Engineering with Computers (2022) 38:4167–4182
https://doi.org/10.1007/s00366-022-01733-3

ORIGINAL ARTICLE

Data-driven modeling of the mechanical behavior of anisotropic soft biological tissue

Vahidullah Tac[1] · Vivek D. Sree[1] · Manuel K. Rausch[2,3,4] · Adrian B. Tepole[1,5]

Received: 12 January 2022 / Accepted: 15 August 2022 / Published online: 16 September 2022

Abstract
Closed-form constitutive models are currently the standard approach for describing soft tissues' mechanical behavior. However, there are inherent pitfalls to this approach. For example, explicit functional forms can lead to poor fits, non-uniqueness of those fits, and exaggerated sensitivity to parameters. Here we overcome some of these problems by designing deep neural networks (DNN) to replace such explicit expert models. One challenge of using DNNs in this context is the enforcement of stress-objectivity. We meet this challenge by training our DNN to predict the strain energy and its derivatives from (pseudo)-invariants. Thereby, we can also enforce polyconvexity through physics-informed constraints on the strain-energy and its derivatives in the loss function. Direct prediction of both energy and derivative functions also enables the computation of the elasticity tensor needed for a finite element implementation. Then, we showcase the DNN's ability by learning the anisotropic mechanical behavior of porcine and murine skin from biaxial test data. Through this example, we find that a multi-fidelity scheme that combines high fidelity experimental data with a low fidelity analytical approximation yields the best performance. Finally, we conduct finite element simulations of tissue expansion using our DNN model to illustrate the potential of data-driven approaches such as ours in medical device design. Also, we expect that the open data and software stemming from this work will broaden the use of data-driven constitutive models in soft tissue mechanics.

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

1 Introduction

Skin is the largest organ in the body and understanding its mechanical properties is a crucial step in many biomedical applications, from prosthesis design to surgical intervention [1]. 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 behavior [2]. The mechanical properties of skin are actually common across many soft connective tissues [3, 4]. Traditionally, the mechanics of skin and other soft tissues has been modelled using expert-constructed constitutive equations [5–7]. In this approach, a closed-form expression describing the main features of the mechanics 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. Inherent restrictions of the explicit functional form can result in poor fitting and high sensitivity to parameters [8]. Unfortunately, even considering just skin out of all connective soft tissue, there

✉ Adrian B. Tepole
abuganza@purdue.edu

Vahidullah Tac
vtac@purdue.edu

Vivek D. Sree
vdharman@purdue.edu

Manuel K. Rausch
manuel.rausch@utexas.edu

[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] Department of Biomedical Engineering, The University of Texas at Austin, Austin, TX, USA

[4] Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, TX, USA

[5] Weldon School of Biomedical Engineering, Purdue University, West Lafayette, IN, USA

is currently no consensus on the choice of model that is most suitable in a particular application [9–11].

A new, emergent approach to material modeling is the use of data-driven methods [12, 13]. Among them, deep neural networks (DNN) have been successfully employed to describe the mechanical behavior of several materials [14–17]. In this approach, there is no need to limit the model to an analytical representation, which results in more accurate predictions than traditional models [18–20]. Physics constraints such as objectivity of the stress and convexity of the hyperelastic strain energy potential are naturally satisfied by most closed-form constitutive models [21]. These constraints are embedded into data-driven methods either during the design of the algorithm itself [22], or as a penalty [23]. Drawbacks of existing approaches stem from scarcity of high fidelity test data to train the data-driven models [19]. This limitation is particularly prevalent in soft tissue mechanics. Additionally, there is a lack of data-driven material model software that can function in standard finite element solvers, which severely limits the applicability of these emerging methods to biomedical applications.

The route followed in our current work lies between the purely data-driven approach and the expert modeling approach. Closed-form material models already include knowledge of physics relevant to soft tissue, observations of the underlying microstructure, and intuition from the modeller regarding the main features of the material response. For example, to model skin, we have assumed hyperelasticity and used expert-designed strain energy functions to fit murine and porcine skin data [24]. However, the error in the fits can be undesirable, the parameters non-unique, and the predictions can be highly sensitive to the parameters [25]. Here we design DNN constitutive models and train them on multi-fidelity data: analytical strain energy functions serve as low fidelity approximations, high fidelity experimental measurements complement the data set. This approach is based on the recent literature that shows the advantage of multi-fidelity schemes over single fidelity approaches [16, 17, 26].

The proposed DNNs output the strain energy and its derivatives with respect to the isochoric strain invariants, including anisotropy, satisfying stress-objectivity a priori. The loss function is designed to impose polyconvexity of strain energy. The multi-output design in which both the energy and derivatives are predicted independently by the DNN, but coupled through additional loss terms, provides more flexibility during training and enables the computation of the the stress and elasticity tensors. As a result, we are able to implement a DNN user material (UMAT) subroutine for the widely used nonlinear finite element package Abaqus [27], and showcase its potential to impact skin therapeutics through simulations of tissue expansion. 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 (Fig. 1).

2 Methods

2.1 Constitutive equations for a hyperelastic material with two families of fibers

In this study we use a 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 [5],

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

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

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

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

$$\hat{\mathbf{F}} = J^{-1/3}\mathbf{F}, \tag{3}$$

$$\hat{\mathbf{C}} = \hat{\mathbf{F}}^{\mathsf{T}}\hat{\mathbf{F}} = J^{-2/3}\mathbf{C}\,. \tag{4}$$

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 [28, 29], arriving at

$$\mathbf{S} = 2\frac{\partial \Psi}{\partial \mathbf{C}} = \mathbf{S}_{\mathrm{iso}} + \mathbf{S}_{\mathrm{vol}}\,, \tag{5}$$

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

The following definition of the pressure has been introduced $p = d\Psi_{\mathrm{vol}}/dJ$. 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. $\hat{\Psi}_1 = \partial\Psi_{\mathrm{iso}}/\partial\hat{I}_1$, $\hat{\Psi}_2 = \partial\Psi_{\mathrm{iso}}/\partial\hat{I}_2$,

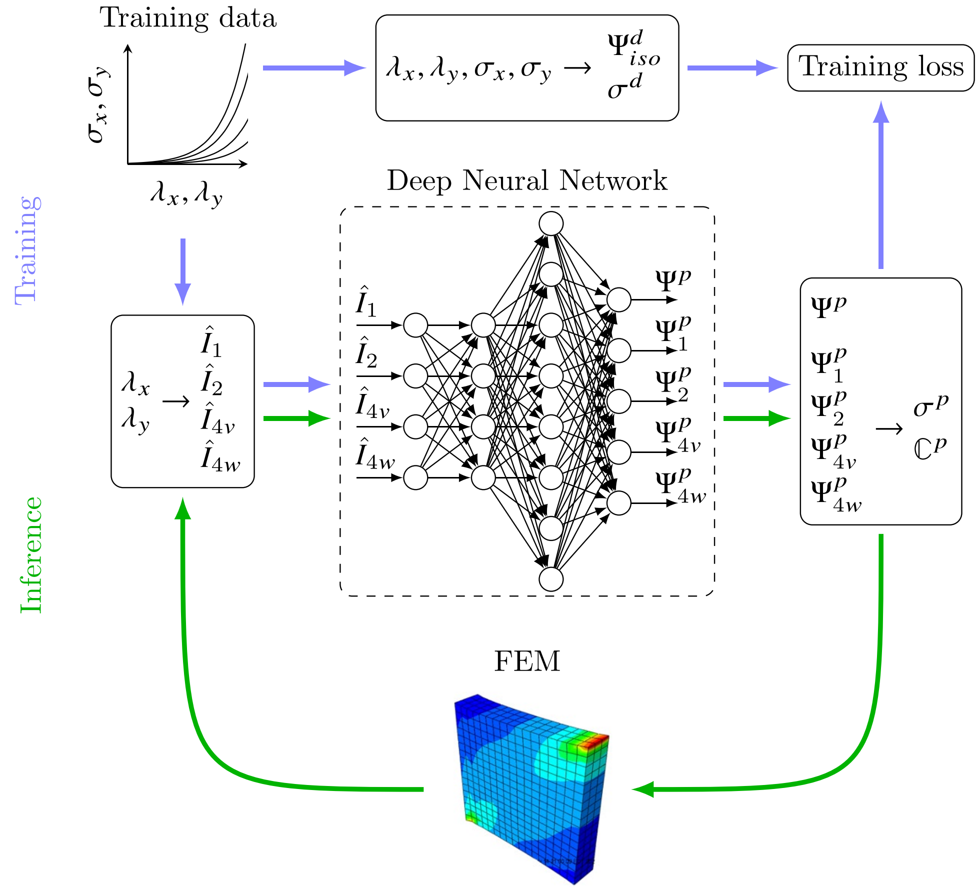
Training data
σ_x, σ_y
λ_x, λ_y
$\lambda_x, \lambda_y, \sigma_x, \sigma_y \rightarrow$ Ψ^d_{iso} σ^d
Training loss
Training
Deep Neural Network
λ_x λ_y $\rightarrow$ $\hat{I}_1$ $\hat{I}_2$ $\hat{I}_{4v}$ $\hat{I}_{4w}$
$\hat{I}_1$
$\hat{I}_2$
$\hat{I}_{4v}$
$\hat{I}_{4w}$
Ψ^p
Ψ^p_1
Ψ^p_2
Ψ^p_{4v}
Ψ^p_{4w}
Ψ^p Ψ^p_1 Ψ^p_2 Ψ^p_{4v} Ψ^p_{4w} $\rightarrow$ σ^p $\mathbb{C}^p$
Inference
FEM

