Computer Methods in Applied Mechanics and Engineering 460 (2026) 119080

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

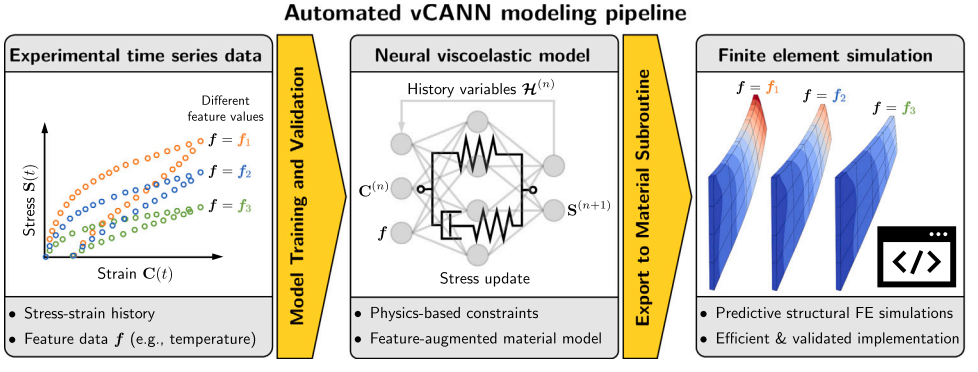
journal homepage: www.elsevier.com/locate/cma

Thermodynamically consistent viscoelastic constitutive artificial neural networks: Automating the pipeline from experimental data to finite element simulations

Kian P. Abdolazizi [a,*], Roland C. Aydin [a,b], Christian J. Cyron [a,b,*], Kevin Linka [a,c]

[a] Institute for Continuum and Material Mechanics, Hamburg University of Technology, Eißendorfer Straße 42, 21073 Hamburg, Germany
[b] Institute of Material Systems Modeling, Helmholtz-Zentrum Hereon, Max–Planck-Straße 1, 21502 Geesthacht, Germany
[c] Department of Computational Mechanics in Medicine, Institutes of Applied Medical Engineering, RWTH Aachen University, Pauwelsstraße 20, 52074 Aachen, Germany

GRAPHICAL ABSTRACT



ARTICLE INFO

Dataset link: https://github.com/KianAbd/vCANN_FEM

Keywords:
Internal state variables
Fiber-reinforced materials
Abaqus user subroutine (UMAT)
Scientific machine learning
Soft tissue biomechanics
Elastomer mechanics

ABSTRACT

Viscoelastic constitutive artificial neural networks (vCANNs) leverage neural networks for data-driven modeling of the viscoelastic behavior of materials. Here, we propose a thermodynamically consistent extension of vCANNs that captures anisotropic, nonlinear, and time-dependent material behavior. A key strength of this approach is its ability to incorporate arbitrary auxiliary features—such as temperature, microstructural descriptors, or processing parameters—directly into neural constitutive laws. We propose an automated computational pipeline for the generation and implementation of such constitutive laws within the proposed framework into finite element (FE) simulations without manual model design. The proposed framework is validated across a broad range of representative material tests, including the nonlinear, thermo-viscoelastic response of soft polymers and arterial tissue with fiber dispersion.

* Corresponding authors.
E-mail addresses: kian.abdolazizi@tuhh.de (K.P. Abdolazizi), christian.cyron@tuhh.de (C.J. Cyron).

https://doi.org/10.1016/j.cma.2026.119080
Received 15 February 2026; Received in revised form 22 April 2026; Accepted 12 May 2026
Available online 23 May 2026

Hyperelasticity

In addition, we demonstrate the accuracy and robustness in FE simulations using benchmark problems such as Cook's membrane. The results underscore the flexibility, physical plausibility, and numerical stability of vCANNs as a powerful class of constitutive models for modern FE simulations enhanced by machine learning.

1. Introduction

Constitutive models are the backbone of any finite element (FE) simulation, providing accurate representations of the complex mechanical behavior of various materials. This pivotal role allows researchers to tailor simulations to a wide range of materials, from metals [1] and polymers [2] to biological tissues [3,4]. Despite well-established experimental evidence confirming the nonlinear viscoelastic nature of many materials [5–9], most commercial FE software packages offer only pre-implemented linear viscoelastic material models [10–13]. This mismatch often forces users to develop, calibrate, and implement appropriate nonlinear viscoelastic constitutive models themselves—a process requiring extensive expert knowledge that is time-consuming, expensive, and error-prone. In response to these challenges, this work establishes a computational pipeline that fully automates all steps of viscoelastic constitutive modeling from model discovery to FE implementation.

In recent decades, scientists have developed numerous constitutive models for specific material classes [14,15]. Yet, there is still no universally accepted rationale to guide model selection, making it largely dependent on subjective criteria. The emerging field of data-driven constitutive modeling promises to address this problem by generating constitutive models in a largely automated and less biased manner.

The model-free data-driven approach by [16] formulates computations directly from experimental material data by minimizing the distance in phase space between states satisfying conservation laws (compatibility and equilibrium) and a material dataset, eliminating empirical material modeling entirely [17]. This framework has since been extended to viscoelastic solids in the frequency domain [18].

Symbolic regression aims to find a closed-form mathematical model that best fits a given dataset by minimizing model error and controlling complexity through systematic exploration of simple analytical expressions [19–22]. In the specific case of sparse regression, expressions are represented as sparse linear combinations of fixed, generally nonlinear candidate terms in a pre-defined library [23]. The expressiveness of this library depends on user expertise in selecting appropriate candidate terms. Sparse regression is used, for example, in the EUCLID framework [24] to identify material models from full-field data, and has been extended to linear viscoelasticity [25]. Recently, [26] systematically paired sparse regression algorithms with model selection criteria to automate the discovery process for material models.

Neural networks have become widely used in constitutive modeling due to their strong approximation capabilities [27]. However, purely data-driven networks may violate basic physical requirements, motivating physics-informed designs that incorporate theoretical structure. While some approaches enforce these constraints via the loss function [28,29], others enforce them by architectural design. Among the latter are Constitutive Artificial Neural Networks (CANNs) [30–32], initially developed to discover strain energy functions of anisotropic hyperelasticity and subsequently applied to various problems, including quasi-linear viscoelasticity of brain tissue [33] and other biomechanical applications [34,35]. For automated FE integration of anisotropic, hyperelastic CANNs, [36] proposed a material subroutine.

Along similar lines, Physics-augmented Neural Networks (PANNs) [37,38], have been extended to small-strain viscoelasticity [39] and, more recently, to incompressible finite strain viscoelasticity [40] within the generalized standard materials (GSM) framework, with a focus on isotropic materials. Other data-driven approaches to isotropic viscoelasticity within the GSM framework were previously proposed, both in the small-strain regime [41] and finite strain regime [42]. More recently, [43] extended this line of work to anisotropic viscoelasticity at finite strains, relaxing the convexity requirement on the dual potential. Constitutive Kolmogorov–Arnold Networks (CKANs) [44,45] have emerged as a promising approach, combining the flexibility of neural networks with the interpretability of symbolic methods through post-processing symbolification. Building upon [44], [46] extended the CKAN approach to isotropic finite strain viscoelasticity within the GSM framework.

An approach to anisotropic finite strain viscoelasticity, representing (dissipation) potentials by neural ordinary differential equations, was proposed in [47]. Building upon the framework of [48], [49] developed a spline-based approach for isotropic finite viscoelasticity in which both the free energy and the creep potential are represented by B-spline interpolants, calibrated through Finite Element Model Updating using full-field data, providing an open-source FE implementation for the isotropic setting.

A dedicated extension of CANNs to viscoelasticity—viscoelastic constitutive artificial neural networks (vCANNs)—was introduced by [50] to automatically learn nonlinear viscoelastic models from experimental data, demonstrating strong capability in capturing the viscoelastic response of synthetic and biological materials across a wide range of loading scenarios. Their formulation is rooted in the classical viscoelasticity framework of Simo [51], which has become a standard in computational mechanics. Simo's foundational model has inspired numerous extensions [52–55]. However, several studies have reported non-physical energy growth and finite-time blow-up in this model family, raising concerns about its thermodynamic soundness [56]. Recently, [57] resolved these issues through a thermodynamically consistent reformulation, successfully adopted in [58] and enabling robust FE simulations [59,60]. Notably, most existing data-driven approaches to finite-strain viscoelasticity—including [40,42,43,46,47]—rely on the multiplicative split of the deformation gradient $\mathbf{F} = \mathbf{F}_e\mathbf{F}_v$. In contrast, the theoretical framework by [51,52], and its rectification by [57] avoid this

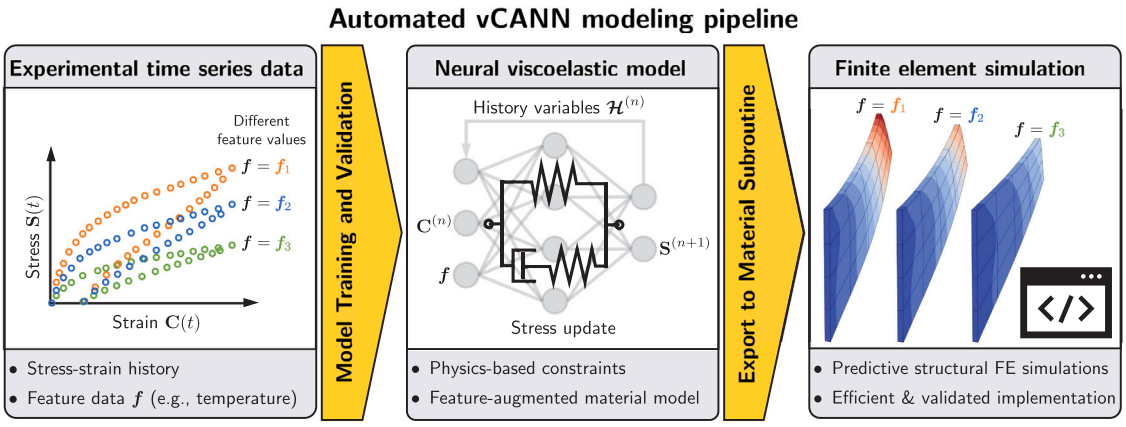


Fig. 1. Automated vCANN modeling pipeline from time-series experiments to predictive finite element (FE) simulations. Left: experimental stress–strain histories are provided as time series, here shown as stress $\mathbf{S}(t)$ versus strain $\mathbf{C}(t)$ in a loading–unloading test for different auxiliary feature values $\boldsymbol{f}$ (e.g., temperature). Middle: a neural viscoelastic constitutive model (vCANN) is trained and validated on complete loading histories to account for history dependence via internal variables $\boldsymbol{\mathcal{H}}$; the learned constitutive law maps the current state $\mathbf{C}^{(n)}$ at time $t^{(n)}$ together with $\boldsymbol{f}$ and history variable $\boldsymbol{\mathcal{H}}^{(n)}$ to the stress response $\mathbf{S}^{(n+1)}$ at time $t^{(n+1)}$ while enforcing physics-based constraints. Right: the trained model is automatically exported to an material subroutine and deployed in FE simulations, enabling predictions of structural stress and displacement fields for different feature values $\boldsymbol{f}$.

split entirely, offering well-known advantages in FE implementation efficiency and the rather uncomplicated treatment of anisotropy. For a thorough discussion on the advantages and disadvantages of the two modeling approaches, we refer to [57].

Despite these advances, no existing framework simultaneously achieves thermodynamic consistency within the rectified formulation of [57], parametrization by arbitrary auxiliary features within this thermodynamically consistent setting, and automated deployment into commercial FE codes without reliance on automatic differentiation. Building upon these theoretical advances of [57], we therefore propose a thermodynamically consistent extension of the vCANN framework [50]. Beyond ensuring physical admissibility, the new formulation allows seamless incorporation of arbitrary auxiliary features—such as temperature, microstructural descriptors, or processing parameters—directly into the neural constitutive law. This enables a single model family to learn highly complex material behavior not only from stress–strain history data but also from diverse data sources, including descriptors that are not inherently physics-based.

While the thermodynamically sound formulation of the proposed framework is a key prerequisite for robust FE simulations, practical use requires reliable deployment of trained models in standard FE environments. We therefore emphasize the automated translation of trained vCANNs—including their learned architecture and parameters—into material subroutines with minimal user intervention. To this end, we develop a dedicated interface that enables seamless integration into standard FE codes. Unlike approaches relying on automatic differentiation, this implementation is directly compatible with standard commercial FE environments and is verified across multiple element formulations, thereby substantially enhancing the practical applicability of vCANNs. As a proof of concept, we demonstrate the workflow in Abaqus [10]; importantly, the approach is designed to be transferable to any FE code supporting custom material routines.

Objective and outline. In this paper, we (i) present a thermodynamically consistent extension of vCANNs for anisotropic nonlinear viscoelasticity that incorporates auxiliary material features, (ii) develop an end-to-end pipeline that learns these models from time-series data and automatically deploys them as FE material subroutines, and (iii) validate the approach in representative material tests and benchmark simulations across different element formulations. Overall, we provide a practical framework for automated modeling and predictive simulation of complex anisotropic nonlinear viscoelastic materials. Fig. 1 illustrates the overall workflow.

This paper is organized as follows. Section 2 introduces the necessary background from continuum mechanics and outlines the constitutive modeling framework of vCANNs. Section 3 presents its implementation in a machine-learning architecture, and Section 4 describes the automated modeling pipeline. Section 5 demonstrates the methodology in numerical experiments and on real-world data. Finally, Section 6 summarizes the main findings and discusses future directions.

Notation. Throughout this work, lowercase italic letters a denote scalars, bold lowercase italic letters $\boldsymbol{a}$ denote vectors, and bold uppercase upright letters $\mathbf{A}$ denote second-order tensors. Fourth-order tensors are written with blackboard symbols, e.g., $\mathbb{A}$. The standard tensor (dyadic) product of two vectors is given by $\boldsymbol{a}\otimes\boldsymbol{b} = a_i b_j \boldsymbol{e}_i \otimes \boldsymbol{e}_j$, where $\{\boldsymbol{e}_i\}_{i=1}^3$ denote Cartesian basis vectors, and the Einstein summation convention applies to all repeated indices. The single contraction of two second-order tensors $\mathbf{A}$ and $\mathbf{B}$ is written $\mathbf{AB} = A_{ik}B_{kj}\boldsymbol{e}_i\otimes\boldsymbol{e}_j$, and their double contraction is written $\mathbf{A} : \mathbf{B} = A_{ij}B_{ij}$. The transpose, inverse, trace, determinant, and cofactor of a second-order tensor $\mathbf{A}$ are written $\mathbf{A}^{\mathrm{T}}$, $\mathbf{A}^{-1}$, $\mathrm{tr}(\mathbf{A})$, $\det(\mathbf{A})$, and $\mathrm{cof}(\mathbf{A}) = \det(\mathbf{A})\mathbf{A}^{-\mathrm{T}}$, respectively. The norm of a second-order tensor $\mathbf{A}$ is defined by $\|\mathbf{A}\| := \mathrm{tr}(\mathbf{A}\mathbf{A}^{\mathrm{T}})^{1/2}$. The second-order identity tensor is $\mathbf{I} = \delta_{ij}\boldsymbol{e}_i\otimes\boldsymbol{e}_j$, where δ_{ij} is the Kronecker delta. Following [61], fourth-order tensors can be constructed from second-order tensors $\mathbf{A}$ and $\mathbf{B}$ by means of two tensor products defined through their

action on an arbitrary second-order tensor $\mathbf{C}$ as $(\mathbf{A}\odot\mathbf{B}):\mathbf{C}=\mathbf{A}(\mathbf{B}:\mathbf{C})$, and $(\mathbf{A}\otimes\mathbf{B}):\mathbf{C}=\mathbf{ACB}$; we note that these definitions may differ from those found elsewhere in the literature. This naturally induces the basis $\{\boldsymbol{e}_i\otimes\boldsymbol{e}_j\odot\boldsymbol{e}_k\otimes\boldsymbol{e}_l\}_{i,j,k,l=1}^{3}$ for fourth-order tensors. For finite element implementations, the component representations of the following tensor products with respect to this basis are important: $[\mathbf{A}\odot\mathbf{B}]_{ijkl}=A_{ij}B_{kl}$, $[\mathbf{A}\otimes\mathbf{B}]_{ijkl}=A_{ik}B_{lj}$, and $[(\mathbf{A}\otimes\mathbf{B})^t]_{ijkl}=A_{il}B_{kj}$, where $(\cdot)^t$ denotes the transposition of a fourth-order tensor. The double contraction of a fourth-order tensor $\mathbb{A}$ in the above basis with a second-order tensor $\mathbf{B}$ is $\mathbb{A}:\mathbf{B}=A_{ijkl}B_{kl}\,\boldsymbol{e}_i\otimes\boldsymbol{e}_j$. The symmetrization of a fourth-order tensor $\mathbb{A}$ is defined as $\mathbb{A}^{\mathrm{s}}:=\frac{1}{2}(\mathbb{A}+\mathbb{A}^t)$. The symmetric fourth-order identity tensor is $\mathbb{I}^{\mathrm{s}}=(\mathbf{I}\otimes\mathbf{I})^{\mathrm{s}}=\frac{1}{2}(\delta_{ik}\delta_{jl}+\delta_{il}\delta_{jk})\,\boldsymbol{e}_i\otimes\boldsymbol{e}_j\odot\boldsymbol{e}_k\otimes\boldsymbol{e}_l$, and $\mathbb{P}:=\mathbb{I}^{\mathrm{s}}-\frac{1}{3}\mathbf{C}^{-1}\odot\mathbf{C}$ denotes the Lagrangian deviatoric projection tensor. The material time derivative of a quantity $(\cdot)$ is denoted $\dot{(\cdot)}:=\mathrm{d}(\cdot)/\mathrm{d}t$. Partial derivatives of a function f with respect to a tensor argument $\mathbf{A}$ are written $\partial f/\partial\mathbf{A}$. Quantities carrying an overline, such as $\bar{\mathbf{F}}$ and $\bar{\mathbf{C}}$, refer to their isochoric counterparts obtained via the multiplicative volumetric–isochoric split. Stress and material tangent tensors carrying an overline are also referred to as fictitious quantities. Quantities carrying a tilde, such as $\tilde{\mathbf{L}}_r$ and $\tilde{\bar{I}}_r$, are associated with the generalized structural tensors introduced to characterize material anisotropy. The superscript $(\cdot)^{\infty}$ indicates equilibrium quantities and $(\cdot)^{\mathrm{neq}}$ non-equilibrium quantities.

2. Theoretical background

2.1. Kinematics

Most soft materials of interest exhibit markedly different responses in volumetric and distortional deformation, especially at large strains. To reflect this separation and to conveniently enforce (near-)incompressibility, we adopt Flory's multiplicative volumetric–isochoric split of the deformation gradient [62,63]. Starting from the deformation gradient $\mathbf{F}$ with $J=\det\mathbf{F}>0$, we define the right Cauchy–Green tensor $\mathbf{C}=\mathbf{F}^{\mathrm{T}}\mathbf{F}$ and decompose $\mathbf{F}$ into a purely volumetric part $J^{1/3}\mathbf{I}$ and an isochoric part

$$\bar{\mathbf{F}}=J^{-1/3}\mathbf{F}\qquad\text{with}\qquad\det\bar{\mathbf{F}}=1. \tag{1}$$

The corresponding isochoric right Cauchy–Green deformation tensor $\bar{\mathbf{C}}$ is defined by

$$\bar{\mathbf{C}}=\bar{\mathbf{F}}^{\mathrm{T}}\bar{\mathbf{F}}=J^{-2/3}\mathbf{C}\qquad\text{with}\qquad\det\bar{\mathbf{C}}=1. \tag{2}$$

We introduce the following useful differentiation relation

$$\frac{\partial\bar{\mathbf{C}}}{\partial\mathbf{C}}=J^{-2/3}\,\mathbb{P}^{\mathrm{T}}. \tag{3}$$

2.2. Material symmetry

To characterize the anisotropy of a material, we introduce preferred material directions represented by the unit direction vectors $\boldsymbol{m}_i\in\mathbb{R}^3$, $i=1,2,\dots,n$, and define the structural tensors

$$\mathbf{L}_0=\frac{1}{3}\mathbf{I},\qquad\mathbf{L}_i=\boldsymbol{m}_i\otimes\boldsymbol{m}_i,\qquad i=1,2,\dots,n, \tag{4}$$

where $\mathbf{L}_0$ is associated with the isotropic part of the material's constitutive behavior. The preferred material directions $\boldsymbol{m}_i$ typically represent directions of reinforcing fibers embedded in a matrix material in the reference configuration.

By forming convex combinations of the structural tensors $\mathbf{L}_i$ (4), we define R symmetric, positive semi-definite generalized structural tensors $\tilde{\mathbf{L}}_r$:

$$\tilde{\mathbf{L}}_r=\sum_{i=0}^{n}w_i^{(r)}\mathbf{L}_i,\quad\text{with}\quad\sum_{i=0}^{n}w_i^{(r)}=1\quad\text{and}\quad w_i^{(r)}\geq 0,\qquad r=1,2,\dots,R, \tag{5}$$

where $w_i^{(r)}$ are scalar weight factors. Using the generalized structural tensors $\tilde{\mathbf{L}}_r$ (5), we define the generalized invariants of $\bar{\mathbf{C}}$ and $\dot{\bar{\mathbf{C}}}=J^{-2/3}\,\mathbb{P}^{\mathrm{T}}:\dot{\mathbf{C}}$ [64]:

$$\tilde{\bar{I}}_r=\mathrm{tr}[\bar{\mathbf{C}}\tilde{\mathbf{L}}_r],\qquad\tilde{\bar{J}}_r=\mathrm{tr}[\bar{\mathbf{C}}^{-1}\tilde{\mathbf{L}}_r],\qquad\mathrm{III}_{\bar{\mathbf{C}}}=\det\bar{\mathbf{C}}=1 \tag{6}$$

$$\dot{\tilde{\bar{I}}}_r=\mathrm{tr}[\dot{\bar{\mathbf{C}}}\tilde{\mathbf{L}}_r],\qquad\dot{\tilde{\bar{J}}}_r=\mathrm{tr}[(\mathrm{cof}\,\dot{\bar{\mathbf{C}}})\tilde{\mathbf{L}}_r],\qquad\mathrm{III}_{\dot{\bar{\mathbf{C}}}}=\det\dot{\bar{\mathbf{C}}},\qquad r=1,2,\dots,R. \tag{7}$$

The following differentiation relations will be useful throughout the remainder

$$\frac{\partial\tilde{\bar{I}}_r}{\partial\bar{\mathbf{C}}}=\tilde{\mathbf{L}}_r,\qquad\frac{\partial\tilde{\bar{J}}_r}{\partial\bar{\mathbf{C}}}=-\bar{\mathbf{H}}_r,\qquad\frac{\partial J}{\partial\mathbf{C}}=\frac{1}{2}J\mathbf{C}^{-1}\qquad\text{with}\qquad\bar{\mathbf{H}}_r:=\bar{\mathbf{C}}^{-1}\tilde{\mathbf{L}}_r\bar{\mathbf{C}}^{-1}. \tag{8}$$

Finally, we introduce the following shorthand notations in terms of tuples:

$$\tilde{\bar{I}}_r:=\big(\tilde{\bar{I}}_r,\dot{\tilde{\bar{I}}}_r\big)\qquad\text{with}\qquad\tilde{\bar{I}}_r:=\big(\tilde{\bar{I}}_r,\tilde{\bar{J}}_r\big),\qquad\dot{\tilde{\bar{I}}}_r:=\big(\dot{\tilde{\bar{I}}}_r,\dot{\tilde{\bar{J}}}_r,\mathrm{III}_{\dot{\bar{\mathbf{C}}}}\big),\qquad r=1,2,\dots,R, \tag{9}$$

$$\tilde{\bar{I}}:=\big(\tilde{\bar{I}},\dot{\tilde{\bar{I}}}\big)\qquad\text{with}\qquad\tilde{\bar{I}}:=\big(\tilde{\bar{I}}_1,\dots,\tilde{\bar{I}}_R\big),\qquad\dot{\tilde{\bar{I}}}:=\big(\dot{\tilde{\bar{I}}}_1,\dots,\dot{\tilde{\bar{I}}}_R\big). \tag{10}$$

The response of an anisotropic material must be invariant under orthogonal transformations $\mathbf{Q}$ in the symmetry group $\mathcal{G}$ of the material. Using the generalized structural tensors (5), we define the symmetry group by

$$\mathcal{G}:=\{\mathbf{Q}\in\mathbf{O}\mathrm{rth}:\mathbf{Q}\tilde{\mathbf{L}}_r\mathbf{Q}^{\mathrm{T}}=\tilde{\mathbf{L}}_r,\quad r=1,2,\dots,R\}, \tag{11}$$

where $\mathbf{O}\mathrm{rth} := \{\mathbf{Q} \in \mathbb{R}^{3\times3} : \mathbf{Q}^{-1} = \mathbf{Q}^{\mathrm{T}}\}$ denotes the orthogonal group. For anisotropic materials, described by a free energy function Ψ, the condition of material symmetry is satisfied if and only if the free energy can be represented as an isotropic tensor function of arguments containing the structural tensors as [61]

$$\Psi(\bar{\mathbf{C}}, \tilde{\mathbf{L}}_r) = \Psi(\mathbf{Q}\bar{\mathbf{C}}\mathbf{Q}^{\mathrm{T}}, \mathbf{Q}\tilde{\mathbf{L}}_r\mathbf{Q}^{\mathrm{T}}), \quad r = 1, 2, \dots, R, \quad \forall \mathbf{Q} \in \mathbf{O}\mathrm{rth}, \tag{12}$$

which can be achieved by formulating Ψ in terms of the generalized invariants $\bar{\tilde{I}}_r$.

For $R = 1$, the anisotropy class is completely governed by the number of distinct eigenvalues of $\tilde{\mathbf{L}}_1$: three distinct eigenvalues yield orthotropy, two coinciding eigenvalues yield transverse isotropy, and all three equal yields isotropy. With $R \geq 2$, triclinic and monoclinic symmetries can additionally be described, depending on the relative orientations of the eigenframes and their eigenvalue multiplicities; formal conditions are given in Theorem 8.6 of [65]. While $R = 2$ suffices to represent all five anisotropy classes, choosing $R > 2$ can be meaningful when individual structural tensors are assigned to mechanically distinct constituents, e.g., separate fiber families. Representing further anisotropy classes necessitates higher-order structural tensors [66]

Remark 1. From the perspective of invariant theory [67,68], the invariant set $\bar{\mathcal{I}}$ utilized in this framework does not, in general, constitute a functional basis. Specifically, the invariants $\mathrm{tr}[\bar{\mathbf{C}}\tilde{\mathbf{L}}_r^2]$ and $\mathrm{tr}[\bar{\mathbf{C}}^{-1}\tilde{\mathbf{L}}_r^2]$ are omitted. Furthermore, for multiple generalized structural tensors ($R \geq 2$), a functional basis also requires the structural coupling invariants $\mathrm{tr}[\bar{\mathbf{C}}\tilde{\mathbf{L}}_i\tilde{\mathbf{L}}_j]$, $1 \leq i < j \leq R$.

However, within the context of our data-driven formulation, we deliberately omit these invariants for three pragmatic reasons: (i) each additional invariant increases the dimensionality of the input space the neural network must map. This is particularly disadvantageous in data-driven mechanics, where comprehensive multi-axial experimental data are typically scarce. Restricting the inputs acts as an inductive bias that prevents the network from overfitting; (ii) the inclusion of coupling invariants can complicate the construction of polyconvex potentials and compromise the numerical stability of finite element simulations; and (iii) this conscious omission is justified by established literature, which demonstrates that the remaining invariants are sufficient for accurate modeling in practice (e.g., [60,69,70]).

2.3. Constitutive modeling framework

The constitutive modeling framework herein builds on [57] and is developed within the theory of irreversible thermodynamics with internal variables. While [57] proposes a thermomechanical theory, we restrict ourselves to the isothermal case to simplify the discussion. We introduce internal variables $\boldsymbol{\Gamma}_{r\alpha}$, $r = 1, \dots, R$, $\alpha = 1, \dots, N$, which are akin to the right Cauchy–Green tensor, and collect them in the tuple $\boldsymbol{\Gamma} := (\boldsymbol{\Gamma}_{r\alpha})$. The meaning of the indices will become clear below. We assume the existence of a total free energy function $\Psi = \Psi\big(\mathbf{C}, \boldsymbol{\Gamma}\big)$ that must satisfy the dissipation inequality [71, p. 173],

$$\mathcal{D} = \frac{1}{2}\,\mathbf{S} : \dot{\mathbf{C}} - \dot{\Psi} \geq 0, \tag{13}$$

where D denotes the non-negative internal dissipation density and $\mathbf{S} = \mathbf{S}(\mathbf{C}, \boldsymbol{\Gamma})$ is the total second Piola–Kirchhoff stress. The evolution of the internal variables is governed by a set of yet-to-be-determined evolution equations of the form $\dot{\boldsymbol{\Gamma}}_{r\alpha} = \dot{\boldsymbol{\Gamma}}_{r\alpha}(\mathbf{C}, \boldsymbol{\Gamma})$.

The multiplicative split (1) suggests an additive decomposition of the free energy into a volumetric equilibrium part $\Psi_{\mathrm{vol}}^{\infty}$ and an isochoric part Ψ_{iso}, as commonly employed for (nearly) incompressible materials,

$$\Psi\big(\mathbf{C}, \boldsymbol{\Gamma}\big) = \Psi_{\mathrm{vol}}^{\infty}(J) + \Psi_{\mathrm{iso}}\big(\bar{\mathbf{C}}, \boldsymbol{\Gamma}\big). \tag{14}$$

Since experimental data suggest that viscous effects in rubber and soft biological tissues are primarily due to isochoric deformations [72,73], we model the volumetric free energy $\Psi_{\mathrm{vol}}^{\infty}$ as fully elastic and independent of the internal variables. Accordingly, we attribute viscous effects exclusively to the isochoric part Ψ_{iso}. As commonly adopted [74], we further decompose Ψ_{iso} into equilibrium contributions $\Psi_{\mathrm{iso}\,r}^{\infty}$ and non-equilibrium contributions $\Psi_{r\alpha}^{\mathrm{neq}}$:

$$\Psi_{\mathrm{iso}}\big(\bar{\mathbf{C}}, \boldsymbol{\Gamma}\big) = \sum_{r=1}^{R}\left[\Psi_{\mathrm{iso}\,r}^{\infty}(\bar{\mathbf{C}}) + \sum_{\alpha=1}^{N}\Psi_{r\alpha}^{\mathrm{neq}}\big(\bar{\mathbf{C}}, \boldsymbol{\Gamma}_{r\alpha}\big)\right]. \tag{15}$$

The generalized Maxwell model depicted in Fig. 2 provides an illustrative interpretation of (15): in this analogy, $\Psi_{\mathrm{iso}\,r}^{\infty}$ corresponds to an equilibrium spring, and each $\Psi_{r\alpha}^{\mathrm{neq}}$ corresponds to an individual Maxwell element. Thus, (15) represents R generalized Maxwell models wired in parallel, each consisting of N Maxwell elements, which offers wide flexibility in describing experimental data. In practice, however, often only a single term ($R = 1$) proves sufficient.

