse methods. NN constitutive models can learn stress–strain behavior from force–displacement measurements using the autoprogressive (AutoP) method without prior assumptions of the underlying constitutive model. However, information about internal structure was required. We invented Cartesian NN constitutive models (CaNNCMs) that learn the spatial variations of material properties. We are presenting the first implementation of CaNNCMs trained with AutoP to develop data-driven models of 2-D linear-elastic materials. Both simulated and experimental force–displacement data were used as input to AutoP to show that CaNNCMs are able to model both continuous and discrete material property distributions with no prior information of internal object structure. Furthermore, we demonstrate that CaNNCMs are robust to measurement noise and can reconstruct reasonably accurate Young’s modulus images from a sparse sampling of measurement data. CaNNCMs are an important step toward clinical use of data-driven elasticity imaging using AutoP.

Index Terms—Machine learning, elastography, finite element analysis, inverse problems.

I. 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

Manuscript received September 19, 2018; revised October 30, 2018 and October 31, 2018; accepted October 31, 2018. Date of publication November 5, 2018; date of current version May 1, 2019. This work was supported by NCI and NIBIB of the National Institutes of Health under Awards R01 CA168575 and R21 EB023402. (Corresponding author: Cameron Hoerig.)

C. Hoerig and M. F. Insana are with the Department of Bioengineering and the Beckman Institute of Advanced Science and Technology, University of Illinois at Urbana–Champaign, Urbana, IL 61801 USA (e-mail: hoerig2@illinois.edu).

J. Ghaboussi is with the Department of Civil and Environmental Engineering, University of Illinois at Urbana–Champaign, Urbana, IL 61801 USA.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TMI.2018.2879495

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 \mathbf{x} , $\boldsymbol{\theta}(\mathbf{x})$,

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

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, $\hat{\mathbf{u}}_k^n$ are the measured displacement vectors, \mathbf{u}_k^n are displacements estimated via a forward problem (e.g., via finite element analysis (FEA)), and f_u is often the L_2 norm of their difference. 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 ($\boldsymbol{\theta}(\mathbf{x}) = E(\mathbf{x})$). That is, if $\boldsymbol{\theta}(\mathbf{x})$ is a vector composed of nonzero elements from the constitutive matrix, then this model assumes Young’s modulus $E(\mathbf{x})$ is the only unknown. With larger deformations applied quickly, tissues exhibit non-linear [1], [2] and viscoelastic [3] 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 [4] proved that a single displacement measurement is insufficient to uniquely estimate Young’s modulus. Barbone and Gokhale [5] later showed that the Young’s modulus distribution can be determined up to a multiplicative constant if multiple displacement measurements are available. Instead of relying on multiple displacement measurements, other approaches to the inverse problem include *a priori* information [6] and/or regularization [7]–[9]. More recently, Tyagi *et al.* [10] demonstrated how measurements of the surface force applied

by the US probe [11], [12] 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 [13], [14]. QUSE has also been extended to the estimation of non-linear and viscoelastic properties of tissues [15]–[22], which provides more diagnostic information than the Young's modulus [2], [23]. For a comprehensive review of model-based methods, see [24].

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) [25]. AutoP combines FEA and artificial neural networks (NNs) to build data-driven constitutive models from force-displacement data [26]. AutoP has been used in many civil and geotechnical engineering applications to model linear, nonlinear, path-dependent, and time-dependent material properties [27]–[34]. 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.

II. METHODS

All prior implementations of AutoP for mechanical modeling have used a form of material property networks (MPN, left side of Fig. 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 (e.g., [32], [35]). Strain values applied to the input of the MPN were scaled by a vector \mathbf{S}^ε whereas the stresses at the output were scaled by the vector \mathbf{S}^σ . 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

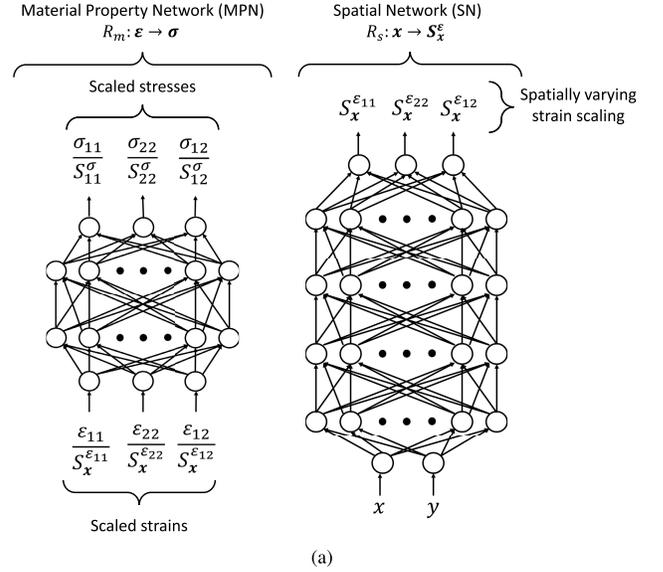


Fig. 1. Structure of the CaNNCM composed of a MPN and SN. The MPN learns a “reference” material property whereas the SN learns spatial variation of the reference.

NNCMs (CaNNCMs) that simultaneously learn material property and geometric information.

The architecture of a CaNNCM is depicted in Fig. 1 and its core theory of operation is described in [36]. It is comprised of both a MPN and spatial network (SN). The structure and function of the MPN is unchanged, 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, $\mathbf{S}^\varepsilon \rightarrow \mathbf{S}_x^\varepsilon$, meaning the SN transforms the strain vectors input to the MPN. The SN can be represented by the function $R_s : \mathbf{x} \rightarrow \mathbf{S}_x^\varepsilon$.

