

Data-driven Modeling of the Mechanical Behavior of Anisotropic Soft Biological Tissue

Vahidullah Tac¹, Vivek D. Sree¹, Manuel K. Rausch², and Adrian B. Tepole^{1,3}

¹*School of Mechanical Engineering, Purdue University, West Lafayette, IN, USA*

²*Department of Aerospace Engineering and Engineering Mechanics, the University of Texas at Austin, Austin, TX, USA*

³*Weldon School of Biomedical Engineering, Purdue University, West Lafayette, IN, USA*

Abstract

Constitutive models that describe the mechanical behavior of soft tissues have advanced greatly over the past few decades. These expert models are generalizable and require the calibration of a number of parameters to fit experimental data. However, inherent pitfalls stemming from the restriction to a specific functional form include poor fits to the data, non-uniqueness of fit, and high sensitivity to parameters. In this study we design and train fully connected neural networks as material models to replace or augment expert models. To guarantee objectivity, the neural network takes isochoric strain invariants as inputs, and outputs the value of strain energy and its derivatives with respect to the invariants. Convexity of the material model is enforced through the loss function. Direct prediction of the derivative functions -rather than just predicting the energy- serves two purposes: it provides flexibility during training, and it enables the calculation of the elasticity tensor through back-propagation. We showcase the ability of the neural network to learn the mechanical behavior of porcine and murine skin from biaxial test data. Crucially, we show that a multi-fidelity scheme which combines high fidelity experimental data with low fidelity analytical data yields the best performance. The neural network material model can then be interpreted as the best extension of an expert model: it learns the features that an expert has encoded in the analytical model while fitting the experimental data better. Finally, we implemented a general user material subroutine (UMAT) for the finite element software Abaqus and thereby make our advances available to the broader computational community. We expect that the methods and software generated in this work will broaden the use of data-driven constitutive models in biomedical applications.

Keywords: Machine Learning, Nonlinear finite elements, Constitutive modeling, Abaqus User Subroutine UMAT, multi-fidelity models, Skin mechanics

Introduction

Skin is the largest organ in the body and understanding its mechanical properties is a crucial step in many biomedical applications, from the design of prostheses to surgical intervention [1, 2]. The tissue microstructure is characterized by the presence of semi-flexible biopolymer fiber networks such as collagen and elastin, which endow skin with nonlinear and anisotropic mechanical behavior [3]. The mechanical properties of skin are actually common across many collagenous soft connective tissue [4, 5]. Traditionally, the mechanical behavior of skin and other soft tissues has been modelled using expert-constructed constitutive equations [6, 7, 8, 9]. In this approach, a functional form describing the main features of the mechanical behavior of a family of materials is constructed first. Then, the free parameters in the equations are fitted to a specific material in the family to obtain a calibrated model. Thus, this approach necessitates choosing a specific functional form for the constitutive model. In turn, inherent restrictions of the functional form can result in poor fitting and poor predictions. Unfortunately, even restricting our attention to skin out of all connective soft tissue, there is currently no consensus on the choice of model that is most suitable in a particular application [10, 11, 12, 13, 14, 15, 16, 17, 18].

A new, emergent approach to material modeling is the use of data-driven methods [17, 19, 20, 21]. Among them, neural network have been used successfully to describe the mechanical behavior of several materials [20, 22, 23, 24]. In this approach, the burden of choosing a suitable analytical representation is no longer present, and since the training

of the neural network is not restricted to a parametric family of functions, it often results in more accurate predictions than traditional models [25, 26, 27]. In particular, a common application of machine learning has been to create macroscale constitutive models out of micromechanical simulations [27, 24].

One potential drawback of this approach is scarcity of high fidelity test data to train the data-driven models. It is often difficult, expensive, and sometimes impossible to gather large amounts of data to train a neural network over the entire input space of deformations. One approach to overcome this problem is to use neural networks in a similar vein as the traditional model development, i.e., training a neural network using data from multiple specimens and multiple loading conditions in order to learn an average response. Then, data of a single material can be fitted by adjusting only a few parameters of the neural network [26].

The route we follow here lies between the purely data-driven approach and the expert modeling approach. Expert models already include knowledge of physics relevant to soft tissue, observations of both the mechanical behavior and underlying microstructure properties, and intuition from the modeller regarding the main features of the material response. For example, to model skin, we have restricted our attention to the framework of hyperelasticity and used expert-designed strain energy functions to fit murine and porcine skin data [15, 28]. However, the error in the fits can be undesirable, the parameters non-unique, and the predictions can be highly sensitive to the material parameters, especially outside of the loading regime used during calibration. Here we propose to use the analytical strain energy functions as low fidelity approximations, and complement the data with high fidelity experimental measurements. This approach is based on the recent literature that shows the advantage of multi-fidelity schemes over single fidelity approaches [24, 23, 29, 16].

Specifically, we train fully connected neural networks that output the strain energy and its derivatives with respect to the isochoric strain invariants, including the effect of anisotropy. The training of the neural network is done with a multi-fidelity dataset that includes low fidelity data from conventional material models, and high fidelity data from biaxial experiments. Furthermore, the loss function is designed to impose convexity constraints on the strain energy, which ensures physically meaningful elasticity tensors. Finally, we implemented a user material (UMAT) subroutine for the widely used finite element package Abaqus [30]. The UMAT reads all the parameters of a fully connected neural network from the input file and evaluates the neural network to construct the stress and elasticity tensors needed in the finite element simulation. The work shown here will extend the reach of machine learning tools to improve the modeling of soft tissue mechanics, in particular through improved constitutive models ready to be used in commercial finite element codes.

Materials and Methods

Constitutive equations for a hyperelastic material with two families of fibers

In this study we use a form of Helmholtz free energy, Ψ , that is a function of the right Cauchy-Green deformation tensor, \mathbf{C} , and two material direction vectors in the reference configuration, \mathbf{v}_0 and \mathbf{w}_0 . This form of the Helmholtz free energy function allows for greater flexibility in recreating the mechanical behavior of materials where more than one family of fibers is present or even when the orientation of fibers is random, which is usually the case in biological tissues. For soft tissues, which we assume to be nearly incompressible, the additive split into isochoric and volumetric parts is used [6],

$$\Psi = \Psi_{\text{iso}}(\hat{I}_1, \hat{I}_2, \hat{I}_{4v}, \hat{I}_{4w}) + \Psi_{\text{vol}}(J), \quad (1)$$

where $J = \sqrt{\det \mathbf{C}}$ is the volume change, and the isochoric strain invariants, \hat{I}_1 , \hat{I}_2 , \hat{I}_{4v} and \hat{I}_{4w} are defined as

$$\begin{aligned} \hat{I}_1 &= \hat{\mathbf{C}} : \mathbf{I} = \text{tr}(\hat{\mathbf{C}}), \\ \hat{I}_2 &= \frac{1}{2}[\hat{I}_1^2 - \text{tr}(\hat{\mathbf{C}}^2)], \\ \hat{I}_{4v} &= \hat{\mathbf{C}} : \mathbf{v}_0 \otimes \mathbf{v}_0 = \hat{\mathbf{C}} : \mathbf{V}_0, \\ \hat{I}_{4w} &= \hat{\mathbf{C}} : \mathbf{w}_0 \otimes \mathbf{w}_0 = \hat{\mathbf{C}} : \mathbf{W}_0. \end{aligned} \quad (2)$$

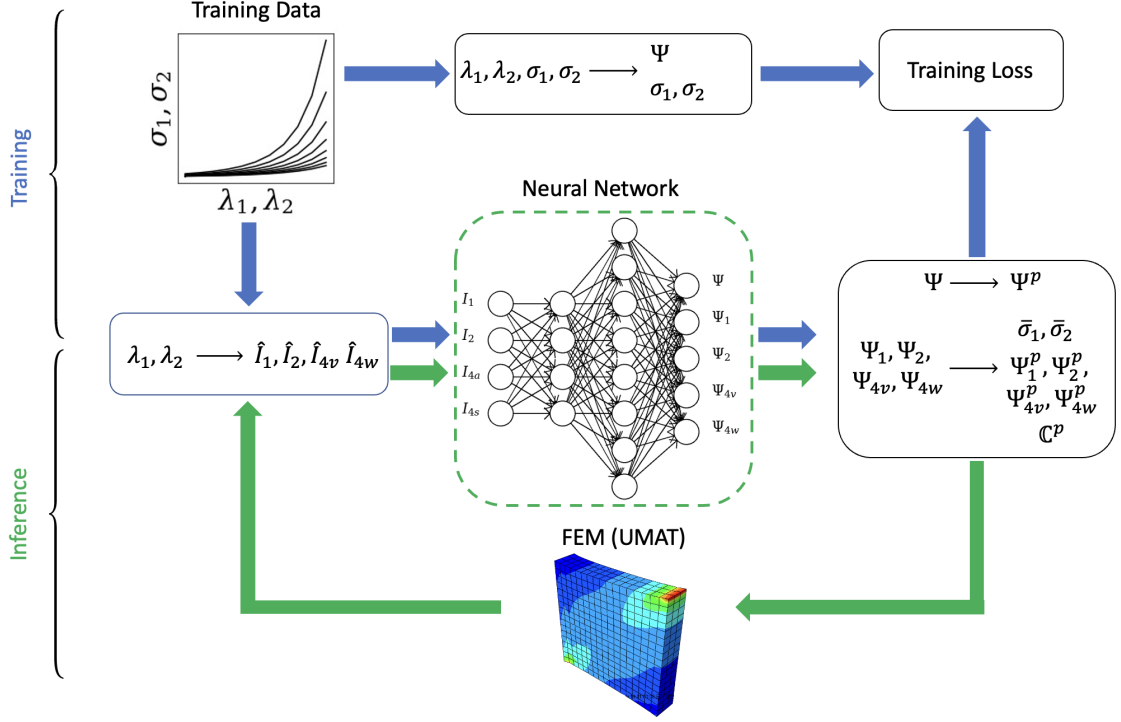


Figure 1: Diagram depicting the training and inference processes of the neural network material model.