Inserting the total free energy (14) into the dissipation inequality (13) and expanding $\dot{\Psi}$ via the chain rule yields

$$\mathcal{D} = \frac{1}{2}\left[\mathbf{S} - \underbrace{2\frac{\partial\Psi_{\mathrm{vol}}^{\infty}}{\partial\mathbf{C}}}_{=:\,\mathbf{S}_{\mathrm{vol}}^{\infty}} - \sum_{r=1}^{R}\left(\underbrace{2\frac{\partial\Psi_{\mathrm{iso}\,r}^{\infty}}{\partial\mathbf{C}}}_{=:\,\mathbf{S}_{\mathrm{iso}\,r}^{\infty}} + \sum_{\alpha=1}^{N}\underbrace{2\frac{\partial\Psi_{r\alpha}^{\mathrm{neq}}}{\partial\mathbf{C}}}_{=:\,\mathbf{S}_{r\alpha}^{\mathrm{neq}}}\right)\right] : \dot{\mathbf{C}} - \sum_{r=1}^{R}\sum_{\alpha=1}^{N}\frac{\partial\Psi_{r\alpha}^{\mathrm{neq}}}{\partial\boldsymbol{\Gamma}_{r\alpha}} : \dot{\boldsymbol{\Gamma}}_{r\alpha} \geq 0 \qquad \forall\,\dot{\mathbf{C}}, \mathbf{C}, \boldsymbol{\Gamma}. \tag{16}$$

Since (16) must hold for all $\dot{\mathbf{C}}$, standard thermodynamic arguments by Coleman, Noll, and Gurtin [75,76] require the bracketed term in (16) to vanish identically, yielding the constitutive relation for the total stress,

$$\mathbf{S} = 2\frac{\partial\Psi}{\partial\mathbf{C}} = \mathbf{S}_{\mathrm{vol}}^{\infty} + \sum_{r=1}^{R}\underbrace{\left(\mathbf{S}_{\mathrm{iso}\,r}^{\infty} + \sum_{\alpha=1}^{N}\mathbf{S}_{r\alpha}^{\mathrm{neq}}\right)}_{=:\,\mathbf{S}_{\mathrm{iso}\,r}}. \tag{17}$$

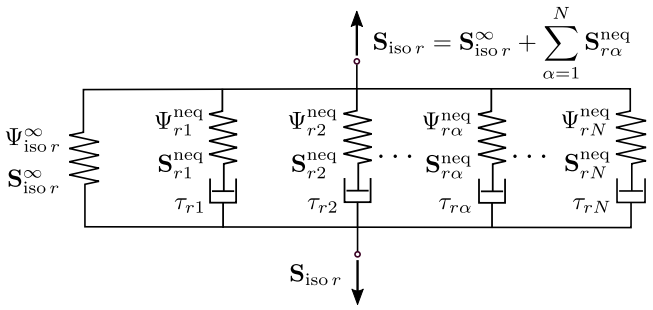


Fig. 2. The generalized Maxwell model causes the total isochoric stress $\mathbf{S}_{\mathrm{iso}\,r}$: the spring on the left symbolizes the equilibrium free energy $\Psi^{\infty}_{\mathrm{iso}\,r}$ and stress $\mathbf{S}^{\infty}_{\mathrm{iso}\,r}$; each Maxwell element is characterized by a non-equilibrium free energy $\Psi^{\mathrm{neq}}_{r\alpha}$ and a strain (rate)-dependent relaxation time $\tau_{r\alpha}$ and produces a non-equilibrium stress $\mathbf{S}^{\mathrm{neq}}_{r\alpha}$.

Here, $\mathbf{S}^{\infty}_{\mathrm{vol}}$ and $\mathbf{S}^{\infty}_{\mathrm{iso}\,r}$ denote volumetric and isochoric equilibrium stresses, respectively, and $\mathbf{S}^{\mathrm{neq}}_{r\alpha}$ are non-equilibrium stresses. Other stress measures, such as the first Piola–Kirchhoff stress $\mathbf{P} = \mathbf{F}\mathbf{S}$ or Cauchy stress $\boldsymbol{\sigma} = J^{-1}\mathbf{F}\mathbf{S}\mathbf{F}^{\mathrm{T}}$, follow accordingly.

Introducing the auxiliary non-equilibrium stress $\mathbf{Q}_{r\alpha}$, the remaining dissipative term in (16) yields the reduced dissipation inequality,

$$\sum_{r=1}^{R}\sum_{\alpha=1}^{N} \mathbf{Q}_{r\alpha} : \frac{1}{2}\dot{\boldsymbol{\Gamma}}_{r\alpha} \geq 0 \qquad \forall\, \mathbf{C}, \boldsymbol{\Gamma}, \qquad \text{with} \qquad \mathbf{Q}_{r\alpha} = -2\frac{\partial \Psi^{\mathrm{neq}}_{r\alpha}}{\partial \boldsymbol{\Gamma}_{r\alpha}}. \tag{18}$$

In the following, we specify functional forms for the free energy contributions that are suitable for integration into a machine learning framework and derive evolution equations for the internal variables that satisfy the reduced dissipation inequality (18).

2.3.1. Anisotropic hyperelasticity

The material response in thermodynamic equilibrium is described by anisotropic hyperelasticity through the equilibrium free energy contributions $\Psi^{\infty}_{\mathrm{vol}}$ and $\Psi^{\infty}_{\mathrm{iso}\,r}$. A sufficient condition for satisfying material objectivity [77] and preserving material symmetry [78] is that the isochoric equilibrium free energy $\Psi^{\infty}_{\mathrm{iso}\,r}$ depends on the generalized invariants (6) [64]. The appropriate functional form of $\Psi^{\infty}_{\mathrm{iso}\,r}$ generally depends on material descriptors such as composition, microstructure, or processing conditions; we collect this information in a feature vector $\boldsymbol{f}$. The volumetric contribution $\Psi^{\infty}_{\mathrm{vol}}$ is used primarily to enforce (near-)incompressibility in a convenient and largely material-agnostic manner [79]; accordingly, we do not parameterize $\Psi^{\infty}_{\mathrm{vol}}$ by $\boldsymbol{f}$ and thus write the total equilibrium free energy as

$$\Psi^{\infty}(\bar{\mathbf{C}}, \boldsymbol{f}) = \Psi^{\infty}_{\mathrm{vol}}(J) + \underbrace{\sum_{r=1}^{R} \Psi^{\infty}_{\mathrm{iso}\,r}\big(\bar{\tilde{I}}_r, \boldsymbol{f}\big)}_{=:\Psi^{\infty}_{\mathrm{iso}\,r}}. \tag{19}$$

This means that each $\Psi^{\infty}_{\mathrm{iso}\,r}$ is associated with different structural tensor $\tilde{\mathbf{L}}_r$. In view of (17), the total equilibrium stress $\mathbf{S}^{\infty}$ follows as

$$\mathbf{S}^{\infty} = 2\frac{\partial \Psi^{\infty}}{\partial \mathbf{C}} = \mathbf{S}^{\infty}_{\mathrm{vol}} + \sum_{r=1}^{R} \mathbf{S}^{\infty}_{\mathrm{iso}\,r} \tag{20}$$

and using (3) and $(8)_3$, the volumetric and isochoric parts can be expressed as

$$\mathbf{S}^{\infty}_{\mathrm{vol}} = 2\frac{\partial \Psi^{\infty}_{\mathrm{vol}}}{\partial \mathbf{C}} = p\,\mathbf{C}^{-1}, \tag{21}$$

$$\mathbf{S}^{\infty}_{\mathrm{iso}\,r} = 2\frac{\partial \Psi^{\infty}_{\mathrm{iso}\,r}}{\partial \mathbf{C}} = J^{-2/3}\mathbb{P} : \bar{\mathbf{S}}^{\infty}_{\mathrm{iso}\,r}, \tag{22}$$

where p denotes the hydrostatic pressure and $\bar{\mathbf{S}}^{\infty}_{\mathrm{iso}}$ is a fictitious isochoric stress. In view of $(8)_{1,2}$, the latter is given by

$$\bar{\mathbf{S}}^{\infty}_{\mathrm{iso}\,r} = 2\frac{\partial \Psi^{\infty}_{\mathrm{iso}\,r}}{\partial \bar{\mathbf{C}}} = 2\left[\frac{\partial \Psi^{\infty}_{\mathrm{iso}}}{\partial \bar{\tilde{I}}_r}\tilde{\mathbf{L}}_r - \frac{\partial \Psi^{\infty}_{\mathrm{iso}}}{\partial \bar{\tilde{J}}_r}\tilde{\mathbf{H}}_r\right]. \tag{23}$$

The volumetric contribution $\Psi^{\infty}_{\mathrm{vol}}$ is handled differently for nearly and perfectly incompressible formulations. For nearly incompressible materials, $\Psi^{\infty}_{\mathrm{vol}}$ is typically prescribed as a generic convex penalty function to approximately enforce incompressibility [80, 81], yielding the hydrostatic pressure

$$p = J\frac{\partial \Psi^{\infty}_{\mathrm{vol}}}{\partial J}. \tag{24}$$

For perfect incompressibility, $\Psi^{\infty}_{\mathrm{vol}}(J) = p(J-1)$ enforces the kinematic constraint with p as a Lagrange multiplier [74, Section 8.5]. In both cases, the material-specific equilibrium response is fully contained in $\Psi^{\infty}_{\mathrm{iso}}$.

In the undeformed reference configuration, $\mathbf{C} = \mathbf{I}$, we require the equilibrium free energy and the equilibrium stress to vanish,

$$\Psi^{\infty}|_{\mathbf{C}=\mathbf{I}} = 0, \qquad \mathbf{S}^{\infty}|_{\mathbf{C}=\mathbf{I}} = \mathbf{0}, \tag{25}$$

which is referred to as the normalization condition.

In FE analysis, it is often advantageous that the equilibrium free energy is polyconvex, i.e., convex in $\mathbf{F}$, $\operatorname{cof}\mathbf{F}$, and $\det\mathbf{F}$, which ensures material stability [82]. Two well-known limitations can arise when combining anisotropy with the volumetric–isochoric split. First, for nearly incompressible formulations, neither $\operatorname{tr}(\operatorname{cof}\bar{\mathbf{C}})$ [81, p. 2776] nor $\operatorname{tr}[(\operatorname{cof}\bar{\mathbf{C}})\mathbf{L}_i]$ [83, pp. 435, 438] is polyconvex, and therefore the convex combinations in $(6)_2$ cannot be expected to be polyconvex either. Second, in nearly incompressible settings, employing $\bar{\mathbf{F}}$ in anisotropic free-energy terms may lead to non-physical responses [84]. In the present work, these issues are avoided by considering perfectly incompressible materials only [85].

2.3.2. Nonlinear viscoelasticity

Following [59, (2.2)], the non-equilibrium free energy $\Psi^{\mathrm{neq}}_{r\alpha}$, associated with the α-th relaxation process, is assumed to have the general form

$$\Psi^{\mathrm{neq}}_{r\alpha}(\bar{\mathbf{C}}, \boldsymbol{\Gamma}_{r\alpha}, \boldsymbol{f}) = \frac{1}{4\mu_{r\alpha}} \left\| \bar{\mathbf{S}}_{r\alpha}(\bar{\bar{I}}_r, \boldsymbol{f}) - \mu_{r\alpha}\left(\boldsymbol{\Gamma}_{r\alpha} - \mathbf{I}\right) \right\|^2, \tag{26}$$

where $\mu_{r\alpha}$ is a constant shear modulus. In analogy to (23), the stress $\bar{\mathbf{S}}_{r\alpha}$ is derived from a potential $\Psi_{r\alpha} = \Psi_{r\alpha}(\bar{\bar{I}}_r, \boldsymbol{f})$, which must be tailored to the material under consideration:

$$\bar{\mathbf{S}}_{r\alpha} = 2\frac{\partial \Psi_{r\alpha}}{\partial \bar{\mathbf{C}}} = \sum_{r=1}^{R} 2\left[\frac{\partial \Psi_{r\alpha}}{\partial \bar{\bar{I}}_r}\tilde{\mathbf{L}}_r - \frac{\partial \Psi_{r\alpha}}{\partial \bar{\bar{J}}_r}\bar{\mathbf{H}}_r\right]. \tag{27}$$

Since $\boldsymbol{\Gamma}_{r\alpha} = \mathbf{I}$ in the undeformed reference configuration, we ensure the normalization condition $\Psi^{\mathrm{neq}}_{r\alpha}|_{\mathbf{C}=\boldsymbol{\Gamma}_{r\alpha}=\mathbf{I}} = 0$ by requiring that $\bar{\mathbf{S}}_{r\alpha}$ vanishes in the reference configuration, i.e., $\bar{\mathbf{S}}_{r\alpha}|_{\mathbf{C}=\mathbf{I}} = \mathbf{0}$.

In view of (26), the auxiliary non-equilibrium stress $\mathbf{Q}_{r\alpha}$ (18) read

$$\mathbf{Q}_{r\alpha} = \bar{\mathbf{S}}_{r\alpha} - \mu_{r\alpha}(\boldsymbol{\Gamma}_{r\alpha} - \mathbf{I}). \tag{28}$$

In view of (17), and using (3) and $(8)_3$, we compute the non-equilibrium stress $\mathbf{S}^{\mathrm{neq}}_{r\alpha}$ by differentiating (26):

$$\mathbf{S}^{\mathrm{neq}}_{r\alpha} = 2\frac{\partial \Psi^{\mathrm{neq}}_{r\alpha}}{\partial \mathbf{C}} = J^{-2/3}\mathbb{P} : \bar{\mathbf{S}}^{\mathrm{neq}}_{r\alpha}, \tag{29}$$

where the fictitious non-equilibrium stress is

$$\bar{\mathbf{S}}^{\mathrm{neq}}_{r\alpha} = \frac{1}{2\mu_{r\alpha}}\bar{\mathbb{C}}_{r\alpha} : \mathbf{Q}_{r\alpha}, \qquad \text{with} \qquad \bar{\mathbb{C}}_{r\alpha} = 2\frac{\partial \bar{\mathbf{S}}_{r\alpha}}{\partial \bar{\mathbf{C}}} \tag{30}$$

By differentiating $\bar{\mathbf{S}}_{r\alpha}$ with respect to $\bar{\mathbf{C}}$ and inserting into (30), $\bar{\mathbf{S}}^{\mathrm{neq}}_{r\alpha}$ explicitly reads

$$\begin{aligned}\bar{\mathbf{S}}^{\mathrm{neq}}_{r\alpha} = \frac{2}{\mu_{r\alpha}}\Bigg[&\frac{\partial^2 \Psi_{r\alpha}}{\partial \bar{\bar{I}}_r^2}\varepsilon_{r\alpha}\tilde{\mathbf{L}}_r + \frac{\partial^2 \Psi_{r\alpha}}{\partial \bar{\bar{J}}_r^2}\gamma_{r\alpha}\bar{\mathbf{H}}_r + \frac{\partial \Psi_{r\alpha}}{\partial \bar{\bar{J}}_r}\left(\bar{\mathbf{C}}^{-1}\mathbf{Q}_{r\alpha}\bar{\mathbf{H}}_r + \bar{\mathbf{H}}_r\mathbf{Q}_{r\alpha}\bar{\mathbf{C}}^{-1}\right)\\ &- \frac{\partial^2 \Psi_{r\alpha}}{\partial \bar{\bar{I}}_r \partial \bar{\bar{J}}_r}\left(\gamma_{r\alpha}\tilde{\mathbf{L}}_r + \varepsilon_{r\alpha}\bar{\mathbf{H}}_r\right)\Bigg] \qquad \text{with} \qquad \gamma_{r\alpha} := \bar{\mathbf{H}}_r : \mathbf{Q}_{r\alpha}, \quad \varepsilon_{r\alpha} := \tilde{\mathbf{L}}_r : \mathbf{Q}_{r\alpha}.\end{aligned} \tag{31}$$

From (31) it is evident that the constant material parameter $\mu_{r\alpha}$ can be absorbed by the material-specific potential $\Psi_{r\alpha}$. Thus, without loss of generality, we set $\mu_{r\alpha} = 1$ MPa.

Recalling the reduced dissipation inequality (18), we can guarantee non-negative dissipation by the evolution equation for the internal variables $\boldsymbol{\Gamma}_{r\alpha}$,

$$\dot{\boldsymbol{\Gamma}}_{r\alpha} = \mathbb{V}^{-1}_{r\alpha} : \mathbf{Q}_{r\alpha} = \frac{1}{\eta_{r\alpha}}[\bar{\mathbf{S}}_{r\alpha} - \mu_{r\alpha}(\boldsymbol{\Gamma}_{r\alpha} - \mathbf{I})], \tag{32}$$

where $\mathbb{V}^{-1}_{r\alpha} = \frac{1}{\eta_{r\alpha}}\mathbb{I}^s$ is the inverse fourth-order viscosity tensor and $\eta_{r\alpha}$ denote scalar viscosities. Evolution equation (32) implies an isotropic viscosity tensor, which is in practice often sufficient [60,86,87]. Nevertheless, the non-equilibrium response exhibits direction-dependent relaxation amplitudes, since the driving force $\mathbf{Q}_{r\alpha}$ in (28) inherits anisotropy from the potentials $\Psi_{r\alpha}$. A generalization to anisotropic viscosity tensors—constructed from structural tensors—is conceptually straightforward but deferred to future work: while additional viscosity parameters would likely improve fits, ensuring that this improvement reflects genuinely learned anisotropic viscosity—rather than merely increased fitting capacity—requires significantly more experimental data than is typically available [88]. The present choice, therefore, prioritizes parsimony.

Since numerous experiments confirm the strain (rate)-dependent viscous character of many important materials, such as elastomers [89,90] and soft biological tissues [6,91], we use strain (rate)-dependent viscosities $\eta_{r\alpha} = \eta_{r\alpha}(\bar{I}_r, \boldsymbol{f})$. We can rewrite the evolution equations by taking the material time derivative on both sides of (28) and (32), and by combining the resulting equations, we obtain

$$\dot{\mathbf{Q}}_{r\alpha} + \frac{\mathbf{Q}_{r\alpha}}{\tau_{r\alpha}} = \dot{\bar{\mathbf{S}}}_{r\alpha}, \tag{33}$$

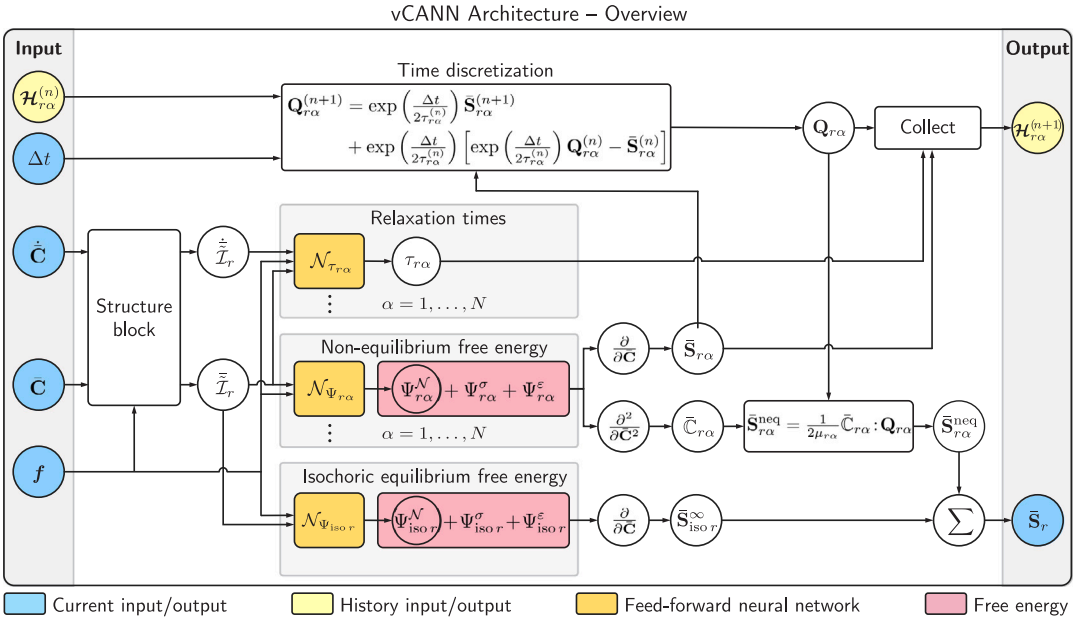


Fig. 3. vCANN architecture: the strain (rate) tensors $\bar{\mathbf{C}}$ and $\dot{\bar{\mathbf{C}}}$, and the feature vector $\boldsymbol{f}$ are input to the structure block, Fig. 4, where generalized structural tensors and invariants are computed. These invariants are fed to the feed-forward neural networks $\mathcal{N}_{\Psi_{\text{iso}\,r}}$, $\mathcal{N}_{\Psi_{r\alpha}}$, and $\mathcal{N}_{\tau_{r\alpha}}$, which learn the free energies $\Psi^{\mathcal{N}}_{\text{iso}\,r}$ and $\Psi^{\mathcal{N}}_{r\alpha}$, and the relaxation times $\tau_{r\alpha}$. The free energies are corrected by $\Psi^{\sigma}_{(\cdot)}$, $\Psi^{\varepsilon}_{(\cdot)}$ to ensure a stress- and energy-free reference configuration. The internal variables $\mathbf{Q}^{(n+1)}_{r\alpha}$ are updated using the time step Δt and the history variables $\boldsymbol{\mathcal{H}}^{(n)}_{r\alpha} = \left(\mathbf{Q}^{(n)}_{r\alpha}, \bar{\mathbf{S}}^{(n)}_{r\alpha}, \tau^{(n)}_{r\alpha}\right)$. Finally, the updated history variables $\boldsymbol{\mathcal{H}}^{(n+1)}_{r\alpha}$ and the current stress $\bar{\mathbf{S}}_r$ are returned. Unless explicitly stated otherwise, all quantities are evaluated at $t^{(n+1)}$.

where $\tau_{r\alpha} = \eta_{r\alpha}(\bar{\mathcal{I}}_r, \boldsymbol{f})/\mu_{r\alpha}$ are non-negative strain (rate)-dependent relaxation times. Additionally, we assume a stress-free reference configuration with the initial conditions $\mathbf{Q}_{r\alpha}|_{t=0} = \mathbf{0}$.

The total stress $\mathbf{S}$ of the proposed vCANN is summarized in Box 1. For comparison, the total stress of the originally proposed vCANN [50] and a brief discussion in the main differences is provided in Appendix A. Appendix B provides the model formulation for compressible materials.

Box 1: Proposed constitutive modeling framework

$$\mathbf{S} = \mathbf{S}^{\infty}_{\text{vol}} + \sum_{r=1}^{R}\left(\mathbf{S}^{\infty}_{\text{iso}\,r} + \sum_{\alpha=1}^{N}\mathbf{S}^{\text{neq}}_{r\alpha}\right), \qquad \mathbf{S}^{\infty}_{\text{iso}\,r} = J^{-2/3}\mathbb{P} : \bar{\mathbf{S}}^{\infty}_{\text{iso}\,r}, \qquad \mathbf{S}^{\text{neq}}_{r\alpha} = J^{-2/3}\mathbb{P} : \bar{\mathbf{S}}^{\text{neq}}_{r\alpha}$$

$$\bar{\mathbf{S}}^{\infty}_{\text{iso}\,r} = \frac{\partial \Psi^{\infty}_{\text{iso}\,r}}{\partial \bar{\mathbf{C}}} \qquad \bar{\mathbf{S}}^{\text{neq}}_{r\alpha} = \frac{1}{2\mu_{r\alpha}}\bar{\mathbb{C}}_{r\alpha} : \mathbf{Q}_{r\alpha}, \qquad \dot{\mathbf{Q}}_{r\alpha} + \frac{\mathbf{Q}_{r\alpha}}{\tau_{r\alpha}} = \dot{\bar{\mathbf{S}}}_{r\alpha}, \qquad \bar{\mathbf{S}}_{r\alpha} = 2\frac{\partial \Psi_{r\alpha}}{\partial \bar{\mathbf{C}}}.$$

3. Thermodynamically consistent vCANN architecture

Having established the constitutive framework of vCANNs, we now embed it in a neural network architecture that enforces the key physical constraints by construction. Specifically, we represent the equilibrium free energies $\Psi^{\infty}_{\text{iso},r}$, the auxiliary free energies $\Psi_{r\alpha}$, the strain (rate)-dependent relaxation times $\tau_{r\alpha}$, and—if feature-dependent—the preferred material directions $\boldsymbol{m}_i$ and corresponding weight factors $w^{(r)}_i$ by separate feed-forward neural networks (FFNNs). This section describes how these sub-networks are designed to satisfy the constitutive requirements summarized in Section 2.3 and outlines the algorithmic update of the internal variables. An overview of the resulting architecture is provided in Fig. 3.

Although we do not limit the depth and width of each sub-network, small networks with a single hidden layer and up to 10 neurons have proven sufficient for most applications. We implemented the vCANN framework using the machine learning library TensorFlow [92].

3.1. Equilibrium free energy

Volumetric part. In the neural network architecture, we do not introduce an FFNN for the volumetric contribution $\Psi^{\infty}_{\text{vol}}$ in (19). Instead, for nearly incompressible simulations, $\Psi^{\infty}_{\text{vol}}(J)$ is prescribed by a standard convex penalty function from the literature [80,81].

By construction, these penalty functions yield an energy- and stress-free reference configuration, i.e., $\Psi^{\infty}_{\mathrm{vol}}(1) = 0$ and $\mathbf{S}^{\infty}_{\mathrm{vol}}|_{\mathbf{C}=\mathbf{I}} = \mathbf{0}$. For perfectly incompressible simulations, $\Psi^{\infty}_{\mathrm{vol}}$ represents the kinematic constraint functions and is handled by a numerical scheme, e.g., the Lagrange-multiplier method. Thus, only the isochoric part is represented by trainable sub-networks.

Isochoric part. The isochoric material behavior in thermodynamic equilibrium is fully characterized by the free energy contributions $\Psi^{\infty}_{\mathrm{iso}\,r}$. These are defined by

$$\Psi^{\infty}_{\mathrm{iso}\,r}(\bar{\bar{I}}_r, \boldsymbol{f}) := \Psi^{\mathcal{N}}_{\mathrm{iso}\,r}(\bar{\bar{I}}_r, \boldsymbol{f}) + \Psi^{\sigma}_{\mathrm{iso}\,r}(\bar{\bar{I}}_r, \boldsymbol{f}) + \Psi^{\varepsilon}_{\mathrm{iso}\,r}(\boldsymbol{f}), \qquad r = 1, \dots, R. \tag{34}$$

The strain–energy contribution $\Psi^{\mathcal{N}}_{\mathrm{iso}\,r}$ is represented by FFNN and captures the material's basic constitutive behavior. The FFNN takes as input the generalized invariants $\bar{\bar{I}}_r$ and the feature vector $\boldsymbol{f}$. It can be expected to approximate stresses with reasonable quality; however, an FFNN cannot, in general, be expected to inherently satisfy the normalization condition (25). To ensure that both stress and strain energy vanish in the undeformed reference configuration, we therefore introduce the correction terms $\Psi^{\sigma}_{\mathrm{iso}\,r}$ and $\Psi^{\varepsilon}_{\mathrm{iso}\,r}$, cf. [37]. Notably, these terms are not modeled by an FFNN; instead, they depend on $\Psi^{\mathcal{N}}_{\mathrm{iso}\,r}$ and its derivatives with respect to the generalized invariants, evaluated in the undeformed reference configuration. Consequently, $\Psi^{\sigma}_{\mathrm{iso}\,r}$ and $\Psi^{\varepsilon}_{\mathrm{iso}\,r}$ continuously adapt during training as $\Psi^{\mathcal{N}}_{\mathrm{iso}\,r}$ evolves. They are *not* computed in a post-processing step, but form an integral part of the network architecture (cf. Fig. 3). Details on how to construct these free energies are provided in Appendix C.

By constructing the isochoric equilibrium free energies $\Psi^{\infty}_{\mathrm{iso}\,r}$ in this manner, we inherently ensure thermodynamic consistency, objectivity, material symmetry, symmetry of the second Piola–Kirchhoff stress tensor, and a stress- and energy-free reference configuration.

Beyond these constitutive constraints, it is often desirable—especially for robust FE simulations—that the learned equilibrium free energy admits a polyconvex representation. We therefore design the invariant-based FFNN such that $\Psi^{\infty}_{\mathrm{iso}\,r}$ in (34) is polyconvex for the incompressible setting considered here. Since $\Psi^{\infty}_{\mathrm{iso}\,r}$ is an additive combination of contributions, polyconvexity is preserved if each contribution is polyconvex [93]. In Appendix C, we show that $\Psi^{\sigma}_{\mathrm{iso}\,r}$ is polyconvex by construction, whereas $\Psi^{\varepsilon}_{\mathrm{iso}\,r}$ is a constant and thus does not affect polyconvexity. Hence, it remains to ensure that $\Psi^{\mathcal{N}}_{\mathrm{iso}\,r}$ is polyconvex.

To this end, we exploit that, in the incompressible setting considered here, the generalized invariants $\bar{\bar{I}}_r$ are polyconvex functions of the deformation and use the composition rule: composing a convex function with a convex, monotonically increasing function yields a convex function [94, Section 6.]. Accordingly, it suffices to construct the FFNN representing $\Psi^{\mathcal{N}}_{\mathrm{iso}\,r}$ to be convex and monotonically increasing with respect to its invariant inputs. We enforce this by constraining all network weights to be non-negative and by using convex, non-decreasing activation functions in each layer [95]. Specifically, we employ the softplus activation $y(x) = \ln(1 + \exp(x))$ in the hidden layers and a linear activation $y(x) = x$ in the output layer. Convexity and monotonicity are not required with respect to the feature vector $\boldsymbol{f}$; if needed, the architecture can be adapted to be only partially input-convex [95,96].

3.2. Non-equilibrium free energy

Analogous to the equilibrium free energy contributions $\Psi^{\infty}_{\mathrm{iso}\,r}$ in (34), we define $\Psi_{r\alpha}$ by

$$\Psi_{r\alpha}(\bar{\bar{I}}_r, \boldsymbol{f}) := \Psi^{\mathcal{N}}_{r\alpha}(\bar{\bar{I}}_r, \boldsymbol{f}) + \Psi^{\sigma}_{r\alpha}(\bar{\bar{I}}_r, \boldsymbol{f}) + \Psi^{\varepsilon}_{r\alpha}(\boldsymbol{f}), \qquad r = 1, \dots, R,\ \alpha = 1, \dots, N, \tag{35}$$

where $\Psi^{\mathcal{N}}_{r\alpha}$ is represented by an FFNN and $\Psi^{\sigma}_{r\alpha}$ and $\Psi^{\varepsilon}_{r\alpha}$ are correction terms to ensure a stress- and energy-free reference configuration. Accordingly, $\Psi_{r\alpha}$ also fulfills the same important constitutive requirements as $\Psi^{\infty}_{\mathrm{iso}\,r}$ does.

The choice of the number N of non-equilibrium free energy contributions $\Psi^{\mathrm{neq}}_{r\alpha}$ used to model a material is critical. While increasing N can capture more complex viscoelastic behavior, it also raises model complexity and computational cost. We therefore initialize the vCANN with an upper bound, i.e., an arbitrary maximum number of Maxwell elements $N^{\max}$. To balance accuracy and sparsity, the vCANN then determines the effective number N during training by progressively eliminating Maxwell elements.

Sparsity is promoted via ℓ_1 regularization of the potentials $\Psi_{r\alpha}$, controlled by a sparsity parameter Λ. The larger Λ, the sparser the model. The hyperparameter Λ is optimized iteratively to balance validation loss and model parsimony.

3.3. Relaxation times

We use individual FFNNs $\mathcal{N}_{\tau_{r\alpha}}$ to separately represent the strain (rate)-dependent relaxation times $\tau_{r\alpha}$. To ensure positive relaxation times, we apply a positive activation function in the last layer, e.g., the softplus function. Using the invariant basis $\bar{I}_r$ as input guarantees material objectivity and symmetry.

Relaxation processes in viscoelastic materials occur on different time scales [97], often modeled uniformly distributed on a logarithmic scale [98]. Therefore, we scale the output of the FFNNs learning the relaxation times by time constants $T_{r\alpha}$ which are uniformly distributed on a logarithmic scale in the range $[T_{\min}, T_{\max}]$. Scaling the output ensures an equal representation of the time scales. $T_{\min}$ and $T_{\max}$ are user-defined parameters based on prior knowledge or heuristic expectations about the expected time scales, which can speed up training if appropriately chosen. The time scaling constants $T_{r\alpha}$ are not the relaxation times of our model; the vCANN will learn relaxation times $\tau_{r\alpha}$ different from $T_{r\alpha}$. Please refer to [50, Section 3.3] for details.