While previous work with NNCMs and AutoP demonstrate the ability to learn complex material properties [30]–[33], [37], 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 (1) and the objective function is minimized when the network learns the spatially-varying stress-strain relationship described by the measured data:

$$R_m, R_s = \underset{R_{\theta_m}, R_{\theta_s} \in \mathbb{R}}{\operatorname{argmin}} \sum_{n=1}^{N_p} \sum_{k=1}^{N_d} f_u(\mathbf{u}_k^n \{R_m, R_s\}, \hat{\mathbf{u}}_k^n), \quad (2)$$

where R_{θ_m} refers to the weights of the MPN and R_{θ_s} are the weights of the SN. Here, $f_u(\cdot)$ is the L_1 norm of the difference between measured $\hat{\mathbf{u}}_k^n$ 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 (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 $f_u(\cdot)$ 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. II-A how (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 $f_u(\cdot)$ only indirectly affects 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 $R_m, R_s : \boldsymbol{\varepsilon}(\mathbf{x}) \rightarrow \boldsymbol{\sigma}(\mathbf{x})$. 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 (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.

A. The Autoprogressive Method

Training CaNNCMs requires an adjustment to the AutoP procedure described in prior reports [25], [26]. AutoP uses FEA to compute stresses and strains in response to force and displacement load increments. Internal \mathbf{I}^n and external \mathbf{P}^n forces are balanced for boundary conditions (BCs) applied in the n th load increment in the solution of a FEA [38]:

$$\mathbf{P}^n = \mathbf{I}^n \quad (3)$$

$$\mathbf{I}^n = \mathbf{I}^{n-1} + \mathbf{K}^n \Delta \mathbf{U}^n \quad (4)$$

$$\mathbf{K}^n \Delta \mathbf{U}^n = \mathbf{P}^n - \mathbf{I}^{n-1} \quad (5)$$

$$\mathbf{K}^n = \sum_{e=1}^{N_e} \int_{\Omega_e} \mathbf{B}_e^T \overbrace{\mathbf{D}^n(\mathbf{x})}^{\text{CaNNCM}} \mathbf{B}_e d\Omega_e \quad (6)$$

$$\mathbf{I}^{n-1} = \sum_{e=1}^{N_e} \int_{\Omega_e} \mathbf{B}_e^T \overbrace{\boldsymbol{\sigma}^{n-1}(\mathbf{x})}^{\text{CaNNCM}} d\Omega_e \quad (7)$$

where \mathbf{K}^n is the tangent stiffness matrix computed in the n th load increment, $\Delta \mathbf{U}^n$ is the vector of displacement increments, \mathbf{P}^n is the vector of applied surface forces, and \mathbf{I}^n is the vector of internal resisting forces. In Eq. 5, \mathbf{I}^{n-1} is expressed as the sum over all N_e elements by multiplying the strain-displacement matrix \mathbf{B}_e with the stress vector $\boldsymbol{\sigma}^{n-1}(\mathbf{x})$

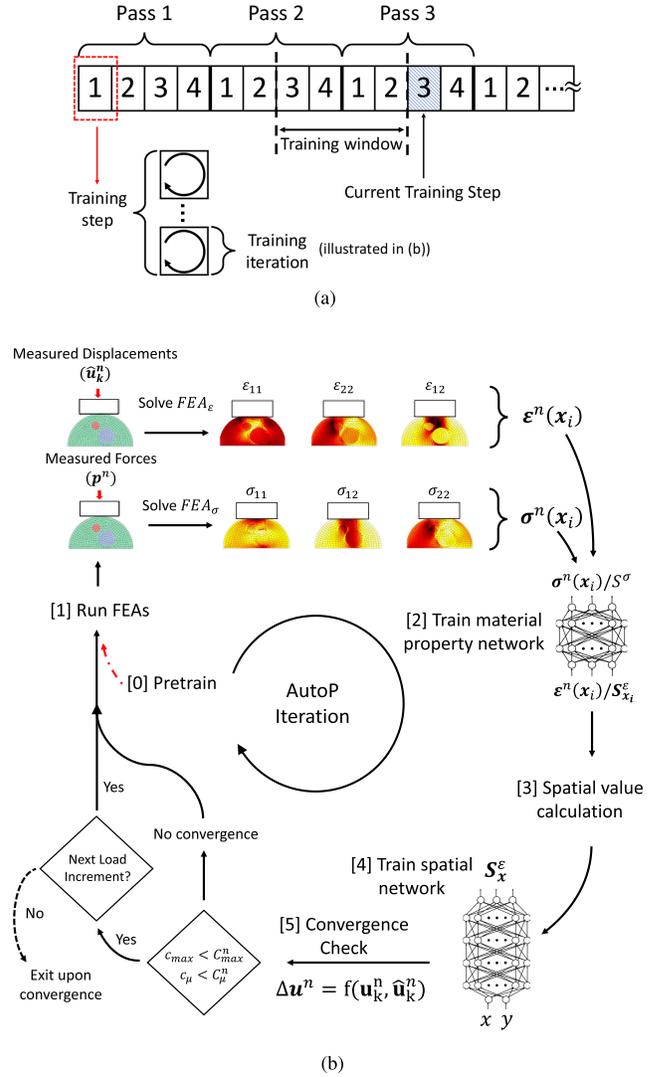


Fig. 2. (a) Hierarchy of training passes, steps, and iterations in AutoP. The training window determines the number of preceding training steps from which stress-strain data are included during stages 2–4 of the current step. (b) One AutoP training iteration using a CaNNCM is illustrated. Adding the spatial network necessitates two extra stages where \mathbf{S}_x^ε 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 [25].

and integrating over the element domains Ω_e . Force BCs reside in \mathbf{P}^n and displacement BCs populate components of $\Delta \mathbf{U}^n$.