The isochoric right Cauchy Green deformation, $\hat{\mathbf{C}}$, can be defined in terms of the isochoric part of the deformation gradient,

$$\hat{\mathbf{F}} = J^{-1/3} \mathbf{F}, \quad (3)$$

$$\hat{\mathbf{C}} = \hat{\mathbf{F}}^\top \hat{\mathbf{F}} = J^{-2/3} \mathbf{C}. \quad (4)$$

The second Piola-Kirchhoff stress tensor, \mathbf{S} , follows from the Doyle-Erickson formula by differentiating the strain energy Ψ with respect to \mathbf{C} and following the procedure outlined by Coleman and Noll [31, 32], arriving at

$$\mathbf{S} = 2 \frac{\partial \Psi}{\partial \mathbf{C}} = \mathbf{S}_{\text{iso}} + \mathbf{S}_{\text{vol}}, \quad (5)$$

$$\mathbf{S}_{\text{iso}} = \hat{\mathbf{S}} : \frac{\partial \hat{\mathbf{C}}}{\partial \mathbf{C}} = J^{-2/3} \hat{\mathbf{S}} : \mathbb{P}_1, \quad \mathbf{S}_{\text{vol}} = 2p \frac{\partial J}{\partial \mathbf{C}} = Jp \mathbf{C}^{-1}. \quad (6)$$

The following definition of the pressure has been introduced $p = d\Psi_{\text{vol}}/dJ$. Additionally, the fictitious second Piola-Kirchhoff stress tensor, $\hat{\mathbf{S}}$, is the result from differentiating the isochoric part of the strain energy with respect to the isochoric invariants, i.e. $\hat{\Psi}_1 = \partial \Psi_{\text{iso}} / \partial \hat{I}_1$, $\hat{\Psi}_2 = \partial \Psi_{\text{iso}} / \partial \hat{I}_2$, $\hat{\Psi}_{4v} = \partial \Psi_{\text{iso}} / \partial \hat{I}_{4v}$ and $\hat{\Psi}_{4w} = \partial \Psi_{\text{iso}} / \partial \hat{I}_{4w}$. The full expansion of the fictitious stress tensor is

$$\hat{\mathbf{S}} = 2 \frac{\partial \Psi_{\text{iso}}}{\partial \hat{\mathbf{C}}} = 2[\hat{\Psi}_1 \mathbf{I} + \hat{\Psi}_2 (\hat{I}_1 \mathbf{I} - \hat{\mathbf{C}}) + \hat{\Psi}_{4v} \mathbf{V}_0 + \hat{\Psi}_{4w} \mathbf{W}_0]. \quad (7)$$

The term *fictitious* originates from the fact that derivatives with respect to the full right Cauchy Green deformation tensor needs the projection with the fourth order tensor

$$\mathbb{P}_1 = \mathbb{I} - \frac{1}{3} \mathbf{C}^{-1} \otimes \mathbf{C},$$

which relates derivatives with respect to the isochoric part of the deformation to derivatives with respect to the total deformation. If the material under consideration is exactly incompressible, i.e. $J = 1$, then $\hat{\mathbf{C}} = \mathbf{C}$, (5) reduces to

$$\mathbf{S} = 2[\hat{\Psi}_1 \mathbf{I} + \hat{\Psi}_2 (\hat{I}_1 \mathbf{I} - \mathbf{C}) + \hat{\Psi}_{4a} \mathbf{V}_0 + \hat{\Psi}_{4s} \mathbf{W}_0] + p \mathbf{C}^{-1}, \quad (8)$$

and the pressure becomes an unknown Lagrange multiplier field.

The UMAT subroutine requires the computation of the Cauchy stress tensor, σ . Thus, for completeness, we state the standard push forward operation for the stress

$$\sigma = \frac{1}{J} \mathbf{F} \mathbf{S} \mathbf{F}^T. \quad (9)$$

The finite element subroutine also requires the computation of the elasticity tensor, $\mathbb{C}_{\text{abaqus}}$ [33]. For ease of derivation, the material version of the elasticity tensor, \mathbb{C} , is introduced first,

$$\mathbb{C} = 2 \frac{\partial \mathbf{S}}{\partial \mathbf{C}} = \mathbb{C}_{\text{iso}} + \mathbb{C}_{\text{vol}}, \quad (10)$$

$$\mathbb{C}_{\text{iso}} = 2 \frac{\partial \mathbf{S}_{\text{iso}}}{\partial \mathbf{C}}, \quad \mathbb{C}_{\text{vol}} = 2 \frac{\partial \mathbf{S}_{\text{vol}}}{\partial \mathbf{C}}. \quad (11)$$

The expressions for the volumetric and isochoric parts of the elasticity tensor, \mathbb{C}_{vol} and \mathbb{C}_{iso} , can be further expanded,

$$\mathbb{C}_{\text{vol}} = J \tilde{p} \mathbf{C}^{-1} \otimes \mathbf{C}^{-1} - 2Jp \mathbf{C}^{-1} \odot \mathbf{C}^{-1}, \quad (12)$$

$$\mathbb{C}_{\text{iso}} = -\frac{2}{3} \mathbf{S}_{\text{iso}} \otimes \mathbf{C}^{-1} + J^{-4/3} \mathbb{P}_1 : \hat{\mathbf{C}} : \mathbb{P}_1^T - \frac{2}{3} \mathbf{C}^{-1} \otimes \mathbf{S}_{\text{iso}} + \frac{2}{3} J^{-2/3} \text{tr}(\hat{\mathbf{S}}) \mathbb{P}_2, \quad (13)$$

where the modified pressure term, $\tilde{p} = p + J dp/dJ$, has been introduced, as well as the special product noted by \odot defined as $(\bullet \odot \circ)_{ijkl} = [(\bullet)_{ik}(\circ)_{jl} + (\bullet)_{il}(\circ)_{jk}]/2$, and an additional fourth order projection tensor \mathbb{P}_2 ,

$$\mathbb{P}_2 = \mathbf{C}^{-1} \odot \mathbf{C}^{-1} - \frac{1}{3} \mathbf{C}^{-1} \otimes \mathbf{C}^{-1}.$$

Finally, the fictitious elasticity tensor, $\hat{\mathbf{C}}$ is obtained from differentiating the fictitious stress tensor with respect to the isochoric part of the deformation tensor,

$$\hat{\mathbf{C}} = 2 \frac{\partial \hat{\mathbf{S}}}{\partial \hat{\mathbf{C}}}.$$

The full expansion of $\hat{\mathbf{C}}$ is available in the Appendix. The only remark needed in the main text is that the tensor $\hat{\mathbf{C}}$ requires the second derivatives of the strain energy function with respect to the isochoric invariants: $\hat{\Psi}_{11} = \partial^2 \hat{\Psi} / \partial^2 \hat{I}_1$, $\hat{\Psi}_{12} = \partial^2 \hat{\Psi} / \partial \hat{I}_2 \partial \hat{I}_1$, $\hat{\Psi}_{14v} = \partial^2 \hat{\Psi} / \partial \hat{I}_{4v} \partial \hat{I}_1$, etc. This point will become important in the design in the neural network later on.

As stated above, the elasticity tensor needed in the UMAT subroutine is associated with the deformed configuration. The push-forward operation for the elasticity tensor yields

$$\mathbb{c} = \frac{1}{J} (\mathbf{F} \otimes \mathbf{F}) : \mathbb{C} : (\mathbf{F} \otimes \mathbf{F})^T \quad (14)$$

where we have introduced the modified dyadic product defined as $(\bullet \otimes \circ)_{ijkl} = (\bullet)_{ik}(\circ)_{jl}$. The tensor \mathbb{c} is related to the Truesdell stress rate; however, Abaqus increments employ the Jaumann stress rate. Therefore, the consistent tangent for Abaqus is not Eq. (14) but rather

$$\mathbb{C}_{\text{abaqus}} = \mathbb{c} + \frac{1}{2} (\sigma \otimes \mathbf{I} + \sigma \otimes \mathbf{I} + \mathbf{I} \otimes \sigma + \mathbf{I} \otimes \sigma). \quad (15)$$

with the modified dyadic product $(\bullet \otimes \circ)_{ijkl} = (\bullet)_{il}(\circ)_{jk}$.

Lastly, we remark that during the training of the neural network, incompressibility can be satisfied exactly. In this case p can be determined from boundary conditions. However, in the UMAT we use the following function for p ,

$$p = K(J - 1), \quad (16)$$

with K the bulk modulus. In this study we set $K = 10$ kPa.

Neural network structure and training

We use a fully connected neural network to learn the mechanical behavior of skin. The neural network takes four inputs, the isochoric strain invariants in (2), and produces five outputs, the strain energy, $\Psi_{\text{iso}}^{\text{p}}$, and its derivatives with respect to the invariants, $\hat{\Psi}_1^{\text{p}}$, $\hat{\Psi}_2^{\text{p}}$, $\hat{\Psi}_{4v}^{\text{p}}$ and $\hat{\Psi}_{4w}^{\text{p}}$. Note that the notation $(\bullet)^{\text{p}}$ is used to denote the predicted values of the neural network. The network architecture is summarized in Table 1.

Training data for the neural network is in the form of stretch, stress and strain energy values. Therefore, the first component of the loss function is simply the comparison of the predicted strain energy, $\Psi_{\text{iso}}^{\text{p}}$, against the observed Ψ_{iso} . Since the derivatives are direct outputs, a regularization term is added to the loss function to enforce that the predicted derivatives coincide with the derivatives of $\Psi_{\text{iso}}^{\text{p}}$. Using back-propagation, the derivatives of $\Psi_{\text{iso}}^{\text{p}}$ with respect to the inputs are denoted as $\hat{\Psi}_i^{\text{diff}}$, with $i = 1, 2, 4v, 4w$. Thus, the first component of the loss function is

$$\mathcal{L}_1 = \frac{1}{N} \sum_{n=1}^N \left[\left((\Psi_{\text{iso}}^{\text{p}})^{(n)} - (\Psi_{\text{iso}})^{(n)} \right)^2 + \sum_{i=1,2,4v,4w} \left((\hat{\Psi}_i^{\text{diff}})^{(n)} - (\hat{\Psi}_i^{\text{p}})^{(n)} \right)^2 \right], \quad (17)$$

where $(\bullet)^{(n)}$ denotes the n^{th} training point, out of a total of N training points.

The second component of the loss results from comparing the stress computed with the neural network outputs against the observed stress σ . The stress, defined in Eqs. (8) and (9), is computed based on the neural network outputs $\Psi_{\text{iso}}^{\text{p}}$, to produce σ^{p} . The loss can then be simply stated as

$$\mathcal{L}_2 = \frac{1}{N} \sum_{n=1}^N \left\| (\sigma^{\text{p}})^{(n)} - (\sigma)^{(n)} \right\|_F \quad (18)$$

where $\|\bullet\|_F$ denotes the Frobenius norm.