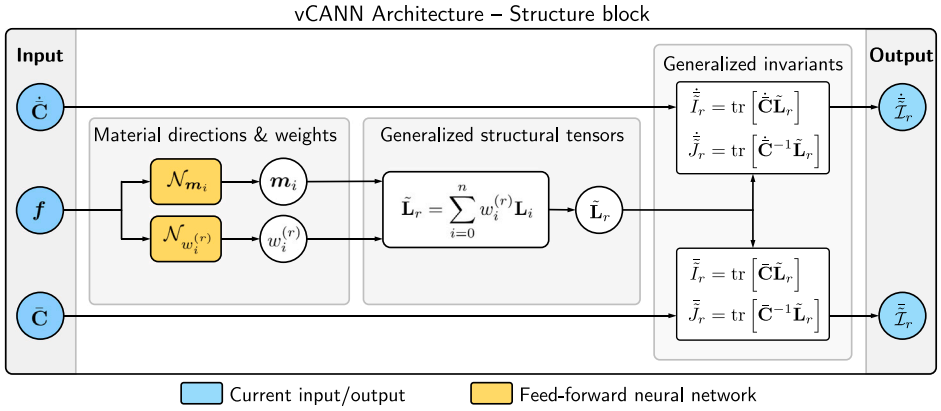


Fig. 4. Structure block related to the generalized tensor $\tilde{\mathbf{L}}_r$: The deformation (rate) tensors $\bar{\mathbf{C}}$ and $\dot{\bar{\mathbf{C}}}$, and the feature vector $\boldsymbol{f}$ are input to the structure block. The neural networks $\mathcal{N}_{\boldsymbol{m}_i}$ and $\mathcal{N}_{w_i^{(r)}}$ learn from $\boldsymbol{f}$ the preferred material directions $\boldsymbol{m}_i$ and weights $w_i^{(r)}$, respectively. From these outputs, we compute the generalized structural tensors $\tilde{\mathbf{L}}_r$, and the generalized invariants $\bar{\bar{I}}_r$ and $\dot{\bar{\bar{I}}}_r$. The generalized invariants are input to the sub-networks of the vCANN in Fig. 3.

3.4. Generalized structural tensors

The generalized structural tensors $\tilde{\mathbf{L}}_r$ depend on the scalar weights $w_i^{(r)}$ and the preferred material directions $\boldsymbol{m}_i$. Both $w_i^{(r)}$ and $\boldsymbol{m}_i$ can differ from material to material. They may also depend on the feature vector $\boldsymbol{f}$. Therefore, for each $\tilde{\mathbf{L}}_r$ one FFNN $\mathcal{N}_{w_i^{(r)}}$ learns the $w_i^{(r)}$, and another FFNN $\mathcal{N}_{\boldsymbol{m}_i}$ learns the three components of the $\boldsymbol{m}_i$, both using $\boldsymbol{f}$ as input, Fig. 4. To satisfy the partition of unity constraint and the positivity constraint, $(5)_{2,3}$, we apply the softmax activation function $y_i(x_i) = \exp(x_i)/\sum_j \exp(x_j)$ to the output of the corresponding FFNNs $\mathcal{N}_{w_i^{(r)}}$. To ensure that the material directions are unit vectors, we apply the activation function $y_i(x_i) = x_i/(\sum_j x_j^2)^{1/2}$ to the output of the corresponding FFNNs $\mathcal{N}_{\boldsymbol{m}_i}$.

However, if the generalized structural tensors do not depend on the feature vector $\boldsymbol{f}$, the scalar weights $w_i^{(r)}$ and the material directions $\boldsymbol{m}_i$ are not outputs of neural networks. Instead, they are incorporated directly into the vCANN architecture as optimizable parameters. This approach is demonstrated in Section 5.3, where the vCANN learns the material's anisotropy from biaxial tension tests alone.

3.5. Algorithmic stress update

Implementing the viscoelastic constitutive law into a machine learning architecture—and a FE code—requires an algorithmic stress update. Specifically, we must provide an algorithmic update of the auxiliary non-equilibrium stress $\mathbf{Q}_{r\alpha}$ governed by the evolution Eq. (33). To do this, we divide the time interval $(0, T]$ into N time intervals $\Delta t := t^{(n+1)} - t^{(n)}$ bounded by generally non-equidistant, discrete time points $\{t^{(n)}\}_{n=0}^{N}$. Inspired by [51,74], we approximate $\mathbf{Q}_{r\alpha}^{(n+1)}$ at time instance $t^{(n+1)}$ using the explicit scheme

$$\mathbf{Q}_{r\alpha}^{(n+1)} = \exp\left(\xi_{r\alpha}\right) \bar{\mathbf{S}}_{r\alpha}^{(n+1)} + \exp\left(\xi_{r\alpha}\right) \left[\exp\left(\xi_{r\alpha}\right) \mathbf{Q}_{r\alpha}^{(n)} - \bar{\mathbf{S}}_{r\alpha}^{(n)}\right], \qquad \xi_{r\alpha} := -\frac{\Delta t}{2\bar{\tau}_{r\alpha}}, \quad \bar{\tau}_{r\alpha} := \frac{\tau_{r\alpha}^{(n+1)} + \tau_{r\alpha}^{(n)}}{2} \tag{36}$$

which depends on history variables $\boldsymbol{\mathcal{H}}_{r\alpha}^{(n)} = \left(\mathbf{Q}_{r\alpha}^{(n)}, \bar{\mathbf{S}}_{r\alpha}^{(n)}, \tau_{r\alpha}^{(n)}\right)$ evaluated at the previous time instance $t^{(n)}$. Assuming sufficiently small time intervals Δt (and a sufficiently smooth problem), we approximated the relaxation time during Δt by $\bar{\tau}_{r\alpha}$ as the average of the relaxation times at the beginning and end of Δt. This algorithmic update is integrated into the upper branch of the vCANN architecture, Fig. 3.

For a detailed summary of the algorithmic stress update, we refer to Appendix E, with the corresponding consistent tangent modulus given in Appendix F.

3.6. Derivatives of feed-forward neural networks

To compute the stress tensor, we require first- and second-order partial derivatives of the free energies $\Psi_{\mathrm{iso}\,r}^{\infty}$ and $\Psi_{r\alpha}$ with respect to the generalized invariants, cf. (23) and (31). For the algorithmically consistent tangent moduli derived in Appendix F.1, third-order derivatives of $\Psi_{r\alpha}$ and first-order derivatives of $\tau_{r\alpha}$ are additionally required. During training, all derivatives are evaluated via automatic differentiation.

However, automatic differentiation is not available in most commercial or academic FE packages, underscoring the need for analytical derivative formulations. Furthermore, automatic differentiation evaluates derivatives through computational graphs

whose size increases rapidly—essentially exponentially—with the derivative order [99]. As a result, computing stresses and tangent moduli becomes costly. Since these quantities must typically be evaluated millions of times in an FE analysis, computational efficiency is essential.

To overcome these limitations, we employ efficient analytical recurrence relations that enable the exact analytical evaluation of the derivatives of a trained network's outputs with respect to its inputs up to third order. These relations are derived in Appendix D.

4. Automated modeling pipeline

This section summarizes the automated pipeline that links experimental (or synthetic) time-series data to a trained vCANN and its deployment in FE simulations via a material subroutine. The workflow is illustrated in Fig. 1 and consists of four main stages: (i) data preparation, (ii) model configuration and training, (iii) export to a material subroutine, and (iv) FE simulation.

Data preparation. The pipeline takes as input time-series data from mechanical experiments (e.g., loading–unloading or relaxation tests). Each experiment provides the deformation gradient history $\mathbf{F}$, the corresponding stress labels $\hat{\mathbf{P}}$, time history t, and optionally additional n_F auxiliary features collected in a feature vector $\boldsymbol{f}$. For n_H loading histories with n_S discrete time steps each, we use tensors of shape

$$\mathbf{F} \in \mathbb{R}^{(n_H,n_S,3,3)}, \quad \mathbf{P} \in \mathbb{R}^{(n_H,n_S,3,3)}, \quad t \in \mathbb{R}^{(n_H,n_S,1)}, \quad \boldsymbol{f} \in \mathbb{R}^{(n_H,n_S,n_F)}.$$

Training requires a fixed number of discrete time steps n_S per loading history. Experiments may have different physical durations and sampling rates; only the step count must be unified. If an experiment is recorded at fewer (or more) than n_S time points, the user must interpolate and resample the measured quantities (time, deformation, stresses, and features) onto a prescribed set of n_S time points prior to training.

Model configuration and hyperparameters. The user specifies a small set of high-level architectural and training hyperparameters. The most relevant ones are: the number R of generalized structural tensors, the number n of preferred material directions, the maximum number of Maxwell elements $N^{\max}$, the depth and width (layers/neurons) of the feed-forward sub-networks representing $\Psi^{\infty}_{\mathrm{iso}\,r}$, $\Psi_{r\alpha}$, $\tau_{r\alpha}$, $\boldsymbol{m}_i$, and $w_i^{(r)}$, and (optionally) a sparsity parameter Λ for Maxwell-element pruning. The time-normalization constants $T_{r\alpha}$ (Section 3) are selected to span the expected relaxation time scales and can accelerate training when reasonable bounds $[T_{\min}, T_{\max}]$ are available. A dedicated hyperparameter study is beyond the scope of this work.

Training objective. Training minimizes a normalized mean-squared error $\mathcal{L}$ between predicted and labeled stress histories $\mathbf{P}$ and $\hat{\mathbf{P}}$, respectively:

$$\mathcal{L} = \frac{1}{n_H n_S} \sum_{k=1}^{n_H} \sum_{l=1}^{n_S} \left\| \mathbf{P}_{k,l} - \hat{\mathbf{P}}_{k,l} \right\|^2. \tag{37}$$

Unless stated otherwise, experiments at different conditions (e.g., strain rates) are equally weighted in the loss. Since the problem is history-dependent and the network is trained on complete stress histories, the stress at any given time point depends on all preceding states. Consequently, in each optimizer iteration, the evolution equations must be integrated according to (36) over the entire time series, and gradients are backpropagated through these sequential updates. We trained all vCANNs using the L-BFGS-B algorithm, that can handle bound constraints. These constraints are crucial to ensure compliance with the physical constraints discussed in Section 3, such as polyconvexity.

Sparsification of Maxwell elements. To promote parsimonious viscoelastic models, the pipeline supports pruning of Maxwell elements via ℓ_1 regularization. Specifically, the non-equilibrium strain–energy contributions are penalized with an ℓ_1 term, which tends to drive unnecessary contributions toward zero. After training, weights below a small threshold ϵ_{cut} are set to zero and the associated Maxwell elements are removed from the effective model.

Handling of auxiliary features in FE simulations. In the current pipeline, the feature vector $\boldsymbol{f}$ is assumed uniform across the domain, i.e., it takes the same value at each integration point and is exported to the material subroutine along with the network parameters. More general use cases with spatially varying features are supported in principle, since the material subroutine is evaluated at the integration-point level; spatially varying feature fields can be prescribed, for example, via user-defined fields or temperature fields. This enables the incorporation of heterogeneous material information into the constitutive response.

Export to an FE material subroutine. After training, the learned network parameters and architecture metadata are exported into a generic Fortran material subroutine template through a single user call. The export inserts the weights and biases of all sub-networks as well as the relevant dimensions (e.g., R, N, number of layers/neurons per network, number of features) into the material subroutine and initializes the corresponding variables accordingly. The exported code consists of two parts: (i) the material subroutine implementing the constitutive update and algorithmically consistent tangent modulus, and (ii) a utility module for evaluating FFNNs and their derivatives (Jacobian, Hessian, and third-order derivatives) required for stress and tangent computations. The utility module is generic and independent of the particular trained model, while the network parameters are material-specific. At present, we focus on Abaqus/Standard (implicit).

FE simulation and evaluation. The exported material subroutine can be used directly in FE simulations by specifying the material and (if applicable) the auxiliary fields in the input file. The pipeline thus provides an end-to-end route from time-series data to predictive structural simulations. To ensure correctness of the automated deployment, we verify the FE implementation against the TensorFlow reference at the single-element level in the next section.

To ensure accessibility and reproducibility, all codes and FE subroutines required to employ the proposed framework within Abaqus are openly provided, enabling practitioners to integrate advanced viscoelastic neural material models without specialized expertise: https://github.com/KianAbd/vCANN_FEM.

5. Results

This section demonstrates the proposed end-to-end workflow—from time-series training data to FE simulations—and verifies the automatically generated material subroutine against the TensorFlow reference before presenting representative structural examples.

Model training was conducted on a desktop workstation equipped with an Intel Core i9-9920X CPU, without GPU acceleration, and required approximately 10–30 min depending on the size of the training dataset. The largest dataset included $n_H = 15$ loading histories, each of which was discretized into $n_S = 1000$ time steps. In the absence of strong prior knowledge, we used, unless stated otherwise, $R = 1$, $N^{\max} = 3$, and $[T_{\min}, T_{\max}] = [10^0, 10^2]$ s, which yielded good results in all examples. These choices can, however, be adjusted based on the fit quality and the apparent complexity of the material response. In the present study, we set $\Lambda = 0$ and disabled regularization; nevertheless, the pipeline supports this option. We refer to [50, Section 3.3] for details and a thorough discussion on the ℓ_1 regularization. Table J.1 summarizes the hyperparameters of all vCANNs from the following examples.

All examples are based on perfect incompressibility, imposed numerically via a Lagrange multiplier scheme, resulting in a two-field (hybrid) formulation with independent interpolation of the displacement and pressure fields for 3D and plane strain elements; an additional stabilization term prevents equation solver difficulties [100, Section 3.2.3]. In contrast, plane stress and shell elements are purely displacement-based and enforce incompressibility through internal kinematic constraints [101, Section 2.4] and [102, Section 2.2.3].

5.1. Single-element verification and feature augmentation

To verify our material subroutine implementation, we simulated uniaxial tension tests using a single element and compared the results with the vCANN material point implementation in TensorFlow. We assumed isotropy, and used a hybrid brick element (C3D8H) with linear displacement interpolation and an additional constant pressure degree of freedom. Introducing a fixed Cartesian coordinate system with basis vectors $\{\boldsymbol{E}_i\}_{i=1}^3$, the deformation gradient reads

$$\mathbf{F} = \lambda \boldsymbol{E}_1 \otimes \boldsymbol{E}_1 + \lambda^{-1/2} \boldsymbol{E}_2 \otimes \boldsymbol{E}_2 + \lambda^{-1/2} \boldsymbol{E}_3 \otimes \boldsymbol{E}_3, \tag{38}$$

where λ denotes the stretch in the tensile direction $\boldsymbol{E}_1$. The stress-free boundary condition in one of the transverse directions, e.g., $P_{33} = 0$, is used to determine the Lagrange multiplier p in the Tensorflow-based vCANN implementation. The only non-zero stress component is P_{11}.

Before performing these simulations, we first trained the vCANN using experimental data from Very-High-Bond (VHB) 4905, a soft, electro-active polymer. Like many polymers, VHB 4905 exhibits pronounced temperature sensitivity. Consequently, [103] carried out a comprehensive thermo-viscoelastic experimental characterization of VHB 4905. Among other tests, they conducted uniaxial loading–unloading experiments for three stretch rates $\dot{\lambda} \in \{0.03, 0.05, 0.1\}$ s^{-1}, three peak stretch levels $\lambda_{\max} \in \{2, 3, 4\}$, and six temperatures $\Theta \in \{0, 10, 20, 40, 60, 80\}$ °C and recorded the nominal stress component P_{11} in the tensile direction.

For simplicity, we neglect the strain-rate dependence of the relaxation times. To account for the pronounced temperature dependence of the stress response, we incorporated the temperature into the vCANN input space via the feature vector $\boldsymbol{f} = \Theta$. Owing to isotropy, the associated generalized structural tensor (44) reduces to $\tilde{\mathbf{L}}_1 = \frac{1}{3}\mathbf{I}$. Accordingly, the following free energy functions and relaxation times—represented by the FFNNs described in Section 3—are learned from the loading–unloading tests:

$$\Psi^{\infty}_{\text{iso}\,1} = \Psi^{\infty}_{\text{iso}\,1}(\bar{\tilde{I}}_1, \Theta), \qquad \Psi_{1\alpha} = \Psi_{1\alpha}(\bar{\tilde{I}}_1, \Theta), \qquad \tau_{1\alpha} = \tau_{1\alpha}(\bar{\tilde{I}}_1, \Theta), \qquad \alpha = 1, \dots, N^{\max}, \tag{39}$$

where

$$\bar{\tilde{I}}_1 = \left(\bar{\tilde{I}}_1, \bar{\tilde{J}}_1\right), \qquad \bar{\tilde{I}}_1 = \frac{1}{3}\,\mathrm{tr}[\bar{\tilde{\mathbf{C}}}], \qquad \bar{\tilde{J}}_1 = \frac{1}{3}\,\mathrm{tr}[\mathrm{cof}\,\bar{\tilde{\mathbf{C}}}]. \tag{40}$$

As a quantitative measure of the goodness-of-fit, we report the R^2 values for the training and validation datasets in Table 1. Fig. 5 contrasts the descriptive performance of the TensorFlow-based vCANN model (black dashed/dash-dotted curves) with the experimental training data for VHB 4905 (open markers). The match with the training data is excellent. After training, we automatically translated the TensorFlow-based vCANN into a material subroutine and integrated it into Abaqus. We then replicated the uniaxial loading–unloading tests using FE simulations, represented by the solid colored curves in Fig. 5. First, the vCANN responses obtained in Abaqus and in TensorFlow are indistinguishable, verifying the correctness of the FE implementation. Second, the simulations showed no convergence issues—even under sudden load reversals—highlighting the robustness and consistency of the algorithmic tangent modulus.

Analogously, Fig. 6 illustrates the predictive capabilities of both the TensorFlow and FEM implementations on the experimental validation dataset for VHB 4905. Importantly, the vCANN was not exposed during training to any data at a peak stretch $\lambda_{\max} = 3$

Table 1
R^2 values summarizing the descriptive and predictive performance of the vCANN on the VHB 4905 dataset, corresponding to the fits and predictions shown in Figs. 5 and 6.

Θ [°C]	$\lambda_{\max}$	$\dot{\lambda} = 0.03$	$\dot{\lambda} = 0.05$	$\dot{\lambda} = 0.10$
0	2	–	–	0.9791
	3	–	–	0.9842
	4	0.9972	0.9885	0.9970
10	3	–	–	0.9571
	4	–	–	0.9995
20	3	–	–	0.9933
	4	0.9986	0.9970	0.9988
40	2	–	–	0.9966
	3	–	–	0.9889
	4	0.9963	0.9883	0.9950
60	2	–	–	0.9936
	3	–	–	0.9884
	4	0.9905	0.9982	0.9990
80	2	–	–	0.9970
	3	–	–	0.9823
	4	0.9950	0.9910	0.9975

Training Validation

or at a stretch rate $\dot{\lambda} = 0.05\ \mathrm{s}^{-1}$. Nevertheless, the trained vCANN accurately predicted the material response for both the unseen stretch level and stretch rate.

As reported in the literature [40,43], training of neural network-based approaches to inelasticity can be prone to numerical difficulties. To mitigate these issues, pre-training strategies have been proposed, e.g., initializing weights to start from a favorable point, or training on a subset of the data before continuing on the full stress–strain history and/or complete dataset. While no instabilities in the sense of exploding loss values (NaN or Inf) were observed, the model occasionally converged prematurely to a suboptimal local minimum. Restarting training one to three times with newly initialized weights typically resolved this issue.

In addition, applying a small ℓ_2-regularization (10^{-3}) to the weights of the free energies $\Psi_{r\alpha}$ was found beneficial to suppress oscillations in the stress–strain curves, likely caused by the double contraction $\mathbb{C}_{r\alpha} : \mathbf{Q}_{r\alpha}$. Once training had stabilized after a few epochs, this regularization was disabled.

Furthermore, initially training on a reduced number of time steps (here 200) while covering the full strain history proved advantageous. After convergence, the number of time steps was increased to the final value (here 1000), optionally including an intermediate refinement stage.

5.2. Cook's membrane: mesh convergence study

To achieve more realistic FE simulations and demonstrate the material subroutine's versatility, we analyzed Cook's membrane problem. Cook's membrane problem is a well-known benchmark for evaluating the combined bending and shear response. The boundary value problem involves a tapered cantilever, fixed on the left and subjected to a shear load of $F = 0.6$ N on the right, as shown in Fig. 7. The concentrated force is applied by a distributing coupling constraint. This constraint couples the nodes on the right side of the cantilever to a rigid body control point to which the load is applied. The force increases linearly from zero to its final value over 1 s.

In a mesh convergence study, we analyzed the vertical displacement u_y of the upper-right edge node A, see Fig. 7. To this end, we solve Cook's membrane problem using several three-dimensional, plane strain, plane stress, and shell element types. We provide a detailed overview of all tested elements in Appendix G. For each element type, we solve the boundary value problem at two temperature levels, 10 and 60 °C, using increasingly refined meshes of 4, 8, 16, and 32 elements on each side of the cantilever. For three-dimensional elements, we considered a single element through the thickness and applied symmetry boundary conditions in the X–Y plane. Therefore, we considered only half of the thickness and the shear load in the three-dimensional simulations. To describe the material behavior of Cook's membrane, we used the same vCANN of Section 5.1, which was already trained on the uniaxial loading–unloading data of VHB 4905. Exemplarily, Fig. 8 presents the deformed configuration and von Mises stress distributions for meshes with 4 and 32 C3D20H elements per side and the temperature levels 10 and 60 °C. Consistent with the uniaxial test results, for a fixed applied vertical load the membrane exhibits larger deformations at higher temperatures, reflecting the temperature-induced softening of the material.

Fig. 9 shows the vertical displacement u_y of node A, located at the upper right corner, as a function of the number of elements per side for both temperature levels, 10 and 60 °C. The quadratic elements closely match the converged solution, even when relatively coarse meshes are used. In contrast, the triangular plane-strain element CPE3H (linear displacement, constant pressure) yields excessively stiff responses, which can be attributed to its poor bending performance and inadequate representation of incompressibility. This overly stiff behavior of the CPE3H element is well known and documented in the literature [104]. All other

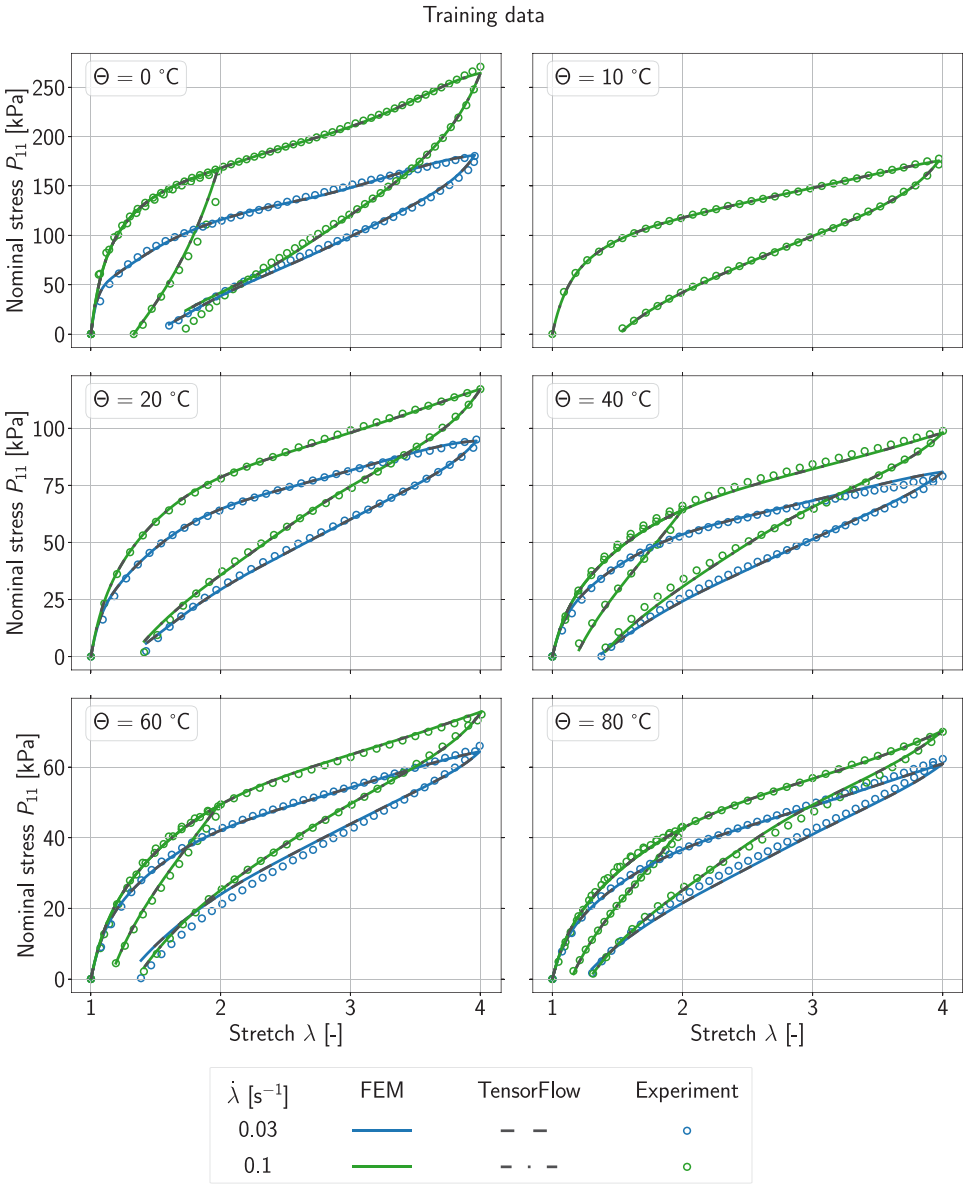


Fig. 5. Descriptive performance of the vCANN on the VHB 4905 training data: nominal stress P_{11} versus stretch λ for uniaxial loading–unloading tests at different temperatures Θ (subfigures), stretch rates $\dot{\lambda}$, and peak stretches $\lambda_{\max}$. Solid colored lines denote the FEM implementation, black dashed/dash-dotted lines denote the TensorFlow implementation, and open markers indicate the experimental data reproduced from [103]. Both implementations yield identical results.

element types approach the same solution upon mesh refinement. For the shell elements, an additional refinement is necessary to achieve convergence. Given that the membrane is subjected to an almost plane stress state, except near the clamping, the tip displacements for the 3D, plane-stress, and shell models are nearly identical. As expected, the tip displacement increases with temperature, Fig. 9. Overall, the observed convergence trends agree with analogous benchmark results based on classical constitutive models

To further highlight the viscoelastic behavior of the material, we performed additional mesh-convergence analyses under cyclic loading. The membrane was subjected to the force–time profile in Fig. 10 at $\Theta = 10$ and 60 °C. For each of the four mesh refinements, we tracked the vertical displacement u_y of node A at both the midpoint and the end of every loading cycle. For clarity, we report only the representative convergence behavior obtained with the quadrilateral plane-strain element CPE4H (linear displacement, constant pressure). As before, the displacements at the selected time points converge with mesh refinement, and the peak tip displacement is larger at the higher temperature. Owing to viscous effects, the membrane does not fully return to its initial configuration at the end

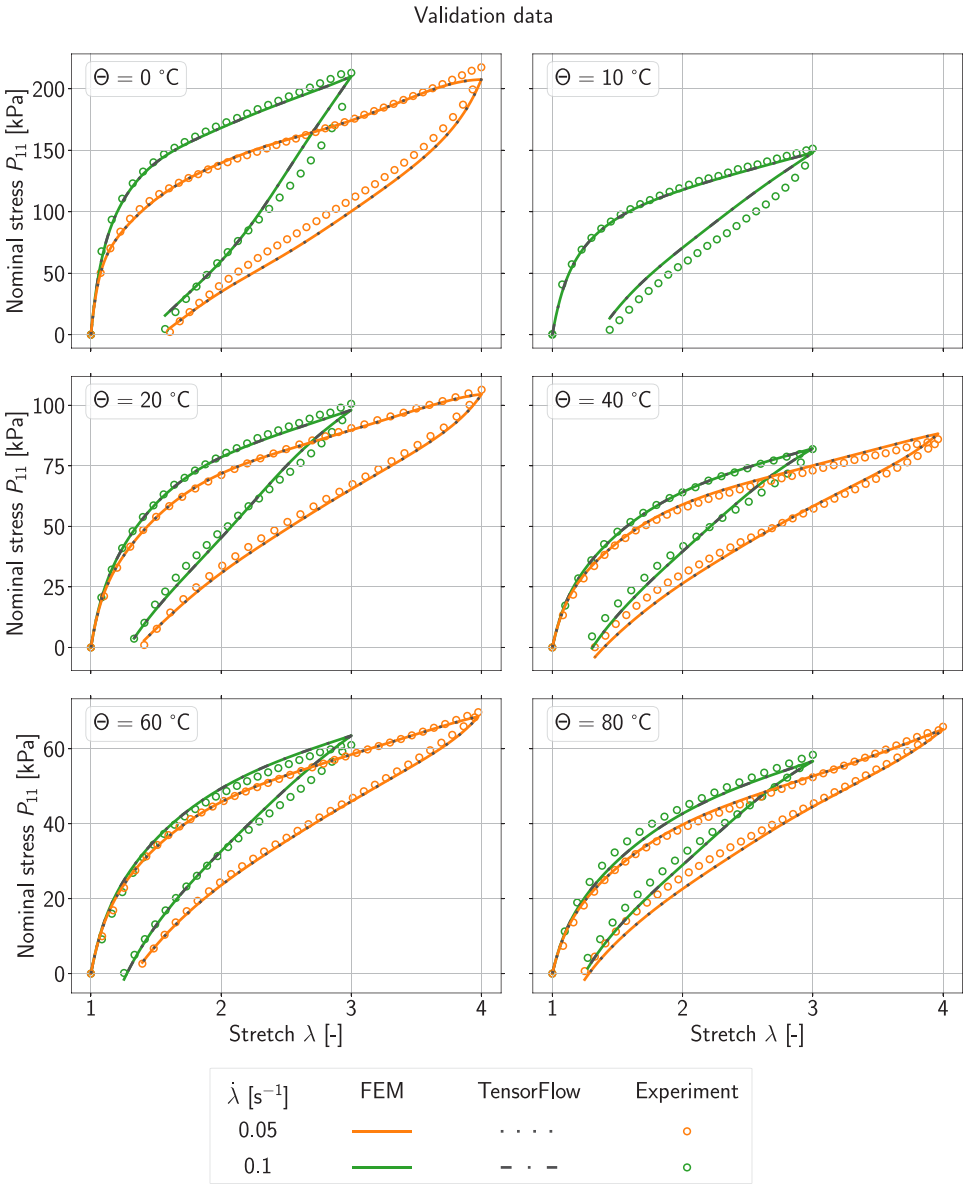


Fig. 6. Predictive performance of the vCANN on the VHB 4905 validation data: nominal stress P_{11} versus stretch λ for uniaxial loading–unloading tests at different temperatures Θ (subfigures), stretch rates $\dot{\lambda}$, and peak stretches $\lambda_{\max}$. Solid colored lines denote the FEM implementation, black dashed/dash-dotted lines denote the TensorFlow implementation, and open markers indicate the experimental data reproduced from [103]. Both implementations yield identical results.

of a cycle when the vertical force F_y is reduced to zero, and this residual deformation is more pronounced at elevated temperatures than at lower ones.

As demonstrated by this example, a key strength of the presented approach is its ability to incorporate arbitrary auxiliary features—here temperature—into neural constitutive laws, enabling automated generation of advanced material routines for FE simulations without manual model design (for an additional example using Shore hardness-dependent Ecoflex silicon polymer data, see Appendix H). Moreover, although the vCANN was trained exclusively on uniaxial data, the corresponding FE simulations ran robustly without convergence issues and produced physically reasonable responses for three-dimensional deformation states. While no quantitative statement on the accuracy of these results can be made in the absence of experimental three-dimensional stress–strain data, this limitation can be addressed by training on synthetically generated data from a reference material model, as demonstrated next.