A forward analysis consists of applying force and/or displacement BCs to the FE model and solving the system of equations (5) for the unknown displacement increments that that satisfy (3). During the analysis, the stiffness matrix $\mathbf{D}^n(\mathbf{x})$ and stress vector are computed using a constitutive model or, in our method, the CaNNCM. To be clear, both the stiffness matrix and stress vector in (6) and (7) are computed from the CaNNCM being trained, not a pre-selected constitutive model as would be done in model-based methods. The analytical expression for $\mathbf{D}^n(\mathbf{x})$ is provided in Appendix B. FEA techniques are thus used to solve the forward problem for $\mathbf{u}_k^n\{R_m, R_s\}$ in (2).

AutoP is organized in a hierarchy of training passes, and iterations as shown in Fig. 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 processing illustrated in Fig. 2b. Many AutoP iterations are performed throughout training, during which the MPN and SN are repeatedly retrained. We will consider the j th 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.

1) 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 pre-training, 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 ($S^\sigma = 1$, $S_x^e = \mathbf{1}$).

2) Estimation of Stresses and Strains: Stage [1] consists of estimating stresses and strains from measurement data. Forces for the n th load increment are applied to the FE model in FEA_σ . Referring back to (5), force measurements are imposed as BCs in \mathbf{P}^n and total mesh deformation is computed. Due to equilibrium conditions relating stresses to forces, all stresses $\sigma^n(\mathbf{x})$ computed throughout the model in FEA_σ are assumed to be physically consistent estimates of the true stress. Similarly, in FEA_ε US probe and internal displacement measurements from the n th load increment are input as components of ΔU^n to compute displacements of the remaining nodes. Due to compatibility requirements relating node displacements to strains, the strains $\varepsilon^n(\mathbf{x})$ computed in FEA_ε are assumed to be physically consistent estimates of the true strains. Recall that R_m^j and R_s^j are invoked by (6)–(7) to solve each FEA.

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

3) Training MPN: A total of N_x stresses $\sigma^n(\mathbf{x}_i)$ and strains $\varepsilon^n(\mathbf{x}_i)$ are computed in FEA_σ and FEA_ε , respectively. Each strain can be scaled by the corresponding $S_{x_i}^e$ computed by R_s^j and input to R_m^j to compute a new stress $\hat{\sigma}^n(\mathbf{x}_i)$. The goal of stage [2] is to adjust the weights of the MPN to minimize the error between $\sigma^n(\mathbf{x})$ and $\hat{\sigma}^n(\mathbf{x})$:

$$R_m^{j+1} = \operatorname{argmin}_{R_{\theta_m} \in \mathbb{R}} \sum_{i=1}^{N_x} \sum_{n=1}^{N_p} \sum_{l=1}^{N_\sigma} f_m(\sigma_l^n(\mathbf{x}_i), \overbrace{R_m^j \{\varepsilon_l^n(\mathbf{x}_i), R_s^j(\mathbf{x}_i)\}}^{\hat{\sigma}_l^n(\mathbf{x}_i)}). \quad (8)$$

$R_s(\mathbf{x}_i)$ is the output of the SN at \mathbf{x}_i and N_σ is the number of stress-strains pairs at \mathbf{x}_i in the n th 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 A. $f_m(\cdot)$ is the L_2 norm of the difference between $\sigma^n(\mathbf{x})$ and $\hat{\sigma}^n(\mathbf{x})$; i.e., the MPN is trained via backpropagation.

Eq. (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 $\varepsilon^n(\mathbf{x})$ to $\sigma^n(\mathbf{x})$ in heterogeneous materials. For example, in the case where $\varepsilon^n(\mathbf{x}_i) = \varepsilon^n(\mathbf{x}_j)$, it is not necessarily true that $\sigma^n(\mathbf{x}_i) = \sigma^n(\mathbf{x}_j)$ (i.e., $R_m : \varepsilon \rightarrow \sigma$ is not bijective). Thus, the SN must supply additional information in the form of S_x^e so that the MPN can determine which stress should be returned for a given strain.

4) Spatial Scaling Calculation: The spatial values S_x^e 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} , $\sigma^n(\mathbf{x})$, and $\varepsilon^n(\mathbf{x})$. The goal is to further minimize the error between $\sigma^n(\mathbf{x})$ and $\hat{\sigma}^n(\mathbf{x})$ by altering the spatial values instead of the weights of the MPN:

$$S_{x_i}^e = \operatorname{argmin}_{\hat{S}_{x_i}^e \in \mathbb{R}} \sum_{n=1}^{N_p} \sum_{l=1}^{N_\sigma} f_m(\sigma_l^n(\mathbf{x}_i), R_m^{j+1} \{\varepsilon_l^n(\mathbf{x}_i), \hat{S}_{x_i}^e\}), \quad (9)$$

where R_m^{j+1} is the output of the MPN retrained in (8).

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

$$R_s^{j+1} = \operatorname{argmin}_{R_{\theta_s} \in \mathbb{R}} \sum_{n=1}^{N_p} \sum_{i=1}^{N_x} f_s(S_{x_i}^e, R_s^j \{x_i\}). \quad (10)$$

Details of solving (9) and (10) are covered in [36].

