ORIGINAL PAPER

Benchmarking physics-informed frameworks for data-driven hyperelasticity

Vahidullah Taç[1] · Kevin Linka[2] · Francisco Sahli-Costabal[3] · Ellen Kuhl[2] · Adrian Buganza Tepole[1,4]

Abstract

Data-driven methods have changed the way we understand and model materials. However, while providing unmatched flexibility, these methods have limitations such as reduced capacity to extrapolate, overfitting, and violation of physics constraints. Recently, frameworks that automatically satisfy these requirements have been proposed. Here we review, extend, and compare three promising data-driven methods: Constitutive Artificial Neural Networks (CANN), Input Convex Neural Networks (ICNN), and Neural Ordinary Differential Equations (NODE). Our formulation expands the strain energy potentials in terms of sums of convex non-decreasing functions of invariants and linear combinations of these. The expansion of the energy is shared across all three methods and guarantees the automatic satisfaction of objectivity, material symmetries, and polyconvexity, essential within the context of hyperelasticity. To benchmark the methods, we train them against rubber and skin stress–strain data. All three approaches capture the data almost perfectly, without overfitting, and have some capacity to extrapolate. This is in contrast to unconstrained neural networks which fail to make physically meaningful predictions outside the training range. Interestingly, the methods find different energy functions even though the prediction on the stress data is nearly identical. The most notable differences are observed in the second derivatives, which could impact performance of numerical solvers. On the rich data used in these benchmarks, the models show the anticipated trade-off between number of parameters and accuracy. Overall, CANN, ICNN and NODE retain the flexibility and accuracy of other data-driven methods without compromising on the physics. These methods are ideal options to model arbitrary hyperelastic material behavior.

Keywords Physics-informed machine learning · Polyconvexity · Nonlinear mechanics · Neural networks · Constitutive models

Introduction

The frontier of biomedical engineering applications such as personalized surgery requires accurate mathematical models of material-specific behavior [1]. Similarly, human-engineered systems based on soft materials also necessitate predictive simulations with high precision [2]. The materials

✉ Adrian Buganza Tepole
  abuganza@purdue.edu

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

2 Department of Mechanical Engineering, Stanford University, Stanford, USA

3 Department of Mechanical and Metallurgical Engineering, Institute for Biological and Medical Engineering, Pontificia Universidad Católica de Chile, Santiago, Chile

4 Weldon School of Biomedical Engineering, Purdue University, West Lafayette, USA

for these applications are extremely nonlinear and undergo large deformations, e.g. rubber and skin. Yet, despite decades of effort developing constitutive equations for these materials, there still isn't a definitive model for them due to inherent limitations of expert-constructed models [3]. Traditional material modeling restricts the prediction of the mechanical response to a narrow set of functional terms, making it nearly impossible to fully capture the data. On the other hand, data-driven methods such as neural networks are universal approximators that can fit mechanical behavior data of complex response almost perfectly [4, 5]. Data-driven methods, on the other hand, have their own drawbacks such as lack of explainability, overfitting, and difficulty in extrapolation to unseen strain states outside the convex hull of the training data. Considering the response of rubbers and many biological materials as hyperelastic, the mechanical response is fully specified by a scalar potential that has to satisfy the conditions of objectivity and polyconvexity [6]. These physics

constraints are crucial for enabling robust large-scale simulations, extrapolate from data, and avoid over-fitting [7]. Yet, off-the-shelf machine learning tools cannot a priori satisfy these conditions. We review and refine three very recent developments in physics-informed machine learning that aim at embedding the objectivity, material symmetries, and polyconvexity constraints as part of the formulation such that they can be satisfied *a priori*. The methods we consider are three: Constitutive Artificial Neural Networks models (CANN) [8], Neural Ordinary Differential Equations constitutive models (NODE) [9], and Input Convex Neural Network constitutive models (ICNN) [10–12].

Characterizing nonlinear materials like rubber and skin involves testing them under a wide set of deformation modes such as uniaxial, biaxial, shear, and sometimes triaxial deformation. The resulting data collected from these tests are stress–strain curves. The direct approach to leverage machine learning on these data is to directly map between strains and stresses [13]. One problem with this strategy is that objectivity and material symmetries are not necessarily preserved. One way of fixing this issue has been to augment the data with superimposed arbitrary rotations [10, 14]. For closed-form constitutive models, fulfilling objectivity and material symmetries has not been a major hurdle. Experts develop strain energies in terms of invariants of the deformation to satisfy these requirements by default [15, 16]. Machine learning methods along these lines have also been proposed [17–20].

The challenge of imposing polyconvexity in data-driven methods is more difficult to address. Polyconvexity of the strain energy (with additional growth conditions) is a sufficient condition for the existence of solutions for boundary value problems in hyperelasticity [21]. Polyconvexity is a flexible framework, compatible with unstable material behavior like buckling. This is in contrast with more restrictive notions like convexity of the strain energy with respect to the deformation gradient, which can violate objectivity [22]. Laxer conditions such as rank-one convexity are weaker than polyconvexity and not sufficient for the existence of global minimizers of the strain energy [23]. Many expert models are based on the notion of polyconvexity (but there are also many examples of popular models which might violate this condition [24]). Different notions of convexity have been considered within data-driven frameworks, but the majority have opted for adding the constraint as a penalty through the loss function [18]. Imposing convexity through the loss function has had some success but faces the challenge of carefully balancing between imposing the constraint and achieving a higher accuracy [7, 25]. Another data-driven approach for modeling arbitrary materials is to select the best model out of a wide library of available models [26, 27]. CANN, ICNN and NODE models are recently developed data-driven frameworks that automatically satisfy the polyconvexity condition [8–10].

CANNs are a new method for automated model discovery that borrow their architecture from traditional feed-forward neural networks but use activation functions that have been used previously in expert-constructed models and that preserve convexity. They also prune the connections between the inputs and subsequent nodes such that the final result is a polyconvex strain energy that encompasses a large family of existing closed-form hyperelastic models [8]. ICNNs are a broader class of feed-forward neural networks that can be used to construct convex function by using convex activation functions and positive weights [28]. ICNN constitutive models build convex functions of invariant inputs [10–12]. Different ICNN models in the literature introduce particular choices for the family of functions that can be interpolated. NODEs tackle the problem differently by leveraging the monotonicity of trajectories of ordinary differential equations (ODEs) in a single variable to interpolate monotic functions associated with derivatives of a strain energy rather than the energy directly [9]. Unlike other approaches, CANNs, ICNNs, and NODEs have physics at the core of the formulation to generate constitutive models that satisfy objectivity and polyconvexity a priori. Yet, there is a gap in our understanding of how these different methods perform on benchmark datasets, and a general need to benchmark machine learning methods in computational mechanics [29, 30].

Rubber modeling was the center of attention for large deformation hyperelasticity in the past century, with tens of constitutive models proposed [31]. Recently, advances in soft robotics has renewed the interest in developing improved high-fidelity simulations of soft robots made of rubbers and other elastomers. For example, applications that aim at produce complex motion such as tentacle grippers, walking soft robots, and rehabilitation soft exoskeletons [32], all require precise modeling of the material response.

Soft tissues made of collagen have remarkable mechanical properties. They show exponential-like stress–strain response and anisotropy. These nonlinearities allow tissues like skin to protect us against environmental harm while allowing interaction and movement [33]. The development of constitutive models for soft tissues, and skin in particular, dates to the seminal work by Lanir and Fung [34, 35], and has resulted in a long list of strain energy functions proposed over the past five decades [3]. New models are being proposed even today [36, 37]. Despite the rich literature on skin and soft tissue modeling, the complexity of the material response in these materials has prevented the emergence of a categorically superior constitutive model.

The manuscript is organized as follows. In the Methods section we first review the basic equations that describe the mechanical behavior of hyperelastic materials with emphasis on strain energy function expansions that satisfy objectivity and polyconvexity requirements. Then, we show how CANN,

ICNN and NODE architectures can be used to create material models within the considered families of elastic potentials. After training the three methods to datasets on rubber and skin, the Results section explores in detail the ability of the models to interpolate and extrapolate, their robustness with respect to model initialization, the regularity of second derivatives of the energy, and the trade-off between number of parameters and model accuracy. We finally discuss the results in the context of other data-driven efforts for computational mechanics.

Methods

Polyconvex strain energy density functions

Consider a motion $\boldsymbol{\varphi}$, the gradient $\mathbf{F} = \nabla\boldsymbol{\varphi}$ contains all the local information about the deformation. Within the framework of hyperelasticity, the strain energy function $\psi(\mathbf{F})$ fully defines the material response. Polyconvexity implies that the energy $\psi(\mathbf{F})$ can be expressed as a convex function in the extended domain $\hat{\psi}(\mathbf{F}, \text{cof}\,\mathbf{F}, \det\mathbf{F})$. Intuitively, this extended domain covers different modes of deformation: $\mathbf{F}$ measures changes in length, $\text{cof}\,\mathbf{F}$ changes in area, and $J = \det\mathbf{F}$ changes in volume. It is difficult to work directly with the deformation gradient and its cofactors as inputs to the strain energy. Instead, the right Cauchy-Green deformation tensor $\mathbf{C} = \mathbf{F}^{\top}\mathbf{F}$ is used because it does not contain information about superimposed rigid body rotations. Furthermore, material symmetries are enforced by working with the invariants

$$
\begin{aligned}
I_1 &= \text{tr}\,\mathbf{C} = \mathbf{C} : \mathbf{I} \\
I_2 &= \frac{1}{2}\left((\text{tr}\,\mathbf{C})^2 - \text{tr}\,\mathbf{C}^2\right) \\
I_3 &= J^2 = \det\mathbf{C} \\
I_{4a} &= \mathbf{C} : \mathbf{a}_0 \otimes \mathbf{a}_0, \quad I_{4s} = \mathbf{C} : \mathbf{s}_0 \otimes \mathbf{s}_0.
\end{aligned} \tag{1}
$$

The last two invariants in Eq. (1) are only relevant for transversely anisotropic materials and depend on the deformation of two material unit vectors $\mathbf{a}_0, \mathbf{s}_0$. For soft tissues, the vectors $\mathbf{a}_0, \mathbf{s}_0$ represent collagen fiber bundle orientations. Furthermore, for nearly incompressible materials such as rubbers and skin, the split between volumetric and isochoric parts is often used. The isochoric part of the deformation is $\bar{\mathbf{F}} = J^{-1/3}\mathbf{F}$, with the corresponding deformation tensor $\bar{\mathbf{C}} = \bar{\mathbf{F}}^{\top}\bar{\mathbf{F}}$. The isochoric invariants follow

$$
\begin{aligned}
\bar{I}_1 &= J^{-2/3} I_1 \\
\bar{I}_2 &= J^{-4/3} I_2 \\
\bar{I}_{4a} &= J^{-2/3} I_{4a}, \quad \bar{I}_{4s} = J^{-2/3} I_{4s}.
\end{aligned} \tag{2}
$$

Based on the split between the isochoric and volumetric parts of the deformation, the energy can be additively decomposed into

$$
\psi = \psi_{\text{iso}}(\bar{I}_1, \bar{I}_2, \bar{I}_{4a}, \bar{I}_{4s}) + \psi_{\text{vol}}(J). \tag{3}
$$

For polyconvexity to be satisfied in this additive split, one requirement is convexity of ψ_{vol} and growth conditions $\psi_{\text{vol}} \to \infty$ as $J \to 0$ or $J \to \infty$. In the case of fully incompressible materials, the volumetric part of the strain energy is replaced by $p(J-1)$, where p is a Lagrange multiplier field that enforces $J = 1$. In simple loading cases such as uniaxial or biaxial deformation, p can be directly determined from boundary conditions. In addition, for incompressible behavior the isochoric part of the energy becomes a function of the original invariants defined in Eq. (1).