Fig. 1 Diagram depicting the training and inference processes of the deep neural network material model. *FEM* Finite Element Method

$\hat{\Psi}_{4v} = \partial\Psi_{\text{iso}}/\partial\hat{I}_{4v}$ and $\hat{\Psi}_{4w} = \partial\Psi_{\text{iso}}/\partial\hat{I}_{4w}$. The full expansion of the fictitious stress tensor is

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

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. Note that if the material under consideration is incompressible, i.e. $J = 1$, then $\hat{\mathbf{C}} = \mathbf{C}$, (5) reduces to

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

and the pressure p becomes an unknown Lagrange multiplier field. In certain cases, p can be solved from boundary conditions. In this study, the nearly incompressible formulation is used in the finite element formulation, while the incompressible formulation is used during neural network training since this constraint can be easily enforce for the plane stress biaxial deformations considered.

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

$$\sigma = \frac{1}{J}\mathbf{F}\mathbf{S}\mathbf{F}^{\top}\,. \tag{9}$$

The finite element subroutine also requires the computation of the elasticity tensor, $\mathbb{c}_{\text{abaqus}}$ [30]. For ease of derivation, the material version of the elasticity tensor, $\mathbb{C}$, is introduced first,

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

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

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

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

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

where the modified pressure term, $\tilde{p} = p + Jdp/dJ$, 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, $\hat{\mathbb{C}}$ is obtained from differentiating the fictitious stress tensor with respect to the isochoric part of the deformation tensor,

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

The full expansion of $\hat{\mathbb{C}}$ is available in the Supplement. The only remark needed in the main text is that the tensor $\hat{\mathbb{C}}$ 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

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

where we have introduced the modified dyadic product defined as $(\bullet\overline{\otimes}\circ)_{ijkl} = (\bullet)_{ik}(\circ)_{jl}$. 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

$$\mathbb{c}_{\text{abaqus}} = \mathbb{c} + \frac{1}{2}(\sigma\overline{\otimes}\mathbf{I} + \sigma\underline{\otimes}\mathbf{I} + \mathbf{I}\overline{\otimes}\sigma + \mathbf{I}\underline{\otimes}\sigma) \,. \tag{15}$$

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

As remarked before, incompressibility is imposed exactly during the training of the neural network, with p determined from boundary conditions. However, in the UMAT we use a volumetric strain energy which leads to the following expression for p,

$$p = K(J-1)\,, \tag{16}$$

with K the bulk modulus. In this study we set $K = 1\,\text{MPa}$.

2.2 Neural network structure and training

We use a fully connected DNN 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, $\Psi^{\text{p}}_{\text{iso}}$, and its derivatives with respect to the invariants, $\hat{\Psi}^{\text{p}}_1$, $\hat{\Psi}^{\text{p}}_2$, $\hat{\Psi}^{\text{p}}_{4v}$ and $\hat{\Psi}^{\text{p}}_{4w}$. Note that the notation $(\bullet)^{\text{p}}$ is used to denote the predicted values of the DNN. The network architecture is summarized in Table 1.

Training data for the DNN is in the form of stretch and stress data, as well as the values of strain energy obtained by the integration of the stretch-stress curves. Therefore, the first component of the loss function is simply the comparison of the predicted strain energy, $\Psi^{\text{p}}_{\text{iso}}$, against the observed $\Psi^{\text{d}}_{\text{iso}}$, where $(\bullet)^{\text{d}}$ is used to refer to data. The first component of the loss function is

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

where $(\bullet)^{(n)}$ denotes the n^{th} training point, out of a total of N training points. The second term in eq. (17) is a regularization term that is added to the loss function to enforce that the predicted derivatives are consistent with the strain energy. In other words, the derivatives that are a direct output of the neural network, $\hat{\Psi}^{\text{p}}_i$, should coincide with the derivatives of $\Psi^{\text{p}}_{\text{iso}}$ calculated using back-propagation, denoted as $\hat{\Psi}^{\text{bp}}_i$ in eq. (17), with $i = 1, 2, 4v, 4w$, and where $(\bullet)^{\text{bp}}$ stands for "back-propagation".

The second component of the loss results from comparing the stress computed with the neural network outputs against the observed stress σ^{d}. The stress, defined in Eqs. (8) and (9), is computed based on the direct strain energy derivatives output by the neural network, $\Psi^{\text{p}}_{\text{i}}$, to produce σ^{p}. The loss for the stress data can then be simply stated as

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

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

Table 1 Neural network architecture

Layer	Number of nodes	Activation function
Input	4	None
Hidden layer 1	4	Sigmoid
Hidden layer 2	8	Sigmoid
Hidden layer 3	8	Sigmoid
Output	5	Linear/quadratic

To guarantee existence of a solution to the boundary value problem, a suitable constraint on the strain energy is that of polyconvexity with respect to the deformation gradient, **F** [21, 31, 32]. An alternative approach is to enforce convexity of the strain energy with respect to **C** [33]. Convexity with respect to **C** can lead to the existence of global minima in boundary value problems under certain conditions [34]. This convexity condition has been employed in constitutive modeling of biological tissues [35] and in numerous studies on data-driven models of hyperelastic materials [19, 33]. For instance, the popular constitutive model by Holzapfel, Gasser and Ogden to capture the mechanical behavior of collagenous tissues was developed to fulfill this condition [5]. However, convexity with respect to **C**, and polyconvexity with respect to **F** are not equivalent. Polyconvexity of the strain energy function with respect to **F**, together with some growth conditions on the strain energy, guarantees the existence of global minimizers to the total potential energy functional [31].

In the current study we enforce convexity of the strain energy function with respect to the deformation invariants of **C** with a global minimum at or below the point $I_1 = 3, I_2 = 3, I_{4v} = 1$ and $I_{4w} = 1$. The invariants are already convex functions of **F** or cof **F** [21, 36]. Thus, the non-decreasing convex function of the invariants, for $I_1 \geq 3, I_2 \geq 3, I_{4v} \geq 1$ and $I_{4w} \geq 1$, together with a suitable volumetric energy convex in J results in polyconvexity of the strain energy with respect to **F**.

For a function to be convex with respect to its arguments, it's Hessian matrix, **H**, must be positive semi-definite [37]. The Hessian matrix of the strain energy as a function of the invariants is

$$\mathbf{H} = \begin{pmatrix} \Psi^{\mathrm{bp}}_{11} & \Psi^{\mathrm{bp}}_{12} & \Psi^{\mathrm{bp}}_{14v} & \Psi^{\mathrm{bp}}_{14w} \\ \Psi^{\mathrm{bp}}_{21} & \Psi_{22^{\mathrm{bp}}} & \Psi^{\mathrm{bp}}_{24v} & \Psi^{\mathrm{bp}}_{24w} \\ \Psi^{\mathrm{bp}}_{4v1} & \Psi^{\mathrm{bp}}_{4v2} & \Psi^{\mathrm{bp}}_{4v4v} & \Psi^{\mathrm{bp}}_{4v4w} \\ \Psi^{\mathrm{bp}}_{4w1} & \Psi^{\mathrm{bp}}_{4w2} & \Psi^{\mathrm{bp}}_{4w4v} & \Psi^{\mathrm{bp}}_{4w4w} \end{pmatrix} . \tag{19}$$

The notation Ψ^{bp}_{ij} indicates the second derivative of the strain energy computed with the neural network by differentiating the outputs $\hat{\Psi}^{\mathrm{p}}_{i}$ with respect to the jth input using back-propagation. We impose positive-definiteness of the Hessian matrix using the principal minor test [38]. 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 an additional loss term,

$$\begin{aligned} \mathcal{L}_3 = {} & \frac{1}{N} \sum_{n=1}^{N} \left|\left| (\mathbf{H})^{(n)} - \left(\mathbf{H}^{\top}\right)^{(n)} \right|\right|_F \\ & + \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{4} \max\left((-\Delta_k^{(n)}, 0) \right) . \end{aligned} \tag{20}$$

Note that non-negative derivatives are obtained by passing the outputs $\hat{\Psi}^{\mathrm{p}}_{i}$ through the quadratic function $y_i = x_i^2$ as indicated in Table 1. The choice of Linear or Quadratic activation functions in the last layer is selected in the examples below depending on whether convexity is imposed or not. The non-negative outputs do not directly enforce convexity. Rather, given that the invariants considered are convex functions of **F** or cof **F**, convex non-decreasing functions of these invariants are needed for polyconvexity. The quadratic activation functions for the $\hat{\Psi}^{\mathrm{p}}_{i}$ outputs in the last layer enforce the non-decreasing condition for the strain energy by restricting the derivatives to be non-negative.