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

$$\Delta \mathbf{u}_k^n = |\mathbf{u}_k^n - \hat{\mathbf{u}}_k^n| = f_u(\mathbf{u}_k^n \{R_m, R_s\}, \hat{\mathbf{u}}_k^n) \quad (11)$$

which is the objective function defined in (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:

$$c_{\max}^n = \frac{\max(\Delta \mathbf{u}_k^n)}{|\max(\mathbf{u}_k^n)|} \quad (12)$$

$$c_\mu^n = \frac{\operatorname{mean}(\Delta \mathbf{u}_k^n)}{|\max(\mathbf{u}_k^n)|}. \quad (13)$$

We define convergence criteria C_{\max}^n and C_μ^n for the n th 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.

B. 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 ($\approx 10.93 \pm 0.57$, 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 [25] and separate samples of each gelatin mixture were stored to independently estimate Young's modulus values via macro-indentation methods [39].

We used the same experimental setup described in [25]. 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 [40] 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. 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 $\pm 4\text{mm}$. 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.

C. 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. 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 fixed to mimic contact between the

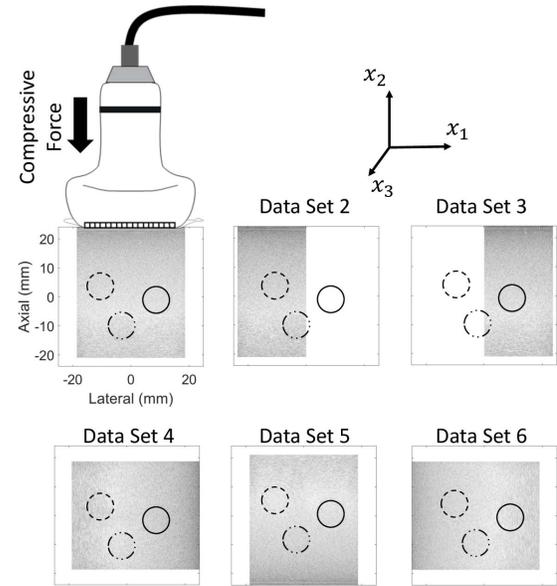


Fig. 3. The cubic phantom was imaged from four sides. Data sets 1–3 were obtained with the US probe along the same surface but at different lateral positions. For sets 4, 5, and 6, the phantom was rotated 90° , 180° and 270° about x_3 , respectively, while keeping the probe laterally centered.

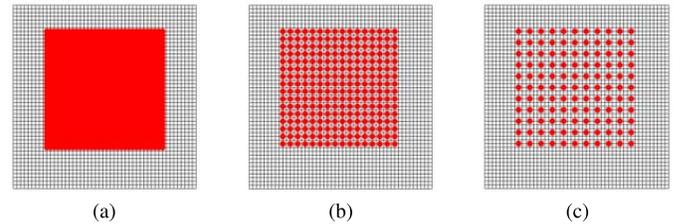


Fig. 4. Rectilinear mesh used in FEM-1. Highlighted nodes indicate locations where displacement data was provided in FEA _{ϵ} . (a) Displacements are given at every node in the ROI, (b) at nodes separated by a minimum of 1.5mm, or (c) at nodes with a minimum 3mm separation.

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 _{σ} and FEA _{ϵ} , which are described in Sect. II-A. Force loads, when applied as boundary conditions (BCs) in FEA _{σ} or in a forward problem, were defined as concentrated forces to the top of the probe model. Similarly, probe displacements in FEA _{ϵ} 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.

D. Simulated Force-Displacement Measurements

First, we tested AutoP employing CaNNCMs and noise-free force and displacement data. Three different material property distributions (Figs. 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 the

TABLE I
FOUR CASES FOR APPLYING AXIAL DISPLACEMENTS IN AutoP.
CASES 2–4 ARE ILLUSTRATED IN FIG. 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

center of a 10 kPa background. Model 2 was a replicate of the gelatin phantom described in Sect. II-B. 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 [41]. Models 1 and 3 were chosen to enable comparison with results reported earlier where stress-strain data were known [36].

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.

E. 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 the supplementary material. 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[†] and the latter as Set 1^{††}.

F. 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 Appendix A. Any differences in how CaNNCMs were trained lie in how the force-displacement data were sampled. We will show in Sect. III 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_c. Table I 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. 4a) indicates all nodes in the ROI were assigned displacements. For Cases 3 and 4 (Figs. 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.

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 $\boldsymbol{\epsilon} = [0.003 \ 0.005 \ 0.0001]$ and computing the stiffness matrix \hat{D}_{ij} using (17) in Appendix B. The Young’s modulus distribution $E(\mathbf{x})$ was then estimated by varying \mathbf{x} in the domain of the mesh and evaluating the function

$$E(\mathbf{x}) = \frac{S_x^\sigma}{S_x^{\epsilon_2}} \hat{D}_{22} (1 - \nu^2), \quad (14)$$

where $\nu = 0.5$ and $S_x^{\epsilon_2}$ is the axial component of the spatial scaling vector at \mathbf{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. Errors were computed as

$$e_x^E = \frac{|E_x^{target} - E_x^{NN}|}{E_x^{target}} \quad (15)$$

where E_x^{target} is the target Young’s modulus distribution and E_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_x^E only within the ROI for all cases.

The following describes each of the CaNNCMs trained.

1) *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.

2) *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.