To ensure polyconvexity of ψ_{iso}, recall that this condition implies a function $\hat{\psi}$ convex on $(\mathbf{F}, \text{cof}\,\mathbf{F}, \det\mathbf{F})$. The invariant I_1 is convex in $\mathbf{F}$, while I_2 is convex in $\text{cof}\,\mathbf{F}$. The anisotropic invariants I_{4a}, I_{4s} are also convex on $\mathbf{F}$. Moreover, the isochoric split preserves polyconvexity of $\bar{I}_1$, $\bar{I}_{4a}$, $\bar{I}_{4s}$, and a simple scaling with a power of J is enough to maintain polyconvexity of $\bar{I}_2$. Thus, a sufficient large family of polyconvex functions has the form

$$
\psi = \psi_1(\bar{I}_1) + \psi_2(\bar{I}_2) + \psi_{4a}(\bar{I}_{4a}) + \psi_{4s}(\bar{I}_{4s}) + \psi_{vol}(J), \tag{4}
$$

with each of the ψ_i a convex non decreasing function of its argument, while, as mentioned previously, ψ_{vol} has to be convex and grow to infinity appropriately with changes in J. Again, for incompressibility, the last term in (4) is replaced with the Lagrange multiplier constraint, and the ψ_i terms can be considered as functions of the invariants in Eq. (1).

Stress predictions from a strain energy potential

Given a strain energy function, the second Piola–Kirchhoff stress follows from the standard Coleman–Noll procedure [6],

$$
\mathbf{S} = 2\frac{\partial\psi}{\partial\mathbf{C}}. \tag{5}
$$

Other stress tensors can be easily obtained with push-forward operations, for instance the nominal or first Piola–Kirchhoff stress $\mathbf{P} = \mathbf{F}\mathbf{S}$, or the Cauchy stress $\boldsymbol{\sigma} = J^{-1}\mathbf{F}\mathbf{S}\mathbf{F}^{\top}$, which appear in the strong form of linear momentum balance in the reference or deformed configurations respectively. Since the energy is in terms of the invariants,

computing the stress requires the standard derivatives

$$\frac{\partial I_1}{\partial \mathbf{C}} = \mathbf{I}$$
$$\frac{\partial I_2}{\partial \mathbf{C}} = \frac{1}{2}(I_1 \mathbf{I} - \mathbf{C})$$
$$\frac{\partial I_3}{\partial \mathbf{C}} = I_3 \mathbf{C}^{-1} \tag{6}$$
$$\frac{\partial I_{4a}}{\partial \mathbf{C}} = \mathbf{a}_0 \otimes \mathbf{a_0} \,, \quad \frac{\partial I_{4s}}{\partial \mathbf{C}} = \mathbf{s}_0 \otimes \mathbf{s_0} \,.$$

The same derivatives as in Eq. (6) apply to the the derivatives of the isochoric invariants with respect to $\bar{\mathbf{C}}$. However, when using the split into volumetric and isochoric parts, we need the projection

$$\frac{\partial \bar{\mathbf{C}}}{\partial \mathbf{C}} = \mathbb{P} = J^{-2/3} \left(\mathbb{I} - \frac{1}{3} \mathbf{C} \otimes \mathbf{C}^{-1} \right) . \tag{7}$$

The tensor $\mathbb{I}$ in Eq. (7) denotes the fourth order identity. Bringing it all together, the second Piola–Kirchhoff stress takes the form

$$\mathbf{S} = 2 \frac{\partial \psi}{\partial \mathbf{C}} = 2 \frac{\partial \psi}{\partial \bar{\mathbf{C}}} : \mathbb{P} + \mathbf{S}_{\text{vol}} = \bar{\mathbf{S}} : \mathbb{P} + \mathbf{S}_{\text{vol}} \,, \tag{8}$$

with

$$\bar{\mathbf{S}} = 2 \frac{\partial \psi_1}{\partial \bar{I}_1} \mathbf{I} + 2 \frac{\partial \psi_2}{\partial \bar{I}_2} (\bar{I}_1 \mathbf{I} - \bar{\mathbf{C}}^{-1}) + 2 \frac{\partial \psi_{4a}}{\partial \bar{I}_{4a}} \mathbf{a}_0 \otimes \mathbf{a}_0$$
$$+ 2 \frac{\partial \psi_{4s}}{\partial \bar{I}_{4s}} \mathbf{s}_0 \otimes \mathbf{s}_0. \tag{9}$$

It is possible to extend Eq. (4) to capture even a wider class of materials. Convex linear combinations of the invariants maintain polyconvexity with respect to $\mathbf{F}$. Therefore, in addition to the invariants in Eq. (1) or their isochoric counterparts Eq. (2), we can consider the mixed invariants

$$K_{ij} = \alpha_{ij} I_i + (1 - \alpha_{ij}) I_j \,, \tag{10}$$

and the corresponding isochoric versions $\bar{K}_{ij}$. The family of strain energies considering these mixed terms has the following structure

$$\psi = \sum \psi_i(\bar{I}_i) + \sum \psi_{ij}(\bar{K}_{ij}) + \psi_{vol}(J) \,. \tag{11}$$

The expression for $\bar{\mathbf{S}}$ in this more general case is analogous to Eq. (9) but with additional terms to account for the ψ_{ij} contributions.

Uniaxial, pure shear, and biaxial loading

For the specific case of isotropic uniaxial deformation of a perfectly incompressible material, the deformation depends on the single stretch λ, and the stress in the direction of the applied stretch is

$$P = 2(\lambda - \lambda^{-2}) \left(\frac{\partial \psi}{\partial I_1} + \frac{1}{\lambda} \frac{\partial \psi}{\partial I_2} \right) . \tag{12}$$

For pure shear deformation of a wide but thin specimen, the nominal stress in the direction of the applied stretch λ is

$$P = 2(\lambda - \lambda^{-3}) \left(\frac{\partial \psi}{\partial I_1} + \frac{\partial \psi}{\partial I_2} \right) . \tag{13}$$

The third loading case of interest for thin incompressible isotropic materials is equibiaxial tension defined by the stretch λ. For this loading, the nominal stress in the two principal directions of applied stretch is the same and equal to

$$P = 2(\lambda - \lambda^{-5}) \left(\frac{\partial \psi}{\partial I_1} + \lambda^2 \frac{\partial \psi}{\partial I_2} \right) . \tag{14}$$

Lastly we consider an incompressible transversely anisotropic material under arbitrary biaxial loading specified by the two stretches λ_x, λ_y. Without loss of generality we set $\mathbf{a}_0 = [1, 0, 0]$, $\mathbf{s}_0 = [0, 1, 0]$. The in plane nominal stresses are

$$P_{xx} = \frac{\partial \psi}{\partial I_1} \lambda_x + \frac{\partial \psi}{\partial I_2} (I_1 \lambda_x - \lambda_x^3) + \frac{\partial \psi}{\partial I_{4a}} \lambda_x - p \lambda_x^{-1}$$
$$P_{yy} = \frac{\partial \psi}{\partial I_1} \lambda_y + \frac{\partial \psi}{\partial I_2} (I_1 \lambda_y - \lambda_y^3) + \frac{\partial \psi}{\partial I_{4s}} \lambda_y - p \lambda_y^{-1} \,, \tag{15}$$

with the pressure Lagrange multiplier solved from the plane stress condition

$$p = \frac{\partial \psi}{\partial I_1} \lambda_z^2 + \frac{\partial \psi}{\partial I_2} (I_1 \lambda_z^2 - \lambda_z^4) \,, \tag{16}$$

and the normal stretch obtained from the incompressibility constraint

$$\lambda_z = \frac{1}{\lambda_x \lambda_y} \,. \tag{17}$$

CANN models

To construct convex non-decreasing functions to represent the energy in Eq. (11), one way is to borrow from the architecture of feed forward neural networks but using only convex non-decreasing activation functions on a polynomial expansion. The method is illustrated in Fig. 1a. Starting from $\mathbf{F}$, the invariants Eq. (1) are computed in a pre-processing step. For ease of implementation and to improve the optimization step during model training, consider the normalized invariants

$$\hat{I}_i = (I_i - a_i)/b_i \tag{18}$$

(a) Constitutive Artificial Neural Networks

(b) Input Convex Neural Networks

(c) Neural Ordinary Differential Equations
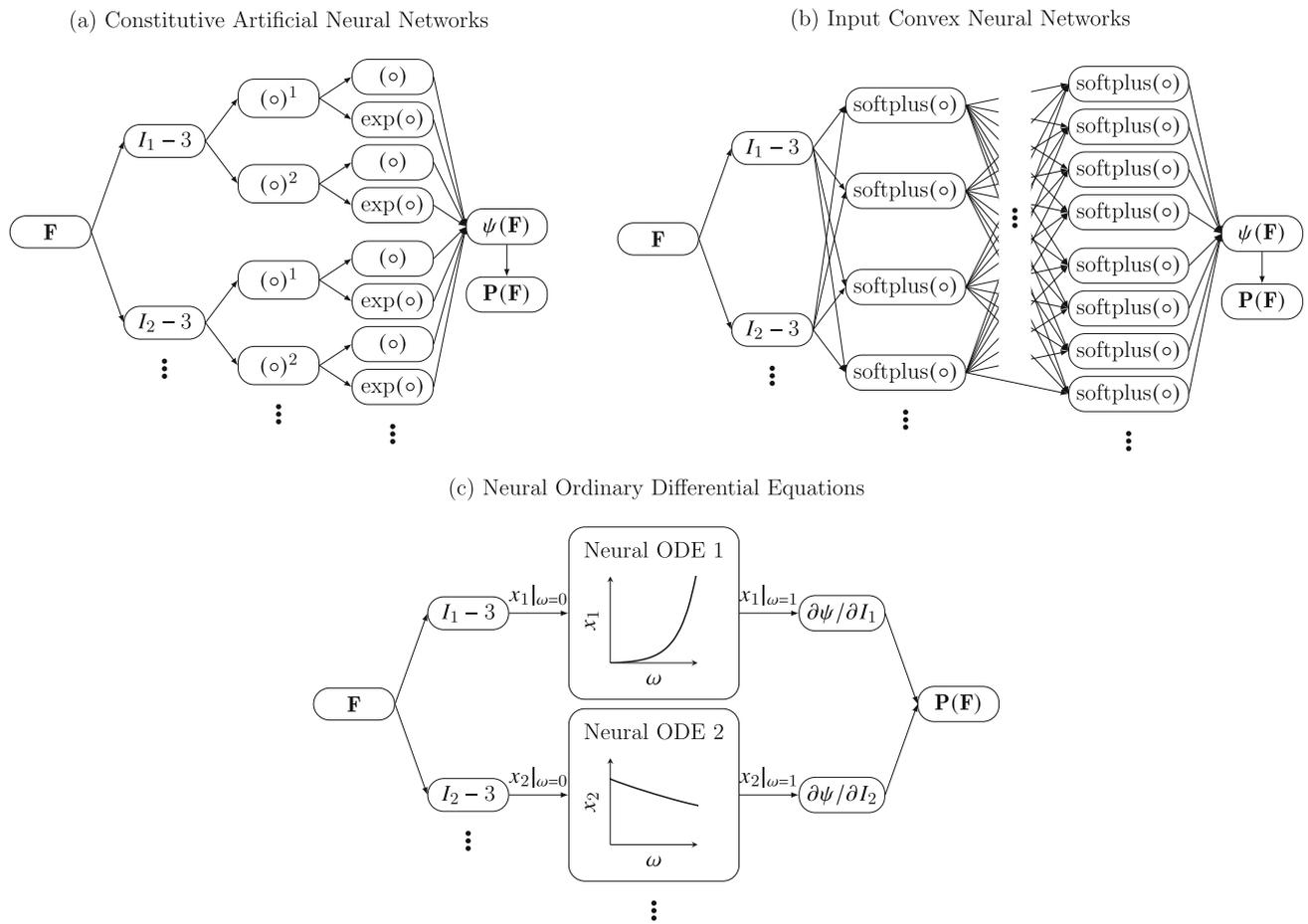
Fig. 1 Diagram depicting the training and inference processes of the deep neural network material model