To ensure optimal convergence of the finite element framework, the strain energy must be a (poly)convex function of the deformation [34]. In the case of a strain energy function defined in terms of the deformation invariants, the strain energy needs to be a convex function of its inputs. For a function to be convex, it's Hessian matrix, \mathbf{H} , must be positive semi-definite. The Hessian matrix is

$$\mathbf{H} = \begin{pmatrix} \Psi_{11}^{\text{diff}} & \Psi_{12}^{\text{diff}} & \Psi_{14v}^{\text{diff}} & \Psi_{14w}^{\text{diff}} \\ \Psi_{21}^{\text{diff}} & \Psi_{22}^{\text{diff}} & \Psi_{24v}^{\text{diff}} & \Psi_{24w}^{\text{diff}} \\ \Psi_{4v1}^{\text{diff}} & \Psi_{4v2}^{\text{diff}} & \Psi_{4v4v}^{\text{diff}} & \Psi_{4v4w}^{\text{diff}} \\ \Psi_{4w1}^{\text{diff}} & \Psi_{4w2}^{\text{diff}} & \Psi_{4w4v}^{\text{diff}} & \Psi_{4w4w}^{\text{diff}} \end{pmatrix}. \quad (19)$$

The notation Ψ_{ij}^{diff} indicates the second derivative of the strain energy computed with the neural network by differentiating the outputs $\hat{\Psi}_i^{\text{p}}$ with respect to the j^{th} input using back-propagation. We impose positive-definiteness of the Hessian matrix using the principal minor test [35]. For a matrix to be positive definite, it has to be symmetric and all its leading principal minors, Δ_k , must be positive. This condition is imposed in terms of two additional loss terms,

$$\mathcal{L}_3 = \frac{1}{N} \sum_{n=1}^N \left\| (\mathbf{H})^{(n)} - (\mathbf{H}^{\text{T}})^{(n)} \right\|_F, \quad (20)$$

$$\mathcal{L}_4 = \frac{1}{N} \sum_{n=1}^N \sum_{k=1}^4 \max \left(-\Delta_k^{(n)}, 0 \right). \quad (21)$$

The total loss is a weighted sum of the terms discussed so far,

$$\mathcal{L} = \mathcal{L}_1 + a_0 \mathcal{L}_2 + a_1 \mathcal{L}_3 + a_2 \mathcal{L}_4. \quad (22)$$

If training data from sources with different fidelities are used, the total loss of the multi-fidelity (mf) dataset is given as a weighted sum of the losses of the high fidelity (hf) and low fidelity (lf) datasets,

$$\mathcal{L}_{\text{mf}} = \mathcal{L}_{\text{lf}} + a_3 \mathcal{L}_{\text{hf}}. \quad (23)$$

The training of the neural network was performed using the Adam optimization algorithm [36]. The initial learning rate was set to 0.0001. The exponential decay rates for first and second moment estimates, β_1 and β_2 , were set to 0.9 and 0.99 respectively. The neural network was trained in 100000 epochs without the use of batching. The training was implemented using Keras [37] with a Tensorflow [38] back-end on a workstation with the following specifications: Intel Xeon E5-1630 3.70 GHz CPU, 16 GB DDR4/2400 MHz random access memory, and Nvidia GeForce GTX 1080 GPU. The values of the weights used in this paper are $a_0 = 50$, $a_1 = 10$, $a_2 = 100$ and $a_3 = 50$.

Table 1: Neural network architecture

Layer	Number of nodes	Activation function
Input	4	None
Hidden layer 1	4	Sigmoid
Hidden layer 2	8	Sigmoid
Output	5	Linear

Synthetic data generation

In the majority of biomedical applications it is difficult to obtain sufficient high fidelity data to train a neural network. The number of measurements might be limited, or the data points may be constrained to a narrow region of the input space. It is then beneficial to make use of low fidelity data if available.

In this study, high fidelity data is in the form of biaxial stress-stretch measurements. However, only two or three curves within the four-dimensional input space defined by the invariants is explored. Therefore, we generate synthetic data using the Gasser-Ogden-Holzapfel (GOH) [8] material model. The GOH model proposes an isochoric strain energy of the form

$$\hat{\Psi}(\mathbf{C}, \mathbf{a}_0) = \hat{\Psi}_{\text{iso}}(\mathbf{C}) + \hat{\Psi}_{\text{aniso}}(\mathbf{C}, \mathbf{a}_0) \quad (24)$$

where $\mathbf{a}_0 = (\sin \theta, \cos \theta, 0)$ is vector denoting the mean fiber direction and parameterized by the angle θ . The functional forms for the GOH strain energy are

$$\hat{\Psi}_{\text{iso}}(\mathbf{C}) = \mu(\hat{I}_1 - 3), \quad (25)$$

$$\hat{\Psi}_{\text{aniso}}(\mathbf{C}, \mathbf{a}_0) = \frac{k_1}{4k_2} \left[\exp(k_2 E^2) - 1 \right], \quad (26)$$

with the generalized fiber strain

$$E = [\kappa \hat{I}_1 + (1 - 3\kappa) \hat{I}_{4a} - 1]. \quad (27)$$

The volumetric term that is the same as the one we used to penalize volume changes in our formulation [6, 8, 16],

$$\Psi_{\text{vol}} = \frac{K}{2} (J - 1)^2. \quad (28)$$

The derivation of the stress tensor for the GOH strain energy is not repeated here, the interested reader is referred to [18, 8].

Synthetic data with the GOH model is generated by fitting the free parameters to the experimental data using the BFGS optimization algorithm in SciPy [39].

Finite element method implementation

We implemented a general neural network material model in a user material subroutine (UMAT) in the nonlinear finite element package Abaqus. The subroutine was written with minimal assumptions to allow for maximal flexibility. The neural network structure, weights and biases, activation functions, etc. are all imported into the subroutine through the input file.

The subroutine performs the following tasks:

1. Read in the architecture, weights and biases, activation function types, etc., as a set of material properties.
2. Pre-process the deformation gradient to obtain the isochoric invariants in Eqs. (2 - 4).

3. Perform the forward propagation of the neural network to obtain the predicted strain energy Ψ^p and its first derivatives Ψ_i^p .
4. Calculate stress using Eqs. (5 - 7, 9).
5. Calculate the second derivatives Ψ_{ij}^{diff} with back-propagation.
6. Compute the consistent tangent $\mathbb{C}_{\text{abaqus}}$ using Eq. (15).

For the forward propagation, let $\mathbf{y}_{i-1} \in \mathbb{R}^m$ be the output of layer $i - 1$ of the neural network with m nodes. Then the output of layer i is given as

$$\mathbf{y}_i = g_i(\mathbf{W}_i^T \mathbf{y}_{i-1} + \mathbf{B}_i), \quad \mathbf{y}_i \in \mathbb{R}^n \quad (29)$$

where $g_i : \mathbb{R} \rightarrow \mathbb{R}$ is the element-wise activation function, $\mathbf{W}_i \in \mathbb{R}^{n \times m}$ is the weights matrix, and $\mathbf{B}_i \in \mathbb{R}^n$ is the biases vector of the i^{th} layer of the network. We use the sigmoid activation function in the hidden layers,

$$g(y) = \frac{1}{1 + e^{-y}}, \quad g'(y) \equiv \frac{dg(y)}{dy} = g(y)(1 - g(y)). \quad (30)$$

For the derivatives, let $\mathbf{J}_{i-1} \in m \times m_0$ be the matrix containing derivatives of the nodes of layer $i - 1$ with respect to the inputs of the neural network. Then

$$\mathbf{J}_i = \text{diag}(g'(\mathbf{y}_i)) \mathbf{W}_i^T \mathbf{J}_{i-1}, \quad \mathbf{J}_i \in \mathbb{R}^{n \times m_0} \quad (31)$$

where m_0 is the number of inputs to the neural network and $\text{diag}(\bullet)$ denotes a diagonal matrix.

Biaxial stress-stretch experiments on porcine and murine skin

We use experimental data from biaxial stress-stretch experiments performed on murine [15] and porcine skin for the training and validation of neural networks. The data is collected in up to 5 different experimental protocols which are defined in Table 2.

Table 2: Experimental loading protocols.

Loading	λ_x	λ_y	σ_z
Off-x	$\sqrt{\lambda}$	λ	0
Off-y	λ	$\sqrt{\lambda}$	0
Equibiaxial	λ	λ	0
Strip-x	λ	1	0
Strip-y	1	λ	0

Results

Performance of the neural network against synthetic data

To test the neural network material model, we first train the network using synthetic data only. We generate eleven curves in the λ_x, λ_y space by first holding $\lambda_x = 1$ while λ_y is increased gradually to $\lambda_y^{(i)}$, with $i = 1, \dots, 11$. The values for the y -stretch are $\lambda_y^{(i)} \in [1, 1.025, 1.05, \dots, 1.25]$. After reaching the corresponding $\lambda_y^{(i)}$ value, λ_y is held constant while λ_x is gradually increased (Figure 2A). These loading curves are representative of the type of test that can be performed experimentally. On the other hand, the neural network takes as inputs the isochoric strain invariants. The invariant space is 4-dimensional, but we plot a 3-dimensional projection in Figure 2B. We use the GOH material model to generate synthetic stress data points and train the neural network. Various components of the loss are plotted in Figure 2C. The predictions of the trained network are plotted against the training data in Figure 2D-F. These results indicate that the neural network is able to recreate expert constitutive models within the training region.

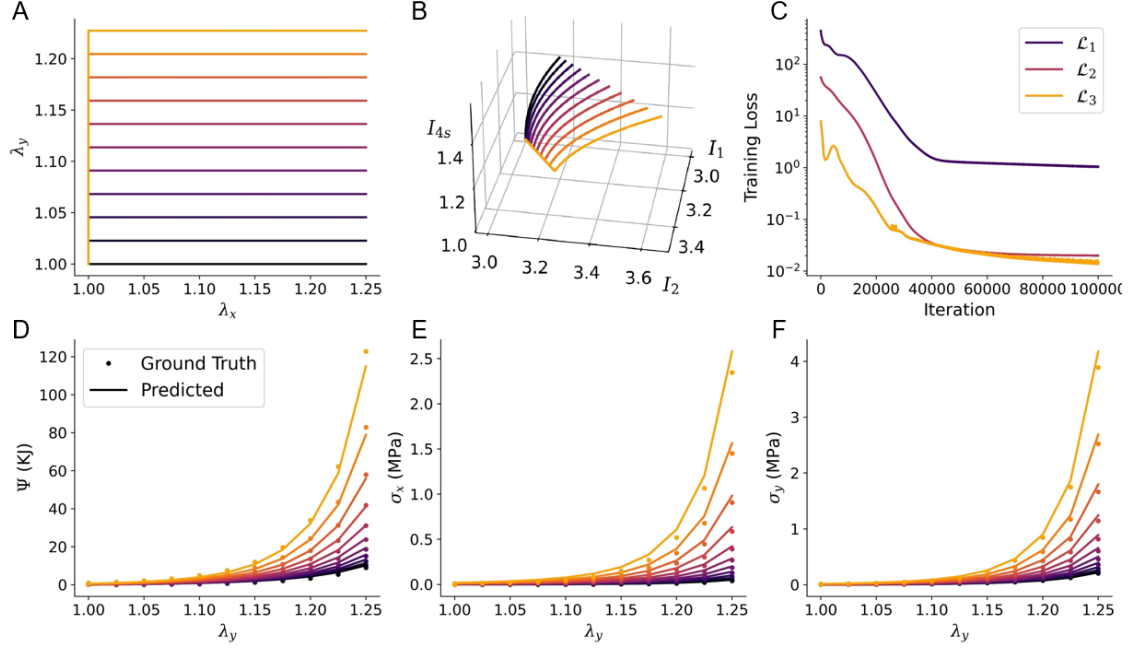


Figure 2: Synthetic data generated to train the neural network and the performance of the neural network compared to the training data. (A) Training data was generated by creating *curves* in the λ_x, λ_y stretch space. (B) The corresponding training data in the invariant space, which is the actual input space for the neural network. The invariant space is four-dimensional but only a three-dimensional projection is shown. (C) Various components of the loss during training. (D) Predicted and ground truth strain energy values throughout the input space. Predicted and ground truth planar stress values in the x (E) and y (F) directions.