3) *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

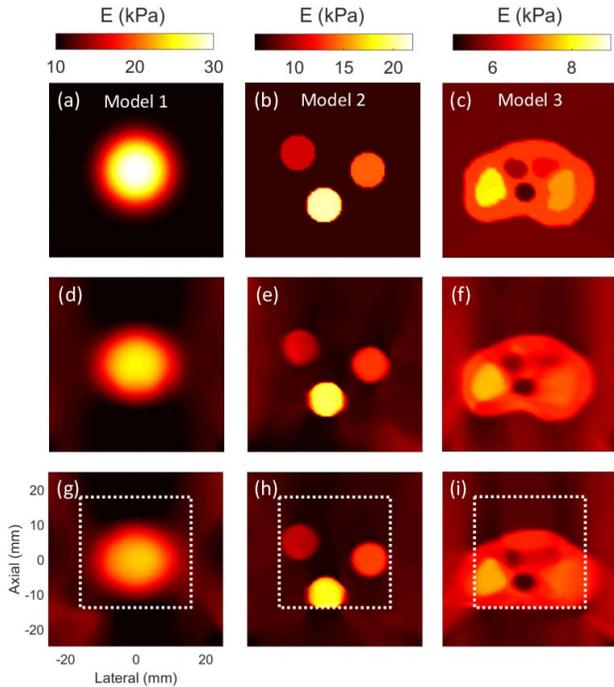


Fig. 5. (Top Row) Target Young's modulus distributions for Models 1–3. Young's modulus reconstructions from CaNNCMs trained with noise-free simulated displacements for Case 1 (Middle row) and Case 2 (Bottom row). The white box indicates the boundaries of the ROI. Color scales at the top of the figure apply to all images within the column.

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.

III. RESULTS

Young's modulus images reconstructed by CaNNCMs trained with noise-free force-displacement data are displayed in Fig. 5. Errors in the modulus estimates computed with (15) are compiled in Table II 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 [36] for the same Model. It could be corrected by increasing the number of iterations performed when solving (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 (9) could reduce the sensitivity to

TABLE II

YOUNG'S MODULUS RECONSTRUCTION ERRORS AND AUTO P RUN TIME. MODELS ARE ILLUSTRATED IN FIGS. 5A-C. SETS ARE ILLUSTRATED IN FIG. 3. CASES ARE DESCRIBED IN TABLE I. SIMULATED RF DATA SETS WITH 30 dB AND 15 dB SNR ARE DENOTED \dagger AND $\dagger\dagger$, RESPECTIVELY. A SUPERSCRIPED LETTER INDICATES SUBFIGURE IN FIG. 8. THE LAST COLUMN INDICATES THE FIGURE NUMBER OF THE CORRESPONDING YOUNG'S MODULUS IMAGE

(Model #){Set}[case]	Modulus Error Mean \pm 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 \dagger }[2]	0.0961 ± 0.1240	80	40	6a
(2){1 \dagger }[3]	0.1325 ± 0.1294	79	40	6b
(2){1 \dagger }[4]	0.0970 ± 0.1126	82	42	6c
(2){1 $\dagger\dagger$ }[2]	0.0914 ± 0.1336	80	40	6d
(2){1 $\dagger\dagger$ }[3] ^(a)	0.0899 ± 0.1111	110	56	6e
(2){1 $\dagger\dagger$ }[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''}[3]	0.4380 ± 0.2163	393	69	7c
(2){1,2,3}[3] ^(c)	0.3228 ± 0.1713	393	69	7f

the number of iterations and noise at the cost of increased computational complexity.

Fig. 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. 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. 6h), but fewer noise artifacts are present. Generally, the trend appears to be that 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

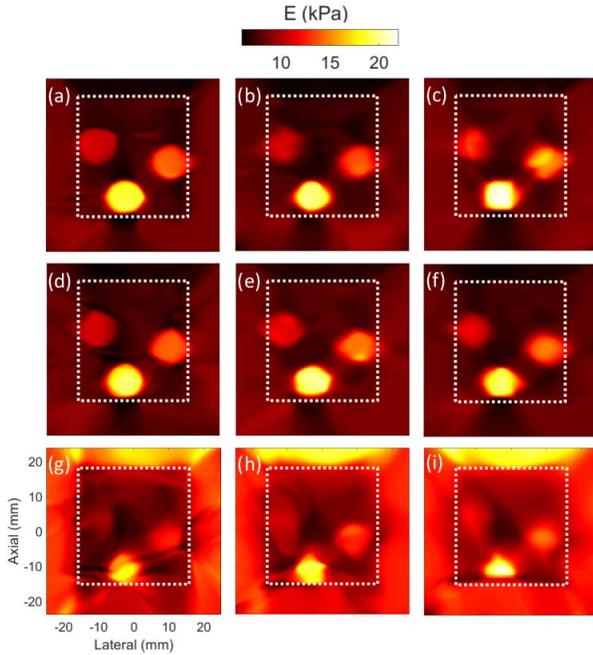


Fig. 6. Comparisons of Young's modulus image reconstructions from RF echo signals for three levels of noise and different displacement sampling densities. Force-displacement measurements from rows 1 and 2 are from simulated echo data at 30 and 15 dB SNR. Row 3 is from experimental phantom measurements with echo SNR in the same range using Data Set 1. Columns from left to right correspond to sampling Cases 2–4.

force-displacement data (Figs. 6g–i), but not when trained with simulated RF Data Sets 1^\dagger or $1^{\dagger\dagger}$ (Figs. 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.