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

$$\mathcal{L} = a_1 \mathcal{L}_1 + a_2 \mathcal{L}_2 + a_3 \mathcal{L}_3 \, . \tag{21}$$

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,

$$\mathcal{L}_{\mathrm{mf}} = \mathcal{L}_{\mathrm{lf}} + a_{hf} \mathcal{L}_{\mathrm{hf}} \, . \tag{22}$$

The training of the DNN was performed using the Adam optimization algorithm [39]. The initial learning rate was set to 4.0e−5. 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 DNN was trained in 100000 epochs without the use of batching. The training was implemented using Keras [40] with a Tensorflow [41] 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 three weights a_1, a_2 and a_3 are set to 0.1, 1.0 and 0.008, respectively, after performing a hyperparameter study as reported in the Supplement.

2.2.1 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 are 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) [6] material model. The GOH model proposes an isochoric strain energy of the form

$$\hat{\Psi}(\mathbf{C}, \mathbf{a}_0) = \hat{\Psi}_{\text{iso}}(\mathbf{C}) + \hat{\Psi}_{\text{aniso}}(\mathbf{C}, \mathbf{v}_0) \tag{23}$$

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

$$\hat{\Psi}_{\text{iso}}(\mathbf{C}) = \mu(\hat{I}_1 - 3)\,, \tag{24}$$

$$\hat{\Psi}_{\text{aniso}}(\mathbf{C}, \mathbf{a}_0) = \frac{k_1}{4k_2}\left[\exp\left(k_2 E^2\right) - 1\right]\,, \tag{25}$$

with the generalized fiber strain

$$E = \left[\kappa \hat{I}_1 + (1 - 3\kappa)\hat{I}_{4v} - 1\right]\,. \tag{26}$$

The volumetric term that is the same as the one we used to penalize volume changes in our formulation [5, 6, 42],

$$\Psi_{\text{vol}} = \frac{K}{2}(J - 1)^2\,. \tag{27}$$

The derivation of the stress tensor for the GOH strain energy is not repeated here, the interested reader is referred to [6, 25].

Synthetic data with the GOH model is generated by fitting the free parameters to the experimental data using the BFGS optimization algorithm in SciPy [43].

2.2.2 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^{p}_i.
4. Calculate stress using Eqs. (5 - 7, 9).
5. Calculate the second derivatives Ψ^{bp}_{ij} 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

$$\mathbf{y}_i = g_i(\mathbf{W}_i^T \mathbf{y}_{i-1} + \mathbf{B}_i), \quad \mathbf{y}_i \in \mathbb{R}^n \tag{28}$$

where $g_i : \mathbb{R} \to \mathbb{R}$ 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,

$$g(y) = \frac{1}{1 + e^{-y}}, \quad g'(y) \equiv \frac{dg(y)}{dy} = g(y)(1 - g(y))\,. \tag{29}$$

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

$$\mathbf{J}_i = \text{diag}(g'(\mathbf{y}_i))\mathbf{W}_i^T \mathbf{J}_{i-1}, \quad \mathbf{J}_i \in \mathbb{R}^{n\times m_0} \tag{30}$$

where m_0 is the number of inputs to the neural network and diag(•) denotes a diagonal matrix.

2.3 Biaxial stress-stretch experiments on porcine and murine skin

We use experimental data from biaxial stress-stretch experiments performed on murine [24] and porcine skin for the training and validation of neural networks. The data are collected in up to 5 different experimental protocols which are defined in Table 2.

2.4 Training data

High fidelity training data used in this study consists of 13 sets of experimental data obtained from 2 pigs and 11 mice. The first porcine dataset consists of 122 data points in the off-x and off-y loading protocols, the second porcine dataset consists of 402 data points which encompasses all 5 loading protocols in Table 2. The murine dataset consists of 549 points in the off-x, off-y and equibiaxial protocols.

Low fidelity data was generated using the GOH material model. For each of the 3 high fidelity datasets, first the free parameters of the GOH model were fitted to the data. Then the model was used to generate 225 synthetic data points for each of the porcine datasets and 165 points for the murine dataset.

During training of the neural networks, the contribution of the high fidelity data are weighted higher than the low

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

fidelity data. This guides the neural network to adhere to the experimental data more closely while approximating the low fidelity data in regions with no high fidelity data. In this study the ratio between the weights was set to 50 : 1.

3 Results

3.1 Performance of the neural network against synthetic data

To test the DNN 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 (Fig. 2a). These loading curves are representative of the the type of test that can be performed experimentally. On the other hand, the DNN takes as inputs the isochoric strain invariants. The invariant space is 4-dimensional, but we plot a 3-dimensional projection in Fig. 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 Fig. 2c. The predictions of the trained network are plotted against the training data in Fig. 2d–f. These results indicate that the DNN is able to recreate almost perfectly the expert constitutive models within the training region.

We also test if the DNN 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

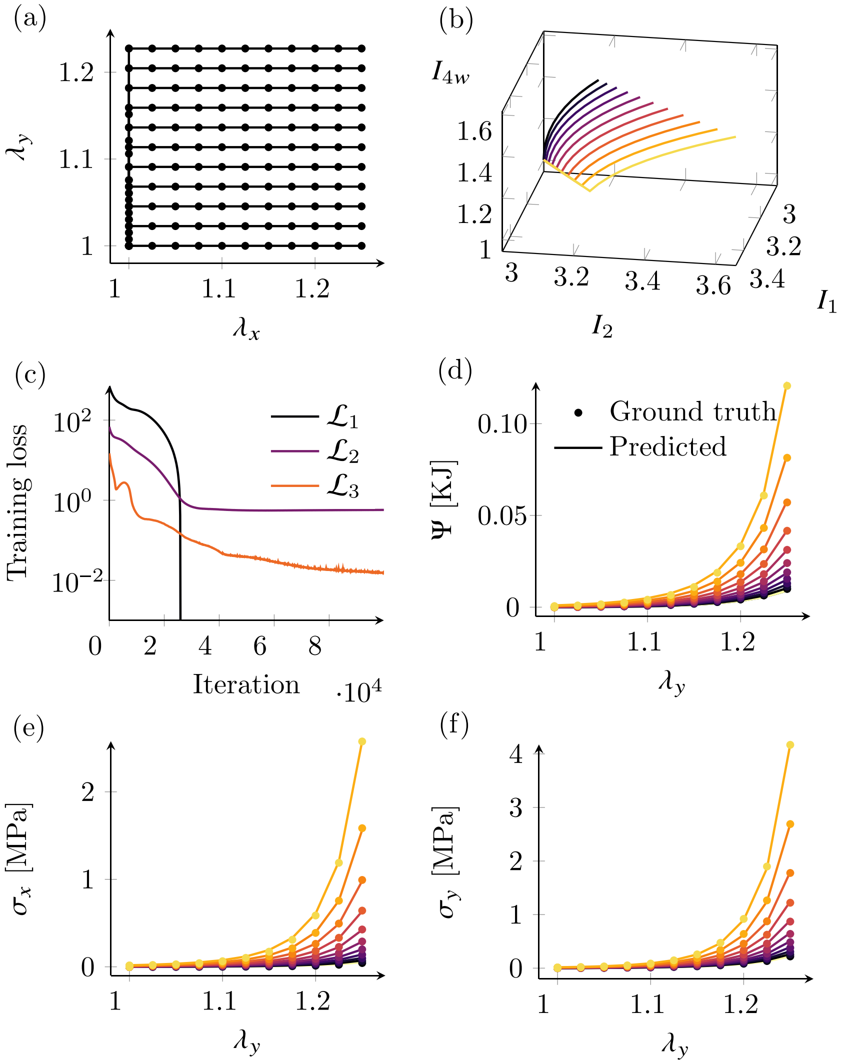

Fig. 2 Synthetic data generated to train the neural network and performance of the neural network compared to the training data. **a** Training data was generated by creating *curves* in the λ_x, λ_y space. **b** Corresponding training data in the invariant space, which is the actual input space for the neural network. The invariant space is four-dimensional but only a three-dimensional projection is shown. The colors of the curves indicate the value of the fourth invariant. **c** Loss during training. **d** Predicted and ground truth strain energy values. Predicted and ground truth planar stress values in the (**e**) x- and (**f**) y-directions. The colors of the curves in (**d**), (**e**) and (**f**) indicate the value of λ_x

$$\mathbf{F} = \begin{pmatrix} \lambda_x & \gamma_{xy} & 0 \\ \gamma_{xy} & \lambda_y & 0 \\ 0 & 0 & \frac{1}{\lambda_x\lambda_y-\gamma_{xy}^2} \end{pmatrix}. \tag{31}$$

The validation dataset is generated from randomly sampling $\lambda_x \in [1, 1.25]$, $\lambda_y \in [1, 1.25]$, $\gamma_{xy} \in [0, 0.3]$. Lastly, we are interested in the potential of the neural network to extrapolate under biaxial deformations. 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 Fig. 3.