where $a_1 = a_2 = 3$, $a_{4a} = a_{4s} = 1$, and b_i is a normalizing constant such that the range of $\hat{I}_i$ is approximately $[0, 3]$. Note that in the case of full incompressibility as assumed from now on, the normalized invariants strictly satisfy $\hat{I}_i \geq 0$. For compressible or nearly incompressible materials, simply replace I_i with the isochoric counterpart $\bar{I}_i$ in Eq. (18). For the mixed invariants, the normalized version is

$$\hat{K}_{ij} = \alpha_{ij}\hat{I}_i + (1 - \alpha_{ij})\hat{I}_j \,, \tag{19}$$

which also satisfies $\hat{K}_{ij} \geq 0$ as along as $\alpha_{ij} \in [0, 1]$. For the general case including anisotropy, the strain energy can be summarized as

$$\begin{aligned}
\psi_{\text{CANN}} &= \sum_{i,a,b} \psi_{i,ab} + \sum_{i,j,a,b,i\neq j} \psi_{ij,ab} \\
&= \sum_{i,a,b} g_{i,ab} f_b(w_{i,ab} P_a(\hat{I}_i)) \\
&\quad + \sum_{i,j,a,b,i\neq j} g_{ij,ab} f_b(w_{i,ab} P_a(\hat{K}_{ij})) \,,
\end{aligned} \tag{20}$$

where $P_a(x) = x^a$ is a basic polynomial expansion with $a \in \{1, 2, 3\}$ in our implementation, $f_b(\circ)$ is the activation function choice in our case either identity $f_1(x) = x$ or exponential $f_2(x) = \exp(x) - 1$. The notation is the same for the mixed invariants. The weights $g_{i,ab}$, $w_{i,ab}$, $g_{ij,ab}$, $w_{ij,ab}$ are the trainable parameters of the model and need to be non-negative to maintain the convex non-decreasing output.

For the rubber examples below, we only use the two main invariants $\hat{I}_1$, $\hat{I}_2$. For the skin example we have two ansatz. The simpler model includes contributions from $\hat{I}_1$, $\hat{I}_2$, $\hat{K}_{4a\,4s}$. The second option for the skin examples takes inputs $\hat{I}_1$, $\hat{I}_2$, $\hat{K}_{12}$, $\hat{K}_{14a}$, $\hat{K}_{14s}$, $\hat{K}_{4a\,4s}$.

Our choice of polynomials and activation functions guarantee the interpolation of convex non-decreasing functions of the inputs $\hat{I}_i$, $\hat{K}_{ij}$ in the domain $\hat{I}_i$, $\hat{K}_{ij} \geq 0$ provided that only non-negative weights are used which is easy to enforce. The non-negative condition on the domain, $\hat{I}_i$, $\hat{K}_{ij} \geq 0$, is trivially satisfied for incompressible materials, and satisfied for compressible or nearly incompressible materials if the isochoric invariants are used in Eq. (18). Thus, CANNs *a priori* result in polyconvex strain energy functions.

ICNN models

This algorithm also relies on building convex functions of the normalized invariants and linear combinations of them. Let X be the input to the first layer, and $\mathbf{Z}_{i-1}$ the output of layer $i - 1$. Then, for layer i the output is

$$\mathbf{Z}_i = \text{softplus}^2(\mathbf{W}_{z,i}^\top \mathbf{Z}_{i-1} + X \exp(\mathbf{W}_{x,i}) + \mathbf{b}_i), \quad (21)$$

parameterized by the weights $\mathbf{W}_{z,i}$, $\mathbf{W}_{x,i}$ and biases $\mathbf{b}_i$. For the first layer we have

$$\mathbf{Z}_1 = \text{softplus}^2(X \exp(\mathbf{W}_{x,1}) + \mathbf{b}_1), \quad (22)$$

while for the last layer

$$Z_n = \exp(\mathbf{W}_{z,n}^\top)\mathbf{Z}_{n-1} + X \exp(W_{x,n}) + \mathbf{b}_n. \quad (23)$$

This architecture retains convexity because $\text{softplus}^2(\circ)$ is a convex non-decreasing function evaluated on linear combinations of the original input and the intermediate layer outputs using non-negative weights (enforced with the $\exp(\circ)$ function). Therefore, ICNNs can be used to create convex non-decreasing functions of the same normalized invariants $\hat{I}_i$ and normalized mixed invariants $\hat{K}_{ij}$ defined in Eqs. (18), (19) for the CANN models. The general expansion is

$$\psi_{\text{ICNN}} = \sum_i \psi_i(\hat{I}_i) + \sum_{i,j,i \neq j} \psi_{ij}(\hat{K}_{ij}). \quad (24)$$

Similarly to the CANNs, for the rubber examples using ICNNs we only consider two functions $\psi_1(\hat{I}_1)$, $\psi_2(\hat{I}_2)$. For the anisotropic examples we have two models. The simpler one uses three functions $\psi_1(\hat{I}_1)$, $\psi_2(\hat{I}_2)$, $\psi_{4a\,4s}(\hat{K}_{4a\,4s})$. The second anisotropic model also includes the mixed terms $\psi_{1\,4a}(\hat{K}_{1\,4a})$, $\psi_{1\,4s}(\hat{K}_{1\,4s})$, $\psi_{1\,2}(\hat{K}_{1\,2})$.

NODE models

In contrast to the two previous methods, NODEs avoid interpolation of the energy and interpolate the derivative functions directly. In the end, the derivatives with respect to the invariants are the ones that enter the definition of the stress, see Eq. (9). Consider the normalized invariant $\hat{I}_i$, the NODE is a feed-forward neural network with weights $\mathbf{W}$ and biases $\mathbf{b}$ that define the function $f(\circ)$ of the ODE

$$\frac{dy_i(\omega)}{d\omega} = f(y_i, \omega), \ y_i(0) = \hat{I}_i, \quad (25)$$

where ω is a pseudo-time auxiliary variable. The output of interest is the solution of the ODE at a fixed pseudo-time. In

this implementation we choose $\omega = 1$,

$$\frac{\partial \psi}{\partial \hat{I}_i} = y_i(1). \quad (26)$$

Note that the output is directly the derivative of the strain energy. The key observation is that trajectories of ODEs do not intersect, thus for two initial conditions $y_i^{(a)}(0) \geq y_i^{(b)}(0)$, the ensuing trajectories continue to satisfy $y_i^{(a)}(\omega) \geq y_i^{(b)}(\omega)$. This implies

$$\frac{\partial \psi}{\partial \hat{I}_i}\bigg|_{\hat{I}_1 = \hat{I}_1^{(a)}} \geq \frac{\partial \psi}{\partial \hat{I}_i}\bigg|_{\hat{I}_1 = \hat{I}_1^{(b)}} \iff \hat{I}_1^{(a)} \geq \hat{I}_1^{(b)}. \quad (27)$$

The monotonicity of the output Eq. (27) is equivalent to convexity of the underlying ψ. For the mixed invariants, the NODE defines the derivative

$$\frac{\partial \psi}{\partial \hat{K}_{ij}} = y_{ij}(1), \quad (28)$$

for an ODE analogous to Eq. (25). Therefore, when using NODE models we do not recover an analytical expression for ψ_{NODE}. Nevertheless, the energy can be integrated if needed

$$\psi_{\text{NODE}} = \sum_i \int_{\hat{I}_i} \frac{\partial \psi}{\partial \hat{I}_i} + \sum_{i,j\,i \neq j} \int_{\hat{K}_{ij}} \frac{\partial \psi}{\partial \hat{K}_{ij}} \quad (29)$$

along a given trajectory over $\hat{I}_i$, $\hat{K}_{ij}$. Even though convexity of ψ_{NODE} with respect to the invariants is ensured by Eq. (27), to construct convex non-decreasing functions the additional restriction of zero biases $\mathbf{b} = 0$ is applied. With this last correction, the energy ψ_{NODE} is automatically polyconvex.

Benchmark datasets and test cases

We consider two datasets in this study, a classicall rubber dataset including uniaxial tension (UT), pure shear (PS), and equibiaxial tension (ET) nominal stress-stretch data from [8]. The other dataset is from porcine skin and consists of three biaxial tests: strip biaxial in the x direction (SX), i.e. $\lambda_x = \lambda$ is applied and the orthogonal direction is kept at $\lambda_y = 1$, strip biaxial in y direction (SY), and equibiaxial tension (EB). Data for skin comes from [7], and is also nominal stress-stretch data.