Young's modulus reconstructions by CaNNCMs trained with experimental measurement data are shown in Fig. 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. 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. 7d, where said inclusion is only partially within view. Material properties estimated by each CaNNCM are consistent, barring Figs. 7g and 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

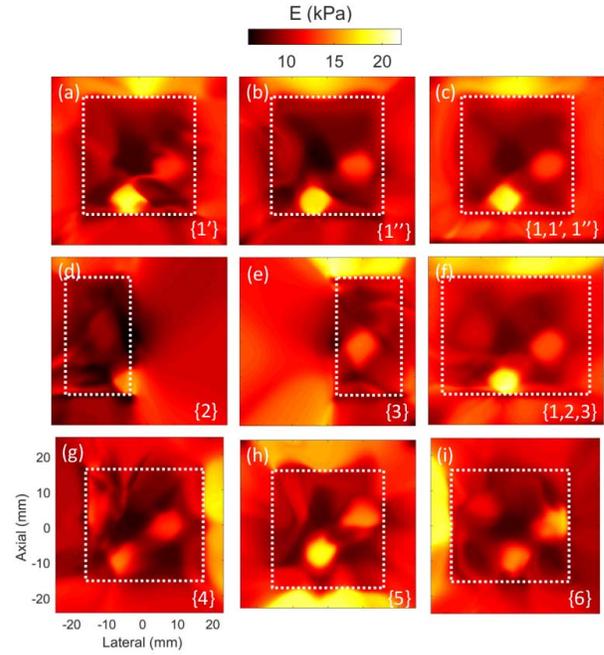


Fig. 7. Young's modulus image reconstruction by CaNNCMs trained from experimental measurements. Bracketed numbers in the lower-right corner of each image indicates the Data Set(s) used to train the CaNNCM. The dotted frame indicates the region over which displacements were estimated.

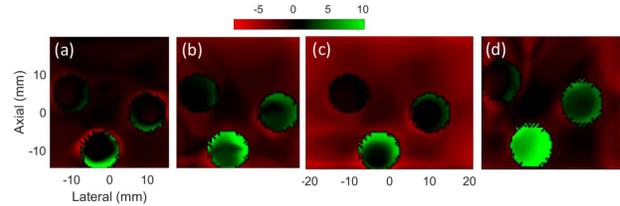


Fig. 8. Maps of Young's modulus error for the four tests indicated by superscript letters in Table II. The error is the difference between the target distribution and that computed by the CaNNCM. Red/green indicates the CaNNCM-estimated Young's modulus was too large/small.

all three inclusions. Furthermore, if we compare displacement errors computed by (12) and (13) for CaNNCMs corresponding to Figs. 7a and 7g, there is no significant difference ($c_{\max}^n = 0.0796$, $c_{\mu}^n = 0.0694$ 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. 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. 8 and the CaNNCMs are indicated in Table II 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. 8d), the CaNNCM more accurately estimates the Young's modulus of the inclusions. The increased error reported in Table II is mostly attributed to over-estimating the Young's modulus of the background gelatin, by up to 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.

IV. 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., [19], [24], [42]). 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 (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 [25]. 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. 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. 6g). That said, a cubic phantom with three parallel cylindrical inclusions limits the conclusions that can be drawn. Comparing Figs. 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. 7c and 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_{\max}^n and c_{μ}^n) 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. 6a–f are much more accurate than Figs. 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.

Developing our approach into a clinically feasible imaging modality will require we 1) reduce the CaNNCM training time and 2) adapt the network architecture to account for 3-D materials exhibiting non-linear and viscoelastic mechanical behaviors. Given that other groups have increased FEA computation speed by several orders of magnitude using GPUs [43]–[45], we believe AutoP training time can be greatly reduced using a custom FEA solver utilizing GPU resources. Changes to the CaNNCM architecture are not likely to require adjustments to the general AutoP stages specified in Sec. II-A. Previous investigations using NNCMs for modeling complex material properties required adjustments to the FEA formulation and

network architecture (e.g., [30]–[33], [37]), but no changes were made to the overall process of AutoP. Due to the significant structural difference of CaNNCMs compared to NNCMs, we provided a new description of AutoP as a method to minimize an objective function that will be applicable to the general CaNNCM architecture comprised of a MPN and SN. Regardless of network changes, the same problem is solved in each stage of AutoP. The differences resides in how to reformulate each problem based on the CaNNCM structure (e.g., how to change (9) to accommodate a SN with different inputs/outputs) . Future investigations can therefore describe these changes as necessary.

V. 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. Current limitations of the method as applied to linear-elastic media suggest a need for 3-D model building and more efficient FEA computation.

APPENDIX A 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 [25]. 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 [37, Algorithm 1] ($N = 50$, $N^\sigma = 8$ due to frame invariance and training window, spatial scaling update rate $\eta = 0.5$). 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 [46]

over 15 epochs. Conversely, the SN training was implemented in TensorFlow using the Adam optimizer [47] (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_{\max}^n, C_μ^n) . 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.