We also test if the neural network performs well outside the training region. We generate three validation datasets. The first validation dataset is built by randomly sampling $\lambda_x \in [1, 1.25]$, $\lambda_y \in [1, 1.25]$ to construct a diagonal deformation gradient of biaxial deformations not seen during training. Then, to test predictions under shear, which are not directly part of the training data, we construct a data set of deformation gradients of the form

$$\mathbf{F} = \begin{pmatrix} \lambda_x & \gamma_{xy} & 0 \\ \gamma_{xy} & \lambda_y & 0 \\ 0 & 0 & \frac{1}{\lambda_x \lambda_y - \gamma_{xy}^2} \end{pmatrix}. \quad (32)$$

The validation dataset is generated from randomly sampling $\lambda_x \in [1, 1.25]$, $\lambda_y \in [1, 1.25]$, $\gamma_{xy} \in [0, 0.5]$. Lastly, we are interested in the potential of the neural network to extrapolate. An additional validation set is constructed by sampling outside the training region: $\lambda_x \in [1, 1.25]$ but $\lambda_y \in [1.25, 1.35]$; $\lambda_y \in [1, 1.25]$ but $\lambda_x \in [1.25, 1.35]$; and $\lambda_x \in [1.25, 1.35]$ and $\lambda_y \in [1.25, 1.35]$. The errors for the validation datasets are shown in Figure 3. It can be seen that the neural network performs well within the training region but worse toward the boundary of the training region. Note that the stress values are very small in magnitude, for small deformations (see Figure 2D-F); thus, even though the absolute error is small for $\lambda_x, \lambda_y < 1.05$, the relative error can achieve close to 100%. This is not a concern since clearly the stresses at these small deformations are two orders of magnitude smaller compared to stresses at larger deformation. For the third validation set, for example, it can be seen that sampling from $\lambda_x \in [1.25, 1.35]$ and $\lambda_y \in [1.25, 1.35]$ leads to errors on the order of 300%. In this region of the input space the stresses are significant. Thus, the neural network has difficulty extrapolating outside of the training region.

Performance against experimental data: multi-fidelity data and convexity constraints

Next, we start training the neural network using experimental data. We want to test the effect of using the experimental data alone (sparse high fidelity data), or combining these data with the low fidelity approximation of the

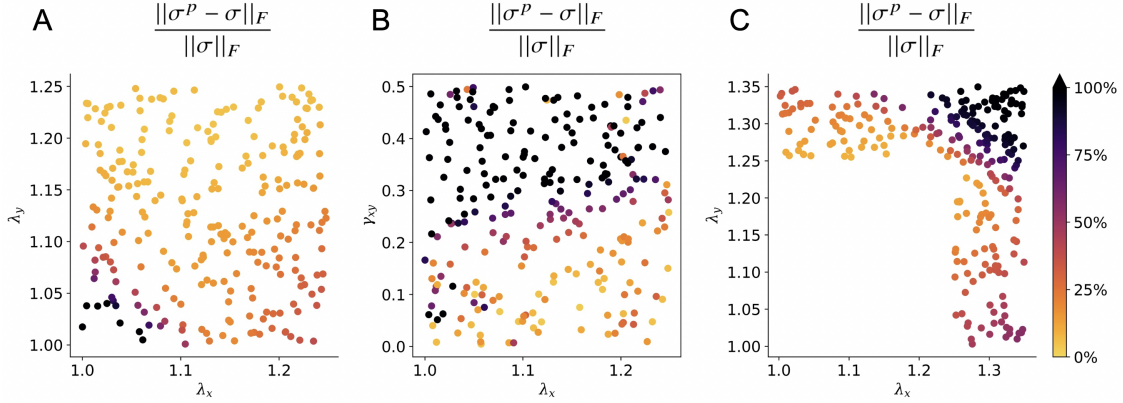


Figure 3: Validation of the neural network trained on synthetic data. Performance of the neural network on points randomly sampled from: (A) $\lambda_x \in [1, 1.25]$ and $\lambda_y \in [1, 1.25]$, (B) $\lambda_x \in [1, 1.25]$, $\lambda_y \in [1, 1.25]$, $\gamma_{xy} \in [0, 0.5]$ and (C) $\lambda_x \in [1, 1.25]$ but $\lambda_y \in [1.25, 1.35]$; $\lambda_y \in [1, 1.25]$ but $\lambda_x \in [1.25, 1.35]$; and $\lambda_x \in [1.25, 1.35]$ but $\lambda_y \in [1.25, 1.35]$

GOH model fit (multi-fidelity data). Concurrently, we want to test if the convexity constraint is required to regularize the fits of the neural network. In Figs. 4 and 5, we show the results for murine skin data and porcine skin data respectively together with error in the predictions which is defined as the average Frobenius norm of the error in stress, $\text{mean}(\|\sigma^P - \sigma\|_F)$.

The first row of Figure 4 corresponds to a neural network that is trained using sparse high-fidelity data where the convexity constraints are not imposed. Figure 4A shows the neural network ability to fit the off-x and off-y data, achieving average errors of 2.415 kPa and 2.246 kPa, respectively. A biaxial test, not used in training, is used to test the predictive capability of the network (Figure 4B). The average error in the validation is 6.137 kPa. Because no convexity constraint is used we can see that it is not satisfied (Figure 4C). Keeping only the high-fidelity data but imposing convexity drastically changes the performance. The training loss is poorer (Figure 4E and F), but the validation error improves (Figure 4G). Moreover, the function is convex over the input space (Figure 4H).

The third and fourth rows of Figure 4 show the results of neural networks trained with multi-fidelity data. It is notable that even though the network of the third row is trained without any convexity constraints, the fact that it is trained on the GOH synthetic data (which is an inherently convex model) helps it achieve better convexity (Figure 4H). Nevertheless, it can be seen that for larger deformations, convexity is lost, as it is evident in the biaxial validation test (Figure 4L). The average error in the validation set for the multi-fidelity case without convexity constraint is 5.415 kPa (Figure 4K), which is comparable to the sparse high fidelity case without convexity requirements (Figure 4C). The results for the neural network trained with multi-fidelity data and with the convexity constraint are shown in last row of Figure 4. The training errors are 1.896 kPa and 4.328 kPa (Figure 4M and N), and the validation error is 5.678 kPa (Figure 4O). Since the convexity is imposed, the loss in the convexity is close to zero over the entire input space (Figure 4P). In summary, the convexity constraint is needed, but it can slightly increase the error of the neural network over the training set. This can be reflective of the uncertainties in the experimental data collection, or limitations of the hyperelastic framework. Training with multi-fidelity (given the convexity constraint) yields the best performance: the error in the training data is comparable to the high fidelity case, but the validation error improves.

In Figure 5 we further study the effects of augmenting the training data and how the neural network differs from relying solely on the expert model. For this we focus on porcine skin. We train two neural networks, one of them is trained with the experimental data only (Figure 5A-F), whereas the other is trained on the augmented data (Figure 5G-L). In Figure 5C and D it can be seen that trained only on experimental data, the neural network achieves a low average error of 5.410 kPa, compared to the GOH fit which is 53.164 kPa. Thus, the neural network outperforms the GOH model. This is not surprising since the only task of the neural network is to interpolate the experimental data and satisfy convexity. The contour plots in Figure 5B,D and F shows the difference between the neural network as compared with GOH material model throughout the input space. It can be seen that the two models are very different. Without any data on the rest of the input space, it is unclear if the predictions from the neural network are at all accurate. On the other hand, the GOH model has been developed and trained against thousands of tissue biaxial data.

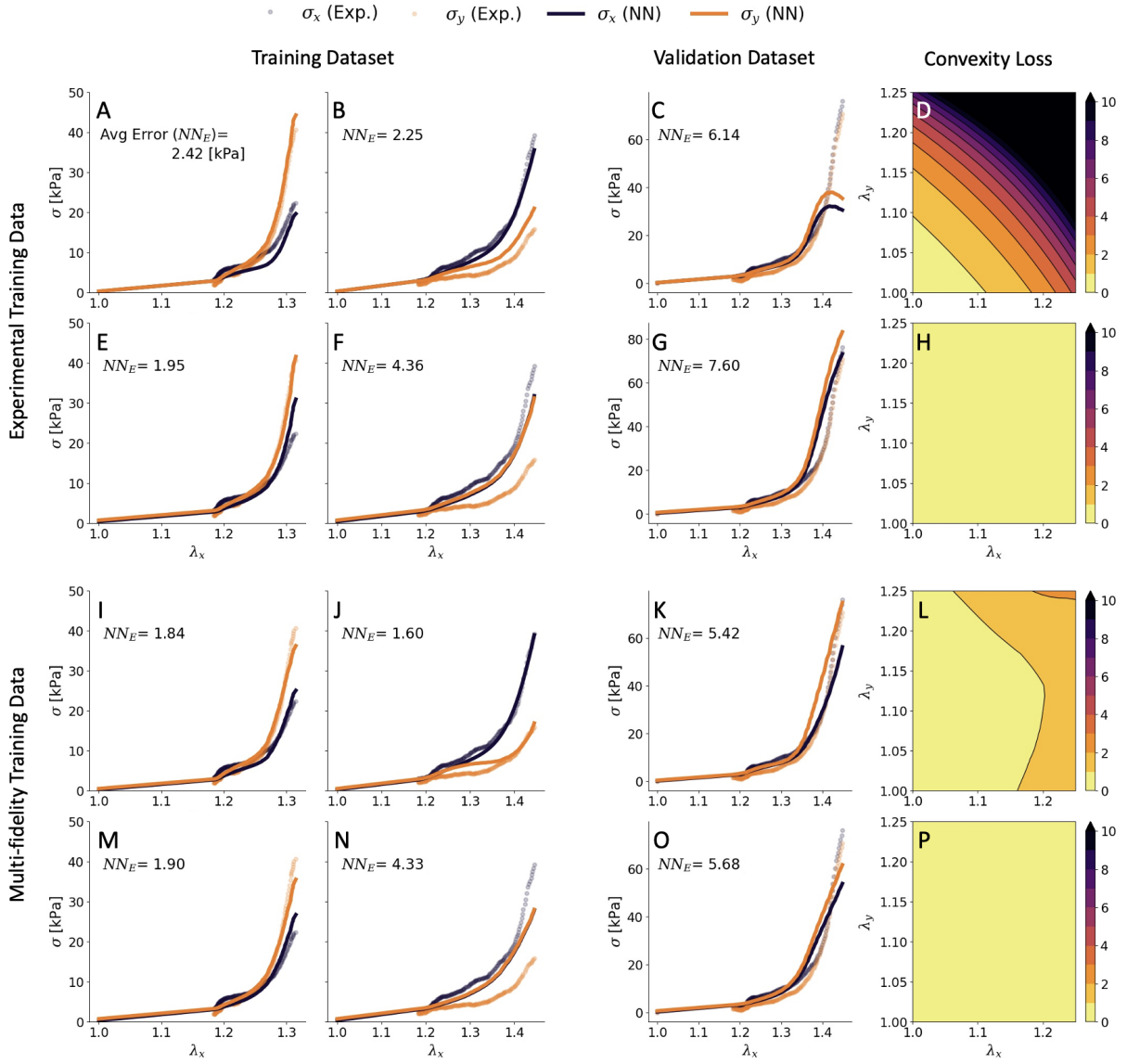


Figure 4: Performance of the neural network on the murine skin data. The first two and the third panels of each row show the predicted stress vs actual stress on the training (Off-x and Off-y) and validation (Equibiaxial) sets, respectively, while the fourth panel shows the convexity loss throughout the input space. Each row corresponds to a separate neural network trained differently. Predictions of neural networks trained with (top to bottom): single-fidelity data and no convexity constraints, single-fidelity data and convexity constraints, multi-fidelity data and no convexity constraints, multi-fidelity data and convexity constraints.