Model calibration

Training of the models consists of finding the set of parameters $\hat{\boldsymbol{\theta}}$ that minimizes the mean squared error (MSE) loss function defined as

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{N}\left[(\mathbf{P}_{xx} - \hat{\mathbf{P}}_{xx}(\boldsymbol{\theta}))^2 + (\mathbf{P}_{yy} - \hat{\mathbf{P}}_{yy}(\boldsymbol{\theta}))^2\right]$$
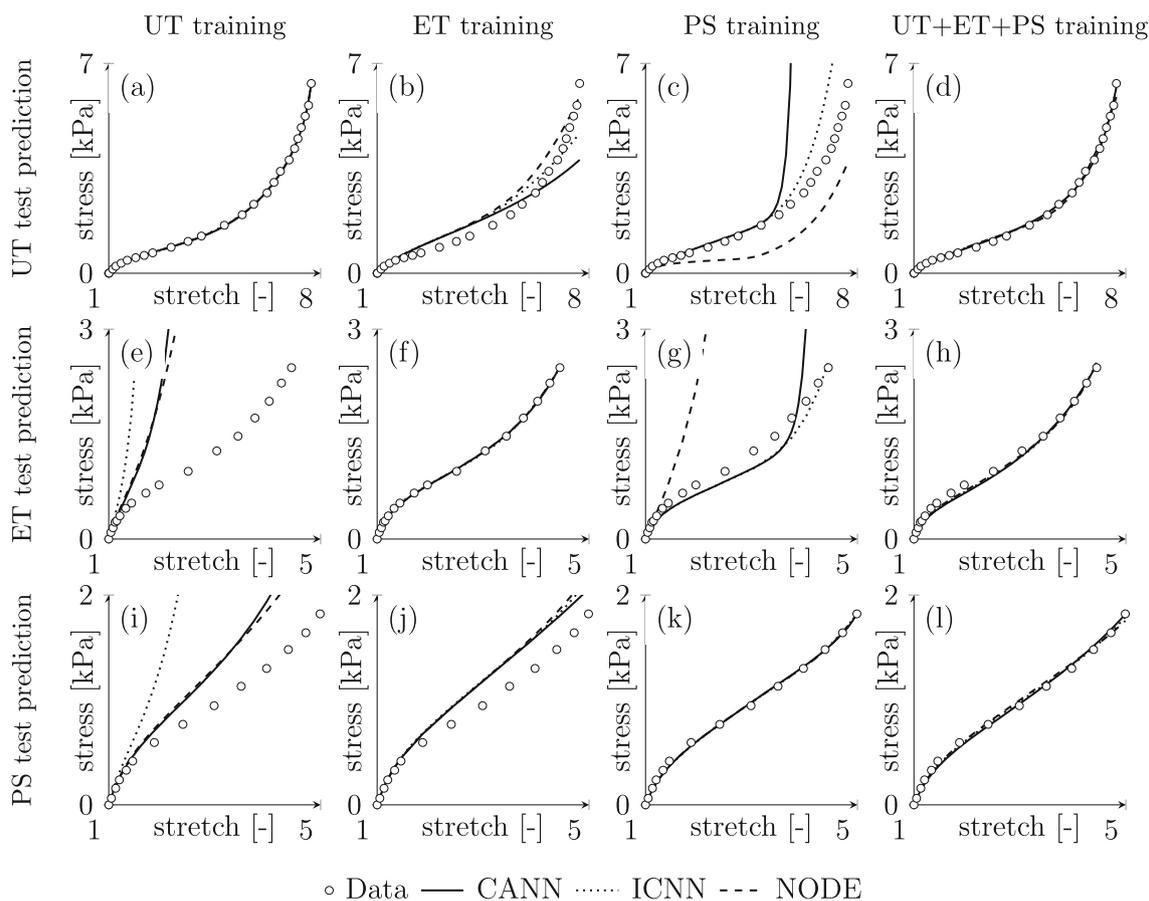
Fig. 2 Performance of CANN, ICNN and NODE moddels on rubber nominal stress-stretch data. Trained on uniaxial tension (UT), models are compared against UT data (**a**) but also against equibiaxial tension (ET) data (**e**) and pure shear (PS) data (**i**). Trained on ET, models are evaluated on UT, ET, PS data (**b**, **f**, **j**) respectively. Similarly, trained on PS data, comparison against UT, ET, PS curves (**c**, **g**, **k**). All three models can capture all three datasets when trained on all data simultaneously (**d**, **h**, **l**)

where $\hat{\mathbf{P}}_{xx}$, $\hat{\mathbf{P}}_{yy} \in \mathbb{R}^N$ are the components of stress as predicted by the models with parameters $\boldsymbol{\theta}$, $\mathbf{P}_{xx}$, $\mathbf{P}_{yy} \in \mathbb{R}^N$ are stress data obtained from biaxial experiments and N is the number of data points.

Training and inference of all the models discussed in this study were performed in Python using JAX high performance numerical computations library. We use the ADAM optimization algorithm with a step size of 2×10^{-4} and parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ (used by JAX). The training is continued for 50,000–100,000 epochs depending on the data used for training. Training times for CANN, ICNN and NODE are less than 20 sec, 50 sec and 7 min, respectively, in the most computationally demanding training case (training anisotropic models with all of the skin data simultaneously) on an Apple M1 Pro CPU.

Results

Performance on rubber dataset

The rubber dataset contains three mechanical tests as described in the Methods Section. To test the ability of the data-driven methods to extrapolate we trained first against one of the three tests and compared against the other two. Results are depicted in the first three columns of Fig. 2. Not surprisingly, all three methods perfectly capture the loading curve on which they are trained on (Fig. 2a, f, k). However, the methods have difficulty extrapolating. Depending on which test was used for training, the performance on the validation data varies. When trained on uniaxial data, predictions on the other two tests are inaccurate, with stiffer predictions in all
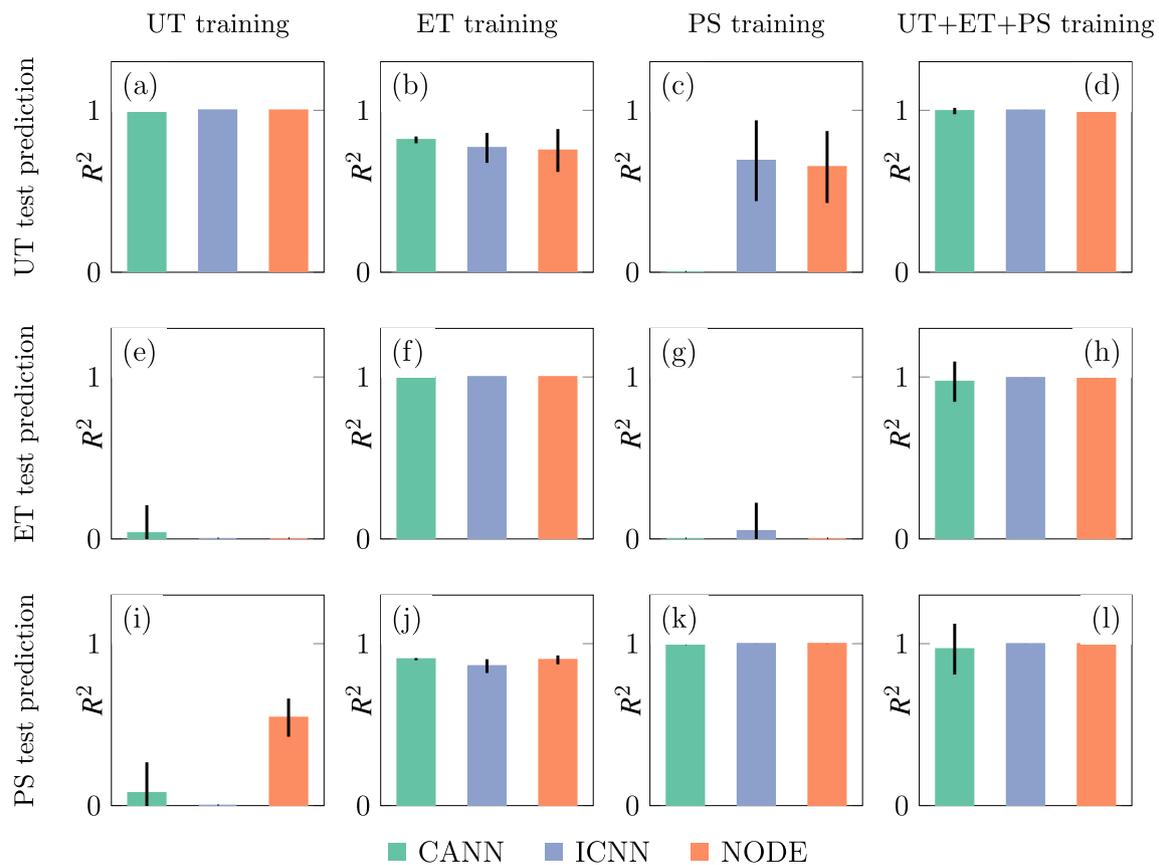
Fig. 3 Performance of the data-driven constitutive models in terms of average R^2 values from 10 training with different initialization. CANN, ICNN and NODE moddels trained on uniaxial tension (UT) are compared against UT data (**a**), equibiaxial tension (ET) data (**e**), and pure shear (PS) data (**i**). Trained on ET, models are evaluated on UT, ET, PS data (**b**, **f**, **j**) respectively. Trained on PS data, comparison is done against UT, ET, PS curves (**c**, **g**, **k**). All three models have R^2 values near one for all data when trained on all data simultaneously (**d**, **h**, **l**)

cases compared to the data (Fig. 2e, i). The ICNN trained on pure-shear data is still able to capture the response in biaxial and uniaxial loading (Fig. 2c, g). In contrast, the CANN model trained on PS data can predict UT and ET data up to an intermediate stretch after which the prediction exponential increases and diverges from the data. The NODE trained on PS data performs well on the UT dataset but not on the ET dataset. Equibiaxial training appears to be the best for extrapolating for all three methods. The prediction for ET data matches closely the experiments, see Fig. 2f, and the predictions for uniaxial and pure shear qualitatively match the observed response albeit with some error (Fig. 2b, j). To verify that the methods are indeed able to capture the entire response of the material, the last column of Fig. 2 shows predictions when CANN, ICNN, NODE models are trained on all data at once. All three methods flawlessly interpolate the entire dataset (Fig. 2d, h, l).