APPENDIX B COMPUTING STIFFNESS MATRIX FROM CANNCM

Hashash *et al.* [48] derived an analytical function for computing the stiffness matrix D_{ij} from the weights and activations of a MPN. Note that the “activation” of a node is the weighted sum of the inputs to said node before passing through the activation function. The same analytical function can be used to compute the stiffness matrix for a CaNNCM in response to an input strain with a minor adjustment:

$$D_{ij}(\mathbf{x}) = \frac{\partial \sigma_i(\mathbf{x})}{\partial \varepsilon_j(\mathbf{x})} \frac{S_i^\sigma}{S_x^{\varepsilon_j}} = \frac{S_i^\sigma}{S_x^{\varepsilon_j}} \hat{D}_{ij}(\mathbf{x}) \quad (16)$$

where

$$\begin{aligned} \hat{D}_{ij} &= (1 - \tanh(f_i)^2) \\ &\times \sum_{c=1}^{N_{h_2}} \left[w_{ic} (1 - \tanh(f_c)^2) \right. \\ &\times \sum_{b=1}^{N_{h_1}} \left\{ w_{cb} (1 - \tanh(f_b)^2) \right. \\ &\times \left. \left. \sum_{a=1}^{N_i} \left(w_{ba} (1 - \tanh(\varepsilon_j)^2) \right) \right\} \right]. \quad (17) \end{aligned}$$

In (17), the values f_i , f_c , 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} .

ACKNOWLEDGMENT

The authors would like to thank H. Hashemi and H. Rivaz for providing the source code for their displacement estimation algorithm.

REFERENCES

- [1] A. Samani and D. Plewes, “A method to measure the hyperelastic parameters of *ex vivo* breast tissue samples,” *Phys. Med. Biol.*, vol. 49, no. 18, p. 4395, 2004.
- [2] J. J. O’Hagan and A. Samani, “Measurement of the hyperelastic properties of 44 pathological *ex vivo* breast tissue samples,” *Phys. Med. Biol.*, vol. 54, no. 8, p. 2557, 2009.