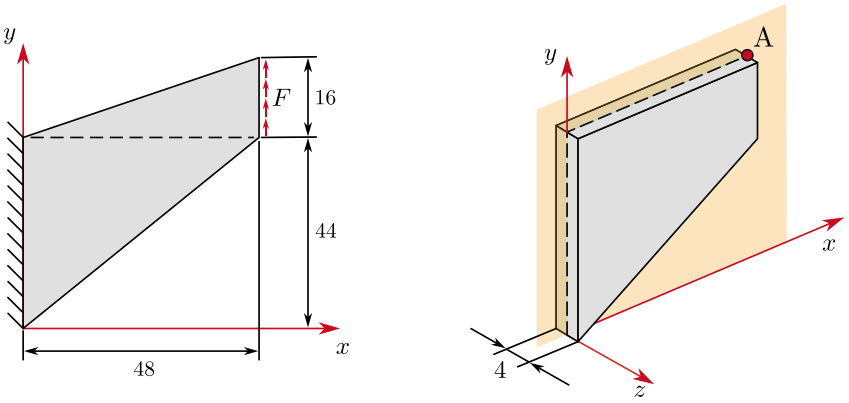
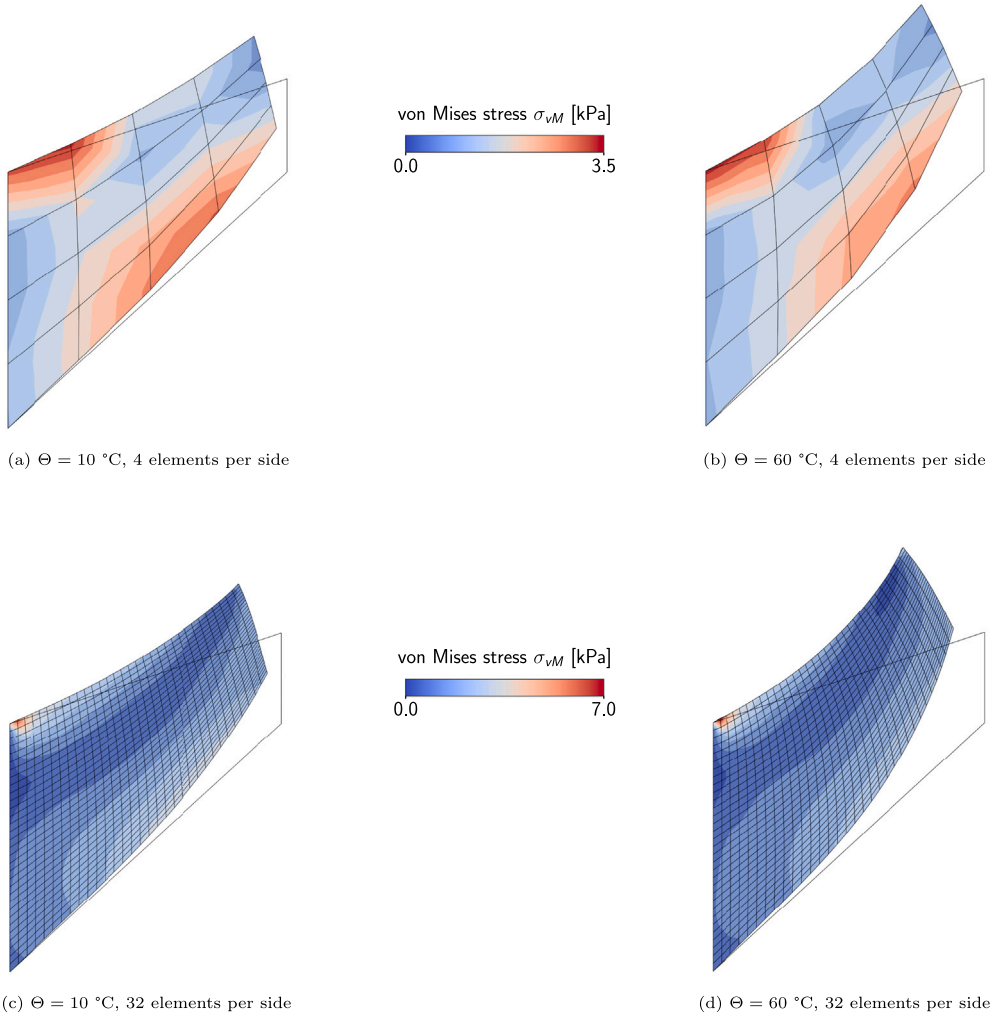


Fig. 7. Cook's membrane problem: initial geometry and loading. All dimensions are in [mm]. The X–Y plane (orange) is the symmetry plane. The vertical displacement u_y of the upper right edge node A is analyzed.



(a) $\Theta = 10$ °C, 4 elements per side

(b) $\Theta = 60$ °C, 4 elements per side

(c) $\Theta = 10$ °C, 32 elements per side

(d) $\Theta = 60$ °C, 32 elements per side

Fig. 8. Cook's membrane benchmark: von Mises stress distribution on the deformed configuration meshed with three-dimensional hybrid continuum elements C3D8H at different temperature levels Θ: (a) 4 elements per side at 10 °C, (b) 4 elements per side at 60 °C, (c) 32 elements per side at 10 °C, and (d) 32 elements per side at 60 °C. The wireframes represent the initial configurations.

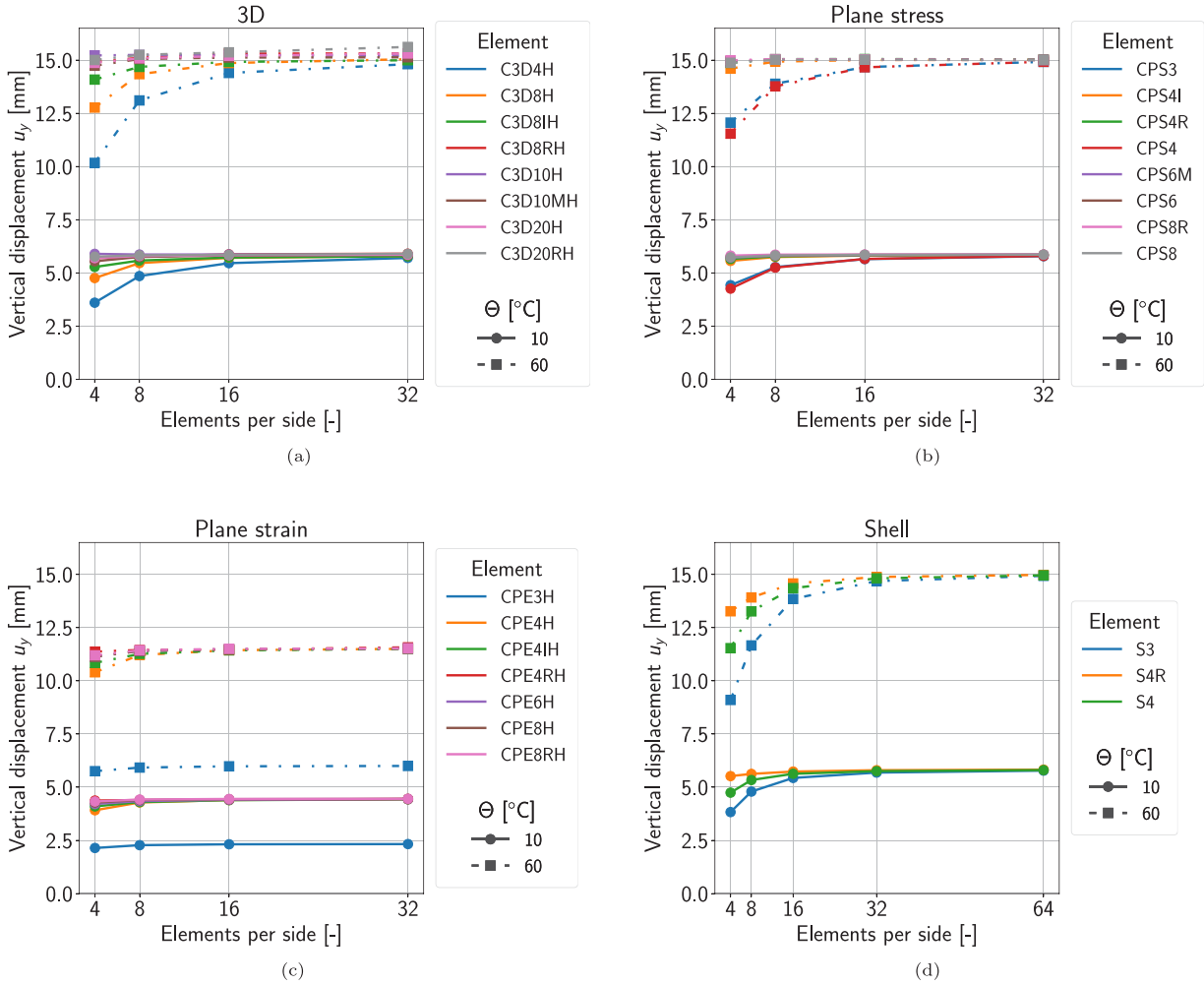


Fig. 9. Cook's membrane; mesh convergence study at temperature levels 10 and 60 °C: vertical displacement u_y of the upper right edge node A versus the number of elements per side: (a) three-dimensional continuum elements, (b) plane stress elements, (c) plane strain elements, and (d) shell elements.

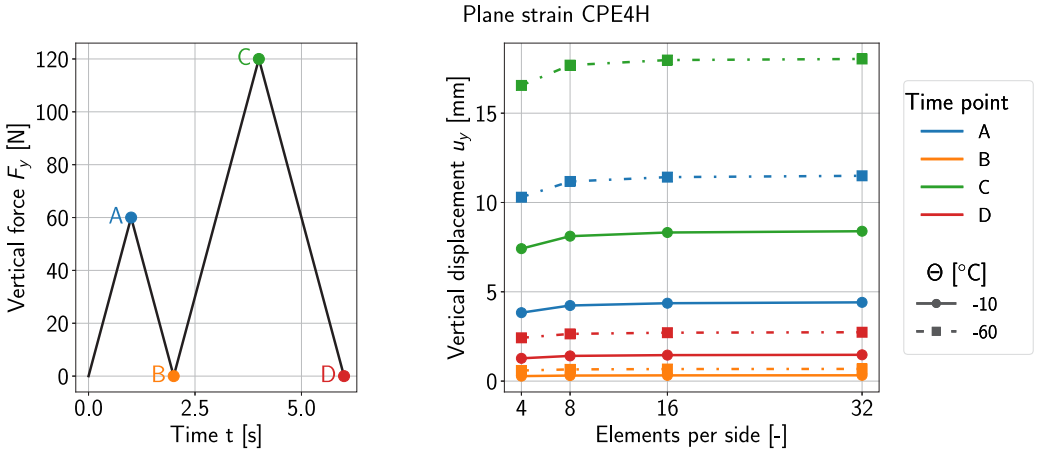


Fig. 10. Cook's membrane under cyclic loading; mesh convergence study of the quadrilateral plane strain hybrid element CPE4H. Left: force-time history and time points A–D at which the vertical displacements u_y of node A are extracted; right: vertical displacements u_y of node A versus the number of elements per side.

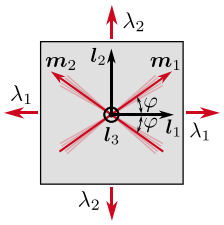


Fig. 11. A material sample reinforced by two mechanically equivalent fiber families $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ is stretched along the principal material directions $\boldsymbol{l}_1$ and $\boldsymbol{l}_2$ by the principal stretches λ_1 and λ_2. Each fiber family forms an angle φ with $\boldsymbol{l}_1$.

5.3. Multiaxial loading of a simplified artery model

In the previous section, we investigated Cook's membrane problem using a vCANN trained solely on uniaxial tensile data. While this demonstrated robust FE performance, it remains unclear how well such a model generalizes to three-dimensional multiaxial deformation states. We now address this question and show that accurate predictions can be achieved using only a small number of selected experiments. We proceed as follows. First, in silico biaxial experiments are performed using a known ground truth constitutive model to generate synthetic training data. Both isotropic and anisotropic ground truth models are considered. A vCANN is then trained on these synthetic training data. Finally, the same three-dimensional boundary value problem is solved using both the ground truth model and the trained vCANN, and the resulting stress fields are compared.

5.3.1. Ground truth model

We describe the equilibrium response of the ground-truth (GT) model using an anisotropic, polyconvex, isochoric free energy function proposed by [105]:

$$\Psi_{\text{iso}}^{\infty,\text{GT}} = \sum_{r=1}^{R} \frac{\mu_r}{2} \left[\frac{1}{\gamma_r + 1} \left(\bar{I}_r^{\gamma_r+1} - 1 \right) + \frac{1}{\delta_r + 1} \left(\bar{J}_r^{\delta_r+1} - 1 \right) \right], \tag{41}$$

where $\mu_r \geq 0$ are stiffness-like parameters and the shape parameters $\gamma_r, \delta_r \geq 0$ control the influence of the corresponding invariants. In the GT model, anisotropy is introduced—as a special case of the vCANN framework—by two mechanically equivalent fiber families $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$, with $w_1^{(r)} = w_2^{(r)}$, oriented along two distinct in-plane directions, which yields orthotropic material symmetry. The principal material directions $\{\boldsymbol{l}_i\}_{i=1}^3$ are defined as the eigenvectors of the associated generalized structural tensor $\tilde{\mathbf{L}}_r$; in particular, $\boldsymbol{l}_1$ and $\boldsymbol{l}_2$ coincide with the angle bisectors of $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$, while $\boldsymbol{l}_3$ is normal to the fiber plane, cf. Fig. 11. Letting 2φ denote the angle between the two fiber directions, the fiber directions can be expressed in terms of the principal directions as

$$\boldsymbol{m}_1 = \cos\varphi\, \boldsymbol{l}_1 + \sin\varphi\, \boldsymbol{l}_2, \qquad \boldsymbol{m}_2 = \cos\varphi\, \boldsymbol{l}_1 - \sin\varphi\, \boldsymbol{l}_2, \tag{42}$$

and hence the generalized structural tensor reads [64]:

$$\tilde{\mathbf{L}}_r = \frac{1 - 2w_1^{(r)}}{3} \mathbf{I} + w_1^{(r)} \left(\boldsymbol{m}_1 \otimes \boldsymbol{m}_1 + \boldsymbol{m}_2 \otimes \boldsymbol{m}_2 \right) \tag{43}$$

$$= \frac{1}{3} \left[1 + (6\cos^2\varphi - 2) w_1^{(r)} \right] \boldsymbol{l}_1 \otimes \boldsymbol{l}_1 + \frac{1}{3} \left[1 + (6\sin^2\varphi - 2) w_1^{(r)} \right] \boldsymbol{l}_2 \otimes \boldsymbol{l}_2 + \frac{1}{3} \left[1 - 2w_1^{(r)} \right] \boldsymbol{l}_3 \otimes \boldsymbol{l}_3. \tag{44}$$

The spectral form of $\tilde{\mathbf{L}}_r$ (44) is convenient for prescribing the material symmetry in a numerical model at the material points, when the eigenvector basis aligns with the local basis of a curvilinear coordinate system, e.g., circumferential/axial/radial directions in cylindrical coordinates, cf. Fig. 12. Note that for the GT model in (41), we employ only single term, i.e., $R = 1$.

The non-equilibrium response was modeled by $\alpha = 3$ Maxwell elements. Following the identical polymer chain assumption [57, 106], the auxiliary free energies $\Psi_{r\alpha}$ were assumed to be proportional to the isochoric equilibrium free energy:

$$\Psi_{\alpha}^{\text{GT}} = \beta_\alpha \Psi_{\text{iso}}^{\infty,\text{GT}}, \qquad \alpha = 1, 2, 3, \tag{45}$$

where $\beta_\alpha \geq 0$ are dimensionless strain–energy factors and the associated relaxation times τ_α are assumed to be constant. In Table I.1, we list the numerical values of the material parameters of the orthotropic and isotropic GT models used in the following examples. The isotropic case is discussed in Appendix I.

5.3.2. Biaxial multi-step relaxation tests

Constitutive models fitted to data from a single deformation mode often fail to generalize to multiaxial deformation states, which motivates the use of multiaxial stress–strain data. For anisotropic materials, this is commonly achieved with biaxial tensile tests [107,108], see Fig. 11. In the following, we align the tensile directions with the principal material directions $\boldsymbol{l}_1$ and $\boldsymbol{l}_2$ introduced above. For an incompressible material, the deformation gradient thus reads

$$\mathbf{F} = \lambda_1 \boldsymbol{l}_1 \otimes \boldsymbol{l}_1 + \lambda_2 \boldsymbol{l}_2 \otimes \boldsymbol{l}_2 + (\lambda_1 \lambda_2)^{-1} \boldsymbol{l}_3 \otimes \boldsymbol{l}_3, \tag{46}$$

Table 2
Common biaxial test protocols. The stretches λ_i are functions of the stretch λ. The stretch in the thickness direction follows from the incompressibility condition, i.e., $\lambda_3 = (\lambda_1\lambda_2)^{-1}$.

Protocol	$\lambda_1(\lambda)$	$\lambda_2(\lambda)$	$\lambda_3(\lambda)$
Equi-biaxial	λ	λ	$1/\lambda^2$
Strip-x (pure shear)	λ	1	$1/\lambda$
Strip-y (pure shear)	1	λ	$1/\lambda$
Off-x	λ	$\sqrt{\lambda}$	$\lambda^{-3/2}$
Off-y	$\sqrt{\lambda}$	λ	$\lambda^{-3/2}$

where λ_1 and λ_2 are prescribed stretches and $\lambda_3 = (\lambda_1\lambda_2)^{-1}$ follows from incompressibility. Although λ_1 and λ_2 can be prescribed independently [109], practical protocols typically impose specific relationships between them, Table 2. The stress-free boundary condition in the thickness direction $\boldsymbol{I}_3$, $P_{33} = 0$, is used to determine the Lagrange multiplier p in the Tensorflow-based vCANN implementation. The only non-zero (nominal) stress components are P_{11} and P_{22}.

For viscoelastic materials, multiaxial loading alone is not sufficient, because both the equilibrium and the non-equilibrium response must be characterized. In this context, (multi-step) relaxation tests are well established [89,110]. In multi-step relaxation tests, a deformation is applied during a load phase and subsequently held constant during a hold phase after which the next deformation is applied and so forth. Each hold phase provides (i) the transient stress decay that characterizes the non-equilibrium response and (ii) for sufficiently long holding times, the asymptotic stress that approximates the equilibrium response at the corresponding strain state. By combining several strain levels within a single experiment, biaxial multi-step relaxation tests efficiently provide equilibrium information across a range of strain states while simultaneously probing the transient non-equilibrium response.

This structure aligns naturally with the modular architecture of vCANNs, which separates the equilibrium free energy from the non-equilibrium contributions. For the single-element tests (Section 5.1) and Cook's membrane (Section 5.2), we trained vCANNs directly on loading–unloading data. Here, to exploit the information provided by multi-step relaxation tests, we adopt a staggered training approach: we first train only the equilibrium free energy using the asymptotic stresses of the individual strain steps (alternatively, quasi-static data or multiple single-step relaxations with negligible viscous effects can be used). We then fix the equilibrium parameters and train the full vCANN on the complete stress–time histories of the relaxation test to identify the non-equilibrium response. Although this staggered strategy is not mandatory, it makes targeted use of the available data, improves interpretability, and can enhance generalizability; therefore, we adopt it for the remainder of this study.

Following the above, for each biaxial loading protocol in Table 2, we generated synthetic training data by simulating multi-step relaxation tests using the GT models. In each in silico-test, nine equal stretch steps were applied at a constant stretch rate over a loading period of 10 s, followed by a relaxation period of 180 s. Although the stretch steps are equally spaced in stretch space, the corresponding generalized invariant pairs are non-uniformly spaced in invariant space due to the highly nonlinear mapping between the two. From each test, time series data comprising 1000 data points, non-uniformly spaced in time, were sampled. To capture the steep stress increase and decay at the onset of the loading and relaxation phases, respectively, the data were sampled more densely in these regions, with decreasing density toward the end of each phase. The final stress at the end of the relaxation period was taken as the equilibrium stress for the staggered training strategy.

5.3.3. Boundary value problem

After training a vCANN on the synthetic data, we solve the same boundary problem in Abaqus using both the GT model and the vCANN. We consider the boundary value problem of a simplified artery segment model, see Fig. 12. A hollow cylinder, fixed at the bottom surface ($u_1 = u_2 = u_3 = 0$), is subjected to an internal pressure of $p = 0.5$ MPa. Additionally, the top surface is displaced by $u_2 = 3$ mm in the $\boldsymbol{E}_2$-direction and rotated by $\varphi_2 = 1.5$ rad about the $\boldsymbol{E}_2$-axis. The deformation is applied via a coupling constraint between the top surface and the control point B, ensuring that the top surface maintains its original shape and remains parallel to the $(\boldsymbol{E}_1, \boldsymbol{E}_3)$-plane throughout the deformation. The pressure, displacement, and rotation are increased linearly from zero to their prescribed values over a period of 5 s and subsequently reduced linearly to zero during the next 5 s. This loading–unloading cycle is repeated twice. The geometry is discretized using 4550 C3D8H elements.

The artery model is reinforced by two helically arranged fiber families, Fig. 12. At each material point, we introduce a local orthonormal basis of the cylindrical coordinate system, which coincides with the principal material directions $\{\boldsymbol{l}_i\}_{i=1}^3$ and is aligned with the circumferential, axial, and radial directions of the cylinder in the reference configuration. The two mechanically equivalent fiber families $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ lie symmetrically in the $(\boldsymbol{l}_1, \boldsymbol{l}_2)$-plane and form an angle $\varphi = 26.25°$ with the circumferential $\boldsymbol{l}_1$-direction. In practice, the structural parameters φ and $w_1^{(r)}$ can be determined from histological image data or learned from the feature vector $\boldsymbol{f}$. However, since the synthetic training data lack such information, we treat φ and $w_1^{(r)}$ as additional optimizable parameters in the vCANN and identify them from the mechanical training data alone.

5.3.4. Descriptive performance on training data

We employed a vCANN with a single generalized Maxwell model ($R = 1$) associated with the orthotropic generalized structural tensor $\tilde{\mathbf{L}}_1$ (43). Accordingly, the following free energy functions and relaxation times—represented by the FFNNs described in Section 3—as well as the fiber angle φ and the weight factor $w_1^{(1)}$ are learned from the synthetic training data:

$$\Psi^{\infty}_{\mathrm{iso}\,1} = \Psi^{\infty}_{\mathrm{iso}\,1}(\bar{\tilde{I}}_1), \qquad \Psi_{1\alpha} = \Psi_{1\alpha}(\bar{\tilde{I}}_1), \qquad \tau_{1\alpha} = \tau_{1\alpha}(\bar{\tilde{I}}_1), \qquad \alpha = 1, \dots, N^{\max}, \tag{47}$$

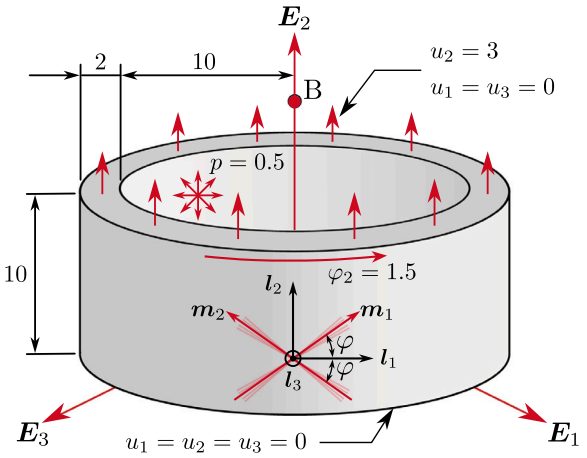


Fig. 12. Initial geometry and loading of the artery model. All dimensions are in [mm]. The top surface is displaced by $u_2 = 3$ mm in the $\boldsymbol{E}_2$-direction and rotated by $\varphi_2 = 1.5$ rad about the $\boldsymbol{E}_2$-axis. A pressure of $p = 0.5$ MPa is applied to the inner surface of the cylinder.

where

$$\bar{\tilde{I}}_1 = (\bar{\tilde{I}}_1, \bar{\tilde{J}}_1), \qquad \bar{\tilde{I}}_1 = \mathrm{tr}[\bar{\mathbf{C}}\tilde{\mathbf{L}}_1], \qquad \bar{\tilde{J}}_1 = \mathrm{tr}[(\mathrm{cof}\,\bar{\mathbf{C}})\tilde{\mathbf{L}}_1]. \tag{48}$$

For simplicity, we neglect the strain-rate dependence of the relaxation times and omit the feature vector $\boldsymbol{f}$, since no additional information is available to characterize the data.

We trained the vCANN exclusively on the five multi-step relaxation tests, and the results show excellent agreement with the training data (Fig. 13). We also analyzed the mean fiber directions and the generalized structural tensor $\tilde{\mathbf{L}}_1$ learned by the vCANN. To provide a geometric interpretation of $\tilde{\mathbf{L}}_1$, we compute the directional anisotropy factor $\omega(\boldsymbol{N}) = \boldsymbol{N}^{\mathrm{T}}\tilde{\mathbf{L}}_1\boldsymbol{N}$, where $\boldsymbol{N}$ is a unit vector representing an arbitrary direction, cf. [111]. When plotted over all directions in three-dimensional space, the distance from the origin to the surface ω visualizes the degree of anisotropy as a function of direction. Fig. 14 shows the relative error in ω between the GT model and the vCANN, evaluated on the surface of the GT model to illustrate the spatial error distribution. The maximum relative error is approximately 1%, and it is mostly below 0.5% elsewhere. These results are consistent with the identified structural parameters: the vCANN predicts a fiber angle of $\varphi = 26.41°$, close to the ground-truth value $\varphi = 26.25°$, and a weight factor $w_1^{(1)} = 0.233$, close to $w_1^{(1)} = 0.23$.

5.3.5. Predictive performance in FE simulation

The vCANN was then automatically translated into a material subroutine and integrated into the FE simulation of the anisotropic artery model. First, we examine in Fig. 15 the contour plots of relative error in von Mises stress between the vCANN prediction and the GT model at the time points of load reversal. Even during critical load reversals, the error does not exceed 5.2%. For a comprehensive evaluation, Fig. 16 shows the empirical probability density function and the empirical cumulative distribution function of the relative error in von Mises stress. These functions take into account the relative errors in each integration point in all time instances. Notably, the cumulative distribution function indicates that in 90% of the cases, the error is less than 1.0%, and in 60% of the cases, the error is even less than 0.2%. Finally, we examined the global structural response of the artery model by considering the reaction force and moment in the $\boldsymbol{E}_2$-direction at control point B, Fig. 17. Similarly, the prediction of the structural response by the vCANN is in remarkable alignment with the reference, Fig. 17.

In summary, the vCANN generalizes significantly well from just a few biaxial multi-step relaxation tests to the structural level and even to the integration point level under complex loading not present in the training data. These results are particularly notable, given the artery model was cyclically loaded, but the training data comprised only relaxation tests. In Appendix I, we analyze the same boundary value problem for the artery segment using an isotropic GT model and achieve even stronger performance in predicting the material response in the FE simulations.

Improvements can be expected by refining the step size of the multi-step relaxation test in the stretch space. Here, we sampled the deformation steps uniformly spaced in the stretch space. However, the highly nonlinear mapping between stretch and invariant space results in a non-uniform spacing in the invariant space. In addition, performing biaxial tensile tests with stretch ratios other than those listed in Table 2, or pre-stretched biaxial tests, can help to cover larger portions of the invariant space [112, Figure 21], providing a complete representation of the mechanical response of the material across different deformation modes.

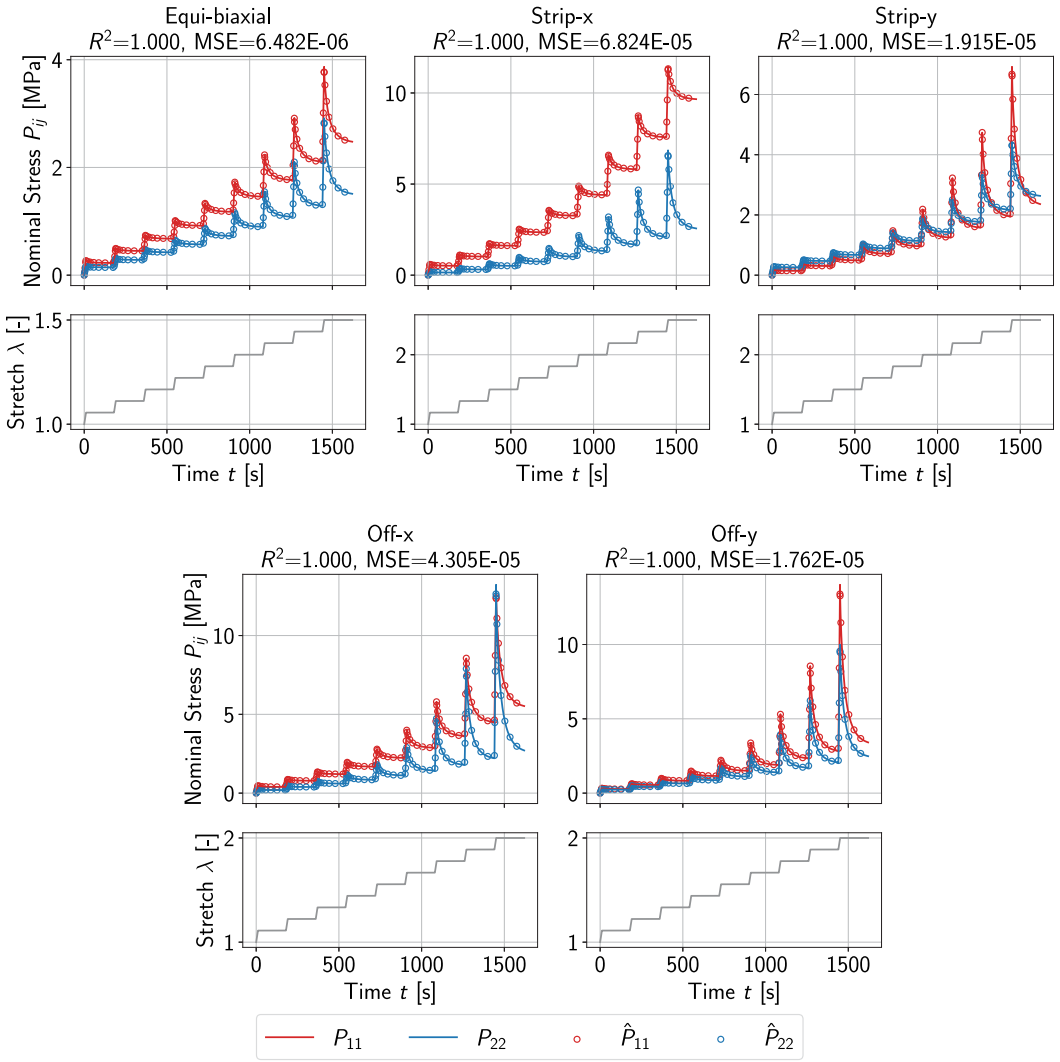


Fig. 13. Solid curves denote the vCANN fit; dotted curves denote training data from the anisotropic ground truth model (shown with every tenth data point for clarity). Bottom: Stretch histories of the multi-step relaxation tests.

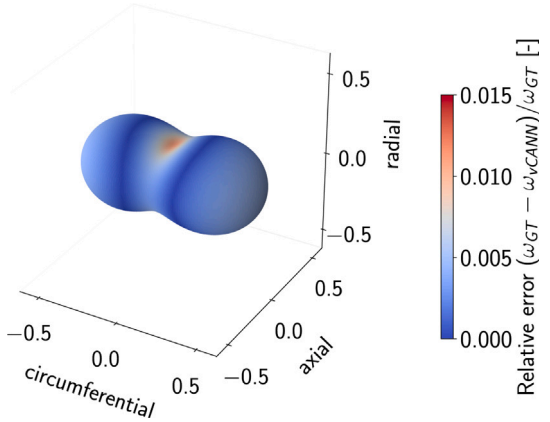


Fig. 14. Relative error in the directional anisotropy factor ω between the ground truth model and the vCANN. The directional anisotropy factor is computed via $\omega(\boldsymbol{N}) = \boldsymbol{N}^{\mathrm{T}}\tilde{\mathbf{L}}_1\boldsymbol{N}$, where $\boldsymbol{N}$ is a unit vector representing an arbitrary direction.