Results in Fig. 2 are representative, yet, they correspond to single fit from the CANN, ICNN and NODE models. To show the robust performance of the data-driven methods, we repeat the training 50 times and compute R^2 values for each trained model. The R^2 values are shown in Fig. 3 in a layout analogous to the representative training Fig. 2. The R^2 values confirm the previous observations from Fig. 2. For uniaxial training, R^2 values on UT data are approximately one always (Fig. 3a) but there is little predictive performance on the other two tests (Fig. 3e, i). For PS training we confirm that NODE and ICNN are able to capture the PS response (Fig. 3k), the UT response (Fig. 3c), but not the ET data (Fig. 3g). CANN models can capture the pure shear response just as well (Fig. 3k), but unable to extrapolate to the other two loading cases (Fig. 3c, g). With 50 instances of model fitting, we can confidently state that equibiaxial tests are indeed the ones that allow the three machine learning models to better extrapolate to other loading cases. R^2 values in Fig. 3b, f, j are always greater than 0.656, with narrow standard deviations. Figure 3d, h, l also confirms that when trained on all data at once, CANN, ICNN and NODE have no trouble fitting the data, achieving R^2 on average 0.971, 0.997, 0.997 for each of the methods respectively.
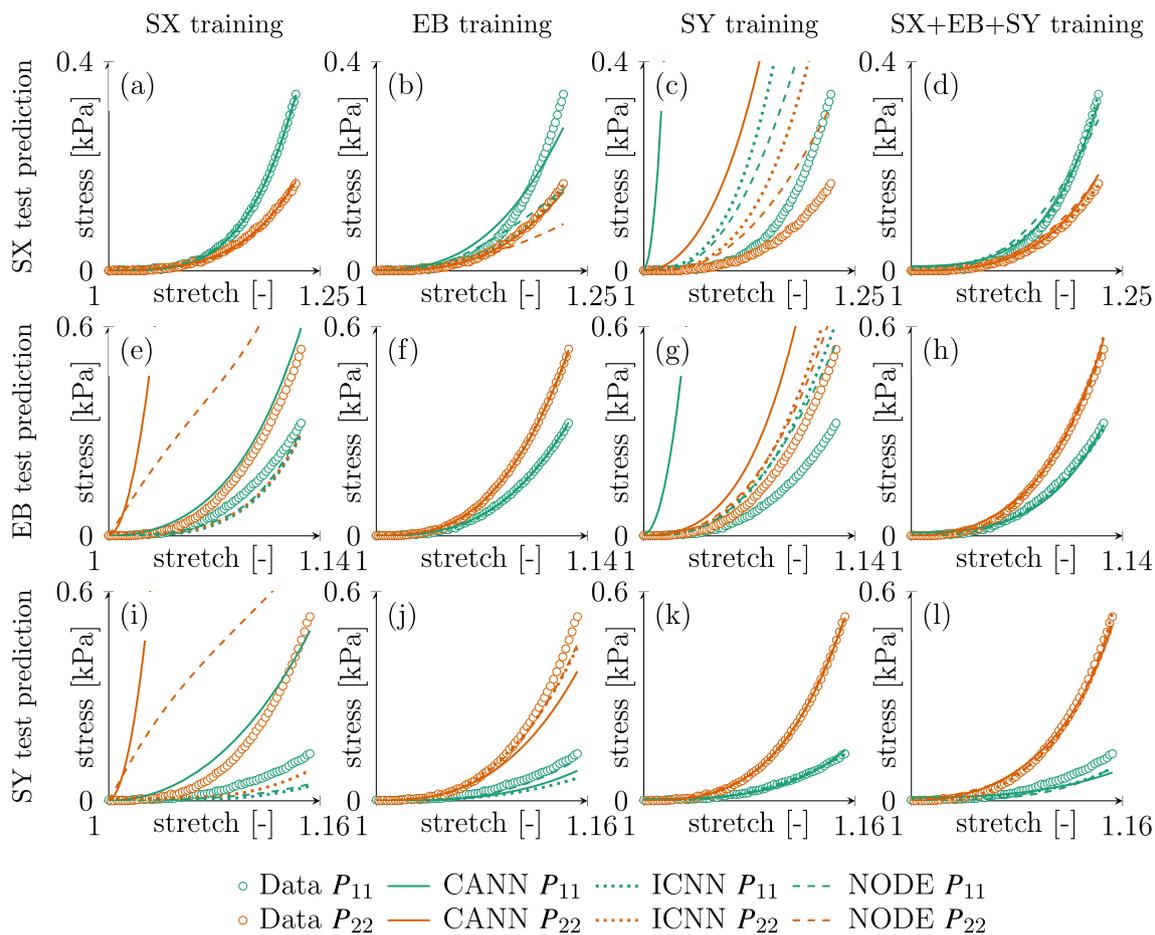
Fig. 4 Performance of CANN, ICNN and NODE models against the skin mechanics dataset. Trained on strip x (SX) data, the three models were compared to SX (**a**), equibiaxial (EB) (**e**) and strip y (SY) data (**i**). Trained on EB data, comparison to SX, EB, SY data is shown in (**b**, **f**, **j**). Trained on SY data, comparison to SX, EB, SY is shoin in (**c**, **g**, **k**). Trained on all data simultaneously, the three methods can capture SX (**d**), EB (**h**) and SY response (l)

Performance on skin dataset

The anisotropy of skin leads to poorer capacity of the three algorithms for extrapolation. Trained with either strip biaxial in x, strip biaxial y, or equibiaxial tension, the three methods can capture the response they are trained on but unable to extrapolate, as illustrated in the first three columns of Fig. 4. To capture the transversely anisotropic response of skin, the number of parameters and flexibility of the functional space available to the three data-driven methods enables them to produce complex response, but at the same time it leads to unconstrained and poor predictions outside of the training region. For instance, trained on SX data, predictions under SX loading are remarkably accurate (Fig. 4a), but CANN models tend to predict stiffer responses in EB and SY loading (Fig. 4e, i); NODE predics stiffer response in SY loading (Fig. 4i) but accurate response in EB loading (Fig. 4e), and ICNN performs well in EB loading (Fig. 4e) but predicts soft

response compared to the data in SY loading (Fig. 4i). To verify if the models are able to capture the entire dataset we trained CANN, ICNN, NODE modes with all the data simultaneously and show the fits in Fig. 4d, h, l. All three methods can capture the response when trained on all data, however, fits are not perfect compared to the individual test fitting in Fig. 4a, f, k. The poorer performance in the simultaneous fitting is consistent between all three methods and suggests that the data themselves might be inconsistent with the assumption of hyperelasticity, that there are experimental errors, or that the functional space available to the data-driven models needs to be even richer.

A more quantitative analysis of the performance is reported in Fig. 5, which shows R^2 values computed after 10 instances of model training with different, random initialization. Just as observed in the representative fits of Fig. 4, the R^2 scores on the loading used for training are near one (Fig. 5a, f, k), but they are low or even near zero for the
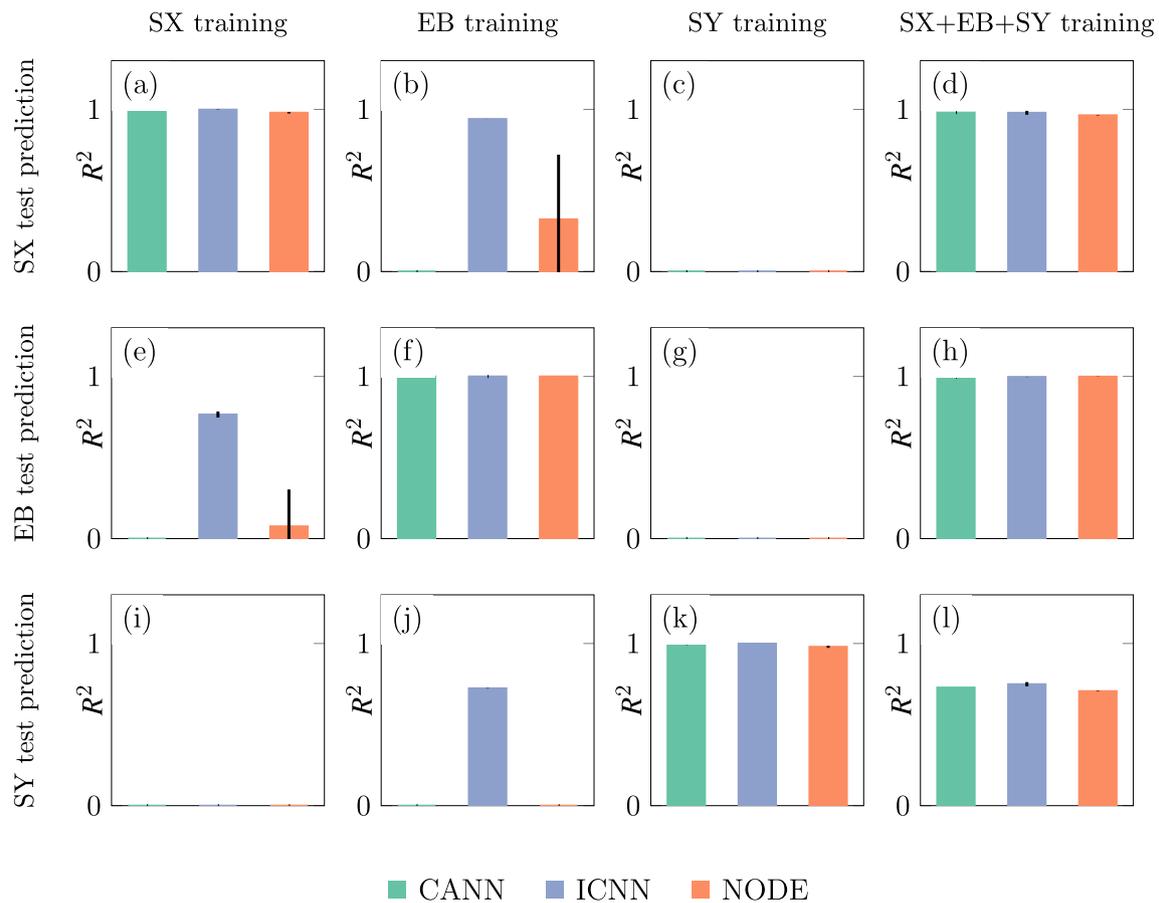
Fig. 5 Performance of the data-driven methods on the skin dataset in terms of R^2 scores for data from three type of tests: strip biaxial x (SX), equibiaxial (EB), strip biaxial in y direction (SY). Trained on SX data (**a**), the models are tested agains EB (**e**) and SY (**i**) data. Trained on EB data (**f**), the models are tested on SX (**b**) and SY (**j**) loading. When trained on SY data (**k**), the models cannot capture SX (**c**) or EB (**g**) response. When trained on all data simultaneously, CANN, ICNN, and NODE models can capture all three types of loading: SX (**d**), EB (**h**), SY (**l**)

validation cases (Fig. 5b, c, e, g, i, j). Surprisingly, there is still some information from the equibiaxial test (Fig. 5f) that is useful for extrapolation to the strip biaxial loading cases (Fig. 5b ,j). Training on the strip biaxial tests, SY has no information for the SX or EB data (Fig. 5c, g), whereas SX training does lead to some $R^2 > 0$ for EB (Fig. 5e) but not SY data (Fig. 5i). The methods are able to consistently interpolate the entire data from all three tests regardless of random initialization (Fig. 5d, h, l).

Extrapolation beyond training range

In the previous sections we have already tested if the models trained on a given loading scenario can predict the observed response under a completely different loading mode. Another test for extrapolation is to train the models up to a certain range of stretches and test their capacity to predict accurate stresses beyond the training range. Figure 6 shows the results

for the rubber dataset. For each of the loading cases (UT, ET, PS), only the first 50% of the data was used in training. The stress was then predicted for the remaining range. For an additional comparison, in this case we also trained an unrestricted feed forward neural network (NN). It is clear that the automatically polyconvex models can extrapolate beyond the training range. The unrestricted NN, on the other hand, exhibits a non-convex response immediately after the training range, with stresses decaying under increased deformation and falling down to near zero in the ET and PS cases.