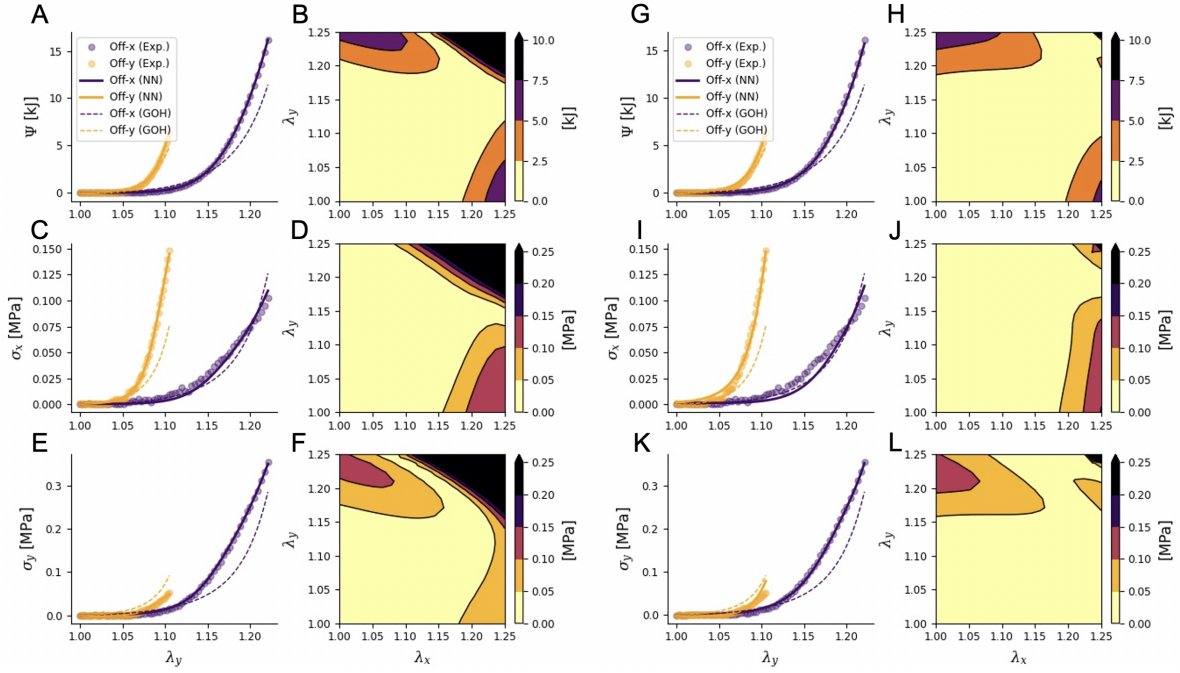


Figure 5: Performance of neural networks trained on single-fidelity (A-F) and multi-fidelity (G-L) training data. The scatter plots compare predicted strain energy and stress values to experimental data as well as GOH model outputs. The contour plots show the difference between the corresponding outputs of the GOH model and the neural network.

It is reasonable to expect that the GOH model, even though it cannot fit any particular dataset as well as the neural network, it can be trusted to guide the neural network away from the training region. We show that training the neural network on the augmented data, the loss is on average 11.705 kPa against the experimental data (Figure 5C and E), which is still lower with respect to using the GOH model alone. However, as looking at the contours in Figure 5H, J and L we see that the neural network now follows the GOH model closely on the rest of the input space.

Therefore, the neural network trained with augmented data is at the very least the best version of the GOH model. It performs better than the GOH material model around the high fidelity data points while approximating the GOH model elsewhere.

The last test of the neural network model is also done with porcine experimental data. In this case we have five different biaxial experiments (see Table 2). We are interested in determining which biaxial tests are the most informative for the neural network material model. Thus, we train the neural network with different combination of experimental data and validate against the rest of the data (Figure 6). We do the same training and testing with the GOH model. In Figure 6A and B we train the material models with only two datasets, and test against the other three. In the training set, as expected from our previous result, the neural network outperforms the GOH model. Here we see that the neural network continues to outperform the GOH model even in the validation dataset. Figure 6C, D show the result of training the models with three of the five biaxial curves, and validating against the remaining two. Again, the neural network obviously outperforms the GOH model in the training set. However, surprisingly, when the neural network is trained with off-biaxial data and tested against strip-biaxial data, it is now outperformed by the GOH model during the validation step (Figure 6C). In Figure 6E we train against four tests, and validate against the equibiaxial test. The validation error is comparable between the neural network and the GOH model. Yet, the superior performance of the neural network on the training data confirms that data-driven models are a preferable alternative to expert constructed constitutive models.

Finite element method implementation

We show a number of basic finite element simulations to test the capabilities of the neural network material model as a UMAT subroutine for Abaqus. The neural network trained on murine skin data (augmented data with convexity

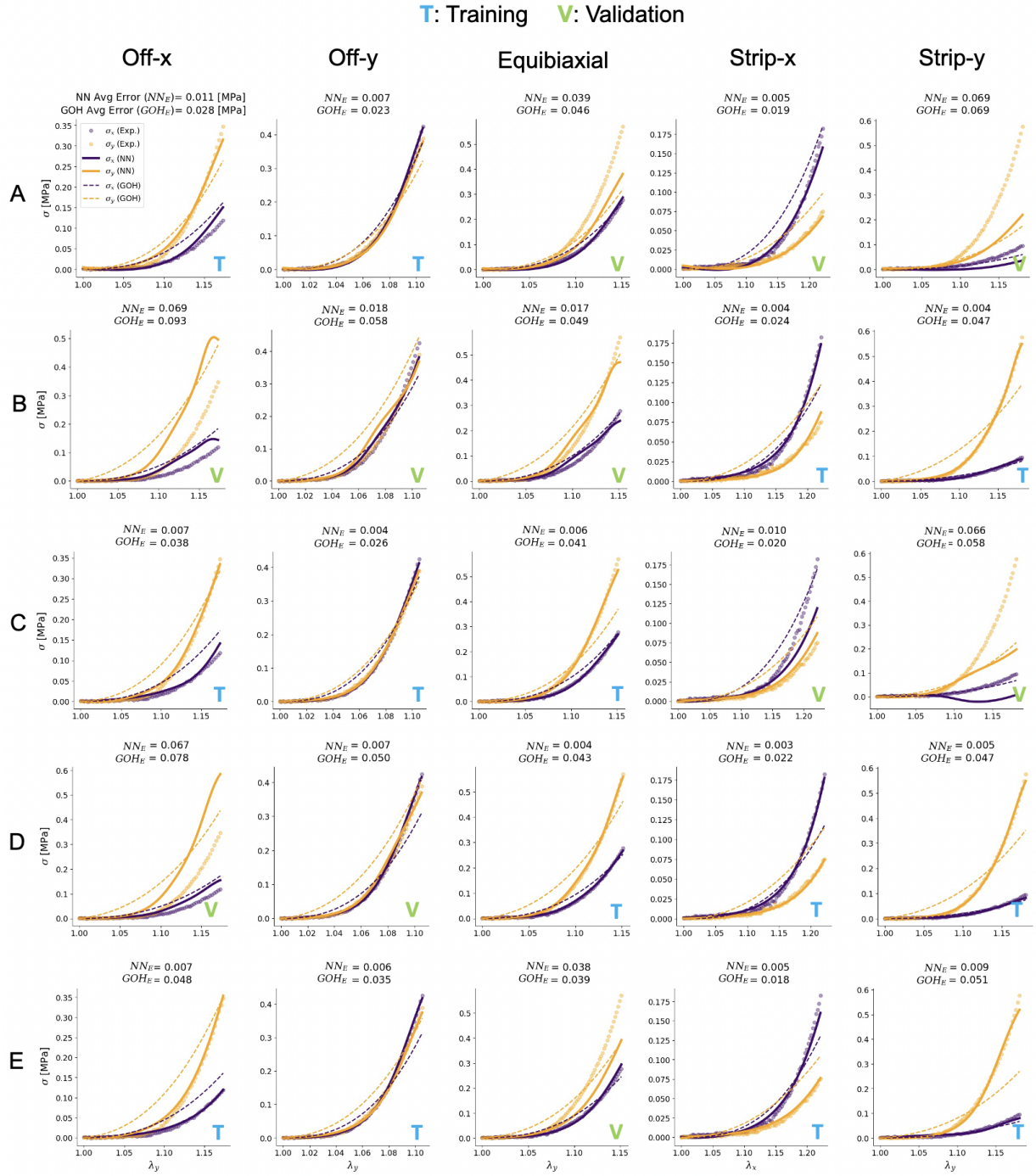


Figure 6: Comparison of predicted stress vs actual stress vs GOH fits for various training/validation splits of porcine experimental data. Predictions of a neural network trained on (A) Off-x and Off-y data, (B) Off-x, Off-y and Equibiaxial data, (C) Off-x, Off-y, Strip-x and Strip-y data, (D) Strip-x and Strip-y data, and (E) Equibiaxial, Strip-x and Strip-y data.

constraint) is defined in the input file. We first consider a rectangular block $5 \times 5 \times 1 \text{ cm}^3$. Boundary conditions, mesh and results for a uniaxial extension simulation with $\lambda_x = 1.2$ are depicted in Figure 7A. The result is a homogeneous stress distribution of $\sigma_x = 1.329 \text{ kPa}$, $\sigma_y = \sigma_z = 0$, consistent with the results in Fig. 4, confirming that the UMAT subroutine is working as intended.