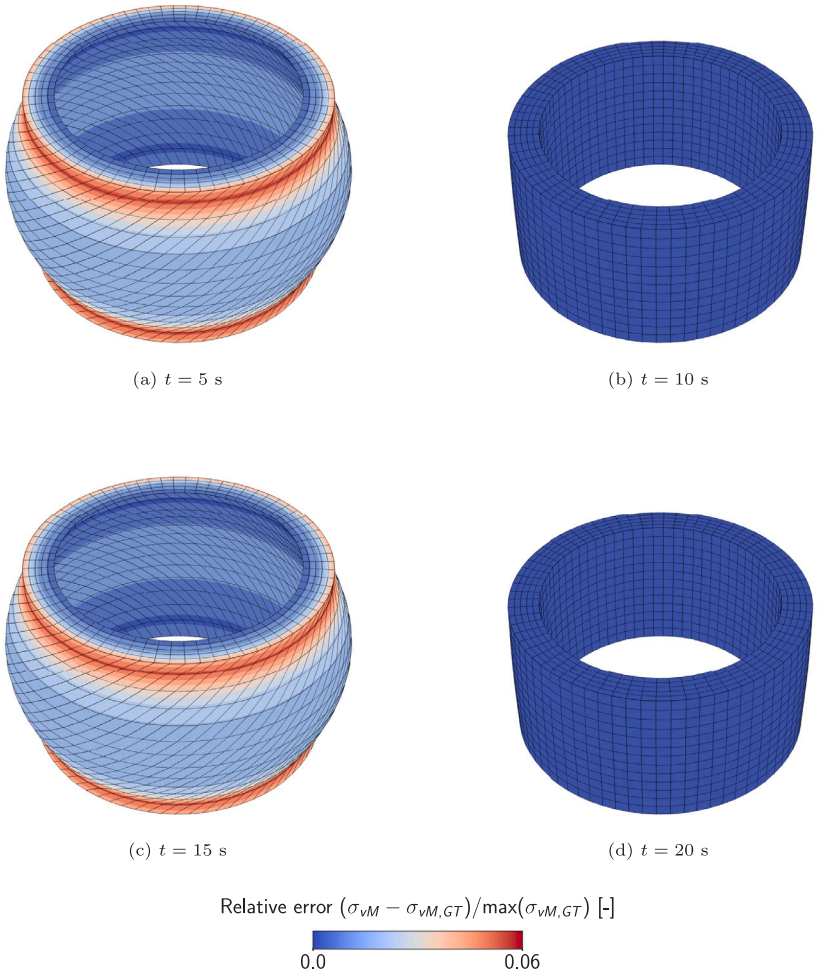


Fig. 15. Anisotropic artery model: relative error in von Mises stress between the vCANN prediction σ_{vM} and the ground truth solution $\sigma_{vM,GT}$ at the time points of load reversal.

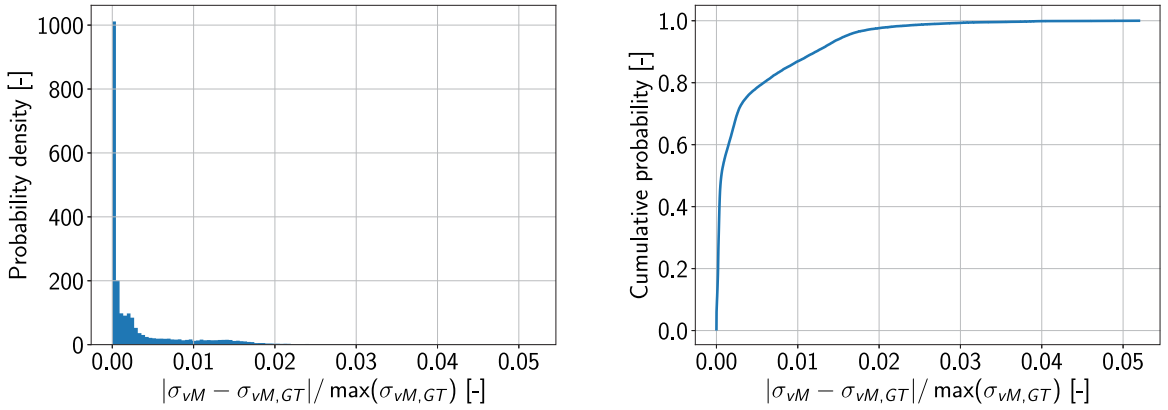


Fig. 16. Anisotropic artery model: empirical probability density function (left) and empirical cumulative distribution function (right) of the relative error in von Mises stress between the vCANN prediction σ_{vM} and the ground truth solution $\sigma_{vM,GT}$. The errors were calculated in each time instance in all integration points.

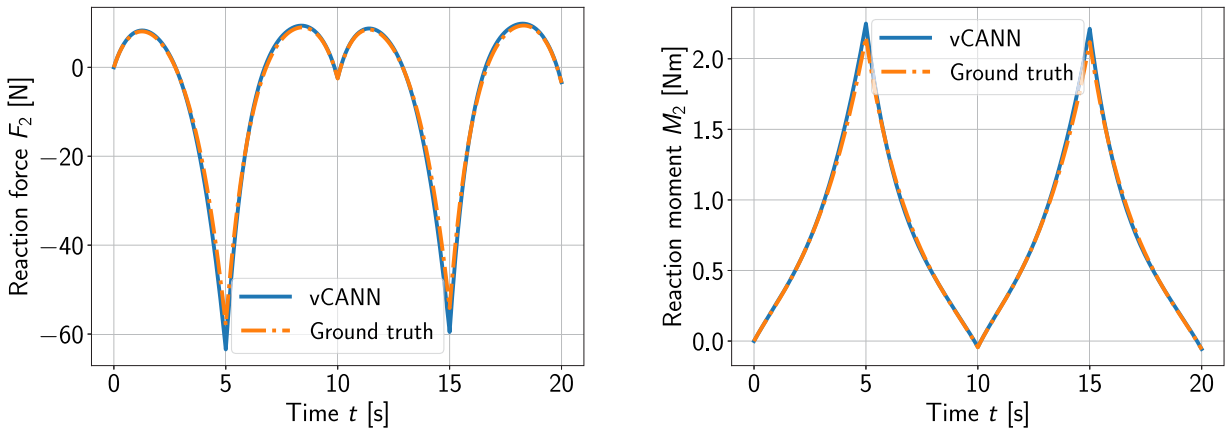


Fig. 17. Anisotropic artery model: time history of the vertical reaction force F_2 (left) and the torsional reaction moment M_2 (right) recorded at the coupling point B. Solid curves represent the vCANN predictions, and dash-dotted curves are the ground truth solutions.

6. Conclusion

Viscoelastic constitutive artificial neural networks (vCANNs) leverage neural networks for data-driven modeling of the viscoelastic behavior of materials. Herein, we propose a thermodynamically consistent extension of vCANNs that captures anisotropic, nonlinear, and time-dependent material behavior. A key strength of this approach is its ability to incorporate arbitrary auxiliary features—such as temperature, microstructural descriptors, or additional state variables—directly into neural constitutive laws. We propose an automated pipeline for the generation and implementation of such constitutive laws into FE simulations by way of material subroutines. We demonstrated that a staggered training scheme efficiently exploits available experimental data, improving the interpretability and potentially generalizability of the models. The automated pipeline minimizes user bias and involvement. We validated it across a broad range of representative examples, including nonlinear thermo-viscoelastic soft polymers, and arterial tissue with fiber dispersion. Through numerical examples, we demonstrated the ability of vCANNs to generalize from the integration point to the structural level. In all examples, the predictions closely resembled the reference stress response, even for complex, three-dimensional, anisotropic stress–strain histories not present in the training data. We also demonstrated the feasibility of reliably detecting anisotropy from stress–strain data alone, without microstructural information.

Overall, this study provides a largely automated solution to the complex task of deriving a nonlinear viscoelastic constitutive law and implementing it in a FE simulation code. By minimizing user involvement and bias, we anticipate that our approach will improve the efficiency and reliability of FE simulations in various engineering domains.

CRediT authorship contribution statement

Kian P. Abdolazizi: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Roland C. Aydin:** Writing – review & editing. **Christian J. Cyron:** Writing – review & editing, Supervision, Resources, Methodology, Funding acquisition, Conceptualization. **Kevin Linka:** Writing – review & editing, Supervision, Resources, Methodology, Funding acquisition, Conceptualization.

Code availability

The source code is openly provided at: https://github.com/KianAbd/vCANN_FEM.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used OpenAI's ChatGPT-5.2 to improve the language and readability of the text. After using this tool, they reviewed and edited the content as needed and take full responsibility for the published article's content.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

K. Linka gratefully acknowledges financial support from Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 533187597; 517243167. C.J. Cyron greatfully acknowledges support of the European Research Council (ERC) under the European Union's Horizon Europe research and innovation programme (grant agreement No 101167207/MechVivo). The authors sincerely thank Ju Liu, Southern University of Science and Technology (SUSTech) in Shenzhen, China, for helpful discussions.

Appendix A. Comparison of the vCANN frameworks

In line with [57, Section 5], we propose two main modifications to the vCANN framework introduced in [50] and summarized in Box 2.

The first modification concerns the evolution of the internal variable $\mathbf{Q}_{r\alpha}$. In the original vCANN, its evolution is driven by $g_{r\alpha}\mathbf{S}^{\infty}_{\mathrm{iso}\,r}$, where $g_{r\alpha}$ is a strain (rate)-dependent relaxation coefficient. In the proposed framework, the evolution is instead driven by the fictitious stress $\bar{\mathbf{S}}_{r\alpha}$.

The second modification concerns the relationship between the internal variable $\mathbf{Q}_{r\alpha}$ and the non-equilibrium stress $\mathbf{S}^{\mathrm{neq}}_{r\alpha}$. In the original vCANN, these two quantities are identified directly, i.e., $\mathbf{S}^{\mathrm{neq}}_{r\alpha} = \mathbf{Q}_{r\alpha}$. In the proposed framework, they are in general distinct and are related by $\bar{\mathbf{S}}^{\mathrm{neq}}_{r\alpha} = \frac{1}{2\mu_{r\alpha}}\bar{\mathbb{C}}_{r\alpha} : \mathbf{Q}_{r\alpha}$, see (30). The two quantities coincide only under the restrictive assumption that the auxiliary potential takes the quadratic form $\Psi_{r\alpha} = \frac{\mu_{r\alpha}}{4}|\bar{\mathbf{C}} - \mathbf{I}|^2$, which implies $\bar{\mathbb{C}}_{r\alpha} = 2\mu_{r\alpha}\mathbb{I}^{\mathrm{s}}$ [57, Section 2.4.4]. Together, these two modifications resolve the concerns about the thermodynamic soundness of [51,52] on which the original vCANN [50] was based.

An important consequence of the second modification is its impact on FE implementation. In most constitutive frameworks, the stress is obtained from first-order derivatives of a potential with respect to its strain arguments, and the consistent tangent modulus requires second-order derivatives. In the proposed framework, however, the non-equilibrium stress $\bar{\mathbf{S}}^{\mathrm{neq}}_{r\alpha}$ involves the elasticity tensor $\bar{\mathbb{C}}_{r\alpha} = 4\,\frac{\partial^2\Psi_{r\alpha}}{\partial\bar{\mathbf{C}}^2}$, which is itself a second-order derivative of $\Psi_{r\alpha}$. Consequently, the consistent tangent modulus requires *third-order* derivatives of the neural network potential $\Psi_{r\alpha}$ with respect to its inputs. For more details, we refer to [50].

Box 2: Initial vCANN

$$\mathbf{S} = \mathbf{S}^{\infty}_{\mathrm{vol}} + \sum_{r=1}^{R}\left(\mathbf{S}^{\infty}_{\mathrm{iso}\,r} + \sum_{\alpha=1}^{N}\mathbf{S}^{\mathrm{neq}}_{r\alpha}\right), \qquad \mathbf{S}^{\infty}_{\mathrm{iso}\,r} = J^{-2/3}\mathbb{P} : \bar{\mathbf{S}}^{\infty}_{\mathrm{iso}\,r}, \qquad \mathbf{S}^{\mathrm{neq}}_{r\alpha} = \mathbf{Q}_{r\alpha},$$

$$\dot{\mathbf{Q}}_{r\alpha} + \frac{\mathbf{Q}_{r\alpha}}{\tau_{r\alpha}} = (g_{r\alpha}\mathbf{S}^{\infty}_{\mathrm{iso}\,r})^{\cdot}, \qquad \bar{\mathbf{S}}^{\infty}_{\mathrm{iso}\,r} = \frac{\partial\Psi^{\infty}_{\mathrm{iso}\,r}}{\partial\bar{\mathbf{C}}}.$$

Appendix B. Compressible materials

Here, we briefly summarize the governing equations for compressible materials.

B.1. Material symmetry: generalized structural tensors and invariants

Analogous to (6) and (7), we define the generalized invariants of $\mathbf{C}$ and $\dot{\mathbf{C}}$ as

$$\tilde{I}_r = \mathrm{tr}\left[\mathbf{C}\tilde{\mathbf{L}}_r\right], \qquad \tilde{J}_r = \mathrm{tr}\left[(\mathrm{cof}\,\mathbf{C})\tilde{\mathbf{L}}_r\right], \qquad \mathrm{III}_{\mathbf{C}} = \det\mathbf{C} = J^2, \tag{B.1}$$

$$\dot{\tilde{I}}_r = \mathrm{tr}\left[\dot{\mathbf{C}}\tilde{\mathbf{L}}_r\right], \qquad \dot{\tilde{J}}_r = \mathrm{tr}\left[(\mathrm{cof}\,\dot{\mathbf{C}})\tilde{\mathbf{L}}_r\right], \qquad \mathrm{III}_{\dot{\mathbf{C}}} = \det\dot{\mathbf{C}}, \qquad r = 1, 2, \dots, R. \tag{B.2}$$

The generalized invariants (B.1) are convex with respect to $\mathbf{F}$, $\mathrm{cof}\,\mathbf{F}$, and $\det\mathbf{F}$, and thus provide a suitable basis for formulating polyconvex free energy functions [64]. The following differentiation relations will be useful:

$$\frac{\partial\tilde{I}_r}{\partial\mathbf{C}} = \tilde{\mathbf{L}}_r, \qquad \frac{\partial\tilde{J}_r}{\partial\mathbf{C}} = \tilde{J}_r\mathbf{C}^{-1} - \mathrm{III}_{\mathbf{C}}\,\mathbf{H}_r, \qquad \frac{\partial\mathrm{III}_{\mathbf{C}}}{\partial\mathbf{C}} = \mathrm{III}_{\mathbf{C}}\,\mathbf{C}^{-1}, \qquad \text{with} \quad \mathbf{H}_r := \mathbf{C}^{-1}\tilde{\mathbf{L}}_r\mathbf{C}^{-1}. \tag{B.3}$$

Finally, we introduce the following shorthand notations in terms of tuples:

$$\mathcal{I}_r := \left(\tilde{I}_r, \dot{\tilde{I}}_r\right) \qquad \text{with} \qquad \tilde{I}_r := \left(\tilde{I}_r, \tilde{J}_r\right), \qquad \dot{\tilde{I}}_r := \left(\dot{\tilde{I}}_r, \dot{\tilde{J}}_r, \mathrm{III}_{\mathbf{C}}\right), \qquad r = 1, 2, \dots, R, \tag{B.4}$$

$$\mathcal{I} := \left(\tilde{I}, \dot{\tilde{I}}\right) \qquad \text{with} \qquad \tilde{I} := \left(\tilde{I}_1, \dots, \tilde{I}_R\right), \qquad \dot{\tilde{I}} := \left(\dot{\tilde{I}}_1, \dots, \dot{\tilde{I}}_R\right). \tag{B.5}$$

B.2. Constitutive modeling framework

In analogy to (14), we assume a free energy function that is additively decomposed into equilibrium and non-equilibrium contributions Ψ^{∞}_r and $\Psi^{\mathrm{neq}}_{r\alpha}$:

$$\Psi\left(\mathbf{C}, \boldsymbol{\Gamma}, \boldsymbol{f}\right) = \sum_{r=1}^{R}\left[\Psi^{\infty}_r\left(\tilde{I}_r\right) + \sum_{\alpha=1}^{N}\Psi^{\mathrm{neq}}_{r\alpha}\left(\tilde{I}_r, \boldsymbol{\Gamma}_{r\alpha}, \boldsymbol{f}\right)\right]. \tag{B.6}$$

The non-equilibrium contribution is defined as

$$\Psi_{r\alpha}^{\mathrm{neq}}\left(\tilde{I}_r, \mathbf{\Gamma}_{r\alpha}, \boldsymbol{f}\right) = \frac{1}{4\mu_{r\alpha}} \left\| \mathbf{S}_{r\alpha}\left(\tilde{I}_r, \boldsymbol{f}\right) - \mu_{r\alpha}\left(\mathbf{\Gamma}_{r\alpha} - \mathbf{I}\right) \right\|^2, \tag{B.7}$$

where $\mu_{r\alpha}$ are constant shear moduli. The analogy to the generalized Maxwell model (Fig. 2) still holds. Note for compressible materials, an additional invariant, $-2J$ (with $J = \sqrt{\mathrm{III}_{\mathbf{C}}}$), should be fed into the free energy functions. See [37, (3.2)] for more information.

Following standard thermodynamic arguments, (16) and (17), the total stress $\mathbf{S}$ is the sum of equilibrium and non-equilibrium contributions $\mathbf{S}_r^{\infty}$ and $\mathbf{S}_{r\alpha}^{\mathrm{neq}}$, respectively:

$$\mathbf{S} = \sum_{r=1}^{R} \left(\mathbf{S}_r^{\infty} + \sum_{\alpha=1}^{N} \mathbf{S}_{r\alpha}^{\mathrm{neq}} \right). \tag{B.8}$$

The equilibrium contributions are defined as

$$\mathbf{S}_r^{\infty} = 2\frac{\partial \Psi_r^{\infty}}{\partial \mathbf{C}} = 2\left[\frac{\partial \Psi_r^{\infty}}{\partial \tilde{I}_r} \tilde{\mathbf{L}}_r + \frac{\partial \Psi_r^{\infty}}{\partial \tilde{J}_r} \left(\tilde{J}_r \mathbf{C}^{-1} - \mathrm{III}_{\mathbf{C}}\, \mathbf{H}_r \right) + \mathrm{III}_{\mathbf{C}} \frac{\partial \Psi_r^{\infty}}{\partial \mathrm{III}_{\mathbf{C}}} \mathbf{C}^{-1} \right]. \tag{B.9}$$

In (B.7), the stresses $\mathbf{S}_{r\alpha}$ are derived from potentials $\Psi_{r\alpha} = \Psi_{r\alpha}\left(\tilde{I}_r, \boldsymbol{f}\right)$, in analogy to (B.9):

$$\mathbf{S}_{r\alpha} = 2\frac{\partial \Psi_{r\alpha}}{\partial \mathbf{C}} = 2\left[\frac{\partial \Psi_{r\alpha}}{\partial \tilde{I}_r} \tilde{\mathbf{L}}_r + \frac{\partial \Psi_{r\alpha}}{\partial \tilde{J}_r} \left(\tilde{J}_r \mathbf{C}^{-1} - \mathrm{III}_{\mathbf{C}}\, \mathbf{H}_r \right) + \mathrm{III}_{\mathbf{C}} \frac{\partial \Psi_{r\alpha}}{\partial \mathrm{III}_{\mathbf{C}}} \mathbf{C}^{-1} \right]. \tag{B.10}$$

Introducing the auxiliary stress

$$\mathbf{Q}_{r\alpha} = -2\frac{\partial \Psi_{r\alpha}^{\mathrm{neq}}}{\partial \mathbf{\Gamma}_{r\alpha}} = \mathbf{S}_{r\alpha} - \mu_{r\alpha}\left(\mathbf{\Gamma}_{r\alpha} - \mathbf{I}\right), \tag{B.11}$$

the non-equilibrium contributions in (B.8) follow as

$$\mathbf{S}_{r\alpha}^{\mathrm{neq}} = 2\frac{\partial \Psi_{r\alpha}^{\mathrm{neq}}}{\partial \mathbf{C}} = \frac{1}{2\mu_{r\alpha}} \mathbb{C}_{r\alpha} : \mathbf{Q}_{r\alpha}, \qquad \mathbb{C}_{r\alpha} := 2\frac{\partial \mathbf{S}_{r\alpha}}{\partial \mathbf{C}}. \tag{B.12}$$

In the compressible case, the normalization condition (25) likewise applies to the equilibrium free energy part. Additionally, it is often required that the equilibrium free energy satisfies the growth condition, ensuring that the energy approaches infinity as the material either shrinks to zero or expands to infinite volume [74]:

$$\Psi_r^{\infty} \to \infty \qquad \text{as} \qquad J \to 0^+ \quad \text{or} \quad J \to \infty. \tag{B.13}$$

Appendix C. Normalization and growth conditions

Classical constitutive models enforce a stress- and energy-free reference configuration by carefully constructing and combining appropriate basis functions. In contrast, a free energy function represented by an FFNN does not inherently satisfy these requirements. Therefore, in (34) and (35), we introduced correction terms for the neural network-based free energies $\Psi_{\mathrm{iso}\,r}^{\infty}$ and $\Psi_{r\alpha}$, respectively, to enforce the normalization conditions. In addition, for compressible materials it may be important to satisfy the growth condition (B.13). In this appendix, we introduce the required correction terms for the free energy functions. For brevity, we focus on $\Psi_{\mathrm{iso}\,r}^{\infty}$; the construction for $\Psi_{r\alpha}$ is analogous.

C.1. (Nearly) incompressible case

C.1.1. Volumetric part

For perfectly incompressible materials, $\Psi_{\mathrm{vol}}^{\infty}$ represents the kinematic constraint $\Psi_{\mathrm{vol}}^{\infty}(J) = p(J-1)$, which is enforced numerically, e.g., by the Lagrange-multiplier method. For nearly incompressible materials, volumetric deformations are small ($J \approx 1$) and $\Psi_{\mathrm{vol}}^{\infty}$ is introduced as a generic, material-agnostic penalty function [79] to approximately enforce incompressibility. This motivates the pragmatic choice of prescribing an analytical from the extensive catalog reported in the literature [80,81], rather than representing $\Psi_{\mathrm{vol}}^{\infty}$ by a neural network. By construction, such penalty functions can be chosen convex and to satisfy the normalization conditions.

As an example, a suitable penalty function and its associated stress contribution read (cf. [81])

$$\Psi_{\mathrm{vol}}^{\infty}(J) = \kappa\left(J^2 + \frac{1}{J^2} - 2 \right), \qquad \mathbf{S}_{\mathrm{vol}}^{\infty} = J\frac{\partial \Psi_{\mathrm{iso}}^{\infty}}{\partial J}\mathbf{C}^{-1} = 2\kappa\left(J^2 - \frac{1}{J^2} \right)\mathbf{C}^{-1}, \tag{C.1}$$

where $\kappa > 0$ denotes the compression modulus and is chosen sufficiently large to enforce near-incompressibility.

C.1.2. Isochoric part

In (34), we defined the isochoric equilibrium free energies $\Psi_{\mathrm{iso}\,r}^{\infty}$ as

$$\Psi_{\mathrm{iso}\,r}^{\infty}(\bar{\bar{I}}_r, \boldsymbol{f}) := \Psi_{\mathrm{iso}\,r}^{\mathcal{N}}(\bar{\bar{I}}_r, \boldsymbol{f}) + \Psi_{\mathrm{iso}\,r}^{\sigma}(\bar{\bar{I}}_r, \boldsymbol{f}) + \Psi_{\mathrm{iso}\,r}^{\varepsilon}(\boldsymbol{f}), \tag{C.2}$$

where $\Psi_{\mathrm{iso}\,r}^{\mathcal{N}}$ is a convex, non-decreasing free energy function represented by the invariant-based FFNN, and $\Psi_{\mathrm{iso}\,r}^{\sigma}$ and $\Psi_{\mathrm{iso}\,r}^{\varepsilon}$ are correction terms that ensure a stress- and energy-free reference configuration, respectively. The correction terms are defined below, and their derivation follows [37].

Stress normalization. To determine $\Psi^{\sigma}_{\mathrm{iso}\,r}$, we first compute the stress $\mathbf{S}^{\mathcal{N}}_{\mathrm{iso}\,r}$ implied by $\Psi^{\mathcal{N}}_{\mathrm{iso}\,r}$. In analogy to (22) and (23), we obtain

$$\mathbf{S}^{\mathcal{N}}_{\mathrm{iso}\,r} = 2\frac{\partial \Psi^{\mathcal{N}}_{\mathrm{iso}\,r}}{\partial \mathbf{C}} = 2J^{-2/3}\left[\frac{\partial \Psi^{\mathcal{N}}_{\mathrm{iso}\,r}}{\partial \bar{\bar{I}}_r}\left(\tilde{\mathbf{L}}_r - \frac{1}{3}\bar{\bar{I}}_r\,\bar{\mathbf{C}}^{-1}\right) - \frac{\partial \Psi^{\mathcal{N}}_{\mathrm{iso}\,r}}{\partial \bar{\bar{J}}_r}\left(\bar{\mathbf{H}}_r - \frac{1}{3}\bar{\bar{J}}_r\,\bar{\mathbf{C}}^{-1}\right)\right]. \tag{C.3}$$

In view of (6), the generalized invariants evaluate in the reference configuration to $\bar{\bar{I}}_r = \bar{\bar{J}}_r = J = 1$, and hence

$$\mathbf{S}^{\mathcal{N}}_{\mathrm{iso}\,r}\Big|_{\mathbf{C}=\mathbf{I}} = \bar{\Delta}_r\left(\tilde{\mathbf{L}}_r - \frac{1}{3}\mathbf{I}\right), \qquad \text{with} \qquad \bar{\Delta}_r := \left(\frac{\partial \Psi^{\mathcal{N}}_{\mathrm{iso}\,r}}{\partial \bar{\bar{I}}_r} - \frac{\partial \Psi^{\mathcal{N}}_{\mathrm{iso}\,r}}{\partial \bar{\bar{J}}_r}\right)\Bigg|_{\mathbf{C}=\mathbf{I}}. \tag{C.4}$$

To compensate for this generally non-zero stress contribution, we construct the stress-normalization term $\Psi^{\sigma}_{\mathrm{iso}\,r}$ such that its associated stress, $\mathbf{S}^{\sigma}_{\mathrm{iso}\,r}$, exactly evaluates to $-\mathbf{S}^{\mathcal{N}}_{\mathrm{iso}\,r}|_{\mathbf{C}=\mathbf{I}}$ in the reference configuration:

$$\Psi^{\sigma}_{\mathrm{iso}\,r}(\bar{\bar{I}}_r, \boldsymbol{f}) = \bar{\alpha}_r\left(\bar{\bar{I}}_r - 1\right) + \bar{\beta}_r\left(\bar{\bar{J}}_r - 1\right), \tag{C.5}$$

with

$$\bar{\alpha}_r := \max(0, -\bar{\Delta}_r), \qquad \bar{\beta}_r := \max(0, \bar{\Delta}_r). \tag{C.6}$$

During training, the non-negative parameters $\bar{\alpha}_r$ and $\bar{\beta}_r$ evolve, since they depend on $\Psi^{\mathcal{N}}_{\mathrm{iso}\,r}$. After training, they are treated as constants (for fixed $\boldsymbol{f}$). Note that $\Psi^{\sigma}_{\mathrm{iso}\,r}$ is convex and monotonically increasing in its arguments, and therefore admits a polyconvex representation whenever its arguments do. The corresponding stress evaluates to

$$\mathbf{S}^{\sigma}_{\mathrm{iso}\,r} = 2J^{-2/3}\left[\bar{\alpha}_r\left(\tilde{\mathbf{L}}_r - \frac{1}{3}\bar{\bar{I}}_r\,\bar{\mathbf{C}}^{-1}\right) - \bar{\beta}_r\left(\bar{\mathbf{H}}_r - \frac{1}{3}\bar{\bar{J}}_r\,\bar{\mathbf{C}}^{-1}\right)\right], \tag{C.7}$$

which indeed cancels (C.4) in the reference configuration. Note that $\Psi^{\sigma}_{\mathrm{iso}\,r}$ and vanishes in the reference configuration.

Energy normalization. Since $\Psi^{\infty}_{\mathrm{iso}\,r}$ vanishes in the reference configuration by construction, the normalization term to ensure an energy-free reference configuration is simply

$$\Psi^{\varepsilon}_{\mathrm{iso}\,r}(\boldsymbol{f}) := -\Psi^{\mathcal{N}}_{\mathrm{iso}\,r}(\bar{\bar{I}}_r, \boldsymbol{f})\Big|_{\mathbf{C}=\mathbf{I}}, \tag{C.8}$$

which motivates the structure (C.2) of $\Psi^{\infty}_{\mathrm{iso}\,r}$. Since $\Psi^{\varepsilon}_{\mathrm{iso}\,r}$ is constant with respect to $\mathbf{C}$, it does not contribute to the stress, i.e., $\mathbf{S}^{\varepsilon}_{\mathrm{iso}\,r} = 2\partial\Psi^{\varepsilon}_{\mathrm{iso}\,r}/\partial\mathbf{C} = \mathbf{0}$.

Total equilibrium stress. Finally, by adding (C.3) and (C.7), the total isochoric equilibrium stress reads

$$\begin{aligned}\mathbf{S}^{\infty}_{\mathrm{iso}\,r} &= 2\frac{\partial \Psi^{\infty}_{\mathrm{iso}\,r}}{\partial \mathbf{C}} = \mathbf{S}^{\mathcal{N}}_{\mathrm{iso}\,r} + \mathbf{S}^{\sigma}_{\mathrm{iso}\,r} \\ &= 2J^{-2/3}\left[\left(\frac{\partial \Psi^{\mathcal{N}}_{\mathrm{iso}\,r}}{\partial \bar{\bar{I}}_r} + \bar{\alpha}_r\right)\left(\tilde{\mathbf{L}}_r - \frac{1}{3}\bar{\bar{I}}_r\,\bar{\mathbf{C}}^{-1}\right) - \left(\frac{\partial \Psi^{\mathcal{N}}_{\mathrm{iso}\,r}}{\partial \bar{\bar{J}}_r} + \bar{\beta}_r\right)\left(\bar{\mathbf{H}}_r - \frac{1}{3}\bar{\bar{J}}_r\,\bar{\mathbf{C}}^{-1}\right)\right],\end{aligned} \tag{C.9}$$

from which we deduce by comparison with that

$$\frac{\partial \Psi^{\infty}_{\mathrm{iso}\,r}}{\partial \bar{\bar{I}}_r} = \frac{\partial \Psi^{\mathcal{N}}_{\mathrm{iso}\,r}}{\partial \bar{\bar{I}}_r} + \bar{\alpha}_r, \qquad \frac{\partial \Psi^{\infty}_{\mathrm{iso}\,r}}{\partial \bar{\bar{J}}_r} = \frac{\partial \Psi^{\mathcal{N}}_{\mathrm{iso}\,r}}{\partial \bar{\bar{J}}_r} + \bar{\beta}_r. \tag{C.10}$$

C.2. Compressible case

Similar to (C.2), we define the equilibrium free energy as

$$\Psi^{\infty}_{r}(\tilde{I}_r, \boldsymbol{f}) := \Psi^{\mathcal{N}}_{r}(\tilde{I}_r, \boldsymbol{f}) + \Psi^{\sigma}_{r}(\tilde{I}_r, \boldsymbol{f}) + \Psi^{\varepsilon}_{r}(\boldsymbol{f}) + \Psi^{\uparrow}_{r}(J), \tag{C.11}$$

where $\Psi^{\mathcal{N}}_{r}$ is a convex, non-decreasing free energy function represented by an invariant-based FFNN, cf. 3.1. The terms Ψ^{σ}_{r} and Ψ^{ε}_{r} enforce a stress- and energy-free reference configuration, respectively. The growth function $\Psi^{\uparrow}_{r}$ serves solely to satisfy the growth condition and is therefore largely material-agnostic [79]. In practice, $\Psi^{\uparrow}_{r}$ can be prescribed analytically by selecting a suitable form from the same catalog of volumetric penalty functions [80,81], for example (C.1), rather than representing it by a neural network. Such choices can be made convex (and hence polyconvex) and to satisfy the normalization conditions by construction. With these choices, Ψ^{∞}_{r} admits a polyconvex representation; details on the normalization properties and polyconvexity of the individual contributions are discussed next.