For the skin dataset, results from training with the first 50% of the data and predicting the remaining 50% are shown in Fig. 7. In contrast to the rubber case, here the unrestricted NN does not show evident unphysical behavior. The CANN method leads to the best predictions outside the training range. The ICNN method produces stiffer responses than the real data, while the NODE method tends to predict softer responses. Much like other constitutive models, the methods

Fig. 6 Extrapolation performance of CANN, ICNN, NODE and an unrestricted NN when trained with rubber data. Trained on the first 50% of uniaxial tension (UT) data (**a**), equibiaxial tension (ET) data (**b**) and pure shear (PS) data (**c**) and tested with the remaining 50%
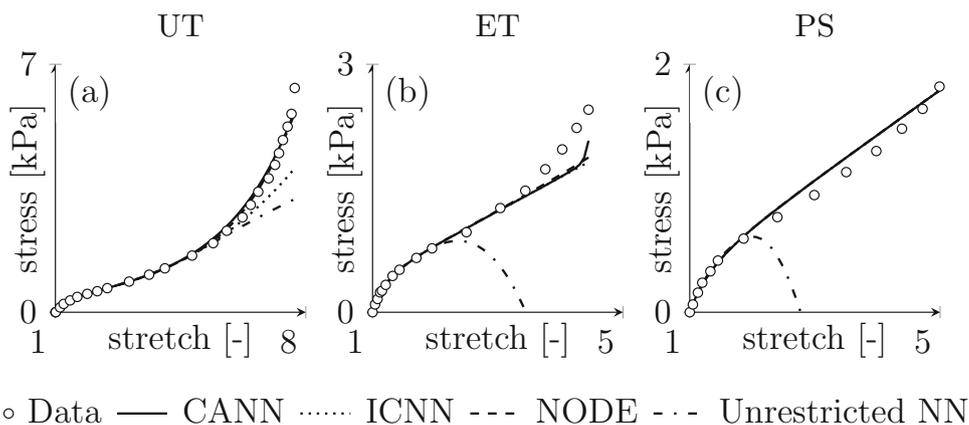
Fig. 7 Extrapolation performance of CANN, ICNN, NODE and an unrestricted NN when trained with skin data. Trained on the first 50% of strip biaxial x (SX) data (**a**), equibiaxial (EB) data (**b**) and strip biaxial y (SY) data (**c**) and tested with the remaining 50%
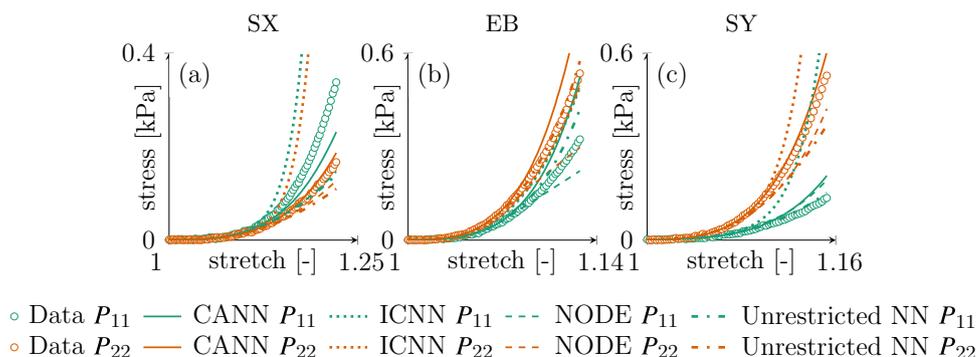
studied here should be used primarily within their training range. Nevertheless, even if accuracy is gradually lost outside the training range, CANN, ICNN and NODE models can still make reasonable predictions.

Regularity of second derivatives

Thus far we have focused on the performance of the data-driven models to capture stress-stretch data, which directly relates to strain energy derivatives. However, using these highly nonlinear model in large scale physics solvers, either implicit dynamics or equilibrium, requires computation of second derivatives of the energy. Therefore, even though second derivatives are not related to any data, we are interested in the regularity of the second derivatives for CANN, ICNN and NODE models.

For the rubber benchmark the models are based on the interpolation of two functions $\partial\psi(I_1)/\partial I_1$, $\partial\psi(I_2)/\partial I_2$. Figure 8 shows the second derivatives $\partial^2\psi(I_1)/\partial I_1^2$, $\partial^2\psi(I_2)/\partial I_2^2$ with the same layout as Figs. 2 and 3. It is surprising that even though all three methods capture the stress data quite well, they differ substantially in terms of their second derivatives. This reflects that there are many strain energy functions $\psi(I_1, I_2)$ that are polyconvex and that can capture the stress-stretch data under uniaxial, pure shear, and equibiaxial loading. The CANN, ICNN and NODE are suited to

capture different functions within the large space of functions available to each method. The consistent trend in Fig. 8 is that the CANN models lead often to exponential second derivatives because one of the two key activation functions is the exponential. In contrast, the NODE model is the one with the smallest second derivatives in all cases. For all the three methods, the second derivatives are smooth functions.

For the skin benchmark, there are more functions being interpolated by the three data-driven frameworks. As a result, Fig. 9 shows the second derivatives $\partial^2\psi/\partial I_1^2$, $\partial^2\psi/\partial I_2^2$, $\partial^2\psi/\partial I_{4a}^2$, and $\partial^2\psi/\partial I_{4b}^2$. Consistent with the rubber data NODE second derivatives are the smallest out of the three methods. The second derivatives might increase for some initial range of deformation but tend to smaller values toward the end of the testing ranges (Fig. 9i–l). The CANN (Fig. 9a–d) and ICNN methods ((Fig. 9e–h) have increasing second derivatives over the range of the invariants. Also similar to the rubber benchmark, here we see that even though all three methods perform similarly on the stress-stretch predictions (see Fig. 4), they do so by interpolating different functions $\psi(I_1, I_2, I_{4a}, I_{4s})$.

Model efficiency

A key question and common criticism of data-driven models, particularly neural network-based models, is that increasing
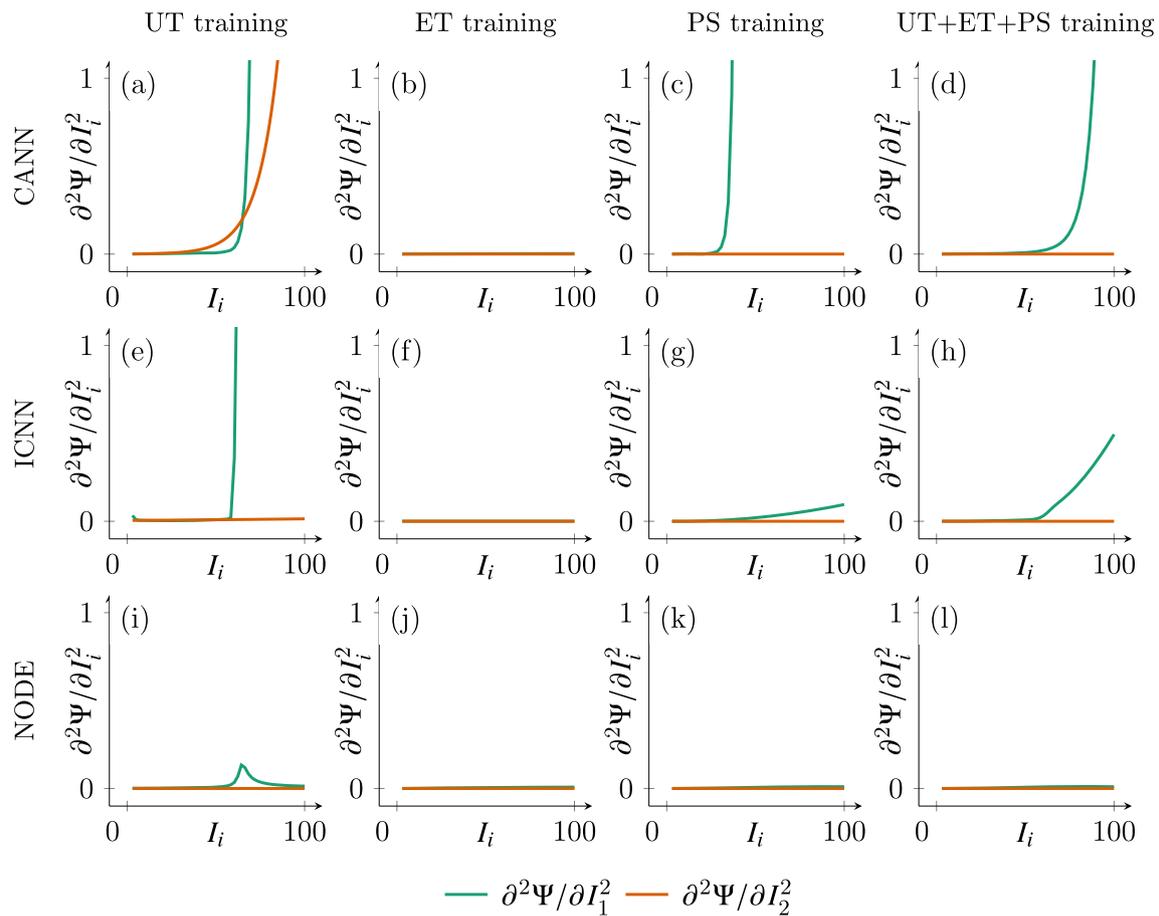
Fig. 8 Second derivatives of strain energy functions predicted by the data-driven models trained with rubber data. Training with UT only, predictions are done for CANN (**a**), ICNN (**e**) and NODE (**i**) models. Similarly, second derivatives for the three methods are shown for ET training only (**b**, **f**, **j**), PS training only (**c**, **g**, **k**) and tained against all data (**d**, **h**, **l**)

the number of trainable parameters logically allows the methods to capture the limited data better and better, but at the risk of over-fitting. The polyconvexity constraint, enforced exactly for CANN, ICNN and NODE models, prevents non-physical extrapolation, much like expert models. On the other hand, expert models and some non-parametric data-driven methods [38] are generally very efficient and capture the data reasonably well with very few parameters. We test how efficiently can the data-driven models interpolate the data, i.e. we ask how does the error decrease as a function of the number of trainable parameters. We use the mean absolute error as the metric of error in this study, which is defined as $MAE = \frac{1}{N}(|\mathbf{P}_{11} - \hat{\mathbf{P}}_{11}| + |\mathbf{P}_{22} - \hat{\mathbf{P}}_{22}|)$ where $\hat{\mathbf{P}}_{11}$ and $\hat{\mathbf{P}}_{22}$ are the predicted values of stress and N is the number of data points. We favor an absolute measure of error to a relative measure due to the fact that experimental data is noisy and the magnitudes of stress in low stretch regions are extremely low. This combination causes relative measurements of error

to be unreasonably high even if the model captures the mean behavior very well.

Figure 10 shows the efficiency plots for the rubber benchmark. The structure of the CANN model is between that of a neural network and an expert model. As a result, there is a single point for the CANN model for each of the plots in Fig. 10. For ICNN and NODE models, the error decays with increasing number of parameters, as expected. When there are 52 trainable parameters, the NODE and ICNN show similar performance in all the training cases. However, the drop in the error is more pronounced for the ICNN compared to the NODE framework. This suggests that the NODE model can capture the data well even with very few parameters.