The stress $\hat{\sigma}^{\mathrm{p}}$ and $\hat{\sigma}^{\mathrm{d}}$ are normalized such that each entry of these tensors is obtained by subtracting the mean and dividing by the standard deviation of the stress values over the validation dataset, e.g. $\hat{\sigma}^{\mathrm{p}}_{ij} = (\sigma^{\mathrm{p}}_{ij} - \sigma^{\mathrm{d,avg}}_{ij})/\Sigma^{\mathrm{d}}_{ij}$, with $\sigma^{\mathrm{d,avg}}_{ij}$ the mean of that stress component over the validation data, and Σ^{d}_{ij} the corresponding standard deviation. If the data were normal, then the normalized quantities would be almost entirely in the range $[-3, 3]$. Even though the data are not normal, this is a useful scaling of the stress. Relative errors are high in the low strain region for which the stress is negligible (orders of magnitude lower than in the high stress regions), and absolute errors are higher in regions of high stress even if the relative error is small. The normalized stress better captures the performance of the DNN over the input space. It can be seen that the DNN performs well within the training region but worse toward the boundary of the training region.

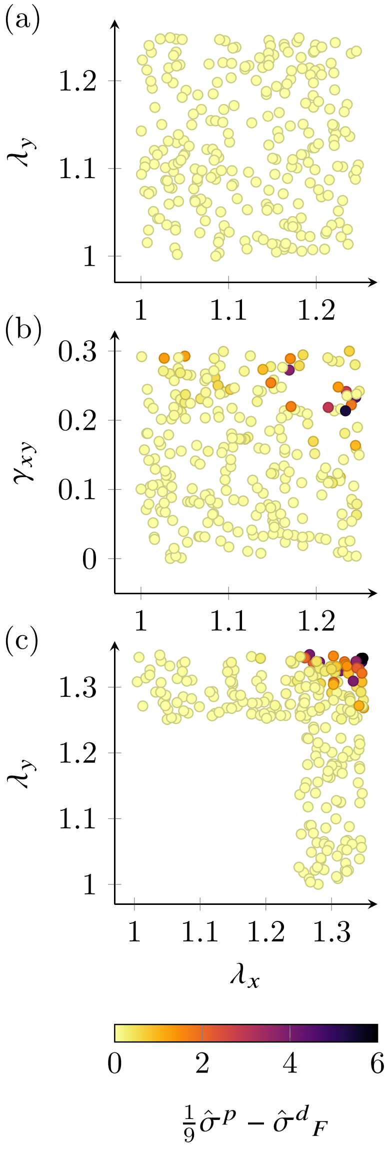

Fig. 3 Validation of the neural network trained on synthetic data. Performance of the neural network on points randomly sampled from: **a** $\lambda_x \in [1, 1.25]$ and $\lambda_y \in [1, 1.25]$, **b** $\lambda_x \in [1, 1.25]$, $\lambda_y \in [1, 1.25]$, $\gamma_{xy} \in [0, 0.3]$ and **c** $\lambda_x \in [1, 1.25]$ 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 $\lambda_y \in [1.25, 1.35]$. The colorbar indicates the error of each point as defined at the bottom of the figure

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

Next, we start training the DNN 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 GOH model fit (multi-fidelity data). Concurrently, we want to test if the polyconvexity constraint is required to regularize the fits of the DNN. 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, mean($\|\sigma^p - \sigma\|_F$).

The first row of Fig. 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 DNN ability to fit the off-x and off-y data, achieving average errors of 4.56 kPa and 5.25 kPa, respectively. A biaxial test, not used in training, is used to test the predictive capability of the network (Fig. 4b). The average error in the validation is 7.33 kPa. Because no convexity constraint is used we can see that it is not satisfied (Fig. 4c). Keeping only the high-fidelity data but imposing convexity changes the performance. The training and validation loss are poorer (Fig. 4e–g), but the function is convex over the input space (Fig. 4h).

The third and fourth rows of Fig. 4 show the results of DNNs 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 (Fig. 4l). The average error in the

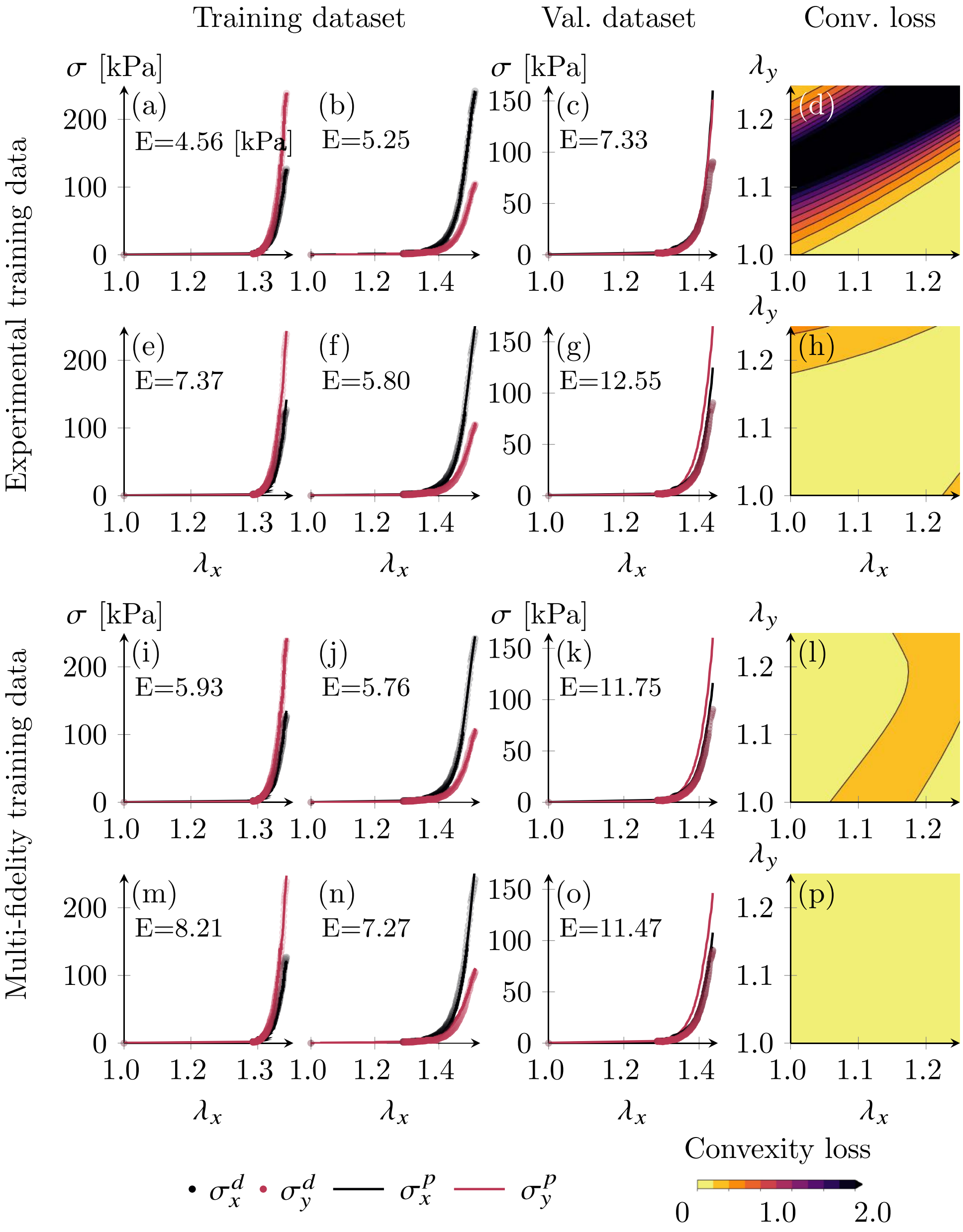

Fig. 4 Performance of the DNN on the murine skin data and average prediction error, E. The plots in each row show the predicted stress vs actual stress on the training (Off-x and Off-y) and validation (Equibiaxial), while the last column shows the convexity loss throughout the input space. Each row corresponds to a separate DNN. Predictions of DNN 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

validation set for the multi-fidelity case without convexity constraint is 11.75 kPa (Fig. 4k), which is worse than the sparse high fidelity case without convexity requirements (Fig. 4c). The results for the neural network trained with multi-fidelity data and with the convexity constraint are shown in last row of Fig. 4. The training errors are 8.21 kPa and 7.27 kPa (Fig. 4(m) and (n)), and the validation error is 11.47 kPa (Fig. 4o). Since the convexity is imposed, the loss in the convexity is close to zero over the entire input space (Fig. 4p). In summary, polyconvexity is a useful framework to guarantee existence of minimizers for problems in elasticity, 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. Additional 10 murine datasets are shown in the Supplemental Material, showing that the DNN can easily fit a wide variety of skin samples.