Stress normalization. In analogy to (B.9), the stress associated with $\Psi^{\mathcal{N}}_{r}$ reads

$$\mathbf{S}^{\mathcal{N}}_{r} = 2\left[\frac{\partial \Psi^{\mathcal{N}}_{r}}{\partial \tilde{I}_r}\tilde{\mathbf{L}}_r + \frac{\partial \Psi^{\mathcal{N}}_{r}}{\partial \tilde{J}_r}\left(\tilde{J}_r\mathbf{C}^{-1} - \mathrm{III}_{\mathbf{C}}\,\mathbf{H}_r\right)\right] + 2\,\mathrm{III}_{\mathbf{C}}\frac{\partial \Psi^{\mathcal{N}}_{r}}{\partial \mathrm{III}_{\mathbf{C}}}\mathbf{C}^{-1}. \tag{C.12}$$

In view of (B.1), the generalized invariants evaluate in the reference configuration to $\tilde{I}_r = \tilde{J}_r = \mathrm{III}_{\mathbf{C}} = 1$, and thus

$$\mathbf{S}^{\mathcal{N}}_{r}\Big|_{\mathbf{C}=\mathbf{I}} = 2\left[\Delta_r\,\tilde{\mathbf{L}}_r + \gamma_r\,\mathbf{I}\right] + 2\delta_r\,\mathbf{I}, \tag{C.13}$$

where

$$\Delta_r := \left(\frac{\partial \Psi_r^{\mathcal{N}}}{\partial \tilde{I}_r} - \frac{\partial \Psi_r^{\mathcal{N}}}{\partial \tilde{J}_r} \right) \Big|_{\mathbf{C}=\mathbf{I}}, \qquad \gamma_r := \frac{\partial \Psi_r^{\mathcal{N}}}{\partial \tilde{J}_r} \Big|_{\mathbf{C}=\mathbf{I}}, \qquad \delta_r := \frac{\partial \Psi_r^{\mathcal{N}}}{\partial \mathrm{III}_{\mathbf{C}}} \Big|_{\mathbf{C}=\mathbf{I}}. \tag{C.14}$$

To compensate for this generally non-zero stress contribution, we construct the stress-normalization term Ψ_r^{σ} as

$$\Psi_r^{\sigma}(\tilde{I}_r, \boldsymbol{f}) = \alpha_r(\tilde{I}_r - 1) + \beta_r(\tilde{J}_r - 1) - 2\,(\beta_r + \gamma_r + \delta_r)\,(J - 1), \tag{C.15}$$

where

$$\alpha_r := \max(0, -\Delta_r), \qquad \beta_r := \max(0, \Delta_r), \tag{C.16}$$

are non-negative constants (for fixed $\boldsymbol{f}$). Using the differentiation relations (B.3), we obtain

$$\mathbf{S}_r^{\sigma} = 2\left[\alpha_r \tilde{\mathbf{L}}_r + \beta_r \left(\tilde{J}_r \mathbf{C}^{-1} - \mathrm{III}_{\mathbf{C}}\, \mathbf{H}_r\right) - J(\beta_r + \gamma_r + \delta_r)\, \mathbf{C}^{-1}\right], \tag{C.17}$$

which cancels (C.13) in the reference configuration. Note that Ψ_r^{σ} is convex and monotonically increasing in its invariant arguments; consequently, it admits a polyconvex representation.

Energy normalization. Since both Ψ_r^{σ} and $\Psi_r^{\uparrow}$ vanish in the reference configuration by construction, the correction term ensuring an energy-free reference configuration is defined as

$$\Psi_r^{\varepsilon}(\boldsymbol{f}) := -\Psi_r^{\mathcal{N}}(\tilde{I}_r, \boldsymbol{f})\Big|_{\mathbf{C}=\mathbf{I}}. \tag{C.18}$$

This motivates the structure (C.11) of Ψ_r^{∞}. Since Ψ_r^{ε} is constant with respect to $\mathbf{C}$, it does not contribute to the stress, i.e., $\mathbf{S}_r^{\varepsilon} = \partial \Psi_r^{\varepsilon} / \partial \mathbf{C} = \mathbf{0}$.

Total equilibrium stress. Adding $(\text{C.1})_2$, (C.12), and (C.17), the total equilibrium stress becomes

$$\begin{aligned} \mathbf{S}_r^{\infty} &= 2\frac{\partial \Psi_r^{\infty}}{\partial \mathbf{C}} = \mathbf{S}_r^{\mathcal{N}} + \mathbf{S}_r^{\sigma} + \mathbf{S}^{\uparrow} \\ &= 2\left[\left(\frac{\partial \Psi_r^{\mathcal{N}}}{\partial \tilde{I}_r} + \alpha_r\right)\tilde{\mathbf{L}}_r + \left(\frac{\partial \Psi_r^{\mathcal{N}}}{\partial \tilde{J}_r} + \beta_r\right)\left(\tilde{J}_r \mathbf{C}^{-1} - \mathrm{III}_{\mathbf{C}}\, \mathbf{H}_r\right)\right] \\ &\quad + 2\,\mathrm{III}_{\mathbf{C}}\left[\frac{\partial \Psi_r^{\mathcal{N}}}{\partial \mathrm{III}_{\mathbf{C}}} + J^{-1}\left(\frac{1}{2}\frac{\partial \Psi_r^{\uparrow}}{\partial J} - (\delta_r + \beta_r + \gamma_r)\right)\right]\mathbf{C}^{-1}, \end{aligned} \tag{C.19}$$

from which we deduce

$$\frac{\partial \Psi_r^{\infty}}{\partial \tilde{I}_r} := \frac{\partial \Psi_r^{\mathcal{N}}}{\partial \tilde{I}_r} + \alpha_r, \quad \frac{\partial \Psi_r^{\infty}}{\partial \tilde{J}_r} := \frac{\partial \Psi_r^{\mathcal{N}}}{\partial \tilde{J}_r} + \beta_r, \quad \frac{\partial \Psi_r^{\infty}}{\partial \mathrm{III}_{\mathbf{C}}} := \frac{\partial \Psi_r^{\mathcal{N}}}{\partial \mathrm{III}_{\mathbf{C}}} + J^{-1}\left(\frac{1}{2}\frac{\partial \Psi_r^{\uparrow}}{\partial J} - (\delta_r + \beta_r + \gamma_r)\right). \tag{C.20}$$

Appendix D. Derivatives of feed-forward neural networks

We derive efficient recurrence relations to compute the first-, second-, and third-order derivatives of an FFNN output vector with respect to its input vector, as required for the FE implementation of vCANNs. These relations are implemented in a Fortran utility module that is called by the material subroutine to evaluate the derivatives without relying on automatic differentiation.

D.1. General

Let $\mathcal{N} : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$, $\boldsymbol{x}^L = \mathcal{N}(\boldsymbol{x}^0)$ denote an FFNN with L layers, where the input vector is $\boldsymbol{x}^0 \in \mathbb{R}^{n_0}$ and the output vector is $\boldsymbol{x}^L \in \mathbb{R}^{n_L}$. For each layer l, let the weight matrix be $\boldsymbol{W}^l \in \mathbb{R}^{n_l \times n_{l-1}}$ and the bias vector be $\boldsymbol{b}^l \in \mathbb{R}^{n_l}$. We denote their elements by W_{ij}^l and b_i^l, respectively. The pre-activation vector $\boldsymbol{z}^l \in \mathbb{R}^{n_l}$ and the output vector $\boldsymbol{x}^l \in \mathbb{R}^{n_l}$ are defined component-wise as

$$z_i^l = \sum_{j=1}^{n_{l-1}} W_{ij}^l x_j^{l-1} + b_i^l, \tag{D.1}$$

$$x_i^l = \sigma_l(z_i^l), \qquad l = 1, \dots, L. \tag{D.2}$$

Here $\sigma_l : \mathbb{R} \to \mathbb{R}$ denotes the activation function in layer l, applied element-wise to $\boldsymbol{z}^l$. In the following, we *do not* employ the Einstein summation convention. Whenever summation is required, it is explicitly indicated using a summation symbol.

D.2. Jacobians

We introduce shorthand notations for the first, second, and third derivatives of x_i^l with respect to z_i^l:

$$d_i^l = \frac{\partial x_i^l}{\partial z_i^l} = \sigma_l'(z_i^l), \tag{D.3}$$

$$d^l_{2,i} = \frac{\partial^2 x^l_i}{\partial {z^l_i}^2} = \sigma_l''(z^l_i), \tag{D.4}$$

$$d^l_{3,i} = \frac{\partial^3 x^l_i}{\partial {z^l_i}^3} = \sigma_l'''(z^l_i), \tag{D.5}$$

where σ_l', σ_l'', and σ_l''' denote the first, second, and third derivatives of the activation function in layer l, applied element-wise to $\boldsymbol{z}^l$.

Differentiating (D.1) with respect to x^{l-1}_j gives

$$\frac{\partial z^l_i}{\partial x^{l-1}_j} = W^l_{ij}. \tag{D.6}$$

Applying the chain rule to (D.2) and using (D.3) together with (D.6), yields the partial derivatives of the outputs of layer l with respect to its inputs,

$$\frac{\partial x^l_i}{\partial x^{l-1}_j} = \frac{\partial x^l_i}{\partial z^l_i}\frac{\partial z^l_i}{\partial x^{l-1}_j} \tag{D.7}$$

$$= d^l_i W^l_{ij}. \tag{D.8}$$

Layer Jacobian. The layer Jacobian $\boldsymbol{J}^l \in \mathbb{R}^{n_l \times n_0}$ collects all first-order derivatives of the layer output $\boldsymbol{x}^l$ with respect to the network input $\boldsymbol{x}^0$:

$$\boldsymbol{J}^l := \frac{\partial \boldsymbol{x}^l}{\partial \boldsymbol{x}^0}, \qquad J^l_{ij} := \frac{\partial x^l_i}{\partial x^0_j}. \tag{D.9}$$

By the chain rule,

$$J^l_{ij} = \sum_{k=1}^{n_{l-1}} \frac{\partial x^l_i}{\partial x^{l-1}_k}\frac{\partial x^{l-1}_k}{\partial x^0_j}. \tag{D.10}$$

Inserting (D.8) yields the recursion for the layer Jacobian

$$\boxed{J^l_{ij} = \sum_{k=1}^{n_{l-1}} d^l_i W^l_{ik} J^{l-1}_{kj}, \qquad l = 1, \dots, L.} \tag{D.11}$$

The recursion starts from $\boldsymbol{J}^0 \equiv \boldsymbol{I}_{n_0}$.

Network Jacobian. The network Jacobian $\boldsymbol{J}$ is obtained at the final layer L:

$$\boldsymbol{J} := \frac{\partial \boldsymbol{x}^L}{\partial \boldsymbol{x}^0} = \boldsymbol{J}^L. \tag{D.12}$$

D.3. Hessian

Layer Hessian. The layer Hessian $\boldsymbol{H}^l \in \mathbb{R}^{n_l \times n_0 \times n_0}$ collects all second-order derivatives of the layer output $\boldsymbol{x}^l$ with respect to the network input $\boldsymbol{x}^0$:

$$\boldsymbol{H}^l := \frac{\partial^2 \boldsymbol{x}^l}{\partial \boldsymbol{x}^0 \partial \boldsymbol{x}^0}, \qquad H^l_{aij} := \frac{\partial^2 x^l_a}{\partial x^0_i \partial x^0_j}. \tag{D.13}$$

This tensor is symmetric in the last two indices: $H^l_{aij} = H^l_{aji}$.

Auxiliary quantities. Define the Jacobian of the pre-activations z^l_a with respect to the network input by

$$M^l_{ai} := \frac{\partial z^l_a}{\partial x^0_i} = \sum_{r=1}^{n_{l-1}} \frac{\partial z^l_a}{\partial x^{l-1}_r}\frac{\partial x^{l-1}_r}{\partial x^0_i} = \sum_{r=1}^{n_{l-1}} W^l_{ar} J^{l-1}_{ri}. \tag{D.14}$$

Using (D.3) and (D.14), the elements of the layer Jacobian can alternatively be expressed as

$$J^l_{ai} = \frac{\partial x^l_a}{\partial z^l_a}\frac{\partial z^l_a}{\partial x^0_i} \tag{D.15}$$

$$= d^l_a M^l_{ai}. \tag{D.16}$$

Similarly, by (D.4) and (D.14),

$$\frac{\partial d^l_a}{\partial x^0_j} = \frac{\partial d^l_a}{\partial z^l_a}\frac{\partial z^l_a}{\partial x^0_j} \tag{D.17}$$

$$= d^l_{2,a} M^l_{aj}. \tag{D.18}$$

Finally, in view of (D.14), define the Hessian of the pre-activations z^l_a with respect to the network inputs by

$$K^l_{aij} := \frac{\partial M^l_{ai}}{\partial x^0_j} = \sum_{r=1}^{n_{l-1}} W^l_{ar} \frac{\partial J^{l-1}_{ri}}{\partial x^0_j} = \sum_{r=1}^{n_{l-1}} W^l_{ar} H^{l-1}_{rij}. \tag{D.19}$$

Recurrence. Differentiating (D.16) with respect to x^0_j yields

$$H^l_{aij} = \frac{\partial J^l_{ai}}{\partial x^0_j} \tag{D.20}$$

$$= \frac{\partial}{\partial x^0_j} \left(d^l_a M^l_{ai} \right) \tag{D.21}$$

$$= \frac{\partial d^l_a}{\partial x^0_j} M^l_{ai} + d^l_a \frac{\partial M^l_{ai}}{\partial x^0_j}. \tag{D.22}$$

Inserting (D.18)–(D.19) gives the recursion for the layer Hessian

$$\boxed{H^l_{aij} = d^l_{2,a} M^l_{ai} M^l_{aj} + d^l_a \left(\sum_{r=1}^{n_{l-1}} W^l_{ar} H^{l-1}_{rij} \right), \qquad l = 1, \dots, L.} \tag{D.23}$$

The recursion starts from $\boldsymbol{H}^0 \equiv \boldsymbol{0}$.

Network Hessian. The network Hessian $\boldsymbol{H}$ is obtained from (D.23) at the final layer L:

$$\boldsymbol{H} := \frac{\partial^2 \boldsymbol{x}^L}{\partial \boldsymbol{x}^0 \partial \boldsymbol{x}^0} = \boldsymbol{H}^L. \tag{D.24}$$

D.4. Third derivative

Layer third derivative. The layer third derivative tensor $\boldsymbol{T}^l \in \mathbb{R}^{n_l \times n_0 \times n_0 \times n_0}$ collects all third-order derivatives of the layer output $\boldsymbol{x}^l$ with respect to the network input $\boldsymbol{x}^0$:

$$\boldsymbol{T}^l := \frac{\partial^3 \boldsymbol{x}^l}{\partial \boldsymbol{x}^0 \partial \boldsymbol{x}^0 \partial \boldsymbol{x}^0}, \qquad T^l_{aijk} := \frac{\partial^3 x^l_a}{\partial x^0_i \partial x^0_j \partial x^0_k}. \tag{D.25}$$

This tensor is symmetric in the last three indices: $T^l_{aijk} = T^l_{ajik} = T^l_{aikj} = \cdots$.

Auxiliary quantities. Using the chain rule, (D.5) and (D.14),

$$\frac{\partial d^l_{2,a}}{\partial x^0_k} = \frac{\partial d^l_{2,a}}{\partial z^l_a} \frac{\partial z^l_a}{\partial x^0_k} \tag{D.26}$$

$$= d^l_{3,a} M^l_{ak}. \tag{D.27}$$

By means of (D.19), the third derivatives of the pre-activations z^l_a with respect to the inputs reads

$$Q^l_{aijk} := \frac{\partial K^l_{aij}}{\partial x^0_k} = \sum_{r=1}^{n_{l-1}} W^l_{ar} \frac{\partial H^{l-1}_{rij}}{\partial x^0_k} = \sum_{r=1}^{n_{l-1}} W^l_{ar} T^{l-1}_{rijk}. \tag{D.28}$$

Recurrence. Differentiating (D.23) with respect to x^0_k yields

$$T^l_{aijk} = \frac{\partial H^l_{aij}}{\partial x^0_k} \tag{D.29}$$

$$= \frac{\partial}{\partial x^0_k} \left(d^l_{2,a} M^l_{ai} M^l_{aj} + d^l_a K^l_{aij} \right) \tag{D.30}$$

$$= \frac{\partial d^l_{2,a}}{\partial x^0_k} M^l_{ai} M^l_{aj} + d^l_{2,a} \frac{\partial M^l_{ai}}{\partial x^0_k} M^l_{aj} + d^l_{2,a} M^l_{ai} \frac{\partial M^l_{aj}}{\partial x^0_k} + \frac{\partial d^l_a}{\partial x^0_k} K^l_{aij} + d^l_a \frac{\partial K^l_{aij}}{\partial x^0_k}. \tag{D.31}$$

Inserting (D.14), (D.18), (D.27), and (D.28) gives the recursion for the layer third derivative tensors

$$\boxed{\begin{aligned} T^l_{aijk} &= d^l_{3,a} M^l_{ak} M^l_{ai} M^l_{aj} \\ &\quad + d^l_{2,a} \left(M^l_{aj} K^l_{aik} + M^l_{ai} K^l_{ajk} + M^l_{ak} K^l_{aij} \right) + d^l_a \left(\textstyle\sum_{r=1}^{n_{l-1}} W^l_{ar} T^{l-1}_{rijk} \right), \qquad l = 1, \dots, L. \end{aligned}} \tag{D.32}$$

The recursion starts from $\boldsymbol{T}^0 \equiv \boldsymbol{0}$.

Network third derivative. The network third derivative tensor $\boldsymbol{T}$ is obtained from (D.32) at the final layer L:

$$\boldsymbol{T} := \frac{\partial^3 \boldsymbol{x}^L}{\partial \boldsymbol{x}^0 \partial \boldsymbol{x}^0 \partial \boldsymbol{x}^0} = \boldsymbol{T}^L. \tag{D.33}$$

Appendix E. Algorithmic stress update

Algorithm 1 provides an overview of the algorithmic update of the Cauchy stress $\boldsymbol{\sigma}$ for (nearly) incompressible materials. Only where necessary to avoid confusion, we indicate whether a quantity is evaluated at time $t^{(n+1)}$ or $t^{(n)}$. Unless explicitly indicated otherwise by a superscript, every quantity is evaluated at $t^{(n+1)}$.

Algorithm 1: Computation of the Cauchy stress tensor σ for (nearly) incompressible materials. Unless explicitly indicated otherwise by a superscript, every quantity is evaluated at $t^{(n+1)}$.

Given: $\mathbf{F}$, $\dot{\mathbf{F}}$, Δt, $\boldsymbol{f}$, $\mathcal{H}_{r\alpha}^{(n)} = \left(\mathbf{Q}_{r\alpha}^{(n)}, \bar{\mathbf{S}}_{r\alpha}^{(n)}, \tau_{r\alpha}^{(n)}\right)$

// **Generalized structural tensors**

for $r = 1$ **to** R **do**

  $\mathcal{N}_{w_i^{(r)}}(\boldsymbol{f})$ and $\mathcal{N}_{\boldsymbol{m}_i}(\boldsymbol{f})$: $\boldsymbol{m}_i$, $w_i^{(r)}$ ▷ Preferred material directions and weights (5)

  $\tilde{\mathbf{L}}_r = \sum_{j=0}^{n} w_i^{(r)} \mathbf{L}_i$ ▷ Generalized structural tensors (5)

end

J, $\mathbf{C}$, $\mathbf{C}^{-1}$, $\bar{\mathbf{C}}$, $\bar{\mathbf{C}}^{-1}$, $\dot{\bar{\mathbf{C}}}$

for $r = 1$ **to** R **do**

  // **Equilibrium stress**

  $\bar{I}_r = \bar{\tilde{I}}_r \cup \bar{\dot{\tilde{I}}}_r$ ▷ Generalized invariants (6), (7)

  $\mathcal{N}_{\Psi_{\mathrm{iso}\,r}}(\bar{\tilde{I}}_r, \boldsymbol{f})$: $\frac{\partial \Psi_{\mathrm{iso}\,r}^{\mathcal{N}}}{\partial \bar{\tilde{I}}_r}$, $\frac{\partial \Psi_{\mathrm{iso}\,r}^{\mathcal{N}}}{\partial \bar{\tilde{J}}_r}$ ▷ Network Jacobians (D.11)

  $\frac{\partial \Psi_{\mathrm{iso}\,r}^{\infty}}{\partial \bar{\tilde{I}}_r} = \frac{\partial \Psi_{\mathrm{iso}\,r}^{\mathcal{N}}}{\partial \bar{\tilde{I}}_r} + \bar{\alpha}_r$, $\frac{\partial \Psi_{\mathrm{iso}\,r}^{\infty}}{\partial \bar{\tilde{J}}_r} = \frac{\partial \Psi_{\mathrm{iso}\,r}^{\mathcal{N}}}{\partial \bar{\tilde{J}}_r} + \bar{\beta}_r$ ▷ Stress normalization (C.10)

  $\bar{\mathbf{S}}_{\mathrm{iso}\,r}^{\infty} = 2\left[\frac{\partial \Psi_{\mathrm{iso}\,r}^{\infty}}{\partial \bar{\tilde{I}}_r}\tilde{\mathbf{L}}_r - \frac{\partial \Psi_{\mathrm{iso}\,r}^{\infty}}{\partial \bar{\tilde{J}}_r}\bar{\mathbf{H}}_r\right]$ ▷ Fictitious equilibrium stress (23)

  // **Non-equilibrium stress**

  for $\alpha = 1$ **to** N **do**

    $\mathcal{N}_{\tau_{r\alpha}}(\bar{I}_r, \boldsymbol{f})$: $\tau_{r\alpha}^{(n+1)}$ ▷ Relaxation times

    $\mathcal{N}_{\Psi_{r\alpha}}(\bar{\tilde{I}}_r, \boldsymbol{f})$: $\frac{\partial \Psi_{r\alpha}^{\mathcal{N}}}{\partial \bar{\tilde{I}}_r}$, $\frac{\partial \Psi_{r\alpha}^{\mathcal{N}}}{\partial \bar{\tilde{J}}_r}$, $\frac{\partial^2 \Psi_{r\alpha}^{\mathcal{N}}}{\partial \bar{\tilde{I}}_r^2}$, $\frac{\partial^2 \Psi_{r\alpha}^{\mathcal{N}}}{\partial \bar{\tilde{J}}_r^2}$, $\frac{\partial^2 \Psi_{r\alpha}^{\mathcal{N}}}{\partial \bar{\tilde{I}}_r \partial \bar{\tilde{J}}_r}$ ▷ Network Jacobians and Hessians (D.11), (D.23)

    $\frac{\partial \Psi_{r\alpha}}{\partial \bar{\tilde{I}}_r}$, $\frac{\partial \Psi_{r\alpha}}{\partial \bar{\tilde{J}}_r}$, $\frac{\partial^2 \Psi_{r\alpha}}{\partial \bar{\tilde{I}}_r^2}$, $\frac{\partial^2 \Psi_{r\alpha}}{\partial \bar{\tilde{J}}_r^2}$, $\frac{\partial^2 \Psi_{r\alpha}}{\partial \bar{\tilde{I}}_r \partial \bar{\tilde{J}}_r}$ ▷ Stress normalization (C.10)

    $\bar{\mathbf{S}}_{r\alpha}^{(n+1)} = 2\left[\frac{\partial \Psi_{r\alpha}}{\partial \bar{\tilde{I}}_r}\tilde{\mathbf{L}}_r - \frac{\partial \Psi_{r\alpha}}{\partial \bar{\tilde{J}}_r}\mathbf{H}_r\right]$ ▷ Fictitious stress (27)

    $\mathbf{Q}_{r\alpha}^{(n+1)} = \exp(\xi_{r\alpha})\bar{\mathbf{S}}_{r\alpha}^{(n+1)} + \exp(\xi_{r\alpha})\left[\exp(\xi_{r\alpha})\mathbf{Q}_{r\alpha}^{(n)} - \bar{\mathbf{S}}_{r\alpha}^{(n)}\right]$ ▷ Update internal variable (36)

    $\bar{\mathbf{S}}_{r\alpha}^{\mathrm{neq}} = \frac{2}{\mu_{r\alpha}}\left[\frac{\partial^2 \Psi_{r\alpha}}{\partial \bar{\tilde{I}}_r^2}(\tilde{\mathbf{L}}_r : \mathbf{Q}_{r\alpha}^{(n+1)})\tilde{\mathbf{L}}_r + \frac{\partial^2 \Psi_{r\alpha}}{\partial \bar{\tilde{J}}_r^2}\left(\bar{\mathbf{H}}_r : \mathbf{Q}_{r\alpha}^{(n+1)}\right)\bar{\mathbf{H}}_r + \frac{\partial \Psi_{r\alpha}}{\partial \bar{\tilde{J}}_r}\left(\bar{\mathbf{C}}^{-1}\mathbf{Q}_{r\alpha}^{(n+1)}\bar{\mathbf{H}}_r + \bar{\mathbf{H}}_r\mathbf{Q}_{r\alpha}^{(n+1)}\bar{\mathbf{C}}^{-1}\right)\right.$

    $\left. - \frac{\partial^2 \Psi_{r\alpha}}{\partial \bar{\tilde{I}}_r \partial \bar{\tilde{J}}_r}\left(\left(\bar{\mathbf{H}}_r : \mathbf{Q}_{r\alpha}^{(n+1)}\right)\tilde{\mathbf{L}}_r + (\tilde{\mathbf{L}}_r : \mathbf{Q}_{r\alpha}^{(n+1)})\bar{\mathbf{H}}_r\right)\right]$ ▷ Fictitious non-equilibrium stress, (31)

  end

end

$\mathbf{S}_{\mathrm{vol}}^{\infty} = p\mathbf{C}^{-1}$ (if nearly incompressible $p = J\frac{\partial \Psi_{\mathrm{vol}}^{\infty}}{\partial J}$) ▷ Volumetric equilibrium stress (21)

$\mathbf{S}_{\mathrm{iso}\,r}^{\infty} = J^{-2/3}\,\mathbb{P} : \bar{\mathbf{S}}_{\mathrm{iso}\,r}^{\infty}$ ▷ Isochoric equilibrium stress (22)

$\mathbf{S}_{r\alpha}^{\mathrm{neq}} = J^{-2/3}\,\mathbb{P} : \bar{\mathbf{S}}_{r\alpha}^{\mathrm{neq}}$ ▷ Isochoric non-equilibrium stress (29)

$\mathbf{S} = \mathbf{S}_{\mathrm{vol}}^{\infty} + \sum_{r=1}^{R}\left(\mathbf{S}_{\mathrm{iso}\,r}^{\infty} + \sum_{\alpha=1}^{N}\mathbf{S}_{r\alpha}^{\mathrm{neq}}\right)$. ▷ Total stress (17)

$\sigma = J^{-1}\mathbf{F}\mathbf{S}\mathbf{F}^{\mathrm{T}}$ ▷ Cauchy stress

return σ, $\mathcal{H}_{r\alpha}^{(n+1)} = \left(\mathbf{Q}_{r\alpha}^{(n+1)}, \bar{\mathbf{S}}_{r\alpha}^{(n+1)}, \tau_{r\alpha}^{(n+1)}\right)$

Appendix F. Algorithmic tangent modulus

In this section, we derive the algorithmically consistent Lagrangian tangent modulus for (nearly) incompressible materials.

F.1. Algorithmic tangent modulus

Only where necessary to avoid confusion, we indicate whether a quantity is evaluated at time $t^{(n+1)}$ or $t^{(n)}$. Unless explicitly stated otherwise, each quantity is evaluated at $t^{(n+1)}$.

Based on the additive decomposition of the stress (17), the total Lagrangian tangent modulus $\mathbb{C}$ is:

$$\mathbb{C} = 2\frac{\partial \mathbf{S}}{\partial \mathbf{C}} = \mathbb{C}^{\infty}_{\mathrm{vol}} + \sum_{r=1}^{R}\left(\mathbb{C}^{\infty}_{\mathrm{iso}\,r} + \sum_{\alpha=1}^{N}\mathbb{C}^{\mathrm{neq}}_{r\alpha}\right), \tag{F.1}$$

where

$$\mathbb{C}^{\infty}_{\mathrm{vol}} = 2\frac{\partial \mathbf{S}^{\infty}_{\mathrm{vol}}}{\partial \mathbf{C}}, \qquad \mathbb{C}^{\infty}_{\mathrm{iso}\,r} = 2\frac{\partial \mathbf{S}^{\infty}_{\mathrm{iso}\,r}}{\partial \mathbf{C}}, \qquad \mathbb{C}^{\mathrm{neq}}_{r\alpha} = 2\frac{\partial \mathbf{S}^{\mathrm{neq}}_{r\alpha}}{\partial \mathbf{C}}. \tag{F.2}$$

In the following, we compute each tangent modulus individually.

1. Volumetric equilibrium tangent modulus $\mathbb{C}^{\infty}_{\mathrm{vol}}$. It can be shown, cf. [74, p. 254], that

$$\mathbb{C}^{\infty}_{\mathrm{vol}} = \left[J^2\frac{\partial^2 \Psi^{\infty}_{\mathrm{vol}}}{\partial J^2} + J\frac{\partial \Psi^{\infty}_{\mathrm{vol}}}{\partial J}\right]\mathbf{C}^{-1}\odot\mathbf{C}^{-1} - 2J\frac{\partial \Psi^{\infty}_{\mathrm{vol}}}{\partial J}\,\mathbf{C}^{-1}\otimes\mathbf{C}^{-1}. \tag{F.3}$$

The user only needs to provide the volumetric equilibrium tangent modulus $\mathbb{C}^{\infty}_{\mathrm{vol}}$ for nearly incompressible materials. For perfectly incompressible materials, Abaqus computes the volumetric tangent modulus internally, consistent with the Lagrange-multiplier method used to enforce incompressibility [10].