The efficiency trends are not preserved for the anisotropic skin data as shown in Fig. 11. In this case, in order to explore the effect of the number of parameters on the accuracy of the methods we follow two strategies: reducing the ansatz by interpolating only the functions in (4), or using the full expansion (11) but changing the number of trainable param-
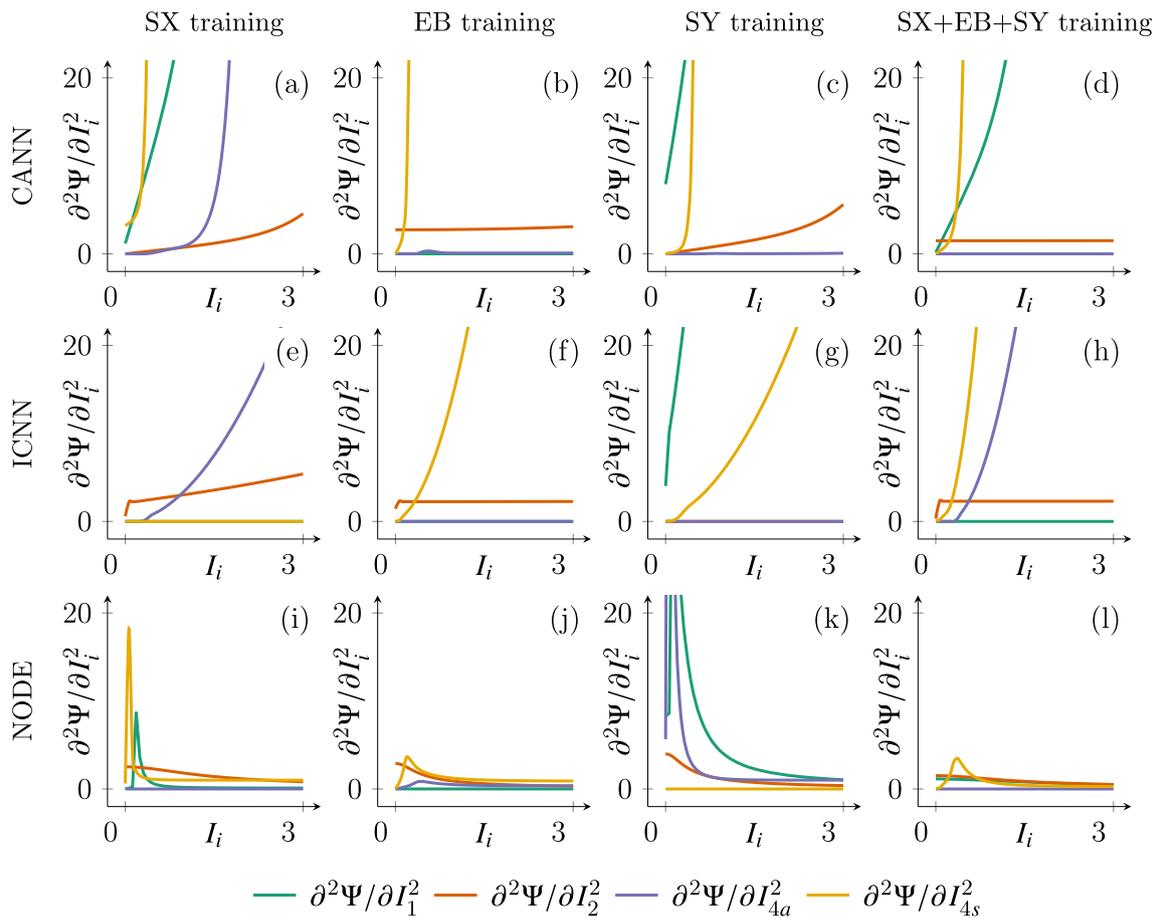
Fig. 9 Second derivatives of strain energy when trained with skin data. CANN predictions (**a–d**), ICNN predictions (**e–h**), and NODE predictions (**d–l**). Columns correspond to the data used in training. First three columns correspond to either SX, EB, or SY data only. Last column shows predictions when models are trained on all data simultaneously
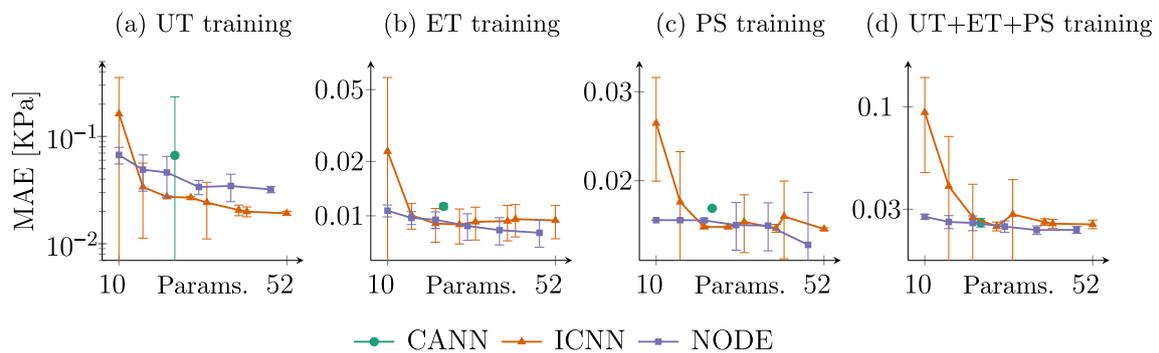
Fig. 10 Model efficiency for the rubber dataset depicted in terms of mean absolute error (MAE) against number of trainable parameters. Columns correspond to the type of data used for training: UT (**a**), ET (**b**), PS (**c**), or all data used simultaneously during training (**d**). Note that for the CANN model the number of parameters is fixed and a single point is shown for the CANN model in each panel. The ICNN and NODE are neural network-based models and the number of trainable parameters increases with number of layers and layer depth
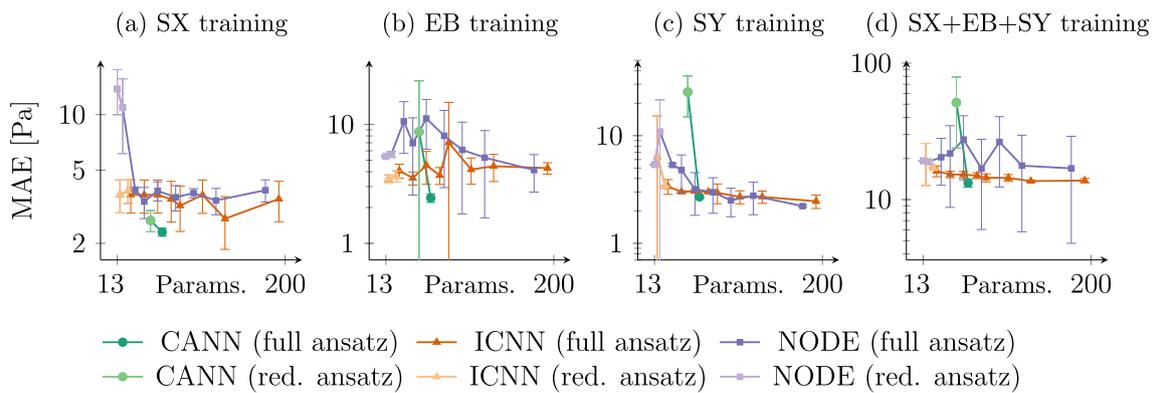
Fig. 11 CANN, ICNN and NODE model efficiency for the skin benchmark shown as mean absolute error (MAE) against number of trainable parameters and model complexity. For each of the three models, two ansatz are used: a reduced expansion based on (4), or a full expansion (11). Plots show efficiency corresponding to different training cases: SX (**a**), EB (**b**), SY (**c**), or all data simultaneously (**d**)

eters. For the CANN model, which has a fixed number of parameters when considering either (4) or (11), we observe that the full ansatz has lower errors than the reduced one for all training cases. The flexibility of the framework increases by including the mixed terms, which helps with capturing the data better. This is consistent with the development of mixed invariant terms in popular closed-form constitutive equations such as the Gasser-Ogden-Holzapfel model [39]. The ICNN and NODE also show decreasing errors when going from the reduced model (4) to the model including mixed invariants (11). The improvement is much more pronounced for the NODE model compared to the ICNN one. In contrast to the rubber dataset, for skin, increasing the number of parameters of the neural networks used in the NODE models leads to a large decrease in error. The ICNN model error decreases only slightly with increasing number of parameters. At the upper end of the range considered, i.e. approaching 200 parameters, both ICNN and NODE perform similarly. The most efficient of the methods for skin data is the CANN, which achieves the lowest errors with the lowest number of parameters.

Discussion

This manuscript analyzes the performance of three data-driven methods for isotropic and anisotropic hyperelastic materials that automatically satisfy objectivity, material symmetries, and polyconvexity of the strain energy. Traditional closed-form models rely on selecting few functional terms to capture the response of a material in a parsimonious way with few parameters to fit. Closed-form expressions are an elegant solution but also have the major downside of sacrificing accuracy. Data-driven methods have the flexibility to perfectly interpolate data. However, a paucity in their adoption is the difficulty to guarantee basic physics constraints

that are front and center in the design of expert models [16, 40]. Objectivity, material symmetries, and polyconvexity are key requirements to represent realistic materials. The CANN, ICNN, and NODE models studied here are constructed in such a way that they a priori satisfy these essential physics constraints. Therefore, these three methods have the potential to revolutionize modeling and simulation of highly nonlinear materials. We show that the three methods can interpolate rubber and skin benchmark datasets for isotropic and anisotropic hyperelasticity. They capture the data almost perfectly and have some capacity to extrapolate when trained on part of the data. The second derivatives are smooth, which is needed for equilibrium and implicit dynamic solvers. The models show the expected trade-off between accuracy and number of parameters. The physics-constrained methods outperform unconstrained neural networks, particularly during extrapolation tests. Overall, either of the three modeling frameworks, CANN, ICNN or NODE based model, is suitable for fully data-driven material modeling.

The methods we analyze here stand in contrast to other recent developments in data-driven computational mechanics. The most obvious way of leveraging machine learning tools is to directly interpolate strain–stress data. There are methods along those lines developed in recent years [41]. Model-free alternatives also use strain–stress data directly and use different regularization techniques to deal with noisy data and outliers [42, 43]. One limitation of these approaches is the inability to extrapolate. Another problem of dealing with stress data is that objectivity and polyconvexity are not satisfied a priori. Data-driven models that capture the strain energy function are more similar to expert models [19]. CANN, ICNN and NODE fall on this category. For the data-driven models that interpolate the strain energy, one option to impose polyconvexity is through the loss function [7, 25]. In contrast, CANN, ICNN and NODE models

automatically satisfy polyconvexity. The original version of these methods was introduced in recent publications [8–12]. However, benchmarking of the original formulations is challenging because different expansions of the energy were used in each case. In this work we have re-formulated the methods such that the same invariants and energy terms are used consistently across the three models. With this implementation we show that, because the constraints are embedded in the methods, there is no trade-off between model accuracy and enforcing the physics. The three methods can get accurate representation of the data, show positive second derivatives of the energy with respect to the invariants, and perform robustly even with random initialization. It should be noted that, while on the one hand using a specific expansion of the energy in Eqs. (4), (11), allowed us to compare across the three methods, other expansions are possible [44]. We did not explore this further because the functional forms were already general enough to capture most closed-form constitutive models for rubbers and skin [3, 31]. The Eqs. (4), (11) restrict which deformation invariants are used to build the potentials, but aside of enforcing the convexity conditions, they can interpolate highly nonlinear functions of those invariants, including and beyond existing closed-form models.

The ideas behind each method are different and this translates into slight differences in their performance. CANN leverages the structure of feed-forward neural networks but uses a fixed number of available terms. Fitting a CANN involves finding the weights of the different terms. As a result, CANNs produce parsimonious models but are inherently limited by the number of functional terms. Despite the fixed structure, CANNs perform well on the benchmark datasets of this study. Other data-driven model identification algorithms relying on richer data (e.g. strain field) have also been proposed in recent years to perform sparse-regression material modeling [26] or ICNN model training [27]. ICNNs rely on building convex functions by using nested linear combinations of convex non-decreasing functions in every layer of an otherwise conventional feed-forward neural network structure. Different ICNN constitutive models have been proposed and the particular implementation shown here mainly differs through the expansion Eqs. (4), (11). NODEs deal directly with the energy derivative functions and leverage monotonicity of ODEs to get monotonic derivative functions (which implies convex functions). Because ICNNs and NODEs have an inner structure that resembles standard neural networks, they have more freedom to adjust the number parameters by changing the number of layers or the depth within a layer. The efficiency plots reflect the trade-off between accuracy and number of parameters. The increased flexibility of ICNN and NODE models comes at the cost of reduced interpretability. As the number of parameters increases, the difference between NODE and ICNN models vanishes. All three methods can accurately and efficiently capture the data.