In Fig. 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 DNNs, one of them is trained with the experimental data only (first and third columns of Fig. 5), whereas the other is trained on the augmented data (second and fourth columns of Fig. 5). In Fig. 5c it can be seen that trained only on experimental data, the neural network achieves a low average error of 5.13 kPa, compared to the GOH fit which is 22.54 kPa. Thus, the neural network outperforms the GOH model. This is not surprising

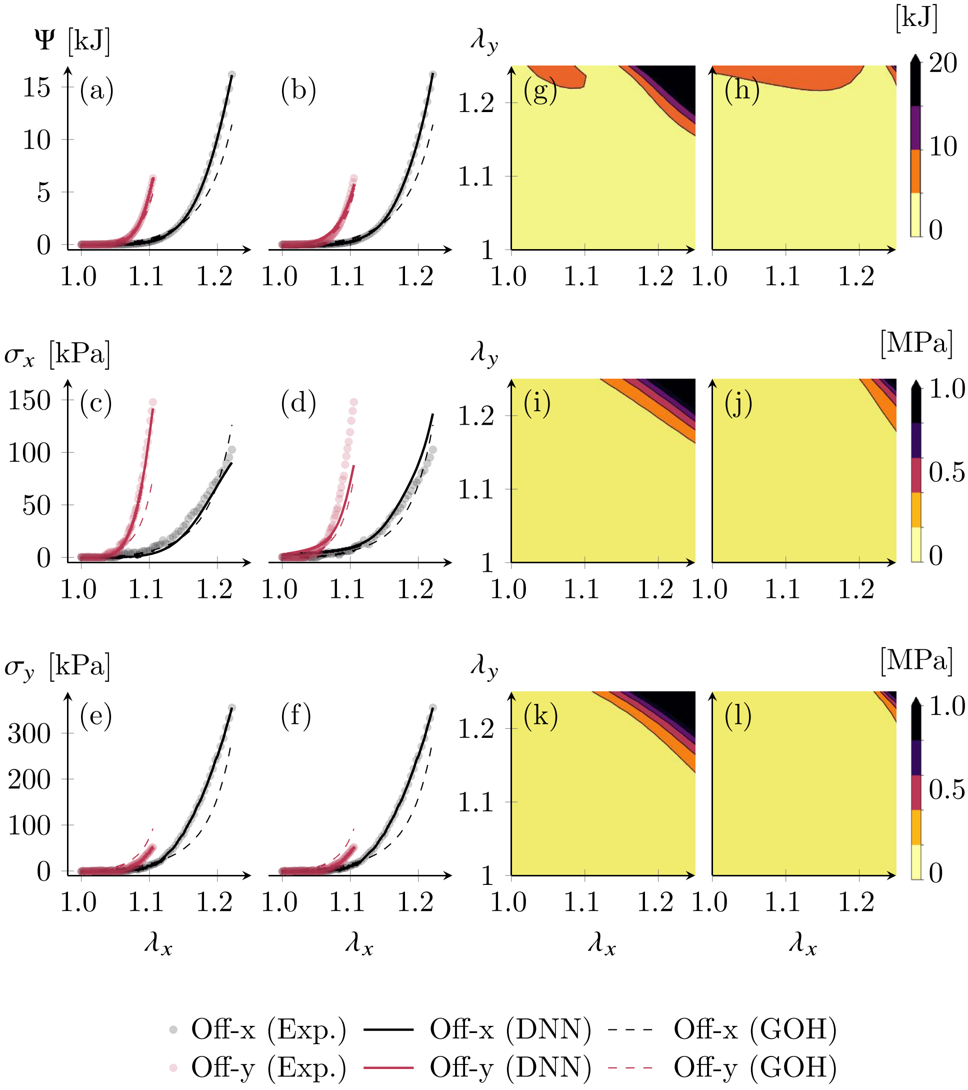

Fig. 5 Performance of DNNs trained on single-fidelity (first and third columns) and multi-fidelity (second and fourth columns) training data. The scatter plots compare predicted strain energy and stress values to experimental data as well as GOH model outputs. The contour plots show the difference between the corresponding outputs of the GOH model and the DNN

since the only task of the neural network is to interpolate the experimental data and satisfy convexity. The contour plots in Fig. 5g–l 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 differ toward the boundary of the deformation space considered. Surprisingly, the models agree with each other in large portions of the input space even when the DNN is trained independently of the GOH model (Fig. 5g–k). Based on the validation examples with the synthetic dataset, we know that the DNN does not extrapolate well outside of the training region. On the other hand, the GOH model has been developed and trained against thousands of tissue biaxial data. It is reasonable to expect that the GOH model, even though it cannot fit any particular dataset as well as the DNN, 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 12.81 kPa against the experimental data (Fig. 5c and e), which is higher than the single-fidelity DNN but still lower with respect to using the GOH model alone. However, as looking at the contours in Fig. 5h, j and l we see that the neural network now follows the GOH model even more closely on the entire input space.

Therefore, the DNN trained with augmented data are 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 DNN material 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 DNN material model. Thus, we train the DNN with different combination of experimental data and validate against the rest of the data (Fig. 6). We do the same training and testing with the GOH model. In

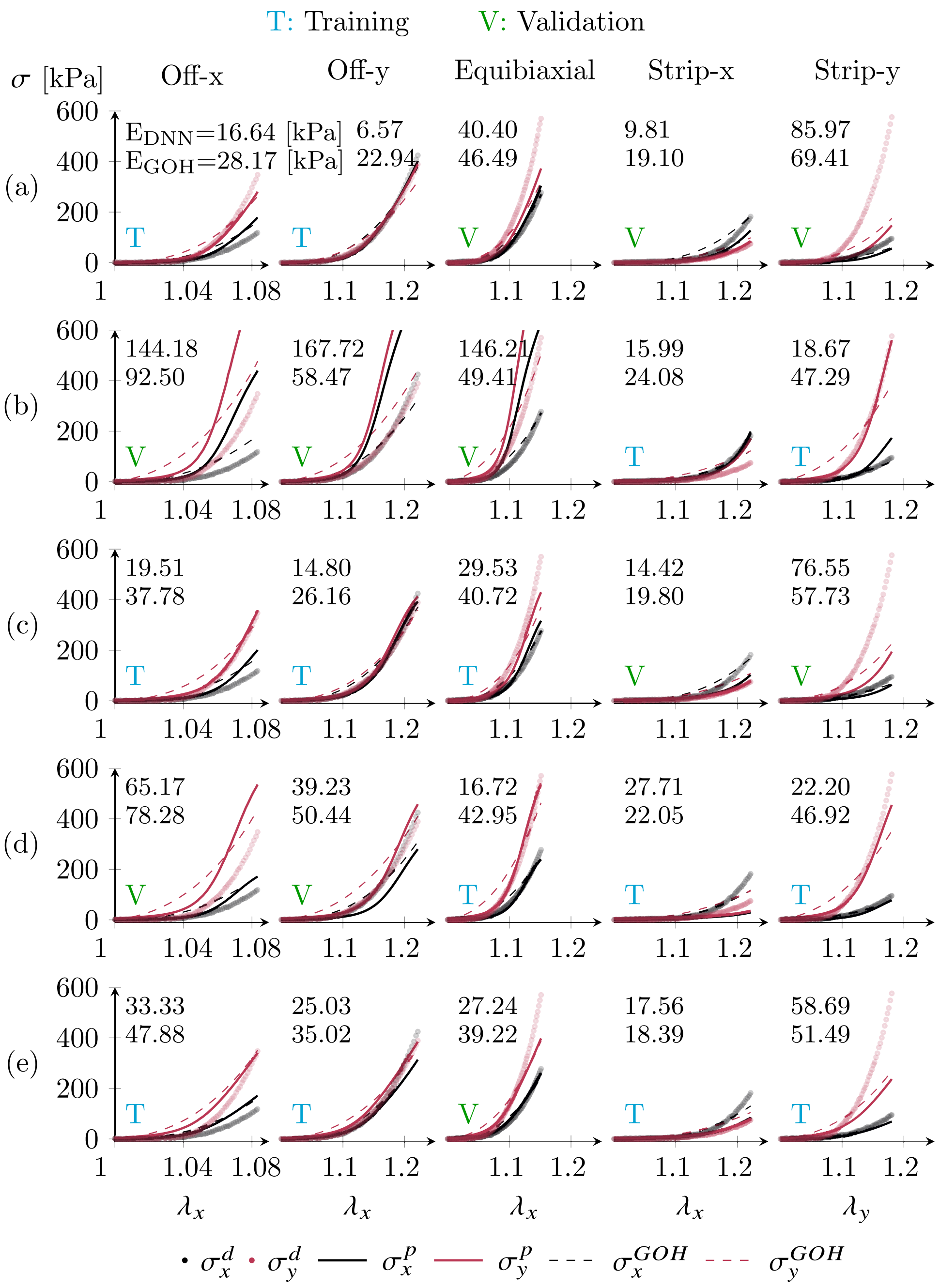

Fig. 6 Comparison of predicted stress vs actual stress vs GOH fits for various training/validation splits of porcine experimental data. Predictions of a neural network trained on **a** Off-x and Off-y data, **b** Strip-x and Strip-y data, **c** Off-x, Off-y and Equibiaxial data, **d** Equibiaxial, Strip-x and Strip-y data and **e** Off-x, Off-y, Strip-x and Strip-y data