2. Isochoric equilibrium tangent moduli $\mathbb{C}^{\infty}_{\mathrm{iso}\,r}$. It can be shown, cf. [74, p. 255], that

$$\begin{aligned}\mathbb{C}^{\infty}_{\mathrm{iso}\,r} &= 2\frac{\partial \mathbf{S}^{\infty}_{\mathrm{iso}\,r}}{\partial \mathbf{C}} = 2\frac{\partial}{\partial \mathbf{C}}\left(J^{-2/3}\,\mathbb{P} : \bar{\mathbf{S}}^{\infty}_{\mathrm{iso}\,r}\right)\\ &= J^{-4/3}\,\mathbb{P} : \bar{\mathbb{C}}^{\infty}_{\mathrm{iso}\,r} : \mathbb{P}^{\mathrm{T}} - \frac{2}{3}\left(\mathbf{S}^{\infty}_{\mathrm{iso}\,r}\odot\mathbf{C}^{-1} + \mathbf{C}^{-1}\odot\mathbf{S}^{\infty}_{\mathrm{iso}\,r}\right) + \frac{2}{3}J^{-2/3}\left(\bar{\mathbf{S}}^{\infty}_{\mathrm{iso}\,r} : \mathbf{C}\right)\left[\left(\mathbf{C}^{-1}\otimes\mathbf{C}^{-1}\right)^{\mathrm{s}} - \frac{1}{3}\mathbf{C}^{-1}\odot\mathbf{C}^{-1}\right]\end{aligned} \tag{F.4}$$

where the fictitious isochoric equilibrium tangent modulus $\bar{\mathbb{C}}^{\infty}_{\mathrm{iso}\,r}$ can be expressed as

$$\begin{aligned}\bar{\mathbb{C}}^{\infty}_{\mathrm{iso}\,r} &= 2\frac{\partial \bar{\mathbf{S}}^{\infty}_{\mathrm{iso}\,r}}{\partial \bar{\mathbf{C}}}\\ &= 4\left[\frac{\partial^2 \Psi^{\infty}_{\mathrm{iso}\,r}}{\partial \bar{\bar{I}}_r^2}\tilde{\mathbf{L}}_r\odot\tilde{\mathbf{L}}_r + \frac{\partial^2 \Psi^{\infty}_{\mathrm{iso}\,r}}{\partial \bar{\bar{J}}_r^2}\bar{\mathbf{H}}_r\odot\bar{\mathbf{H}}_r + \frac{\partial \Psi^{\infty}_{\mathrm{iso}\,r}}{\partial \bar{\bar{J}}_r}\left(\bar{\mathbf{C}}^{-1}\otimes\bar{\mathbf{H}}_r + \bar{\mathbf{H}}_r\otimes\bar{\mathbf{C}}^{-1}\right)^{\mathrm{s}} - \frac{\partial^2 \Psi^{\infty}_{\mathrm{iso}\,r}}{\partial \bar{\bar{I}}_r\partial \bar{\bar{J}}_r}\left(\tilde{\mathbf{L}}_r\odot\bar{\mathbf{H}}_r + \bar{\mathbf{H}}_r\odot\tilde{\mathbf{L}}_r\right)\right].\end{aligned} \tag{F.5}$$

3. Non-equilibrium tangent moduli $\mathbb{C}^{\mathrm{neq}}_{r\alpha}$. In analogy to (F.4), the non-equilibrium tangent moduli are given by

$$\mathbb{C}^{\mathrm{neq}}_{r\alpha} = J^{-4/3}\,\mathbb{P} : \bar{\mathbb{C}}^{\mathrm{neq}}_{r\alpha} : \mathbb{P}^{\mathrm{T}} - \frac{2}{3}\left(\mathbf{S}^{\mathrm{neq}}_{r\alpha}\odot\mathbf{C}^{-1} + \mathbf{C}^{-1}\odot\mathbf{S}^{\mathrm{neq}}_{r\alpha}\right) + \frac{2}{3}J^{-2/3}\left(\bar{\mathbf{S}}^{\mathrm{neq}}_{r\alpha} : \mathbf{C}\right)\left[\left(\mathbf{C}^{-1}\otimes\mathbf{C}^{-1}\right)^{\mathrm{s}} - \frac{1}{3}\mathbf{C}^{-1}\odot\mathbf{C}^{-1}\right]. \tag{F.6}$$

Consequently, it remains to compute the fictitious tangent moduli $\bar{\mathbb{C}}^{\mathrm{neq}}_{r\alpha}$. Differentiating (31) with respect to $\bar{\mathbf{C}}$ yields:

$$\begin{aligned}\bar{\mathbb{C}}^{\mathrm{neq}}_{r\alpha} &= 2\frac{\partial \bar{\mathbf{S}}^{\mathrm{neq}}_{r\alpha}}{\partial \bar{\mathbf{C}}} = \frac{1}{\mu_{r\alpha}}\frac{\partial}{\partial \bar{\mathbf{C}}}\left(\bar{\mathbb{C}}_{r\alpha} : \mathbf{Q}_{r\alpha}\right)\\ &= \frac{4}{\mu_{r\alpha}}\left\{\frac{\partial^3 \Psi_{r\alpha}}{\partial \bar{\bar{I}}_r^3}\varepsilon_{r\alpha}\tilde{\mathbf{L}}_r\odot\tilde{\mathbf{L}}_r - \frac{\partial^3 \Psi_{r\alpha}}{\partial \bar{\bar{J}}_r^3}\gamma_{r\alpha}\bar{\mathbf{H}}_r\odot\bar{\mathbf{H}}_r - \frac{\partial^3 \Psi_{r\alpha}}{\partial \bar{\bar{I}}_r^2\partial \bar{\bar{J}}_r}\varepsilon_{r\alpha}\tilde{\mathbf{L}}_r\odot\bar{\mathbf{H}}_r + \frac{\partial^3 \Psi_{r\alpha}}{\partial \bar{\bar{I}}_r\partial \bar{\bar{J}}_r^2}\gamma_{r\alpha}\bar{\mathbf{H}}_r\odot\tilde{\mathbf{L}}_r\right.\\ &\quad - \frac{\partial^2 \Psi_{r\alpha}}{\partial \bar{\bar{J}}_r^2}\left[\gamma_{r\alpha}\left(\bar{\mathbf{C}}^{-1}\otimes\bar{\mathbf{H}}_r + \bar{\mathbf{H}}_r\otimes\bar{\mathbf{C}}^{-1}\right)^{\mathrm{s}} + \bar{\mathbf{H}}_r\odot\left(\bar{\mathbf{C}}^{-1}\mathbf{Q}_{r\alpha}\bar{\mathbf{H}}_r + \bar{\mathbf{H}}_r\mathbf{Q}_{r\alpha}\bar{\mathbf{C}}^{-1}\right)\right]\\ &\quad - \frac{\partial^2 \Psi_{r\alpha}}{\partial \bar{\bar{J}}_r^2}\left(\bar{\mathbf{C}}^{-1}\mathbf{Q}_{r\alpha}\bar{\mathbf{H}}_r + \bar{\mathbf{H}}_r\mathbf{Q}_{r\alpha}\bar{\mathbf{C}}^{-1}\right)\odot\bar{\mathbf{H}}_r + \frac{\partial^2 \Psi_{r\alpha}}{\partial \bar{\bar{I}}_r\partial \bar{\bar{J}}_r}\left(\bar{\mathbf{C}}^{-1}\mathbf{Q}_{r\alpha}\bar{\mathbf{H}}_r + \bar{\mathbf{H}}_r\mathbf{Q}_{r\alpha}\bar{\mathbf{C}}^{-1}\right)\odot\tilde{\mathbf{L}}_r\\ &\quad - \frac{\partial \Psi_{r\alpha}}{\partial \bar{\bar{J}}_r}\left[\left(\bar{\mathbf{C}}^{-1}\otimes\bar{\mathbf{C}}^{-1}\right)^{\mathrm{s}}\mathbf{Q}_{r\alpha}\bar{\mathbf{H}}_r + \bar{\mathbf{H}}_r\mathbf{Q}_{r\alpha}\left(\bar{\mathbf{C}}^{-1}\otimes\bar{\mathbf{C}}^{-1}\right)^{\mathrm{s}}\right.\\ &\quad \left.\left. + \left(\bar{\mathbf{C}}^{-1}\otimes\bar{\mathbf{H}}_r + \bar{\mathbf{H}}_r\otimes\bar{\mathbf{C}}^{-1}\right)^{\mathrm{s}}\mathbf{Q}_{r\alpha}\bar{\mathbf{C}}^{-1} + \bar{\mathbf{C}}^{-1}\mathbf{Q}_{r\alpha}\left(\bar{\mathbf{C}}^{-1}\otimes\bar{\mathbf{H}}_r + \bar{\mathbf{H}}_r\otimes\bar{\mathbf{C}}^{-1}\right)^{\mathrm{s}}\right]\right.\\ &\quad - \frac{\partial^3 \Psi_{r\alpha}}{\partial \bar{\bar{I}}_r^2\partial \bar{\bar{J}}_r}\left[\gamma_{r\alpha}\tilde{\mathbf{L}}_r + \varepsilon_{r\alpha}\bar{\mathbf{H}}_r\right]\odot\tilde{\mathbf{L}}_r + \frac{\partial^3 \Psi_{r\alpha}}{\partial \bar{\bar{I}}_r\partial \bar{\bar{J}}_r^2}\left[\gamma_{r\alpha}\tilde{\mathbf{L}}_r + \varepsilon_{r\alpha}\bar{\mathbf{H}}_r\right]\odot\bar{\mathbf{H}}_r\\ &\quad \left. + \frac{\partial^2 \Psi_{r\alpha}}{\partial \bar{\bar{I}}_r\partial \bar{\bar{J}}_r}\left[\tilde{\mathbf{L}}_r\odot\left(\bar{\mathbf{C}}^{-1}\mathbf{Q}_{r\alpha}\bar{\mathbf{H}}_r + \bar{\mathbf{H}}_r\mathbf{Q}_{r\alpha}\bar{\mathbf{C}}^{-1}\right) + \varepsilon_{r\alpha}\left(\bar{\mathbf{C}}^{-1}\otimes\bar{\mathbf{H}}_r + \mathbf{H}_r\otimes\bar{\mathbf{C}}^{-1}\right)^{\mathrm{s}}\right]\right\}\\ &\quad + \frac{1}{\mu_{r\alpha}}\bar{\mathbb{C}}_{r\alpha} : \frac{\partial \mathbf{Q}_{r\alpha}}{\partial \bar{\mathbf{C}}}.\end{aligned} \tag{F.7}$$

Note that $\bar{\mathbb{C}}_{r\alpha}$ in the last term of (F.7) can be computed from (F.5) by replacing $\Psi^{\infty}_{\mathrm{iso}\,r}$ with $\Psi_{r\alpha}$. The derivative $\partial \mathbf{Q}_{r\alpha}/\partial \bar{\mathbf{C}}$ can be obtained by differentiating the update formula (36):

$$\frac{\partial \mathbf{Q}_{r\alpha}}{\partial \bar{\mathbf{C}}} = \frac{1}{2}\exp\left(\xi_{r\alpha}\right)\left[\bar{\mathbb{C}}_{r\alpha} - \frac{\xi_{r\alpha}}{\bar{\tau}_{r\alpha}}\left(\bar{\mathbf{S}}^{(n+1)}_{r\alpha} + 2\exp\left(\xi_{r\alpha}\right)\mathbf{Q}^{(n)}_{r\alpha} - \bar{\mathbf{S}}^{(n)}_{r\alpha}\right) \odot \frac{\partial \tau_{r\alpha}}{\partial \bar{\mathbf{C}}}\right], \tag{F.8}$$

where

$$\frac{\partial \tau_{r\alpha}}{\partial \bar{\mathbf{C}}} = \frac{\partial \tau_{r\alpha}}{\partial \bar{\bar{I}}_r}\tilde{\mathbf{L}}_r - \frac{\partial \tau_{r\alpha}}{\partial \bar{\bar{J}}_r}\bar{\mathbf{H}}_r. \tag{F.9}$$

F.2. Algorithmic tangent modulus in Abaqus

Implementing a material subroutine in Abaqus requires computing the tangent modulus

$$\mathbb{c}^{\mathrm{Abaqus}} = \frac{1}{J}\mathbb{c}^{\tau JZ}. \tag{F.10}$$

The tangent modulus $\mathbb{c}^{\tau JZ}$ is associated with the Jaumann-Zaremba rate of the Kirchhoff stress $\boldsymbol{\tau}^{\nabla JZ}$,

$$\boldsymbol{\tau}^{\nabla JZ} = \dot{\boldsymbol{\tau}} - \mathbf{w}\boldsymbol{\tau} - \boldsymbol{\tau}\mathbf{w}^{\mathrm{T}} = \mathbb{c}^{\tau JZ} : \mathbf{d}, \tag{F.11}$$

where $\mathbf{d}$ and $\mathbf{w}$ are the symmetric and skew-symmetric parts of the spatial velocity gradient $\mathbf{l}$, i.e., $\mathbf{l} = \mathbf{d} + \mathbf{w}$. To express $\mathbb{c}^{\nabla JZ}$ in terms of $\mathbb{C}$, we first recall the spatial tangent modulus $\mathbb{c}$ relating the Lie time derivative of the Kirchhoff stress tensor $\mathcal{L}_{\mathrm{v}}(\boldsymbol{\tau})$ and the rate of deformation tensor $\mathbf{d}$ by

$$\mathcal{L}_{\mathrm{v}}(\boldsymbol{\tau}) = \dot{\boldsymbol{\tau}} - \mathbf{l}\boldsymbol{\tau} - \boldsymbol{\tau}\mathbf{l}^{\mathrm{T}} = \mathbb{c} : \mathbf{d}. \tag{F.12}$$

By pushing forward the material tangent modulus (F.1) to the current configuration, we obtain the spatial tangent modulus

$$\mathbb{c} = (\mathbf{F} \otimes \mathbf{F}^{\mathrm{T}}) : \mathbb{C} : (\mathbf{F}^{\mathrm{T}} \otimes \mathbf{F}). \tag{F.13}$$

Subtracting (F.12) from (F.11), yields

$$\begin{aligned}\mathbb{c}^{\tau JZ} : \mathbf{d} &= \mathbb{c} : \mathbf{d} + (\mathbf{l} - \mathbf{w})\boldsymbol{\tau} + \boldsymbol{\tau}(\mathbf{l} - \mathbf{w})^{\mathrm{T}} \\ &= \mathbb{c} : \mathbf{d} + \mathbf{d}\boldsymbol{\tau} + \boldsymbol{\tau}\mathbf{d}^{\mathrm{T}} \\ &= \left[\mathbb{c} + \mathbf{I} \otimes \boldsymbol{\tau} + (\boldsymbol{\tau} \otimes \mathbf{I})^{t}\right] : \mathbf{d}.\end{aligned} \tag{F.14}$$

Since (F.14) has to hold for any $\mathbf{d}$, we finally obtain in view of (F.10) and $\boldsymbol{\tau} = J\boldsymbol{\sigma}$,

$$\mathbb{c}^{\mathrm{Abaqus}} = \frac{1}{J}\mathbb{c} + \mathbf{I} \otimes \boldsymbol{\sigma} + (\boldsymbol{\sigma} \otimes \mathbf{I})^{t}. \tag{F.15}$$

The implementation of the algorithmically consistent tangent modulus was verified using the automatic differentiation capabilities of the TensorFlow-based vCANN implementation and, additionally, via a numerical approximation described below in Appendix F.3, which is also optionally available in the material subroutine.

F.3. Numerical approximation of the algorithmic tangent modulus

We have implemented a numerical approximation technique to compute the algorithmic tangent modulus outlined in Appendix F.1. Introduced by Miehe [113], this approach uses a forward difference approximation to replace the computation of the tangent modulus with N additional stress computations, where $N = 4$ for two-dimensional problems and $N = 6$ for three-dimensional problems. For each additional stress calculation, a particular perturbed deformation gradient $\hat{\mathbf{F}}_{(kl)}$, detailed in Algorithm 2, is used. Miehe's numerical approximation refers specifically to the tangent modulus $\mathbb{c}$, which is associated with the Lie time derivative of the Kirchhoff stress. Building on this, Sun [114] subsequently derived a numerical approximation for the tangent modulus $\mathbb{c}^{\tau JZ}$, associated with the Jaumann–Zaremba rate of the Kirchhoff stress, which is used in Abaqus,

$$\mathbb{c}^{\tau JZ}_{ijkl} \approx \frac{1}{J\varepsilon}\left[\hat{J}\hat{\boldsymbol{\sigma}}^{ij}(\hat{\mathbf{F}}_{(kl)}) - J\boldsymbol{\sigma}^{ij}(\mathbf{F})\right], \tag{F.16}$$

where $\hat{J} = \det(\hat{\mathbf{F}}_{(kl)})$ and ε is a perturbation parameter.

Algorithm 2: Numerical approximation of the tangent modulus $\mathbb{c}^{\tau JZ}$ in Voigt notation. Unless explicitly indicated otherwise by a superscript, every quantity is evaluated at $t^{(n+1)}$.

Input: $\mathbf{F}$, $\dot{\mathbf{F}}$, Δt, J, σ, ε, $\boldsymbol{f}$, $\mathcal{H}_{r\alpha}^{(n)} = \left(\mathbf{Q}_{r\alpha}^{(n)}, \bar{\mathbf{S}}_{r\alpha}^{(n)}, \tau_{r\alpha}^{(n)}\right)$
Output: $\mathbb{c}^{\tau JZ}$
Data: VoigtIndex = [(11), (22), (33), (12), (13), (23)], Eulerian basis $\mathcal{E} = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$

for $B = 1$ **to** 6 **do**
  $(kl) \leftarrow$ VoigtIndex(B)
  $\mathbf{e}_k, \mathbf{e}_l \leftarrow \mathcal{E}(k), \mathcal{E}(l)$
  $\Delta\mathbf{F}_{(kl)} = \frac{\varepsilon}{2}(\mathbf{e}_k \otimes \mathbf{e}_l\mathbf{F} + \mathbf{e}_l \otimes \mathbf{e}_k\mathbf{F})$ ▷ perturbation
  $\hat{\mathbf{F}}_{(kl)} = \mathbf{F} + \Delta\mathbf{F}_{(kl)}$ ▷ perturbed deformation gradient
  $\hat{J} = \det\left(\hat{\mathbf{F}}_{(kl)}\right)$ ▷ perturbed Jacobian determiant
  $\hat{\sigma}\left(\hat{\mathbf{F}}_{(kl)}, \dot{\mathbf{F}}, \Delta t, \boldsymbol{f}, \mathcal{H}_{r\alpha}^{(n)}\right)$ ▷ perturbed Cauchy stress, Algorithm 1
  for $A = 1$ **to** 6 **do**
    $(ij) \leftarrow$ VoigtIndex(A)
    $\mathbb{c}_{A,B}^{\tau JZ} \leftarrow \frac{1}{J\varepsilon}\left[\hat{J}\hat{\sigma}^{ij} - J\sigma^{ij}\right]$
  end
end

Appendix G. Element types tested in Cook's membrane benchmark test

Table G.1 lists all elements tested in Cook's membrane benchmark test in Section 5.2.

Table G.1
Elements tested in Cook's membrane benchmark in Section 5.2 [100]. The letters 'I', 'R', and 'M' in the element names refer to incompatible mode elements, reduced integration, and modified elements, respectively.

Element	Description
3D	
C3D4H	4-node linear tetrahedron, hybrid with linear pressure
C3D8H	8-node linear brick, hybrid with constant pressure
C3D8IH	8-node linear brick, incompatible modes, hybrid with linear pressure
C3D8RH	8-node linear brick, reduced integration, hourglass control, hybrid with constant pressure
C3D10H	10-node quadratic tetrahedron, hybrid with constant pressure
C3D10MH	10-node modified tetrahedron,, hourglass control, hybrid with linear pressure
C3D20H	20-node quadratic brick, hybrid with linear pressure
C3D20H	20-node quadratic brick, reduced integration, hybrid with linear pressure
Plane strain	
CPE3H	3-node linear, hybrid with constant pressure
CPE4H	4-node bilinear, hybrid with constant pressure
CPE4IH	4-node bilinear, incompatible modes, hybrid with linear pressure
CPE4RH	4-node bilinear, reduced integration, hourglass control, hybrid with constant pressure
CPE6H	6-node quadratic, hybrid with linear pressure
CPE8H	8-node biquadratic, hybrid with linear pressure
CPE8RH	8-node biquadratic, reduced integration, hybrid with linear pressure
Plane stress	
CPS3	3-node linear
CPS4	4-node bilinear
CPS4I	4-node bilinear, incompatible modes
CPS4R	4-node bilinear, reduced integration, hourglass control
CPS6	6-node quadratic
CPS6M	6-node modified,, hourglass control
CPS8	8-node biquadratic
CPS8R	8-node biquadratic, reduced integration
Shell	
S3	3-node triangular general-purpose shell, finite membrane strains
S4	4-node general-purpose shell, finite membrane strains
S4R	4-node general-purpose shell, reduced integration, hourglass control, finite membrane strains

Appendix H. Ecoflex silicone polymer: Shore hardness as auxiliary feature

As an additional example of the benefit of auxiliary features in constitutive modeling, we consider Ecoflex, a commercially available silicone elastomer with widespread applications in biomedical engineering, soft robotics, and wearable devices, owing to its biocompatibility, thermal stability, and large stretchability. Ecoflex comes in multiple Shore hardness grades, which directly

Table H.1
Leave-one-out cross-validation R^2 values across four Shore hardness levels, corresponding to the fits and predictions shown in Fig. H.1. Each row corresponds to one fold, where the withheld (validation) Shore hardness is indicated by the fold label. Orange cells denote the validation score; blue cells are training scores.

	R^2			
Data fold	Shore 10	Shore 20	Shore 30	Shore 50
Shore 10 withheld	0.9781	0.9973	0.9998	0.9999
Shore 20 withheld	0.9991	0.9891	0.9999	0.9999
Shore 30 withheld	0.9864	0.9990	0.9755	0.9997
Shore 50 withheld	0.9958	0.9998	0.9998	0.9548

Training Validation

Table I.1
Material parameters for the orthotropic and isotropic ground truth models.

Material parameter	Orthotropy	Isotropy (Appendix I)
Material symmetry parameters		
Weight factor $\omega_1^{(1)}$ [–]	0.23	0.00
Fiber angle φ [°]	26.25	–
Hyperelastic parameters		
Stiffness μ_1 [MPa]	1.00	1.00
First shape parameter γ_1 [–]	1.45	1.45
Second shape parameter δ_1 [–]	0.01	0.01
Viscous parameters		
Relaxation time τ_1 [s]	1.00	1.00
Relaxation time τ_2 [s]	15.98	15.98
Relaxation time τ_3 [s]	50.01	50.01
Strain energy factor β_1 [–]	0.374	1.241
Strain energy factor β_2 [–]	0.332	1.145
Strain energy factor β_3 [–]	0.332	0.374

correlate with its stiffness. The dataset of [115] provides extensive characterization of Ecoflex in uniaxial tensile tests, including loading–unloading experiments performed at varying Shore hardness levels.

We analyzed the virgin loading–unloading curves, using the Shore hardness as an additional input to the vCANN. Shore hardness values of $s \in \{10, 20, 30, 50\}$ and a maximum uniaxial stretch of $\lambda_{\max} = 6$ were considered. Given the limited size of the dataset, a leave-one-out cross-validation (LOOCV) strategy was adopted, in which one Shore hardness level is excluded from training and reserved for validation. This experimental design enables an assessment of the model's ability to interpolate and extrapolate material behavior across Shore hardness levels not seen during training.

To account for the pronounced dependence of the stress response on the Shore hardness s, we incorporated s into the vCANN input space via the feature vector $\boldsymbol{f} = s$. Owing to isotropy, the associated generalized structural tensor (44) reduces to $\tilde{\mathbf{L}}_1 = \frac{1}{3}\mathbf{I}$. Accordingly, the following free energy functions and relaxation times are learned from the loading–unloading tests:

$$\Psi^{\infty}_{\text{iso}\,1} = \Psi^{\infty}_{\text{iso}\,1}(\bar{\tilde{I}}_1, s), \qquad \Psi_{1\alpha} = \Psi_{1\alpha}(\bar{\tilde{I}}_1, s), \qquad \tau_{1\alpha} = \tau_{1\alpha}(\bar{\tilde{I}}_1, s), \qquad \alpha = 1, \dots, N^{\max}, \tag{H.1}$$

where

$$\bar{\tilde{I}}_1 = \left(\bar{\tilde{I}}_1, \bar{\tilde{J}}_1\right), \qquad \bar{\tilde{I}}_1 = \frac{1}{3}\,\mathrm{tr}[\bar{\mathbf{C}}], \qquad \bar{\tilde{J}}_1 = \frac{1}{3}\,\mathrm{tr}[\mathrm{cof}\,\bar{\mathbf{C}}]. \tag{H.2}$$

Fig. H.1 shows the descriptive and predictive performance of the vCANNs—implemented both in FEM and TensorFlow—on the training and validation dataset, respectively, for each of the four data folds. The match with the training data is excellent. Additionally, the vCANNs accurately predict the material response for the Shore hardness levels withheld in the respective training processes. Notably, in extrapolating the stress response for the withheld Shore hardness level of 10, precise predictions are achieved. As a quantitative measure of the goodness-of-fit, we report the R^2 values of each data fold in Table H.1. In addition to the temperature-dependent VHB 4905 dataset in Section 5.1, this example further demonstrates the ability to incorporate arbitrary auxiliary features into the neural constitutive law.

Appendix I. Isotropic artery model

Here, we analyze the same boundary problem as in Section 5.3 using an isotropic ground truth (GT) model (41) with material parameters given in second column of Table I.1.

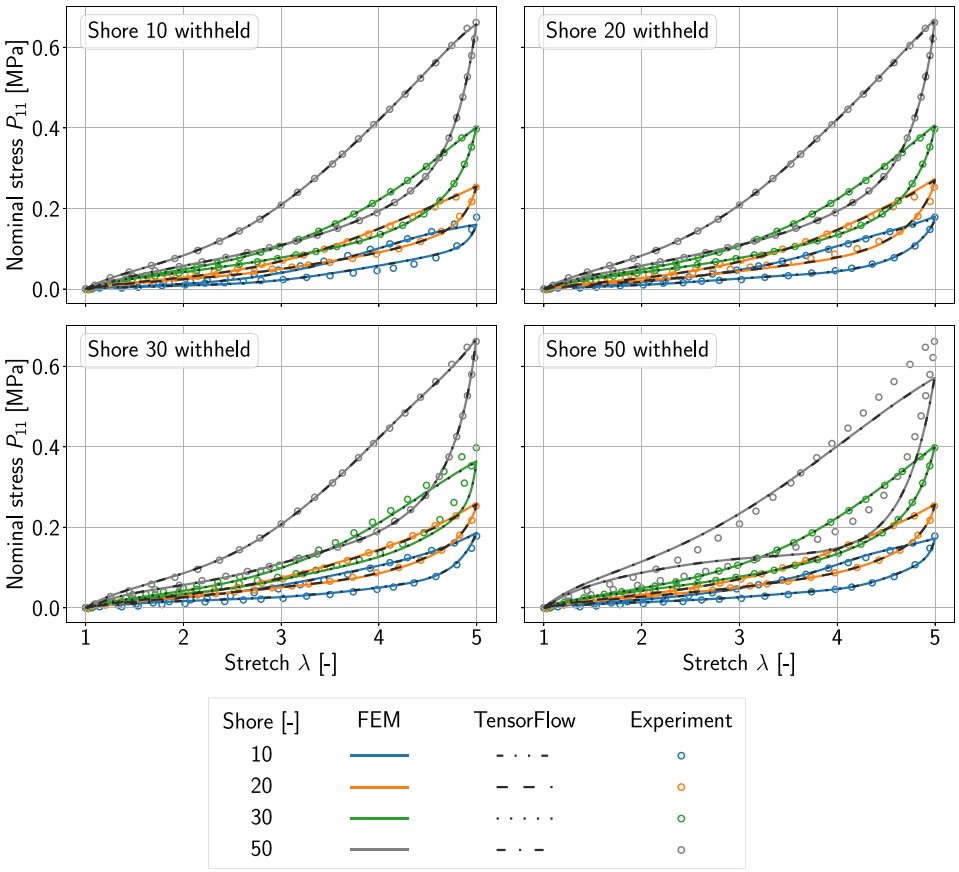


Fig. H.1. Training and validation results for Ecoflex silicone polymer data, using a leave-one-out cross-validation scheme. The subplots show the nominal stress P_{11} versus stretch λ for uniaxial loading–unloading tests at different Shore hardness levels. In each subplot, one hardness level is excluded during training and used for validation. The predictive performance on this withheld data demonstrates the ability of the trained vCANN to predict the behavior of a material with a previously unseen hardness level. Solid colored lines denote the FEM implementation, black dashed/dotted lines represent the TensorFlow implementation, and open markers indicate experimental data reproduced from [103]. Both implementations yield identical results. The R^2 values of the training and validation datasets are provided in Table H.1.

I.1. Descriptive performance on training data

We employed a vCANN with a single generalized Maxwell model ($R = 1$). Owing to isotropy, the associated generalized structural tensor (44) reduces to $\tilde{\mathbf{L}}_1 = \frac{1}{3}\mathbf{I}$. Accordingly, the following free energy functions and relaxation times—represented by the FFNNs described in Section 3—are learned from three synthetic multi-step relaxation tests:

$$\Psi^{\infty}_{\text{iso}\,1} = \Psi^{\infty}_{\text{iso}\,1}(\bar{\tilde{I}}_1), \Psi_{1\alpha} = \Psi_{1\alpha}(\bar{\tilde{I}}_1), \tau_{1\alpha} = \tau_{1\alpha}(\bar{\tilde{I}}_1), \alpha = 1, \dots, N^{\max}, \tag{I.1}$$

where

$$\bar{\tilde{I}}_1 = \left(\bar{\tilde{I}}_1, \bar{\tilde{J}}_1\right), \qquad \bar{\tilde{I}}_1 = \frac{1}{3}\,\text{tr}[\bar{\mathbf{C}}], \qquad \bar{\tilde{J}}_1 = \frac{1}{3}\,\text{tr}[\text{cof}\,\bar{\mathbf{C}}]. \tag{I.2}$$

For simplicity, we neglect the strain-rate dependence of the relaxation times and omit the feature vector $\boldsymbol{f}$, since no additional information is available to characterize the data. Table J.1 lists the remaining hyperparameters of the vCANN.

For an isotropic, incompressible material, each deformation state can be represented in the invariant plane by an invariant pair $(\bar{\tilde{I}}_r, \bar{\tilde{J}}_r)$. The uniaxial and equi-biaxial tension curves bound the set of admissible invariant pairs. Any other admissible invariant pair corresponds to a general biaxial deformation [112], with independent stretches in the two orthogonal loading directions (Fig. 11). Fig. I.1 also illustrates a key limitation of relying on a single experiment to characterize a material: constitutive models fitted to one loading mode (e.g., uniaxial tension) are likely to fail when extrapolated to other deformation states in the invariant plane, highlighting the need for diverse stress–strain data. Accordingly, the vCANN was trained on in silico uniaxial, equi-biaxial, and pure shear (strip-biaxial) multi-step relaxation tests. The resulting fit shows excellent agreement with the training data (Fig. I.2).

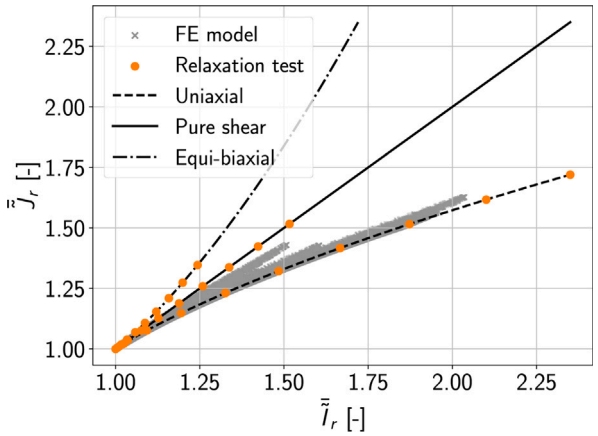


Fig. I.1. Invariant plane: admissible deformation states of an isotropic, incompressible material lie between the uniaxial and equi-biaxial tension curves. The gray crosses denote deformation states in the cylinder simulated with the ground truth model and extracted at the integration points. The orange points denote deformation states encountered during the multi-step relaxation tests.

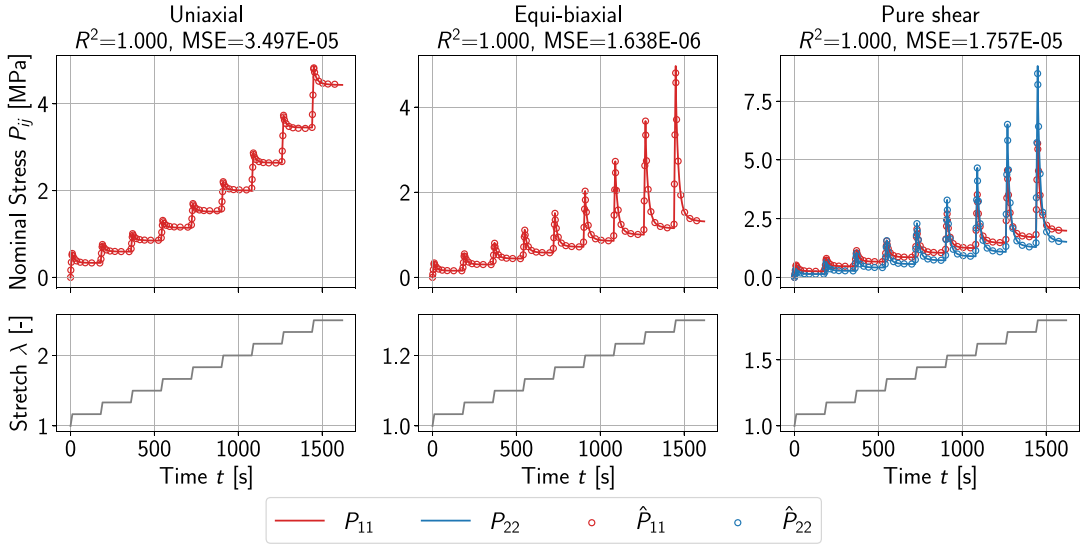


Fig. I.2. Top: solid curves denote the vCANN fit; dotted curves denote the training data from the isotropic ground truth model (every tenth data point is shown for clarity). Bottom: stretch histories of the multi-step relaxation tests.

I.2. Predictive performance in FE simulation

The vCANN was automatically translated into a material subroutine and integrated into the FE simulation of the isotropic artery model. Fig. I.3 shows the relative error in von Mises stress between the vCANN prediction and the GT model at the time points of load reversal. Even during these critical load reversals, the relative error remains below 1.0%.

The cumulative distribution function in Fig. I.4 indicates that in 90% of cases the relative error is less than 0.4%, and in 50% of cases it is below 0.1%. Similarly, the vCANN prediction of the reaction force and moment in the $\boldsymbol{E}_2$-direction is in excellent agreement with the GT solution, with virtually no observable difference (Fig. I.5).

Overall, we conclude that, in the isotropic case, the vCANN generalizes remarkably well from three simple multi-step relaxation tests to the structural level and even to the integration-point level under complex loading. This is particularly notable because the artery model was subjected to cyclic loading, whereas the training data comprised only relaxation tests. Fig. I.1 shows that the training data cover only a limited set of deformation states (orange markers), non-uniformly distributed near the boundaries of the admissible invariant space. In contrast, the invariant pairs observed at integration points in the artery simulation with the GT model (gray markers) cover a substantially larger region of the invariant plane.