A shearing simulation is shown in Figure 7B. In this analysis the -x surface of the prism is clamped and a displacement boundary condition of $U_x = U_y = 5$ is applied on the right surface. The contours of the resulting stress components, σ_x and σ_y are shown in Figure 7. The Appendix shows a simulation with the GOH fit. As discussed in the previous section, the neural network model with the augmented data is, in a way, the best extension of the GOH model: it retains the expert model features but does not suffer from the constraints of an explicit functional form.

The last simulation in Figure 7 is a torsional loading scenario. In this simulation the -x surface of the rectangular prism is clamped and a rotation boundary condition of $UR_x = 1 \text{ rad}$ is imposed on the +x surface. The resulting stresses are presented in Figure 7C. This loading scenario is different from the previous two because it involves significant deformations in the out-of-plane direction. The UMAT subroutine executes without any problems. The three simulations in Figure 7 showcase the robustness and versatility of our neural network UMAT.

Next, we perform a simulation that is much more closely related to skin biomechanics. Tissue expansion is a widely used technique in reconstructive surgery in which a balloon-like device is inserted and inflated subcutaneously to stretch and grow skin [16]. The domain is a $10 \times 10 \times 0.3 \text{ cm}^3$ patch of skin modeled with 3200 brick elements. A rectangular expander of dimensions $8 \times 8 \text{ cm}^2$ underneath the skin mesh is modeled with the fluid cavity feature in Abaqus. The expander is inflated to 20, 40 and 60 cm^3 resulting in the principal strain distributions shown in Figure 8. Once again, the simulation converged without issues and the results align with our previous experimental observations of higher deformation at the apex and less toward the periphery of the expander [40]. The simulation in Figure 8 showcases the ability of our neural network model to be used in realistic finite element simulations through our UMAT.

Note that the simulation in Figure 8 evidences the anisotropy of the model. The fiber directions \mathbf{v} and \mathbf{w} are aligned with the Cartesian basis $[1, 0, 0]$ and $[0, 1, 0]$. The tissue is stiffer in the \mathbf{v} direction, which is why there is a band of higher stress along that direction in Figure 8. To further showcase the anisotropy in the deformation we also plot the corresponding strains (Figure 9).

Discussion

In this study we propose a neural network material model to replace conventional constitutive equations for non-linear materials, in particular soft collagenous tissues. The neural network takes isochoric strain invariants as inputs and produces the isochoric strain energy and its derivatives with respect to the invariants as outputs. With this design, objectivity is satisfied *a priori*. Other efforts in data-driven modeling of materials and structures rely on training directly on the stress data, which requires additional steps to ensure objectivity [41, 42, 43]. Efforts using invariants or principal stretches have also been shown by others [26, 44, 45], and by us as well but for isotropic materials [25]. The key ideas introduced in this paper are the consideration of anisotropy, training with multi-fidelity data, convexity constraints, and design of the neural network architecture to compute not only the stress but the consistent tangent needed in finite element simulations.

Training data for the neural network can consist of both *expensive* (or hard to get) high fidelity data such as results of laboratory experiments, or a combination of high fidelity data supplemented by synthetic data from expert models. A control case shown here is to train the neural network on synthetic data alone. The performance of the neural network with the synthetic data shows that the network can interpolate the expert model. This is not surprising since neural networks are universal approximators [46]. When trained on sparse high fidelity data alone, imposing convexity requirements does not only ensure a physically admissible model, but also improves the performance over the validation set (see Figure 4). This result reflects the smoothness and convexity of the underlying material behavior. However, even with the convexity regularization, there is no guarantee that the neural network predictions will be accurate outside of the training region. In fact, for the control case in which the neural network is trained on synthetic data alone, predictions outside of the training region are not accurate (see Figure 3).

Unfortunately, high fidelity data of soft tissue mechanics is sparse in most applications. In our previous work, typically only three protocols have been performed: off-biaxial x, off-biaxial y and equi-biaxial (see Table 2) [15].

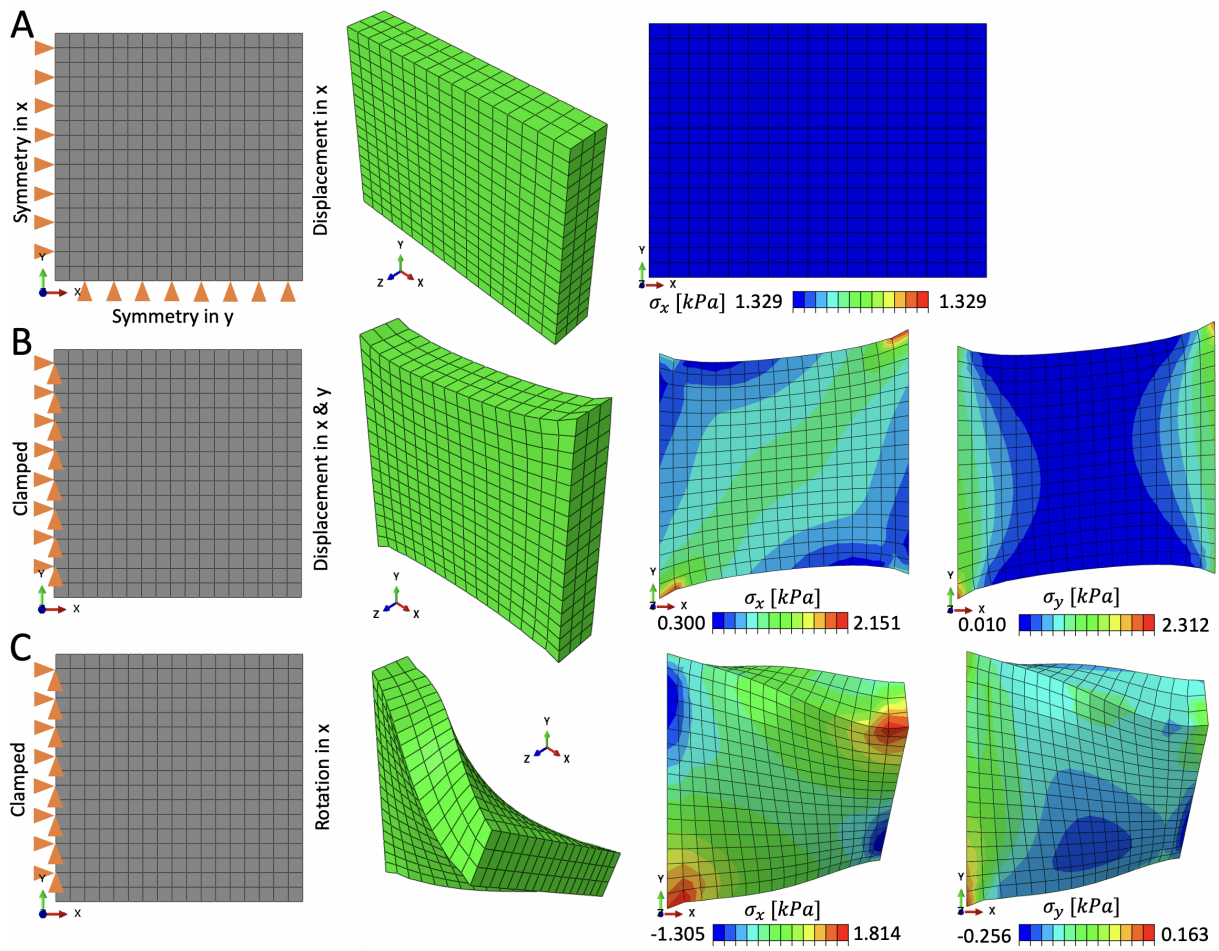


Figure 7: Finite element method simulations using the neural network based material model in UMAT. Boundary conditions, deformed geometry and contours of σ_x under uniaxial loading (A), Boundary conditions, deformed geometry and contours of σ_x and σ_y under shear loading (B), and, Boundary conditions, deformed geometry and contours of σ_x and σ_y under torsional loading (C).

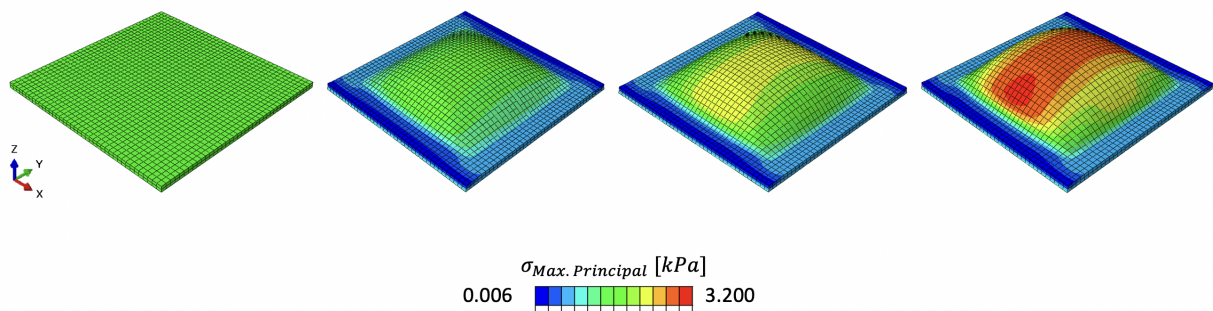


Figure 8: Finite element method simulations of tissue expansion using the neural network based material model in UMAT. From left to right: Undeformed geometry, and, contours of maximum principal stress on deformed geometry after the expander is expanded to 20, 40 and 60 cm³, respectively.

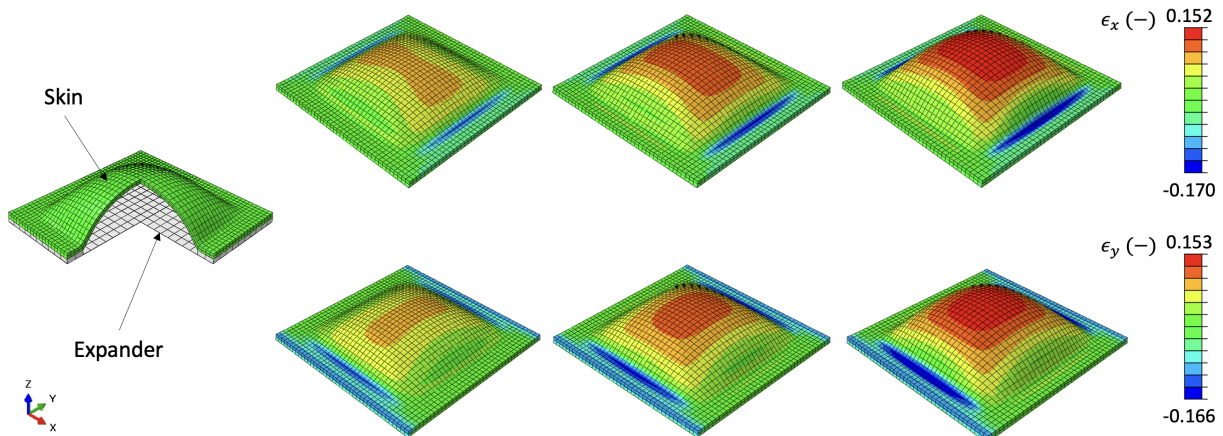


Figure 9: Finite element method simulations of tissue expansion using the neural network based material model in the UMAT. From left to right: model setup, and, contours of strain on deformed geometry after the expander is expanded to 20, 40 and 60 cm³, respectively.