Fig. 6(a) 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. In the validation data we see that the neural network performs similarly to the GOH model in the case in which is trained on Off-x and Off-y data, but is outperformed by the GOH model in the case in which it is trained with Strip-x and Strip-y data. 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 DNN obviously outperforms the GOH model in the training set. During validation, when the DNN is trained on strip biaxial data as well as equibiaxial data it is able to outperform the GOH model in both validation cases (Fig. 6d). In Fig. 6e we train against four tests, and validate against the equibiaxial test. The validation and training errors are lower for the neural network compared to the GOH model. The superior performance of the DNN on the last case confirms that data-driven models are a preferable alternative to expert constructed constitutive models when sufficient training data are available.

3.3 Finite element method implementation

We show a number of basic finite element simulations to test the capabilities of the DNN material model as a UMAT subroutine for Abaqus. The neural network trained on porcine skin data (with convexity constraint) is defined in the input file. We first consider a rectangular block $5 \times 5 \times 1$ cm^3. Boundary conditions, mesh and results for a uniaxial extension simulation with $\lambda_x = 1.2$ are depicted in Fig. 7a. The result is a homogeneous stress distribution of $\sigma_x = 29.9$ kPa, $\sigma_y = \sigma_z = 0$, consistent with the results in Fig. 6, confirming that the UMAT subroutine is working as intended.

A shearing simulation is shown in Fig. 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 Fig. 7. The Supplement shows a simulation with the GOH fit. As discussed in the previous section, the neural network model with the augmented data are, in a way, the best extension of the GOH model: it retains some of the expert model features but does not suffer from the constraints of an explicit functional form.

The last simulation in Fig. 7 is a torsional loading scenario. In this simulation the -x surface of the rectangular prism is clamped and a rotation boundary condition of $UR_x = 1$ *rad* is imposed on the +x surface. The resulting stresses are presented in Fig. 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 Fig. 7 showcase the robustness and versatility of our DNN UMAT. It should also be noted that the DNN material model usually requires approximately twice as much computational time compared to the built-in GOH model. For example, for the torsion problem, the execution time for GOH is 00:01:42, while for the DNN UMAT it is 00:02:24.

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 [42]. The domain is a $10 \times 10 \times 0.3$ cm^3 patch of skin modeled with 3200 brick elements. A rectangular expander of dimensions 8×8 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 Fig. 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 [44]. The simulation in Fig. 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 Fig. 8 evidences the anisotropy of the model. The fiber directions **v** and **w** are aligned

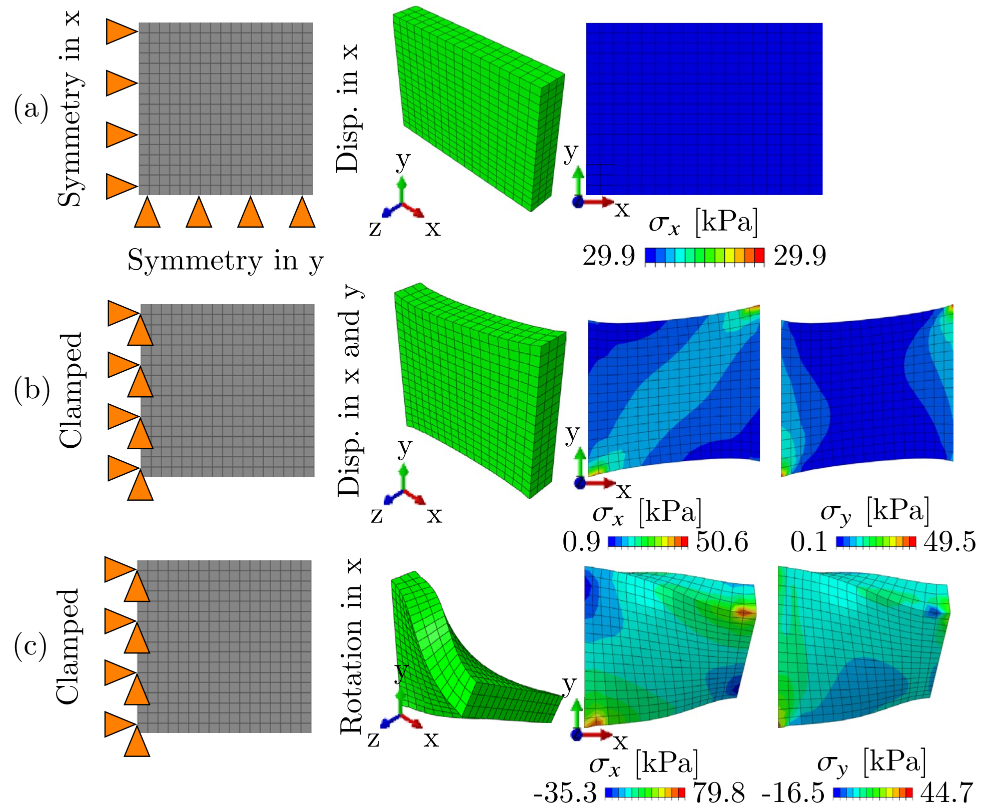

Fig. 7 Finite element method simulations using the DNN material model in UMAT. Boundary conditions, deformed geometry and contours of σ_x under uniaxial loading (**a**), Boundary conditions, deformed geometry and contours of σ_x and σ_y under shear loading (**b**), and, Boundary conditions, deformed geometry and contours of σ_x and σ_y under torsional loading (**c**)

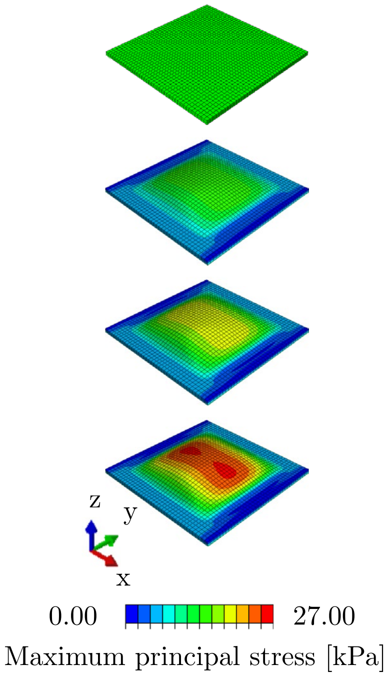

Fig. 8 Finite element method simulations of tissue expansion using the DNN material model in UMAT. From top to bottom: Undeformed geometry, and, contours of maximum principal stress on deformed geometry after the expander is expanded to 20, 40 and 60 cm^3, respectively

with the Cartesian basis [1, 0, 0 and [0, 1, 0]. The tissue is stiffer in the **v** direction, which is why there is a band of higher stress along that direction in Fig. 8. To further showcase the anisotropy in the deformation we also plot the corresponding strains (Fig. 9).

4 Discussion

In this study we propose a deep neural network (DNN) material model to replace conventional constitutive equations for nonlinear materials, in particular soft collagenous tissues. The neural network takes isochoric strain invariants as inputs and produces the isochoric strain energy and its derivatives 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 [45–47]. For instance, additional loss functions to deal with the violation of objectivity have been proposed [48]. Efforts using invariants or principal stretches as inputs with energy as the output have also been shown by others [19, 49, 50], and by us as well but for isotropic materials [18]. Intermediate approaches that map deformation invariants to principal stresses have also been sought, but they are limited to isotropic materials and even in that case require regularization schemes [51]. The key ideas introduced in this paper are the consideration of anisotropy, training with multi-fidelity data -including experimental data-, polyconvexity constraints, and design of the DNN 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 almost perfectly (Fig. 2). This is not surprising since neural networks are universal approximators [52]. Other work using neural network material models have also shown excellent performance against synthetic data [53]. When enough data are available, recent efforts in data-driven computational mechanics have shown that model-free approaches can be used [47, 54]. However, for applications in biomechanics, the anisotropy and high nonlinearity in the materials necessitates large amounts of data from deformations that can cover the entire input space [55, 56]. This is often out of reach soft tissue characterization. Thus, we propose a DNN model that captures the experimental data, but does so constrained by a hyperelastic framework and the condition of polyconvexity of the strain energy.