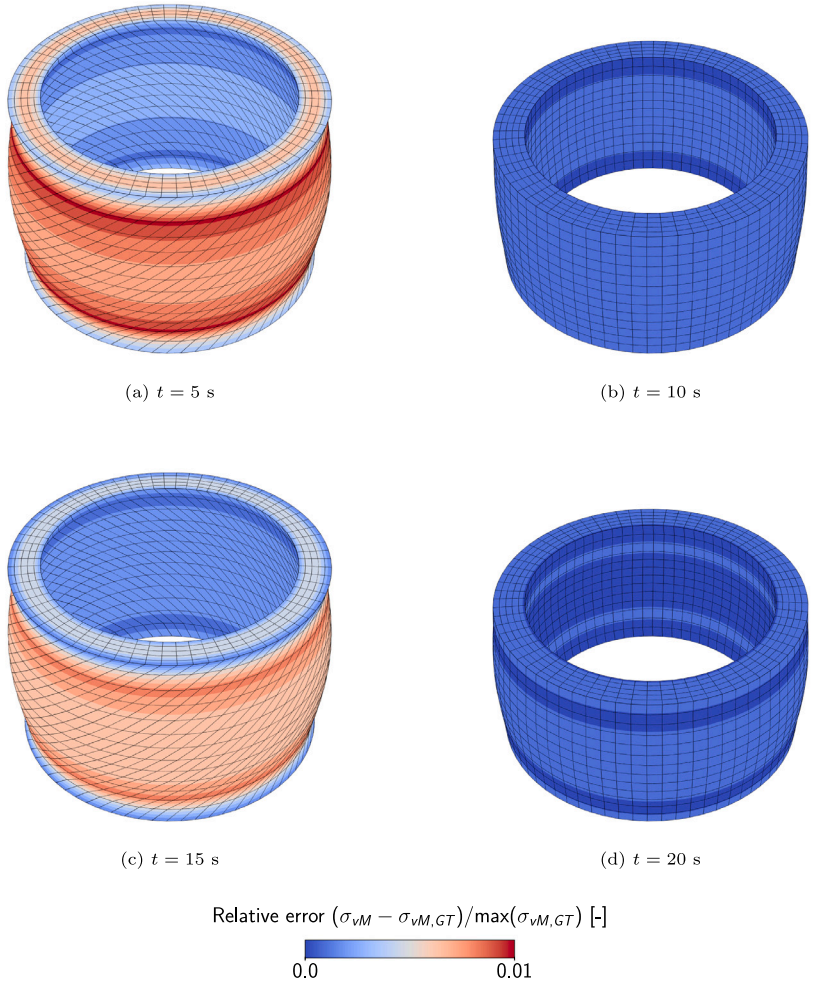


Fig. I.3. Isotropic artery model: relative error in von Mises stress between the vCANN prediction σ_{vM} and the ground truth solution $\sigma_{vM,GT}$ at the time points of load reversal.

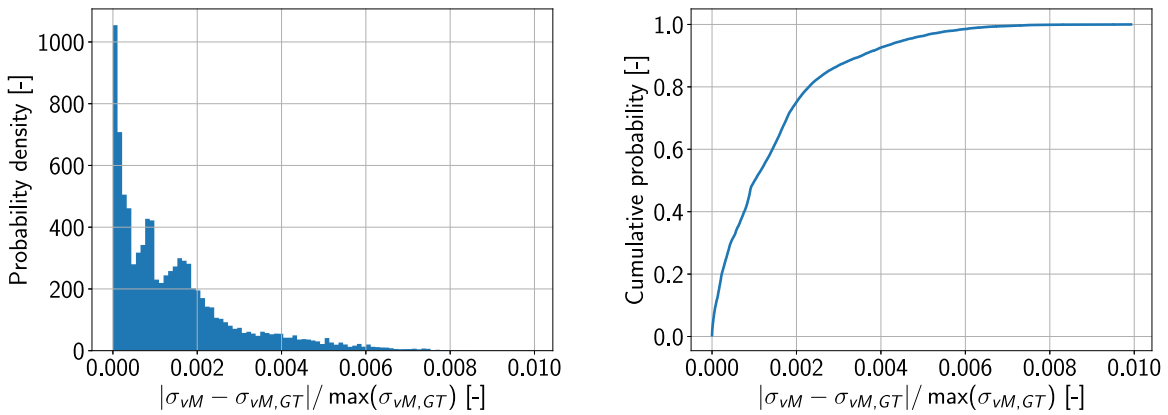


Fig. I.4. Isotropic artery model: empirical probability density function (left) and empirical cumulative distribution function (right) of the relative error in von Mises stress between the vCANN prediction σ_{vM} and the ground truth solution $\sigma_{vM,GT}$. The errors were calculated in each time instance in all integration points.

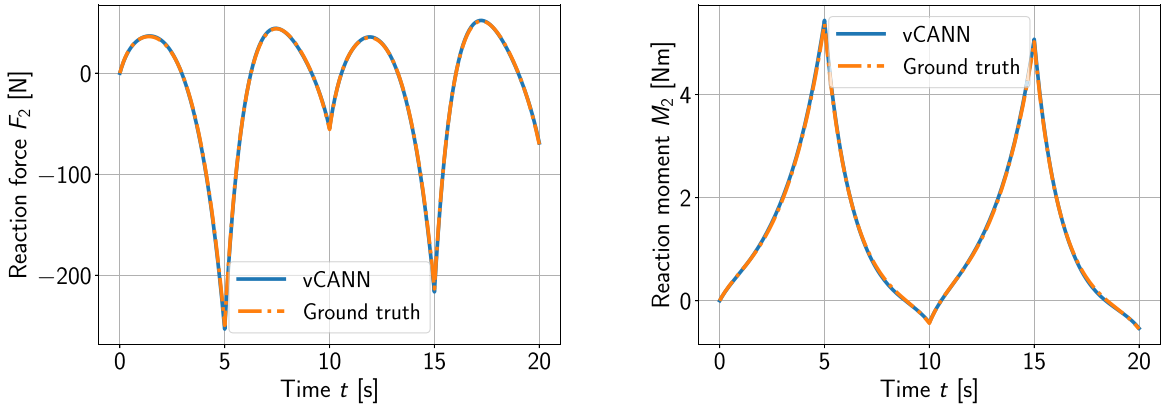


Fig. I.5. Isotropic artery model: time history of the vertical reaction force F_2 (left) and the torsional reaction moment M_2 (right) recorded at the coupling point B. Solid curves represent the vCANN predictions, and dash-dotted curves are the ground truth solutions.

Appendix J. Hyperparameters

We list the hyperparameters of the vCANNs trained in Section 5. The choice of activation functions is detailed in Section 3. The training was terminated based on early stopping. We initialized all weights using the (positive) variance scaling initializer with truncated normal distribution, while biases were initialized to zero.

Table J.1
Hyperparameters and model parameters of all vCANNs trained in this study.

Hyperparameter	Value			
	Section 5.1, 5.2	Section 5.3	Appendix H	Appendix I
General				
Sparsity penalty parameter Λ	0.0	0.0	0.0	0.0
Number of generalized structural tensors R	1	1	1	1
Number of preferred material directions n	0	2	0	0
Equilibrium free energy $\Psi_{\text{iso}}^{\mathcal{N}}$				
Number of neurons per hidden layer of $\mathcal{N}_{\Psi_{\text{iso}}}$	$\{8,\}$	$\{10,\}$	$\{8,\}$	$\{10,\}$
Relaxation times $\tau_{r\alpha}$ *&* *non-equilibrium free energy* $\Psi_{r\alpha}^{\mathcal{N}}$				
Maximal number of Maxwell elements $N^{\max}$	3	3	3	3
Time normalization $[T_{\min}, T_{\max}]$	$[10^0, 10^2]$ s	$[10^0, 10^2]$ s	$[10^0, 10^2]$ s	$[10^0, 10^2]$ s
Number of neurons per hidden layer of $\mathcal{N}_{\Psi_\alpha}$	$\{8,\}$	$\{10,\}$	$\{8,\}$	$\{10,\}$
Number of neurons per hidden layer of $\mathcal{N}_{\tau_{r\alpha}}$	$\{8,\}$	$\{10,\}$	$\{8,\}$	$\{10,\}$

Data availability

To ensure accessibility and reproducibility, all of the codes and FE subroutines necessary for using the proposed framework within Abaqus are openly provided at: https://github.com/KianAbd/vCANN_FEM.

References

[1] Dirk Steglich, Jacques Besson, Inken Reinke, Heike Helmholz, Monika Luczak, Vasil M. Garamus, Björn Wiese, Daniel Höche, Christian J. Cyron, Regine Willumeit-Römer, Strength and ductility loss of Magnesium-Gadolinium due to corrosion in physiological environment: Experiments and modeling, J. Mech. Behav. Biomed. Mater. 144 (2023) 105939.
[2] Christian J. Cyron, Wolfgang A. Wall, Numerical method for the simulation of the Brownian dynamics of rod-like microstructures with three-dimensional nonlinear beam elements, Internat. J. Numer. Methods Engrg. 90 (8) (2012) 955–987.
[3] Kevin Linka, Vu Ngoc Khiêm, Mikhail Itskov, Multi-scale modeling of soft fibrous tissues based on proteoglycan mechanics, J. Biomech. 49 (12) (2016) 2349–2357.
[4] Kevin Linka, Amelie Schäfer, Markus Hillgärtner, Mikhail Itskov, Matthias Knobe, Christiane Kuhl, Lea Hitpass, Daniel Truhn, Johannes Thuering, Sven Nebelung, Towards patient-specific computational modelling of articular cartilage on the basis of advanced multiparametric MRI techniques, Sci. Rep. 9 (1) (2019) 1–13.
[5] Paolo P. Provenzano, Roderic S. Lakes, Thomas Keenan, Ray Vanderby, Nonlinear ligament viscoelasticity, Ann. Biomed. Eng. 29 (10) (2001) 908–914.
[6] Kevin L. Troyer, Christian M. Puttlitz, Human cervical spine ligaments exhibit fully nonlinear viscoelastic behavior, Acta Biomater. 7 (2) (2011) 700–709.

[7] Marcos Latorre, Francisco J. Montáns, Strain-level dependent nonequilibrium anisotropic viscoelasticity: Application to the abdominal muscle, J. Biomech. Eng. 139 (10) (2017) 1–9.
[8] Adela Capilnasiu, Lynne Bilston, Ralph Sinkus, David Nordsletten, Nonlinear viscoelastic constitutive model for bovine liver tissue, Biomech. Model. Mechanobiol. 19 (5) (2020) 1641–1662.
[9] Will Zhang, Adela Capilnasiu, Gerhard Sommer, Gerhard A. Holzapfel, David A. Nordsletten, An efficient and accurate method for modeling nonlinear fractional viscoelastic biomaterials, Comput. Methods Appl. Mech. Engrg. 362 (2020) 112834.
[10] Abaqus Materials Guide, Dassault Systèmes Simulia Corp., Rhode Island, 2022.
[11] Robert L. Taylor, Sanjay Govindjee, FEAP – A finite element analysis program, version 8.6 user manual, 2020, p. 693.
[12] Structural Mechanics Module User's Guide, COMSOL Multiphysics® v. 6.2., COMSOL AB, Stockholm, Sweden, 2023.
[13] FEBio theory manual version 3.4, 2021.
[14] Grégory Chagnon, Marie Rebouah, Denis Favier, Hyperelastic energy densities for soft biological tissues: A review, J. Elasticity 120 (2) (2015) 129–160.
[15] Paul Steinmann, Mokarram Hossain, Gunnar Possart, Hyperelastic models for rubber-like materials: Consistent tangent operators and suitability for Treloar's data, Arch. Appl. Mech. 82 (9) (2012) 1183–1217.
[16] Trenton Kirchdoerfer, Michael Ortiz, Data-driven computational mechanics, Comput. Methods Appl. Mech. Engrg. 304 (2016) 81–101.
[17] Lu Trong Khiem Nguyen, Roland C. Aydin, Christian J. Cyron, Accelerating the distance-minimizing method for data-driven elasticity with adaptive hyperparameters, Comput. Mech. 70 (3) (2022) 621–638.
[18] Hossein Salahshoor, Michael Ortiz, Model-free data-driven viscoelasticity in the frequency domain, Comput. Methods Appl. Mech. Engrg. 403 (2023) 115657.
[19] Daniele Versino, Alberto Tonda, Curt A. Bronkhorst, Data driven modeling of plastic deformation, Comput. Methods Appl. Mech. Engrg. 318 (2017) 981–1004.
[20] G.F. Bomarito, T.S. Townsend, K.M. Stewart, K.V. Esham, J.M. Emery, J.D. Hochhalter, Development of interpretable, data-driven plasticity models with symbolic regression, Comput. Struct. 252 (2021).
[21] Evgeniya Kabliman, Ana Helena Kolody, Johannes Kronsteiner, Michael Kommenda, Gabriel Kronberger, Application of symbolic regression for constitutive modeling of plastic deformation, Appl. Eng. Sci. 6 (2021) 100052.
[22] Rasul Abdusalamov, Markus Hillgärtner, Mikhail Itskov, Automatic generation of interpretable hyperelastic material models by symbolic regression, Internat. J. Numer. Methods Engrg. 124 (9) (2023) 2093–2104.
[23] Josh Bongard, Hod Lipson, Automated reverse engineering of nonlinear dynamical systems, Proc. Natl. Acad. Sci. USA 104 (24) (2007) 9943–9948.
[24] Moritz Flaschel, Siddhant Kumar, Laura De Lorenzis, Unsupervised discovery of interpretable hyperelastic constitutive laws, Comput. Methods Appl. Mech. Engrg. 381 (January 2022) (2021) 113852.
[25] Enzo Marino, Moritz Flaschel, Siddhant Kumar, Laura De Lorenzis, Automated identification of linear viscoelastic constitutive laws with EUCLID, Mech. Mater. 181 (2023) 104643.
[26] Jorge-Humberto Urrea-Quintero, David Anton, Laura De Lorenzis, Henning Wessels, Automated constitutive model discovery by pairing sparse regression algorithms with model selection criteria, Comput. Methods Appl. Mech. Engrg. 449 (2026) 118551.
[27] Kurt Hornik, Maxwell Stinchcombe, Halbert White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (5) (1989) 359–366.
[28] Filippo Masi, Ioannis Stefanou, Paolo Vannucci, Victor Maffi-Berthier, Thermodynamics-based artificial neural networks for constitutive modeling, J. Mech. Phys. Solids 147 (May 2020) (2021) 104277.
[29] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.
[30] Kevin Linka, Markus Hillgärtner, Kian P. Abdolazizi, Roland C. Aydin, Mikhail Itskov, Christian J. Cyron, Constitutive artificial neural networks: A fast and general approach to predictive data-driven constitutive modeling by deep learning, J. Comput. Phys. 429 (2021) 110010.
[31] Kevin Linka, Ellen Kuhl, A new family of constitutive artificial neural networks towards automated model discovery, Comput. Methods Appl. Mech. Engrg. 403 (2023) 115731.
[32] Marius Tacke, Matthias Busch, Kartik Bali, Kian Abdolazizi, Kevin Linka, Christian Cyron, Roland Aydin, Constitutive scientific generative agent (CSGA): Leveraging large language models for automated constitutive model discovery, Mach. Learn. Comput. Sci. Eng. 1 (1) (2025) 23–.
[33] Kevin Linka, Nina Reiter, Jasmin Würges, Martin Schicht, Lars Bräuer, Christian J. Cyron, Friedrich Paulsen, Silvia Budday, Unraveling the local relation between tissue composition and human brain mechanics through machine learning, Front. Bioeng. Biotechnol. 9 (2021) 704738.
[34] Kevin Linka, Cristina Cavinato, Jay D. Humphrey, Christian J. Cyron, Predicting and understanding arterial elasticity from key microstructural features by bidirectional deep learning, Acta Biomater. 147 (2022) 63–72.
[35] Gerhard A. Holzapfel, Kevin Linka, Selda Sherifova, Christian J. Cyron, Predictive constitutive modelling of arteries by deep learning, J. R. Soc. Interface 18 (182) (2021).
[36] Mathias Peirlinck, Juan A. Hurtado, Manuel K. Rausch, Adrián Buganza Tepole, Ellen Kuhl, A universal material model subroutine for soft matter systems, Eng. Comput. 41 (2) (2025) 905–927.
[37] Lennart Linden, Dominik K. Klein, Karl A. Kalina, Jörg Brummund, Oliver Weeger, Markus Kästner, Neural networks meet hyperelasticity: A guide to enforcing physics, J. Mech. Phys. Solids 179 (2023) 105363.
[38] Franz Dammaß, Karl A. Kalina, Markus Kästner, When invariants matter: The role of I1 and I2 in neural network models of incompressible hyperelasticity, Mech. Mater. 210 (2025).
[39] Max Rosenkranz, Karl A. Kalina, Jörg Brummund, Wai Ching Sun, Markus Kästner, Viscoelasticty with physics-augmented neural networks: model formulation and training methods without prescribed internal variables, Comput. Mech. 74 (6) (2024) 1279–1301.
[40] Karl A. Kalina, Jörg Brummund, Markus Kästner, A physics-augmented neural network framework for finite strain incompressible viscoelasticity, Comput. Methods Appl. Mech. Engrg. 455 (2025) 118892.
[41] Moritz Flaschel, Paul Steinmann, Laura De Lorenzis, Ellen Kuhl, Convex neural networks learn generalized standard material models, J. Mech. Phys. Solids 200 (2025).
[42] Hagen Holthusen, Lukas Lamm, Tim Brepols, Stefanie Reese, Ellen Kuhl, Theory and implementation of inelastic constitutive artificial neural networks, Comput. Methods Appl. Mech. Engrg. 428 (2024) 117063.
[43] Hagen Holthusen, Ellen Kuhl, A complement to neural networks for anisotropic inelasticity at finite strains, Comput. Methods Appl. Mech. Engrg. 450 (2026) 118612.
[44] Kian P. Abdolazizi, Roland C. Aydin, Christian J. Cyron, Kevin Linka, Constitutive Kolmogorov–Arnold networks (CKANs): Combining accuracy and interpretability in data-driven material modeling, J. Mech. Phys. Solids 203 (2025) 106212.
[45] Prakash Thakolkaran, Yaqi Guo, Shivam Saini, Mathias Peirlinck, Benjamin Alheit, Siddhant Kumar, Can KAN CANs? Input-convex Kolmogorov-Arnold networks (KANs) as hyperelastic constitutive artificial neural networks (CANs), Comput. Methods Appl. Mech. Engrg. 443 (2025) 118089.
[46] Chenyi Ji, Kian P. Abdolazizi, Hagen Holthusen, Christian J. Cyron, Kevin Linka, Inelastic constitutive Kolmogorov-Arnold networks: A generalized framework for automated discovery of interpretable inelastic material models, 2026.
[47] Vahidullah Taç, Manuel K. Rausch, Francisco Sahli Costabal, Adrian Buganza Tepole, Data-driven anisotropic finite viscoelasticity using neural ordinary differential equations, Comput. Methods Appl. Mech. Engrg. 411 (2023) 116046.
[48] Stefanie Reese, Sanjay Govindjee, A theory of finite viscoelasticity and numerical aspects, Int. J. Solids Struct. 35 (26–27) (1998) 3455–3482.

[49] Simon Wiesheier, Miguel Angel Moreno-Mateos, Paul Steinmann, Data-adaptive spline-based viscoelasticity for soft solids, Comput. Methods Appl. Mech. Engrg. 451 (2026).
[50] Kian P. Abdolazizi, Kevin Linka, Christian J. Cyron, Viscoelastic constitutive artificial neural networks (vCANNs) – A framework for data-driven anisotropic nonlinear finite viscoelasticity, J. Comput. Phys. 499 (2023) 112704.
[51] Juan C. Simo, On a fully three-dimensional finite-strain viscoelastic damage model: Formulation and computational aspects, Comput. Methods Appl. Mech. Engrg. 60 (2) (1987) 153–173.
[52] Gerhard A. Holzapfel, On large strain viscoelasticity: Continuum formulation and finite element applications to elastomeric structures, Internat. J. Numer. Methods Engrg. 39 (22) (1996) 3903–3926.
[53] T. Christian Gasser, Caroline Forsell, The numerical implementation of invariant-based viscoelastic formulations at finite strains. An anisotropic model for the passive myocardium, Comput. Methods Appl. Mech. Engrg. 200 (49–52) (2011) 3637–3645.
[54] Michael Kaliske, Heinrich Rothert, Formulation and implementation of three-dimensional viscoelasticity at small and finite strains, Comput. Mech. 19 (3) (1997) 228–239.
[55] Estefanía Peña, Begoña Calvo, Miguel A. Martínez, Manuel Doblaré, An anisotropic visco-hyperelastic model for ligaments at finite strains. Formulation and computational aspects, Int. J. Solids Struct. 44 (3–4) (2007) 760–778.
[56] Sanjay Govindjee, Trevor Potter, Jon Wilkening, Dynamic stability of spinning viscoelastic cylinders at finite deformation, Int. J. Solids Struct. 51 (21–22) (2014) 3589–3603.
[57] Ju Liu, Marcos Latorre, Alison L. Marsden, A continuum and computational framework for viscoelastodynamics: I. Finite deformation linear models, Comput. Methods Appl. Mech. Engrg. 385 (2021) 114059.
[58] Maximilian P. Wollner, Michele Terzano, Malte Rolf-Pissarczyk, Gerhard A. Holzapfel, A general model for anisotropic pseudo-elasticity and viscoelasticity at finite strains, J. Mech. Phys. Solids 180 (August) (2023) 105403.
[59] Ju Liu, Jiashen Guan, A continuum and computational framework for viscoelastodynamics: II. Strain-driven and energy–momentum consistent schemes, Comput. Methods Appl. Mech. Engrg. 417 (2023) 116308.
[60] Michele Terzano, Maximilian P. Wollner, Manuel P. Kainz, Malte Rolf-Pissarczyk, Nils Götzen, Gerhard A. Holzapfel, Modelling the anisotropic inelastic response of polymeric scaffolds for in situ tissue engineering applications, J. R. Soc. Interface 20 (206) (2023) 20230318.
[61] Mikhail Itskov, Tensor algebra and tensor analysis for engineers, in: Mathematical Engineering, in: Mathematical Engineering, Springer Cham, 2015, pp. 1–295.
[62] Paul J. Flory, Thermodynamic relations for high elastic materials, Trans. Faraday Soc. 57 (1961) 829–838.
[63] Ray W. Ogden, Volume changes associated with the deformation of rubber-like solids, J. Mech. Phys. Solids 24 (6) (1976) 323–338.
[64] Alexander E. Ehret, Mikhail Itskov, A polyconvex hyperelastic model for fiber-reinforced materials in application to soft tissues, J. Mater. Sci. 42 (21) (2007) 8853–8863.
[65] Marc Olive, Boris Kolev, Rodrigue Desmorat, Boris Desmorat, Characterization of the symmetry class of an elasticity tensor using polynomial covariants, Math. Mech. Solids 27 (1) (2022) 144–190.
[66] Karl A. Kalina, Jörg Brummund, WaiChing Sun, Markus Kästner, Neural networks meet anisotropic hyperelasticity: A framework based on generalized structure tensors and isotropic tensor functions, Comput. Methods Appl. Mech. Engrg. 437 (2025) 117725.
[67] Jean-Paul Boehler, in: J.P. Boehler (Ed.), Applications of Tensor Functions in Solid Mechanics, Springer Vienna, Vienna, 1987.
[68] Anthony J.M. Spencer, Continuum theory of the mechanics of fibre-reinforced composites, in: A.J.M. Spencer (Ed.), Journal of Chemical Information and Modeling, Springer Vienna, Vienna, 1984.
[69] T. Christian Gasser, Ray W. Ogden, Gerhard A. Holzapfel, Hyperelastic modelling of arterial layers with distributed collagen fibre orientations, J. R. Soc. Interface 3 (6) (2006) 15–35.
[70] Alexander E. Ehret, Mikhail Itskov, Modeling of anisotropic softening phenomena: Application to soft biological tissues, Int. J. Plast. 25 (5) (2009) 901–919.
[71] Gerhard A. Holzapfel, Thomas C. Gasser, Ray W. Ogden, A new constitutive framework for arterial wall mechanics and a comparative study of material models, J. Elasticity 61 (1–3) (2000) 1–48.
[72] John D. Ferry, Henry S. Myers, Viscoelastic properties of polymers, J. Electrochem. Soc. 108 (7) (1961) 142C.
[73] Juan C. Simo, Thomas J.R. Hughes, Computational inelasticity, in: Computational Inelasticity, in: Interdisciplinary Applied Mathematics, vol. 7, Springer-Verlag, New York, 1998.
[74] Gerhard A. Holzapfel, Nonlinear Solid Mechanics: A Continuum Approach for Engineering Science, John Wiley & Sons Ltd., 2000, p. 456.
[75] Bernard D. Coleman, Walter Noll, The thermodynamics of elastic materials with heat conduction and viscosity, Arch. Ration. Mech. Anal. 13 (1) (1963) 167–178.
[76] Bernard D. Coleman, Morton E. Gurtin, Thermodynamics with internal state variables, J. Chem. Phys. 47 (2) (1967) 597–613.
[77] Ray W. Ogden, Nonlinear elastic deformations, in: Journal of Applied Mechanics, Dover Publications, 1997.
[78] Jin Min Zhang, Jan Rychlewski, Structure tensors for anisotropic solids, Arch. Mech. 42 (3) (1990) 267–277.
[79] Horst Parisch, Festkörper-Kontinuumsmechanik, Vieweg+Teubner Verlag, Wiesbaden, 2003.
[80] Stefan Doll, Karl Schweizerhof, On the development of volumetric strain energy functions, J. Appl. Mech. Trans. ASME 67 (1) (2000) 17–21.
[81] Stefan Hartmann, Patrizio Neff, Polyconvexity of generalized polynomial-type hyperelastic strain energy functions for near-incompressibility, Int. J. Solids Struct. 40 (11) (2003) 2767–2791.
[82] John M. Ball, Convexity conditions and existence theorems in nonlinear elasticity, Arch. Ration. Mech. Anal. 63 (4) (1976) 337–403.
[83] Jörg Schröder, Patrizio Neff, Invariant formulaiton of hyperelastic transverse isotropy based on polyconvex free energy functions, Int. J. Solids Struct. 40 (2) (2003) 401–445.
[84] Carlo Sansour, On the physical assumptions underlying the volumetric-isochoric split and the case of anisotropy, Eur. J. Mech. A Solids 27 (1) (2008) 28–39.
[85] Jörg Helfenstein, Mahmood Jabareen, Edoardo Mazza, Sanjay Govindjee, On non-physical response in models for fiber-reinforced hyperelastic materials, Int. J. Solids Struct. 47 (16) (2010) 2056–2061.
[86] Gerhard A. Holzapfel, T. Christian Gasser, Michael Stadler, A structural model for the viscoelastic behavior of arterial walls: Continuum formulation and finite element analysis, Eur. J. Mech. A Solids 21 (3) (2002) 441–463.
[87] Osman Gültekin, Gerhard Sommer, Gerhard A. Holzapfel, An orthotropic viscoelastic model for the passive myocardium: continuum basis and numerical treatment, Comput. Methods Biomech. Biomed. Eng. 19 (15) (2016) 1647–1664.
[88] Marcos Latorre, Francisco Javier Montáns, Fully anisotropic finite strain viscoelasticity based on a reverse multiplicative decomposition and logarithmic strains, Comput. Struct. 163 (2016) 56–70.
[89] Peter Haupt, Konstantin Sedlan, Viscoplasticity of elastomeric materials: experimental facts and constitutive modelling, Arch. Appl. Mech. 71 (2–3) (2001) 89–109.
[90] A.F.M. Saiful-Amin, Alexander Lion, S. Sekita, Yoshiaki Okui, Nonlinear dependence of viscosity in modeling the rate-dependent response of natural and high damping rubbers in compression and shear: Experimental identification and numerical verification, Int. J. Plast. 22 (9) (2006) 1610–1657.
[91] Dominique P. Pioletti, Lalao R. Rakotomanana, On the independence of time and strain effects in the stress relaxation of ligaments and tendons, J. Biomech. 33 (12) (2000) 1729–1732.

[92] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, Xiaoqiang Zheng, TensorFlow: A system for large-scale machine learning, in: Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016, 2016, pp. 265–283.
[93] Stephen P. Boyd, Lieven Vandenberghe, Convex Optimization, Cambridge University Press, 2004, p. 716.
[94] J. Patrick Wilber, Jay R. Walton, The convexity properties of a class of constitutive models for biological soft issues, Math. Mech. Solids 7 (3) (2002) 217–235.
[95] Brandon Amos, Lei Xu, J. Zico Kolter, Input convex neural networks, in: Proceedings of the 34th International Conference on Machine Learning, PMLR, 2017.
[96] Shenglin Huang, Zequn He, Bryan Chem, Celia Reina, Variational onsager neural networks (VONNs): A thermodynamics-based variational learning strategy for non-equilibrium PDEs, J. Mech. Phys. Solids 163 (2022) 104856.
[97] Roderic S. Lakes, Viscoelastic Materials, Cambridge University Press, 2009.
[98] Julie Diani, Pierre Gilormini, Carole Frédy, Ingrid Rousseau, Predicting thermal shape memory of crosslinked polymer networks from linear viscoelasticity, Int. J. Solids Struct. 49 (5) (2012) 793–799.
[99] Tingxiong Xiao, Runzhao Yang, Yuxiao Cheng, Jinli Suo, SHoP: A deep learning framework for solving high-order partial differential equations, Proc. the AAAI Conf. Artif. Intell. 38 (14) (2024) 16032–16039.
[100] Abaqus Theory Guide, Dassault Systèmes Simulia Corp., Rhode Island, 2022.
[101] David R. Nolan, Caitríona Lally, J. Patrick McGarry, Understanding the deformation gradient in Abaqus and key guidelines for anisotropic hyperelastic user material subroutines (UMATs), J. Mech. Behav. Biomed. Mater. 126 (April 2021) (2022) 104940.
[102] Heleen Fehervary, Lauranne Maes, Julie Vastmans, Gertjan Kloosterman, Nele Famaey, How to implement user-defined fiber-reinforced hyperelastic materials in finite element software, J. Mech. Behav. Biomed. Mater. 110 (103737) (2020).
[103] Zisheng Liao, Mokarram Hossain, Xiaohu Yao, Markus Mehnert, Paul Steinmann, On thermo-viscoelastic experimental characterization and numerical modelling of VHB polymer, Int. J. Non-Linear Mech. 118 (2020) 103263.
[104] Abaqus Benchmark Guide, Dassault Systèmes Simulia Corp., Rhode Island, 2022.
[105] Mikhail Itskov, Nuri Aksel, A class of orthotropic and transversely isotropic hyperelastic constitutive models based on a polyconvex strain energy function, Int. J. Solids Struct. 41 (14) (2004) 3833–3848.
[106] Sanjay Govindjee, Juan C. Simo, Mullins effect and the strain amplitude dependence of the storage modulus, Int. J. Solids Struct. 29 (14–15) (1992) 1737–1751.
[107] Y. Lanir, Y.C. Fung, Two-dimensional mechanical properties of rabbit skin-I. Experimental system, J. Biomech. 7 (1) (1974) 29–34.
[108] Michael S. Sacks, Biaxial mechanical evaluation of planar biological materials, J. Elasticity 61 (1–3) (2000) 199–246.
[109] S. Kawabata, M. Matsuda, K. Tei, H. Kawai, Experimental survey of the strain energy density function of isoprene rubber vulcanizate, Macromolecules 14 (1) (1981) 154–162.
[110] Kevin L. Troyer, Donald J. Estep, Christian M. Puttlitz, Viscoelastic effects during loading play an integral role in soft tissue mechanics, Acta Biomater. 8 (1) (2012) 234–243.
[111] Mikhail Itskov, Alexander E. Ehret, A universal model for the elastic, inelastic and active behaviour of soft biological tissues, GAMM-Mitt. 32 (2) (2009) 221–236.
[112] Michael Johlitz, Stefan Diebels, Characterisation of a polymer using biaxial tension tests. Part I: Hyperelasticity, Arch. Appl. Mech. 81 (10) (2011) 1333–1349.
[113] Christian Miehe, Numerical computation of algorithmic (consistent) tangent moduli in large-strain computational inelasticity, Comput. Methods Appl. Mech. Engrg. 134 (3–4) (1996) 223–240.
[114] Wei Sun, Elliot L. Chaikof, Marc E. Levenston, Numerical approximation of tangent moduli for finite element implementations of nonlinear hyperelastic material models, J. Biomech. Eng. (2009).
[115] Zisheng Liao, Mokarram Hossain, Xiaohu Yao, Ecoflex polymer of different shore hardnesses: Experimental investigations and constitutive modelling, Mech. Mater. 144 (February) (2020) 103366.