Two other tests, strip-biaxial tests in the x - and y -direction are explored here as well. Liu et al. [26] generated datasets by subjecting tissues to seven biaxial tests. Clearly, more coverage of the input space is always better for data-driven approaches. However, this is a challenge, it requires establishment of multiple repeatable protocols and extensive testing of a individual specimens which can introduce unforeseen uncertainties. Expert models of soft tissue mechanics have evolved over the past few decades and reflect our growing understanding of soft tissues. For example, expert models are often based on microstructure observations [47, 48], are physically meaningful [34, 49], and have been carefully designed based on observations of many data [9]. On the other hand, expert models can also present limitations, such as non-uniqueness of fit, high sensitivity to parameters, and inability to fit the data due to the inherent constraints of the functional form [50]. By combining the high fidelity data with an expert model as a low fidelity approximation we aim at getting the best of both, keep data-centric models that can capture the experimental data really well, while at the same time maintaining relatively good performance in regions with scarce high fidelity data.

A strong motivation behind the development of constitutive models of soft tissues is to be able to build predictive finite element simulations to guide device design or treatment planning [51]. Previous work on data driven modeling has fallen short in this regard [41, 26, 52]. We implemented the neural network model in a UMAT subroutine for Abaqus, a popular finite element package in both academia and industry. The UMAT subroutine code was implemented with maximum flexibility in mind. The definition and all parameters of the neural network are fed into the UMAT through the input file. We showcased finite element simulations with the neural network trained on the murine data. The simulations converged stably without any issues in all loading scenarios considered, from simple deformations to realistic applications such as tissue expansion.

Of course, this work is not without limitations. While it is common to model soft tissues within the hyperelastic framework, other physical phenomena will be included in future work, namely viscoelasticity, interstitial flow, and damage. Additionally, a Bayesian framework is needed to account for the inherent uncertainty in material behavior of biological materials. Nevertheless, we anticipate that the general framework introduced here will open up new avenues in data-driven finite element models that balance high-fidelity experimental data with expert knowledge of soft tissue mechanics.

Conclusions

The work presented in this study shows that neural network material models can reliably replace or augment conventional constitutive material models in tissue mechanics analyses. If enough high fidelity data is available, data-driven models can eliminate the burden of choosing a specific functional form and the inherent limitations that come with this choice. However, in most applications, high fidelity data is scarce. Our work demonstrates that a multi-fidelity approach can leverage expert knowledge in the form of synthetic data, while achieving a better fit to

the experimental observations. A strong motivation to develop accurate material models of soft tissue is to build predictive finite element models. We designed the neural network with this application in mind, and implemented a neural network UMAT subroutine for Abaqus, a widely used finite element package.

Data Availability

Murine biaxial stress-strain data are available through Manuel K. Rausch's dataverse (together with other mechanical raw data): <https://dataverse.tdl.org/dataverse/STBML> Code associated with this manuscript is available at: https://github.com/abuganza/NN_aniso_UMAT

Acknowledgements

This work was supported by the National Institute of Arthritis and Musculoskeletal and Skin Diseases, National Institute of Health, United States under award R01AR074525 to Adrian Buganza Tepole and the National Science Foundation through awards 1916663 and 1916665 to Manuel K. Rausch and Adrian Buganza Tepole, respectively.

References

- [1] A. B. Tepole, M. Gart, C. A. Purnell, A. K. Gosain, E. Kuhl, Multi-view stereo analysis reveals anisotropy of prestrain, deformation, and growth in living skin, *Biomechanics and modeling in mechanobiology* 14 (2015) 1007–1019.
- [2] C. G. Neumann, The expansion of an area of skin by progressive distention of a subcutaneous balloon; use of the method for securing skin for subtotal reconstruction of the ear, *Plastic Reconstructive Surgery* 19 (1957) 124–130.
- [3] V. Sherman, Y. Tang, S. Zhao, W. Yang, M. Meyers, Structural characterization and viscoelastic constitutive modeling of skin, *Acta Biomaterialia* 53 (2017) 460–469.
- [4] S. Kakaletsis, W. D. Meador, M. Mathur, G. P. Sugerman, T. Jazwiec, M. Malinowski, E. Lejeune, T. A. Timek, M. K. Rausch, Right ventricular myocardial mechanics: Multi-modal deformation, microstructure, modeling, and comparison to the left ventricle, *Acta Biomaterialia* 123 (2021) 154–166.
- [5] W. D. Meador, M. Mathur, G. P. Sugerman, T. Jazwiec, M. Malinowski, M. R. Bersi, T. A. Timek, M. K. Rausch, A detailed mechanical and microstructural analysis of ovine tricuspid valve leaflets, *Acta biomaterialia* 102 (2020) 100–113.
- [6] G. A. Holzapfel, *Nonlinear Solid Mechanics; A Continuum Approach for Engineering*, John Wiley & Sons, LTD, 2000.
- [7] G. A. Holzapfel, R. W. Ogden, On planar biaxial tests for anisotropic nonlinearly elastic solids. a continuum mechanical framework, *Mathematics and Mechanics of Solids* 14 (2009) 474–489.
- [8] T. C. Gasser, R. W. Ogden, G. A. Holzapfel, Hyperelastic modelling of arterial layers with distributed collagen fibre orientations, *Journal of the royal society interface* 3 (2005) 15–35.
- [9] J. D. Humphrey, R. K. Strumpf, F. C. P. Yin, Determination of a constitutive model for passive myocardium: I. a new functional form, *Journal of Biomechanical Engineering* 112 (1990) 333–339.
- [10] O. A. Shergold, N. A. Fleck, D. Radford, The uniaxial stress versus strain response of pig skin and silicone rubber at low and high strain rates, *International Journal of Impact Engineering* 32 (2006) 1384–1402.
- [11] Y. C. Fung, *Biomechanics; Mechanical Properties of Living Tissues*, Springer Science + Business Media, Inc, 1993.
- [12] A. Delalleau, G. Josse, J. M. Lagarde, H. Zahouani, J. M. Bergheau, A nonlinear elastic behavior to identify the mechanical parameters of human skin in vivo, *Skin Research and Technology* 14 (2007) 152–164.
- [13] R. J. Lapeer, P. D. Gasson, V. Karri, Simulating plastic surgery: From human skin tensile tests, through hyperelastic finite element models to real-time haptics, *Progress in Biophysics and Molecular Biology* 103 (2010) 208–216.
- [14] H. Joodaki, M. B. Panzer, Skin mechanical properties and modeling: A review, *Journal of Engineering in Medicine* 232 (2018).
- [15] W. D. Meador, G. P. Sugerman, H. M. Story, A. W. Steifert, M. R. Bersi, A. B. Tepole, M. K. Rausch, The regional-dependent biaxial behavior of young and aged mouse skin: A detailed histomechanical characterization, residual strain analysis, and constitutive model, *Acta Biomaterialia* 101 (2020) 403–413.
- [16] T. Lee, S. Y. Turin, A. K. Gosain, I. Bilonis, A. B. Tepole, Propagation of material behavior uncertainty in a nonlinear finite element model of reconstructive surgery, *Biomechanics and Modeling in Mechanobiology* 17 (2018) 1857–1873.
- [17] D. Ambrosi, G. Ateshian, E. Arruda, S. Cowin, J. Dumais, A. Goriely, G. Holzapfel, J. Humphrey, R. Kemkemer, E. Kuhl, J. Olberding, L. Taber, K. Garikipati, Perspectives on biological growth and remodeling, *Journal of the Mechanics and Physics of Solids* 59 (2011) 863–883.
- [18] T. Lee, S. Y. Turin, A. K. Gosain, I. Bilonis, A. B. Tepole, Propagation of material behavior uncertainty in a nonlinear finite element model of reconstructive surgery, *Biomechanics and Modeling in Mechanobiology* 17 (2018) 1857–1873.
- [19] B. Rahmani, W. Sun, A kd-tree-accelerated hybrid data-driven/model-based approach for poroelasticity problems with multi-fidelity multi-physics data, *Computer Methods in Applied Mechanics and Engineering* 382 (2021).
- [20] X. Zhang, K. Garikipati, Machine learning materials physics: Multi-resolution neural networks learn the free energy and nonlinear elastic response of evolving microstructures, *Computer Methods in Applied Mechanics and Engineering* 372 (2020) 113362.
- [21] G. C. Peng, M. Alber, A. B. Tepole, W. R. Cannon, S. De, S. Dura-Bernal, K. Garikipati, G. Karniadakis, W. W. Lytton, P. Perdikaris, et al., Multiscale modeling meets machine learning: What can we learn?, *Archives of Computational Methods in Engineering* (2020) 1–21.