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 [24]. Two other tests, strip-biaxial tests in the x- and y-direction are explored here as well. Liu et al. [19] generated datasets by subjecting

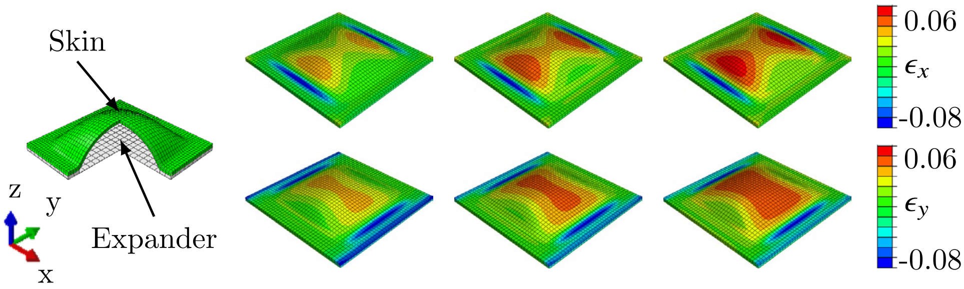

Fig. 9 Finite element method simulations of tissue expansion using the DNN material model in the UMAT. From left to right: model setup, and, contours of strain on deformed geometry after the expander is expanded to 20, 40 and 60 cm^3, respectively

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. Within the hyperelastic framework, expert models of soft tissue mechanics have been developed over the past few decades and reflect our growing understanding of soft tissues. For example, expert models are often based on microstructure observations [9, 57], satisfy physics constraints [21, 35], and have been carefully designed based on observations of many data [7]. On the other hand, expert models have many 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 [8]. 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 with great accuracy, while maintaining relatively good performance in regions with scarce high fidelity data. Of course, this raises the question of how to balance between the two. Here we set a much higher priority for the experimental data, but future studies should quantify the uncertainty of both the data and the models in regions with little experimental data in order to rigorously weight the high and low fidelity models. With the current weights, we showcase the ability of the DNN material model to capture the mechanical response of skin based on data obtained from 2 pigs and 11 mice, demonstrating the applicability of our approach to realistic datasets.

Imposing polyconvexity through the loss function ensures a stable material model suitable for finite element applications, but comes at the expense of fitting error (see Fig. 4). This result points to polyconvexity as a potentially restrictive condition on the data, possibly due to the existence of dissipative phenomena such as viscoelasticity or damage which were not accounted for in the model [2, 58]. Noise can also affect the performance of the machine learning approaches [59]. The Supplement shows that imposing the convexity constraint, the DNN model can capture the synthetic data even in the presence of noise. Other data-driven approaches have considered different convexity constraints. For example, Vlassis et al. [33] check for convexity of the strain energy with respect to the right Cauchy Green deformation tensor $\mathbf{C}$. While convexity with respect to $\mathbf{C}$ is widely used [34], it is not equivalent to polyconvexity with respect to $\mathbf{F}$ [60]. Polyconvexity of the strain energy, together with some growth conditions on the energy, ensures the existence of minimizers for boundary value problems in elasticity [31]. While it is true that polyconvexity is a sufficient but not a necessary condition, it provides enough flexibility, is compatible with phenomena such as buckling [61], and is desirable for finite element implementation. Data-driven work enforcing polyconvexity have also been explored with different approaches by us and others [53, 62].

A strong motivation behind the development of data-driven constitutive models of soft tissues based on experimental tissue testing data is to use the model in predictive finite element simulations to guide device design or treatment planning [63]. Previous work on data driven modeling has fallen short in this regard [19, 53, 64]. The DNN design shown here, including the use of invariants as inputs and prediction of energy and energy derivatives as outputs, allows us to compute not only the stresses but the consistent tangent. Together with the polyconvexity loss, our data-driven framework is uniquely suited for finite element simulations. While it would be possible to predict the energy alone, this introduces noise in the derivatives that need to be regularized as shown in [33]. An alternative framework is to use integrable neural networks [22]. Another method we have explored recently is the use of neural ordinary differential equations to learn the energy derivatives, ignoring the underlying energy function entirely [62]. In [62], the model architecture guarantees that the derivative functions do indeed come from differentiation of an underlying potential even if this potential is not explicitly modeled. The approach followed here is more akin to multifield formulations in elasticity or enhanced strain methods, for which additional degrees of freedom are added together with suitable constraints [65]. We implemented the DNN 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 provided to the UMAT through the input file. We showcased finite element simulations with the neural network trained on the porcine data, 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.

5 Conclusion

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 are 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 are 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 DNN UMAT subroutine for Abaqus, a widely used finite element package.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00366-022-01733-3.

Acknowledgements This work was supported by the National Institute of Arthritis and Musculoskeletal and Skin Diseases, National Institutes of Health, United States under award R01AR074525 to Adrian Buganza Tepole and the National Science Foundation through awards 1916663 and 2127925 to Manuel K. Rausch as well as 1916665 to Adrian Buganza Tepole.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Lee T, Turin SY, Stowers C, Gosain AK, Tepole AB (2021) Personalized computational models of tissue-rearrangement in the scalp predict the mechanical stress signature of rotation flaps. The Cleft Palate-Craniofacial J 58(4):438–445
2. Sherman V, Tang Y, Zhao S, Yang W, Meyers M (2017) Structural characterization and viscoelastic constitutive modeling of skin. Acta Biomaterialia 53:460–469
3. Kakaletsis S, Meador WD, Mathur M, Sugerman GP, Jazwiec T, Malinowski M, Lejeune E, Timek TA, Rausch MK (2021) Right ventricular myocardial mechanics: multi-modal deformation, microstructure, modeling, and comparison to the left ventricle. Acta Biomaterialia 123:154–166
4. Meador WD, Mathur M, Sugerman GP, Jazwiec T, Malinowski M, Bersi MR, Timek TA, Rausch MK (2020) A detailed mechanical and microstructural analysis of ovine tricuspid valve leaflets. Acta Biomaterialia 102:100–113
5. Holzapfel GA (2000) Nonlinear solid mechanics. John Wiley Sons LTD, A Continuum Approach for Engineering
6. Gasser TC, Ogden RW, Holzapfel GA (2005) Hyperelastic modelling of arterial layers with distributed collagen fibre orientations. J R Soc Interface 3(6):15–35
7. Humphrey JD, Strumpf RK, Yin FCP (1990) Determination of a constitutive model for passive myocardium: I. a new functional form. J Biomech Eng 112:333–339
8. Tonge TK, Voo LM, Nguyen TD (2013) 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(4):5926–5942. https://doi.org/10.1016/j.actbio.2012.11.034
9. Limbert G (2019) Skin Biophysics: from experimental characterisation to advanced modelling vol. 22. Springer
10. Jor JW, Parker MD, Taberner AJ, Nash MP, Nielsen PM (2013) Computational and experimental characterization of skin mechanics: identifying current challenges and future directions. Wiley Interdiscipl Rev Syst Biol Med 5(5):539–556
11. Mueller B, Elrod J, Distler O, Schiestl C, Mazza E (2021) On the reliability of suction measurements for skin characterization. J Biomech Eng 143(2):021002
12. Peng GC, Alber M, Buganza Tepole A, Cannon WR, De S, Dura-Bernal S, Garikipati K, Karniadakis G, Lytton WW, Perdikaris P et al (2021) Multiscale modeling meets machine learning: What can we learn? Arch Comput Methods Eng 28(3):1017–37
13. Han Z, De S et al (2019) A deep learning-based hybrid approach for the solution of multiphysics problems in electrosurgery. Comput Methods Appl Mech Eng 357:112603
14. Zhang X, Garikipati K (2020) Machine learning materials physics: Multi-resolution neural networks learn the free energy and nonlinear elastic response of evolving microstructures. Comput Methods Appl Mech Eng 372:113362. https://doi.org/10.1016/j.cma.2020.113362
15. Vlassis NN, Sun W (2021) Sobolev training of thermodynamic-informed neural networks for interpretable elasto-plasticity models with level set hardening. Comput Methods Appl Mech Eng 377:113695
16. Lu L, Dao M, Kumar P, Ramamurty U, Karniadakis GE, Suresh S (2020) Extraction of mechanical properties of materials through deep learning from instrumented indentation. Proc Natl Acad Sci USA 117(3):7052–7062
17. Lejeune E, Zhao B (2021) Exploring the potential of transfer learning for metamodels of heterogeneous material deformation. J Mech Behav Biomed Mater 117:104276. https://doi.org/10.1016/j.jmbbm.2020.104276
18. Leng Y, Calve S, Tepole AB (2021) Predicting the mechanical properties of fibrin using neural networks trained on discrete fiber network data. arXiv preprint arXiv:2101.11712
19. Liu M, Liang L, Sun W (2020) A generic physics-informed neural network-based constitutive model for soft biological tissues. Comput Methods Appl Mech Eng 372:113402
20. Reimann D, Chandra K, Vajragupta N, Glasmachers T, Junker P, Hartmaier A et al (2019) Modeling macroscopic material behavior with machine learning algorithms trained by micromechanical simulations. Front Mater 6:181
21. Ehret AE, Itskov MA (2007) polyconvex hyperelastic model for fiber-reinforced materials in application to soft tissues. J Mater Sci 42(21):8853–8863
22. Teichert GH, Natarajan A, Van der Ven A, Garikipati K (2019) Machine learning materials physics: integrable deep neural networks enable scale bridging by learning free energy functions. Comput Methods Appl Mech Eng 353:201–216
23. Nguyen LTK, Keip M-A (2018) A data-driven approach to nonlinear elasticity. Comput Struct 194:97–115
24. Meador WD, Sugerman GP, Story HM, Steifert AW, Bersi MR, Tepole AB, Rausch MK (2020) The regional-dependent biaxial behavior of young and aged mouse skin: a detailed histomechanical characterization, residual strain analysis, and constitutive model. Acta Biomaterialia 101:403–413
25. Lee T, Turin SY, Gosain AK, Bilionis I, Tepole AB (2018) Propagation of material behavior uncertainty in a nonlinear finite element model of reconstructive surgery. Biomech Model Mechanobiol 17:1857–1873
26. Bonfiglio L, Perdikaris P, Brizzolara S (2020) Multi-fidelity Bayesian optimization of swath hull forms. J Ship Res 64(2):154–170
27. Smith M (2009) ABAQUS/Standard User's Manual, Version 6.9. Dassault Systèmes Simulia Corp, United States
28. Doyle TC, Ericksen JL (1956) Nonlinear elasticity. Adv Appl Mech 4:53–115