The other notable difference between the methods is in the prediction of second derivatives of the strain energy. CANN and ICNN models tend to predict increasing second derivative functions. The NODE, in contrast with the other two methods, tends to predict smaller or vanishing second derivatives towards the end of the training region. This difference likely stems from the fact that ODEs have fixed points. In other words, the derivative predictions converge to a single value, consequently producing vanishing second derivatives. In all cases the second derivatives are smooth functions which is ideal for equilibrium and implicit dynamic solvers [9]. This is in contrast with other data-driven methods that require additional regularization of the derivatives [45]. Another strategy to work with the derivatives of the energy but avoid solving an ODE would be to explore integrable neural networks [46].

The methods we benchmark here have been designed to capture hyperelasticity. More complex material response beyond hyperelasticity can also benefit from the flexibility of data-driven methods. There is still a gap in the development of physics-informed machine learning methods for dissipative materials such as plasticity and viscoelasticity [47]. There have been data-driven methods in this direction, with different degree of built-in physics constraints [45, 48–50]. Therefore, this is a central area for future work that can leverage the three existing frameworks reviewed and refined here. A second extension that is needed is modeling uncertainty in the material response. This is particularly relevant to biological tissue [1]. Neural network-based frameworks can capture perfectly the response of a single material and can easily retrained with new data, but a fully Bayesian approach would allow deeper understanding of population distribution. For example, it would allow us to model how skin properties change with age, sex, or ethnicity. A Bayesian framework would also allow monitoring of epistemic uncertainty to guide data collection and produce trustworthy simulations. The third item we want to highlight is the extension to multi-modality data. The three methods we explore are still based on stress-stretch data. In contrast, some expert models are built around the idea of microstructure modeling, multi-scale simulations, or micromechanics arguments [39]. These ideas have started to permeate intro data-driven modeling [51, 52]. Alternatively, inferring material behavior from full-field displacements and global force data without relying on stress-stretch pairs has also gained attention recently [53, 54]. Physics-informed machine learning methods that can build on CANN, ICNN or NODE frameworks but also leverage images of the tissue microstructure or information about material composition are a natural next step.

Conclusions

We present three fully data-driven and physics-constrained methods for nonlinear material modeling: CANNs, ICNNs, and NODEs. The methods capture hyperelastic material response perfectly on benchmark datasets of rubber and skin under three different loading cases. Evaluating their capacity to extrapolate, their efficiency, and the regularity of their second derivatives, we conclude that even though the methods have different features, they all have comparable low errors which decay with parameter and model complexity, have smooth second derivatives, and have some capacity to extrapolate. In summary, these methods are suitable for high-fidelity modeling of arbitrary material behavior without the need to select closed-form expressions. Code and data are available with this submission and we are confident that these resources complement our detailed analysis and will favor the ongoing development and refinement of data-driven computational mechanics.

Data availability The code associated with this publication is available at https://github.com/tajtac/hypermodelcomp.

References

1. 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
2. Duriez C, Bieze T (2017) Soft robot modeling, simulation and control in real-time. In: Laschi C et al (eds) Soft robotics: trends, applications and challenges. Springer, Berlin, pp 103–109
3. Limbert G (2019) Skin biophysics: from experimental characterisation to advanced modelling, vol 22. Springer, Berlin
4. Leshno M, Lin VY, Pinkus A, Schocken S (1993) Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. Neural Netw 6:861–867
5. Peng GC, Alber M, Tepole AB, Cannon WR, De S, Dura-Bernal S, Garikipati K, Karniadakis G, Lytton WW, Perdikaris P et al (2020) Multiscale modeling meets machine learning: what can we learn? Arch Comput Methods Eng. https://doi.org/10.1007/s11831-020-09405-5
6. Marsden JE, Hughes TJ (1994) Mathematical foundations of elasticity. Courier Corporation, North Chelmsford
7. Tac V, Sree VD, Rausch MK, Tepole AB (2022) Data-driven modeling of the mechanical behavior of anisotropic soft biological tissue. Eng Comput 38:4167–4182
8. Linka K, Kuhl E (2023) A new family of constitutive artificial neural networks towards automated model discovery. Comput Methods Appl Mech Eng 403:115731
9. Tac V, Sahli Costabal F, Tepole AB (2022) Data-driven tissue mechanics with polyconvex neural ordinary differential equations. Comput Methods Appl Mech Eng 398:115248
10. Klein DK, Fernández M, Martin RJ, Neff P, Weeger O (2022) Polyconvex anisotropic hyperelasticity with neural networks. J Mech Phys Solids 159:104703
11. As'ad F, Avery P, Farhat C (2022) A mechanics-informed artificial neural network approach in data-driven constitutive modeling. Int J Numer Methods Eng 123:2738–2759
12. Chen P, Guilleminot J (2022) Polyconvex neural networks for hyperelastic constitutive models: a rectification approach. Mech Res Commun 125:103993
13. Ghaboussi J, Sidarta D (1998) New nested adaptive neural networks (NANN) for constitutive modeling. Comput Geotech 22:29–52
14. 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
15. Holzapfel GA (2000) Nonlinear solid mechanics; a continuum approach for engineering. Wiley, London
16. Ehret AE, Itskov M (2007) A polyconvex hyperelastic model for fiber-reinforced materials in application to soft tissues. J Mater Sci 42:8853–8863
17. 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
18. Vlassis NN, Ma R, Sun W (2020) Geometric deep learning for computational mechanics part I: anisotropic hyperelasticity. Comput Methods Appl Mech Eng 371:113299
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. Fuhg JN, Bouklas N, Jones RE (2022) Learning hyperelastic anisotropy from data via a tensor basis neural network. arXiv preprint arXiv:2204.04529
21. Ball JM (1976) Convexity conditions and existence theorems in nonlinear elasticity. Arch Ration Mech Anal 63:337–403
22. Schröder J (2010) Anisotropic polyconvex energies. In: Schröder J, Neff P (eds) Poly-, quasi-and rank-one convexity in applied mechanics. Springer, Berlin, pp 53–105
23. Schröder J, Neff P (2003) Invariant formulation of hyperelastic transverse isotropy based on polyconvex free energy functions. Int J Solids Struct 40:401–445
24. Gao DY, Neff P, Roventa I, Thiel C (2017) On the convexity of nonlinear elastic energies in the right Cauchy–Green tensor. J Elast 127:303–308
25. Ghaderi A, Morovati V, Dargazany R (2020) A physics-informed assembly of feed-forward neural network engines to predict inelasticity in cross-linked polymers. Polymers 12:2628
26. Flaschel M, Kumar S, De Lorenzis L (2023) Automated discovery of generalized standard material models with EUCLID. Comput Methods Appl Mech Eng 405:115867
27. Thakolkaran P, Joshi A, Zheng Y, Flaschel M, De Lorenzis L, Kumar S (2022) NN-EUCLID: deep-learning hyperelasticity without stress data. J Mech Phys Solids 169:105076
28. Amos B, Xu L, Kolter JZ Input convex neural networks. In: International conference on machine learning, PMLR, pp 146–155
29. Lejeune E (2020) Mechanical MNIST: a benchmark dataset for mechanical metamodels. Extreme Mech Lett 36:100659
30. Kobeissi H, Mohammadzadeh S, Lejeune E (2022) Enhancing mechanical metamodels with a generative model-based augmented training dataset. J Biomech Eng 144:121002
31. Dal H, Açıkgöz K, Badienia Y (2021) On the performance of isotropic hyperelastic constitutive models for rubber-like materials: a state of the art review. Appl Mech Rev 73:020802
32. Rus D, Tolley MT (2015) Design, fabrication and control of soft robots. Nature 521:467–475
33. Jor JW, Parker MD, Taberner AJ, Nash MP, Nielsen PM (2013) Computational and experimental characterization of skin mechan-

34. Lanir Y, Fung Y (1974) Two-dimensional mechanical properties of rabbit skin-II. Experimental results. J Biomech 7:171–182

35. Lanir Y (1983) Constitutive equations for fibrous connective tissues. J Biomech 16:1–12

36. Toaquiza Tubon JD, Moreno-Flores O, Sree VD, Tepole AB (2022) Anisotropic damage model for collagenous tissues and its application to model fracture and needle insertion mechanics. Biomech Model Mechanobiol 21:1–16

37. Chen S, Ní Annaidh A, Roccabianca S (2020) A microstructurally inspired constitutive model for skin mechanics. Biomech Model Mechanobiol 19:275–289

38. Fuhg JN, Bouklas N (2022) On physics-informed data-driven isotropic and anisotropic constitutive models through probabilistic machine learning and space-filling sampling. Comput Methods Appl Mech Eng 394:114915

39. Gasser TC, Ogden RW, Holzapfel GA (2005) Hyperelastic modelling of arterial layers with distributed collagen fibre orientations. J R Soc Interface 3:15–35

40. Fuhg JN, Hamel CM, Johnson K, Jones R, Bouklas N (2022) Modular machine learning-based elastoplasticity: generalization in the context of limited data. Comput Methods Appl Mech Eng 407:115930

41. Kirchdoerfer T, Ortiz M (2016) Data-driven computational mechanics. Comput Methods Appl Mech Eng 304:81–101

42. Bahmani B, Sun W (2022) Manifold embedding data-driven mechanics. J Mech Phys Solids 166:104927

43. Stainier L, Leygue A, Ortiz M (2019) Model-free data-driven methods in mechanics: material data identification and solvers. Comput Mech 64:381–393

44. Steigmann DJ (2003) On isotropic, frame-invariant, polyconvex strain-energy functions. Q J Mech Appl Math 56:483–491

45. 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

46. 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

47. Zhang W, Sommer G, Niestrawska JA, Holzapfel GA, Nordsletten D (2022) The effects of viscoelasticity on residual strain in aortic soft tissues. Acta Biomater 140:398–411

48. Xu K, Tartakovsky AM, Burghardt J, Darve E (2021) Learning viscoelasticity models from indirect data using deep neural networks. Comput Methods Appl Mech Eng 387:114124

49. Chen G (2021) Recurrent neural networks (RNNs) learn the constitutive law of viscoelasticity. Comput Mech 67:1009–1019

50. Fuhg JN, Hamel CM, Johnson K, Jones R, Bouklas N (2023) Modular machine learning-based elastoplasticity: generalization in the context of limited data. Comput Methods Appl Mech Eng 407:115930

51. Holzapfel GA, Linka K, Sherifova S, Cyron CJ (2021) Predictive constitutive modelling of arteries by deep learning. J R Soc Interface 18:20210411

52. 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

53. Flaschel M, Kumar S, De Lorenzis L (2021) Unsupervised discovery of interpretable hyperelastic constitutive laws. Comput Methods Appl Mech Eng 381:113852

54. Chen C-T, Gu GX (2021) Learning hidden elasticity with deep neural networks. Proc Natl Acad Sci 118:e2102721118