- [22] N. N. Vlassis, W. Sun, Sobolev training of thermodynamic-informed neural networks for interpretable elasto-plasticity models with level set hardening, *Computer Methods in Applied Mechanics and Engineering* 377 (2021).
- [23] L. Lu, M. Dao, P. Kumar, U. Ramamurty, G. E. Karniadakis, S. Suresh, Extraction of mechanical properties of materials through deep learning from instrumented indentation, *Proceedings of the National Academy of Sciences of the United States of America* 117 (2020) 7052–7062.
- [24] E. Lejeune, B. Zhao, Exploring the potential of transfer learning for metamodels of heterogeneous material deformation, *Journal of the Mechanical Behavior of Biomedical Materials* 117 (2021) 104276.
- [25] Y. Leng, S. Calve, A. B. Tepole, Predicting the mechanical properties of fibrin using neural networks trained on discrete fiber network data, *arXiv preprint arXiv:2101.11712* (2021).
- [26] M. Liu, L. Liang, W. Sun, A generic physics-informed neural network-based constitutive model for soft biological tissues, *Computer Methods in Applied Mechanics and Engineering* 372 (2020).
- [27] D. Reimann, K. Chandra, N. Vajragupta, T. Glasmachers, P. Junker, A. Hartmaier, et al., Modeling macroscopic material behavior with machine learning algorithms trained by micromechanical simulations, *Frontiers in Materials* 6 (2019) 181.
- [28] T. Han, T. Lee, J. Ledwon, E. Vaca, S. Turin, A. Kearney, A. Gosain, A. B. Tepole, Bayesian calibration of a computational model of tissue expansion based on a porcine animal model, *Manuscript submitted for review*. (2021).
- [29] L. Bonfiglio, P. Perdikaris, S. Brizzolara, Multi-fidelity bayesian optimization of swath hull forms, *Journal of Ship Research* 64 (2020) 154–170.
- [30] M. Smith, ABAQUS/Standard User's Manual, Version 6.9, Dassault Systèmes Simulia Corp, United States, 2009.
- [31] T. Doyle, J. Ericksen, Nonlinear elasticity, volume 4 of *Advances in Applied Mechanics*, Elsevier, 1956, pp. 53–115.
- [32] B. D. Coleman, N. Walter, *The Foundations of Mechanics and Thermodynamics*, Springer Link, 1974.
- [33] How to implement user-defined fiber-reinforced hyperelastic materials in finite element software, *Journal of the Mechanical Behavior of Biomedical Materials* 110 (2020) 103737.
- [34] A. E. Ehret, M. Itskov, A polyconvex hyperelastic model for fiber-reinforced materials in application to soft tissues, *Journal of Materials Science* 42 (2007) 8853–8863.
- [35] J. E. Prussing, The principal minor test for semidefiniteness matrices, *Journal of Guidance, Control and Dynamics* 9 (1985) 121–122.
- [36] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980* (2014).
- [37] F. Chollet, et al., Keras: The python deep learning library, *ascl* (2018) ascl–1806.
- [38] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al., Tensorflow: Large-scale machine learning on heterogeneous distributed systems, *arXiv preprint arXiv:1603.04467* (2016).
- [39] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, Í. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*, *Nature Methods* 17 (2020) 261–272.
- [40] A. B. Tepole, M. Gart, A. K. Gosain, E. Kuhl, Characterization of living skin using multi-view stereo and isogeometric analysis, *Acta biomaterialia* 10 (2014) 4822–4831.
- [41] J. Ghaboussi, D. A. Pecknold, M. Zhang, R. M. Haj-Ali, Autoprogressive training of neural network constitutive models, *International Journal for Numerical Methods in Engineering* 42 (1998) 105–126.
- [42] J. Ghaboussi, D. Sidarta, New nested adaptive neural networks (nann) for constitutive modeling, *Computers and Geotechnics* 22 (1998) 29–52.
- [43] T. Kirchdoerfer, M. Ortiz, Data-driven computational mechanics, *Computer Methods in Applied Mechanics and Engineering* 304 (2016) 81–101.
- [44] L. A. Mihai, T. E. Woolley, A. Gorieli, Stochastic isotropic hyperelastic materials: constitutive calibration and model selection, *Proceedings of the Royal Society A* 474 (2018).
- [45] Y. Dabiri, A. Van der Velden, K. L. Sack, J. S. Choy, G. S. Kassab, J. M. Guccione, Prediction of left ventricular mechanics using machine learning, *Frontiers in Physics* 7 (2019) 117.
- [46] M. Leshno, V. Y. Lin, A. Pinkus, S. Schocken, Multilayer feedforward networks with a nonpolynomial activation function can approximate any function, *Neural networks* 6 (1993) 861–867.
- [47] G. Limbert, *Skin Biophysics: From Experimental Characterisation to Advanced Modelling*, volume 22, Springer, 2019.
- [48] G. S. Kassab, M. S. Sacks, *Structure-based mechanics of tissues and organs*, Springer, 2016.
- [49] G. A. Holzapfel, T. C. Gasser, R. W. Ogden, A new constitutive framework for arterial wall mechanics and a comparative study of material models, *Journal of elasticity and the physical science of solids* 61 (2000) 1–48.
- [50] T. K. Tonge, L. M. Voo, T. D. Nguyen, Full-field bulge test for planar anisotropic tissues: Part ii – a thin shell method for determining material parameters and comparison of two distributed fiber modeling approaches, *Acta Biomaterialia* 9 (2013) 5926–5942.
- [51] B. Baillargeon, N. Rebelo, D. D. Fox, R. L. Taylor, E. Kuhl, The living heart project: a robust and integrative simulator for human heart function, *European Journal of Mechanics-A/Solids* 48 (2014) 38–47.
- [52] M. Cilla, I. Pérez-Rey, M. A. Martínez, E. Peña, J. Martínez, On the use of machine learning techniques for the mechanical characterization of soft biological tissues, *International Journal for Numerical Methods in Biomedical Engineering* 34 (2018) e3121. E3121 cnm.3121.

Appendix

0.1. Explicit expression for the second Piola-Kirchhoff stress

$$\begin{aligned} \frac{\partial \hat{\mathbf{S}}}{\partial \hat{\mathbf{C}}} = & \delta_1 \mathbf{I} \otimes \mathbf{I} + \delta_2 (\hat{\mathbf{C}} \otimes \mathbf{I} + \mathbf{I} \otimes \hat{\mathbf{C}}) + \delta_3 (\mathbf{V}_0 \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{V}_0) + \delta_4 (\mathbf{W}_0 \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{W}_0) + \\ & \delta_5 (\mathbf{V}_0 \otimes \hat{\mathbf{C}} + \hat{\mathbf{C}} \otimes \mathbf{V}_0) + \delta_6 (\mathbf{W}_0 \otimes \hat{\mathbf{C}} + \hat{\mathbf{C}} \otimes \mathbf{W}_0) + \delta_7 (\mathbf{V}_0 \otimes \mathbf{W}_0 + \mathbf{W}_0 \otimes \mathbf{V}_0) + \\ & \delta_8 \mathbf{V}_0 \otimes \mathbf{W}_0 + \delta_9 \mathbf{V}_0 \otimes \mathbf{W}_0 + \delta_{10} \hat{\mathbf{C}} \otimes \hat{\mathbf{C}} + \delta_{11} \mathbb{I} \end{aligned}$$

$$\delta_1 = +2(\hat{\Psi}_{11} + 2\hat{I}_1 \hat{\Psi}_{12} + \hat{\Psi}_2 + \hat{I}_1^2 \hat{\Psi}_{22})$$

$$\delta_2 = -2(\hat{\Psi}_{12} + \hat{I}_1 \hat{\Psi}_{22})$$

$$\delta_3 = +2(\hat{\Psi}_{14v} + \hat{I}_1 \hat{\Psi}_{24v})$$

$$\delta_4 = +2(\hat{\Psi}_{14w} + \hat{I}_1 \hat{\Psi}_{24w})$$

$$\delta_5 = -2\hat{\Psi}_{24v}$$

$$\delta_6 = -2\hat{\Psi}_{24w}$$

$$\delta_7 = +2\hat{\Psi}_{4v4w}$$

$$\delta_8 = +2\hat{\Psi}_{4v4v}$$

$$\delta_9 = +2\hat{\Psi}_{4w4w}$$

$$\delta_{10} = +2\hat{\Psi}_{22}$$

$$\delta_{11} = -2\hat{\Psi}_2$$

0.2. Finite element method analysis results with GOH material model

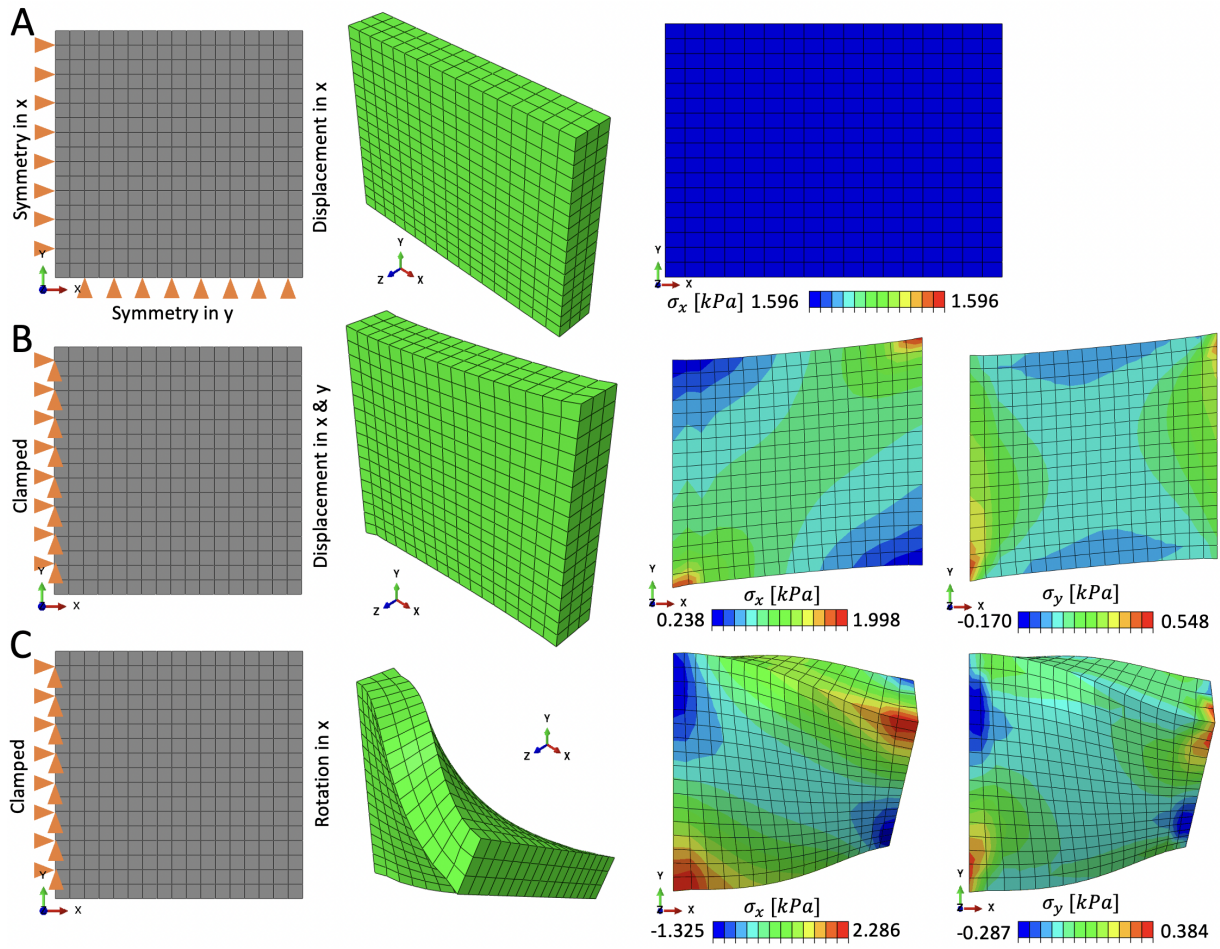


Figure 10: Finite element method simulations using the GOH material model in UMAT. Boundary conditions, deformed geometry and contours of σ_x under uniaxial loading (A), Boundary conditions, deformed geometry and contours of σ_x and σ_y under shear loading (B), and, Boundary conditions, deformed geometry and contours of σ_x and σ_y under torsional loading (C).