29. Coleman BD, Walter N (1974) The foundations of mechanics and thermodynamics. Springer
30. Fehervary H, Maes L, Vastmans J, Kloosterman G, Famaey N (2020) How to implement user-defined fiber-reinforced hyperelastic materials in finite element software. J Mech Behav Biomed Mater 110:103737. https://doi.org/10.1016/j.jmbbm.2020.103737
31. Ball JM (1976) Convexity conditions and existence theorems in nonlinear elasticity. Arch Rational Mech Anal 63(4):337–403
32. Schröder J (2010) Anisotropic polyconvex energies. In: Poly-, quasi-and rank-one convexity in applied mechanics, pp 53–105. Springer
33. Vlassis NN, Ma R, Sun W (2020) Geometric deep learning for computational mechanics part i: anisotropic hyperelasticity. Comput Methods Appl Mech Eng 371:113299. https://doi.org/10.1016/j.cma.2020.113299
34. Gao DY, Neff P, Roventa I, Thiel C (2017) On the convexity of nonlinear elastic energies in the right cauchy-green tensor. J Elasticity 127(2):303–308
35. Holzapfel GA, Gasser TC, Ogden RW (2000) A new constitutive framework for arterial wall mechanics and a comparative study of material models. J Elast Phys Sci Solids 61(1):1–48
36. Schröder J, Neff P, Balzani D (2005) A variational approach for materially stable anisotropic hyperelasticity. Int J Solids Struct 42(15):4352–4371
37. Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press
38. Prussing JE (1985) The principal minor test for semidefiniteness matrices. J Guidance Control Dynm 9(1):121–122
39. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980
40. Chollet F et al (2018) Keras: The python deep learning library. ascl, 1806
41. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, et al. (2016) Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467
42. Lee T, Turin SY, Gosain AK, Bilionis I, Tepole AB (2018) Propagation of material behavior uncertainty in a nonlinear finite element model of reconstructive surgery. Biomech Model Mechanobiol 17:1857–1873
43. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Millman KJ, Mayorov N, Nelson ARJ, Jones E, Kern R, Larson E, Carey CJ, Polat İ, Feng Y, Moore EW, VanderPlas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero EA, Harris CR, Archibald AM, Ribeiro AH, Pedregosa F, van Mulbregt P (2020) SciPy 1.0 Contributors SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat Methods 17:261–272. https://doi.org/10.1038/s41592-019-0686-2
44. Tepole AB, Gart M, Gosain AK, Kuhl E (2014) Characterization of living skin using multi-view stereo and isogeometric analysis. Acta Biomaterialia 10(11):4822–4831
45. Ghaboussi J, Pecknold DA, Zhang M, Haj-Ali RM (1998) Autoprogressive training of neural network constitutive models. Int J Num Methods Eng 42(1):105–126
46. Ghaboussi J, Sidarta DE (1998) New nested adaptive neural networks (nann) for constitutive modeling. Comput Geotech 22(1):29–52. https://doi.org/10.1016/S0266-352X(97)00034-7
47. Kirchdoerfer T, Ortiz M (2016) Data-driven computational mechanics. Comput Methods Appl Mech Eng 304:81–101
48. Heider Y, Wang K, Sun W (2020) So (3)-invariance of informed-graph-based deep neural network for anisotropic elastoplastic materials. Comput Methods Appl Mech Eng 363:112875
49. Mihai LA, Woolley TE, Gorieli A (2018) Stochastic isotropic hyperelastic materials: constitutive calibration and model selection. Proc R Society A 474(2211):20170858
50. Dabiri Y, Van der Velden A, Sack KL, Choy JS, Kassab GS, Guccione JM (2019) Prediction of left ventricular mechanics using machine learning. Front Phys 7:117. https://doi.org/10.3389/fphy.2019.00117
51. Kalina KA, Linden L, Brummund J, Metsch P, Kästner M (2022) Automated constitutive modeling of isotropic hyperelasticity based on artificial neural networks. Comput Mech 69(1):213–232
52. Leshno M, Lin VY, Pinkus A, Schocken S (1993) Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. Neural Netw 6(6):861–867
53. Klein DK, Fernández M, Martin RJ, Neff P, Weeger O (2022) Polyconvex anisotropic hyperelasticity with neural networks. J Mech Phys Solids 159:104703
54. Mora-Macías J, Ayensa-Jiménez J, Reina-Romo E, Doweidar MH, Domínguez J, Doblaré M, Sanz-Herrera JA (2020) A multiscale data-driven approach for bone tissue biomechanics. Comput Methods Appl Mech Eng 368:113136
55. He Q, Laurence DW, Lee C-H, Chen J-S (2021) Manifold learning based data-driven modeling for soft biological tissues. J Biomech 117:110124
56. He X, He Q, Chen J-S (2021) Deep autoencoders for physics-constrained data-driven nonlinear materials modeling. Comput Methods Appl Mech Eng 385:114034
57. Kassab GS, Sacks MS (2016) Structure-based mechanics of tissues and organs. Springer
58. Kumaraswamy N, Khatam H, Reece GP, Fingeret MC, Markey MK, Ravi-Chandar K (2017) Mechanical response of human female breast skin under uniaxial stretching. J Mech Behav Biomed Mater 74:164–175
59. Wang Z, Estrada JB, Arruda EM, Garikipati K (2021) Inference of deformation mechanisms and constitutive response of soft material surrogates of biological tissue by full-field characterization and data-driven variational system identification. J Mech Phys Solids 153:104474
60. Sivaloganathan J, Spector SJ (2018) On the uniqueness of energy minimizers in finite elasticity. J Elasticity 133(1):73–103
61. Bonet J, Gil AJ, Ortigosa RA (2015) computational framework for polyconvex large strain elasticity. Comput Methods Appl Mech Eng 283:1061–1094
62. Tac V, Sahli Costabal F, Tepole AB (2022) Data-driven tissue mechanics with polyconvex neural ordinary differential equations. Comput Methods in Appl Mech and Eng. 398:115248
63. Baillargeon B, Rebelo N, Fox DD, Taylor RL, Kuhl E (2014) The living heart project: a robust and integrative simulator for human heart function. Eur J Mech-A/Solids 48:38–47
64. Cilla M, Pérez-Rey I, Martínez MA, Peña E, Martínez J (2018) On the use of machine learning techniques for the mechanical characterization of soft biological tissues. Int J Num Methods Biomed Eng 34(10):3121
65. Wriggers P, Korelc J (1996) On enhanced strain methods for small and finite deformations of solids. Comput Mech 18(6):413–428

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.