- [3] M. Sridhar, J. Liu, and M. Insana, "Viscoelasticity imaging using ultrasound: Parameters and error analysis," *Phys. Med. Biol.*, vol. 52, no. 9, p. 2425, 2007.
- [4] P. E. Barbone and J. C. Bamber, "Quantitative elasticity imaging: What can and cannot be inferred from strain images," *Phys. Med. Biol.*, vol. 47, no. 12, pp. 2147–2164, 2002.
- [5] 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.
- [6] 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," *Phys. Med. Biol.*, vol. 51, no. 1, pp. 95–112, 2006.
- [7] F. Kallel and M. Bertrand, "Tissue elasticity reconstruction using linear perturbation method," *IEEE Trans. Med. Imag.*, vol. 15, no. 3, pp. 299–313, Jun. 1996.
- [8] U. Albocher, P. E. Barbone, M. S. Richards, A. A. Oberai, and I. Harari, "Approaches to accommodate noisy data in the direct solution of inverse problems in incompressible plane-strain elasticity," *Inverse Problems Sci. Eng.*, vol. 22, no. 8, pp. 1307–1328, 2014.
- [9] Y. Mei, S. Kuznetsov, and S. Goenezen, "Reduced boundary sensitivity and improved contrast of the regularized inverse problem solution in elasticity," *J. Appl. Mech.*, vol. 83, no. 3, p. 031001, 2015.
- [10] M. Tyagi, S. Goenezen, P. E. Barbone, and A. A. Oberai, "Algorithms for quantitative quasi-static elasticity imaging using force data," *Int. J. Numer. Methods Biomed. Eng.*, vol. 30, no. 12, pp. 1421–1436, Dec. 2014.
- [11] Y. Qiu, M. Sridhar, J. K. Tsou, K. K. Lindfors, and M. F. Insana, "Ultrasonic viscoelasticity imaging of nonpalpable breast tumors: Preliminary results," *Acad. Radiol.*, vol. 15, no. 12, pp. 1526–1533, 2008.
- [12] 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 Med. Imag. Rev.*, vol. 2, no. 1, pp. 157–164, 2006.
- [13] Y. Mei, R. Fulmer, V. Raja, S. Wang, and S. Goenezen, "Estimating the non-homogeneous elastic modulus distribution from surface deformations," *Int. J. Solids Struct.*, vol. 83, pp. 73–80, Apr. 2015.
- [14] V. Egorov and A. P. Sarvazyan, "Mechanical imaging of the breast," *IEEE Trans. Med. Imag.*, vol. 27, no. 9, pp. 1275–1287, Sep. 2008.
- [15] S. Goenezen *et al.*, "Linear and nonlinear elastic modulus imaging: An application to breast cancer diagnosis," *IEEE Trans. Med. Imag.*, vol. 31, no. 8, pp. 1628–1637, Aug. 2012.
- [16] J. Palacio-Torralba *et al.*, "Quantitative diagnostics of soft tissue through viscoelastic characterization using time-based instrumented palpation," *J. Mech. Behav. Biomed. Mater.*, vol. 41, pp. 60–149, Jan. 2015.
- [17] 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.
- [18] M. F. Insana, C. Pellot-Barakat, M. Sridhar, and K. K. Lindfors, "Viscoelastic imaging of breast tumor microenvironment with ultrasound," *J. Mammary Gland Biol. Neoplasia*, vol. 9, no. 4, pp. 393–404, 2004.
- [19] S. L. Sridhar, Y. Mei, and S. Goenezen, "Improving the sensitivity to map nonlinear parameters for hyperelastic problems," *Comput. Methods Appl. Mech. Eng.*, vol. 331, pp. 474–491, Apr. 2018.
- [20] M. Bayat *et al.*, "Automated *in vivo* sub-hertz analysis of viscoelasticity (SAVE) for evaluation of breast lesions," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 10, pp. 2237–2247, Oct. 2018.
- [21] 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.
- [22] S. Goenezen, P. Barbone, and A. A. Oberai, "Solution of the nonlinear elasticity imaging inverse problem: The incompressible case," *Comput. Methods Appl. Mech. Eng.*, vol. 200, nos. 13–16, pp. 1406–1420, 2011.
- [23] T. Krouskop, T. M. Wheeler, F. Kallel, B. S. Garra, and T. Hall, "Elastic moduli of breast and prostate tissues under compression," *Ultrason. Imag.*, vol. 20, pp. 260–274, Oct. 1998.
- [24] M. Doyley, "Model-based elastography: A survey of approaches to the inverse elasticity problem," *Phys. Med. Biol.*, vol. 57, no. 3, p. R35, 2012.
- [25] C. Hoerig, J. Ghaboussi, and M. F. Insana, "An information-based machine learning approach to elasticity imaging," *Biomech. Model. Mechanobiol.*, vol. 16, no. 3, pp. 805–822, 2017.
- [26] J. Ghaboussi, D. A. Pecknold, M. Zhang, and R. M. Haj-Ali, "Autoprospective training of neural network constitutive models," *Int. J. Numer. Methods Eng.*, vol. 42, no. 1, pp. 105–126, 1998.
- [27] D. E. Sidarta and J. Ghaboussi, "Constitutive modeling of geomaterials from non-uniform material tests," *Comput. Geotech.*, vol. 22, no. 1, pp. 53–71, 1998.
- [28] Y. M. A. 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 *Proc. Geotech. Eng. Inf. Technol. Age (GeoCongr)*, 2006, pp. 1–6.
- [29] Y. M. A. Hashash, J. Ghaboussi, and S. Jung, "Characterizing granular material constitutive behavior using selfsim with boundary load-displacement measurements," in *Proc. Eng., Construct., Oper. Challenging Environ.*, 2006, pp. 1–8.
- [30] S. Jung and J. Ghaboussi, "Characterizing rate-dependent material behaviors in self-learning simulation," *Comput. Methods Appl. Mech. Eng.*, vol. 196, nos. 1–3, pp. 608–619, 2006.
- [31] S. Jung and J. Ghaboussi, "Inverse identification of creep of concrete from *in situ* load–displacement monitoring," *Eng. Struct.*, vol. 32, no. 5, pp. 1437–1445, 2010.
- [32] J. Kim, J. Ghaboussi, and A. S. Elnashai, "Mechanical and informational modeling of steel beam-to-column connections," *Eng. Struct.*, vol. 32, no. 2, pp. 449–458, 2012.
- [33] 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," *J. Aerosp. Eng.*, vol. 25, no. 2, pp. 256–272, 2012.
- [34] J. Ghaboussi, D. A. Pecknold, Y. Hashash, and T.-H. Kwon, "Accurate determination of intraocular pressure and characterization of mechanical properties of the cornea," U.S. Patent 8070679, Dec. 6, 2011. [Online]. Available: <https://www.google.com/patents/US8070679>
- [35] 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," *J. Earthq. Eng.*, vol. 11, no. 3, pp. 453–467, 2007.
- [36] C. Hoerig, J. Ghaboussi, and M. F. Insana. (Sep. 2018). "Cartesian neural network constitutive models for data-driven elasticity imaging." [Online]. Available: <https://arxiv.org/abs/1809.04121>
- [37] A. H. Gandomi and G. J. Yun, "Coupled SelfSim and genetic programming for non-linear material constitutive modelling," *Inverse Problems Sci. Eng.*, vol. 23, no. 7, pp. 1101–1119, 2015.
- [38] J. Ghaboussi, D. A. Pecknold, and X. S. Wu, *Nonlinear Computational Solid Mechanics*. Boca Raton, FL, USA: CRC Press, 2017.
- [39] K. N. Altahhan, Y. Wang, N. Sobh, and M. F. Insana, "Indentation measurements to validate dynamic elasticity imaging methods," *Ultrason. Imag.*, vol. 38, no. 5, pp. 332–345, 2016.
- [40] H. S. Hashemi and H. Rivaz, "Global time-delay estimation in ultrasound elastography," *IEEE Trans. Ultrason., Ferroelect., Freq. Control*, vol. 64, no. 10, pp. 1625–1636, Oct. 2017.
- [41] C. Hoerig, W. Reyes, L. Fabre, J. Ghaboussi, and M. F. Insana, "Imaging spatially varying biomechanical properties with neural networks," *Proc. SPIE, Med. Imag., Ultrason. Imag. Tomography*, vol. 10139, p. 1013905, Mar. 2017, doi: 10.1117/12.2254331.
- [42] Y. B. Fu, C. K. Chui, C. L. Teo, and E. Kobayashi, "Elasticity imaging of biological soft tissue using a combined finite element and non-linear optimization method," *Inverse Problems Sci. Eng.*, vol. 23, no. 2, pp. 179–196, 2015.
- [43] J. M. Huang, S. K. Ong, and A. Y. C. Nee, "Real-time finite element structural analysis in augmented reality," *Adv. Eng. Softw.*, vol. 87, pp. 43–56, Sep. 2015.
- [44] 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 Anal. Des.*, vol. 105, pp. 79–89, Nov. 2015.
- [45] S. F. Johnsen *et al.*, "NiftySim: A GPU-based nonlinear finite element package for simulation of soft tissue biomechanics," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 10, no. 7, pp. 1077–1095, 2015.
- [46] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The RPROP algorithm," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 1, Mar./Apr. 1993, pp. 586–591.
- [47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, pp. 1–15, Dec. 2014. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [48] Y. M. A. Hashash, S. Jung, and J. Ghaboussi, "Numerical implementation of a neural network based material model in finite element analysis," *Int. J. Numer. Methods Eng.*, vol. 59, no. 7, pp. 989–1005, 2004.