

Can KAN CANs? Input-convex Kolmogorov-Arnold Networks (KANs) as hyperelastic constitutive artificial neural networks (CANs)

Thakolkaran, Prakash; Guo, Yaqi; Saini, Shivam; Peirlinck, Mathias; Alheit, Benjamin; Kumar, Siddhant

DOI

[10.1016/j.cma.2025.118089](https://doi.org/10.1016/j.cma.2025.118089)

Publication date

2025

Document Version

Final published version

Published in

Computer Methods in Applied Mechanics and Engineering

Citation (APA)

Thakolkaran, P., Guo, Y., Saini, S., Peirlinck, M., Alheit, B., & Kumar, S. (2025). Can KAN CANs? Input-convex Kolmogorov-Arnold Networks (KANs) as hyperelastic constitutive artificial neural networks (CANs). *Computer Methods in Applied Mechanics and Engineering*, 443, Article 118089. <https://doi.org/10.1016/j.cma.2025.118089>

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.



Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

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

Can KAN CANs? Input-convex Kolmogorov-Arnold Networks (KANs) as hyperelastic constitutive artificial neural networks (CANs)

Prakash Thakolkaran ^a,^{*}1, Yaqi Guo ^a,¹, Shivam Saini ^a, Mathias Peirlinck ^b, Benjamin Alheit ^{a,b}, Siddhant Kumar ^a,^{*}

^a Department of Materials Science and Engineering, Delft University of Technology, 2628 CD Delft, The Netherlands

^b Department of Biomechanical Engineering, Delft University of Technology, 2628 CD Delft, The Netherlands

ARTICLE INFO

Dataset link: <https://github.com/mmc-group/1CKANs/>

Keywords:

Constitutive modeling
Unsupervised learning
Hyperelasticity
Kolmogorov-arnold network
Convexity

ABSTRACT

Traditional constitutive models rely on hand-crafted parametric forms with limited expressivity and generalizability, while neural network-based models can capture complex material behavior but often lack interpretability. To balance these trade-offs, we present monotonic Input-Convex Kolmogorov-Arnold Networks (ICKANs) for learning polyconvex hyperelastic constitutive laws. ICKANs leverage the Kolmogorov-Arnold representation, decomposing the model into compositions of trainable univariate spline-based activation functions for rich expressivity. We introduce trainable monotonic input-convex splines within the KAN architecture, ensuring physically admissible polyconvex models for isotropic compressible hyperelasticity. The resulting models are both compact and interpretable, enabling explicit extraction of analytical constitutive relationships through a monotonic input-convex symbolic regression technique. Through unsupervised training on full-field strain data and limited global force measurements, ICKANs accurately capture nonlinear stress-strain behavior across diverse strain states. Finite element simulations of unseen geometries with trained ICKAN hyperelastic constitutive models confirm the framework's robustness and generalization capability.

1. Introduction

Constitutive modeling of material behavior has traditionally relied on *a priori* hand-crafted parametric models, with model parameters iteratively calibrated using simple experimental data, such as tension or torsion tests. However, this approach poses challenges in limited generalization of the constitutive model beyond the calibration data and relies on an inefficient, time-consuming trial-and-error process of hand-crafting such parametric models. In recent years, machine learning (ML) and data-driven techniques have revolutionized traditional phenomenological constitutive modeling approaches. Rather than relying on hand-crafted parametric models, these methods extract patterns directly from experimental or multiscale simulation data, enabling the efficient development of physically-consistent, accurate, and generalizable constitutive models [1,2]. Among the modern data-driven approaches, several categories exist, each with its own advantages and limitations. Here, we provide a brief and non-exhaustive review of recent representative works (see [1] for detailed review) to identify existing gaps and highlight the motivation for this study.

* Corresponding authors.

E-mail addresses: P.Thakolkaran@tudelft.nl (P. Thakolkaran), Sid.Kumar@tudelft.nl (S. Kumar).

¹ Equal contributions.

<https://doi.org/10.1016/j.cma.2025.118089>

Received 11 March 2025; Received in revised form 10 May 2025; Accepted 10 May 2025

Available online 29 May 2025

0045-7825/© 2025 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

The *model-free* approaches [3–10] rely on a large catalog of stress–strain pairs and avoid potential errors arising from modeling assumption. In this method, when a new strain query is made, the closest stress–strain pair from the existing dataset is identified, subject to physical compatibility constraints. However, it struggles with poor generalization to stress–strain states beyond the available dataset, lacks the ability to correct for measurement noise, and requires a large amount of data.

In contrast, *model-based* approaches aim to learn physically admissible and consistent surrogate mappings of stress–strain data using various modeling techniques. These methods strive to enhance generalizability, correct for measurement noise, and achieve accurate predictions with minimal data. However, among model-based approaches, state-of-the-art techniques span a broad spectrum in terms of interpretability (i.e., understanding how and why a model makes a certain prediction), generalizability (i.e., extending beyond training data), and model expressivity (i.e., the ability to approximate diverse data distributions and constitutive behaviors).

On one end of the spectrum are techniques based on sparse and symbolic regression. *Sparse regression* techniques [11–17] assume a large catalog of hand-crafted candidate functions and use deterministic or Bayesian methods to select a subset, creating parsimonious and interpretable constitutive models. *Symbolic regression* techniques [18–23] explore permutation and combination of mathematical operations on input data to yield an analytical model that best fits the data. While these approaches offer high interpretability and generalizability as well as analytically verifiable compliance with physical laws, their expressive power is inherently constrained by the predefined set of candidate functions and mathematical operations.

On the other end of the spectrum are techniques based on *Gaussian process regression* (GPR) and *neural networks* (NNs) which are uninterpretable and black-box in nature, but offer higher expressivity. GPR is particularly well-suited for constitutive modeling [24–27] in the low-data regime but encounters computational scaling bottlenecks as the amount of data increases. Neural networks (NNs) as constitutive models overcome this bottleneck by leveraging scalable, highly parameterized architectures (ranging from thousands to millions of trainable parameters) while providing significantly greater expressivity than GPR. NN-based constitutive models have been successfully developed for a wide range of material behaviors — from hyperelasticity [28,29], viscoelasticity [30], and plasticity [31] to multi-physics phenomena [32,33]. These models employ diverse architectures, including recurrent neural networks [34], long short-term memory networks, hypernetworks [35], neural ordinary differential equations [36], sparse networks with interpretable hidden features [37], Bayesian neural networks [38], probabilistic diffusion fields [39], and many more; see [1] for an exhaustive review.

To strike a balance between interpretability and expressivity – in the general ML setting and independent of constitutive modeling – *Kolmogorov-Arnold Networks* (KANs) [40] have recently emerged as a novel NN architecture. The Kolmogorov-Arnold representation theorem [41] states that any multivariate continuous function on a bounded domain can be written as a finite composition of continuous univariate functions and the binary operation of addition. KANs take advantage of the Kolmogorov-Arnold representation theorem to break down high-dimensional functions into simpler univariate functions. Like standard multi-layer perceptrons (MLPs), KANs are fully connected networks. However, they differ in a key way. MLPs learn by adjusting weights and biases, with fixed activation functions providing nonlinearity. In contrast, in KANs, the activation functions themselves are trainable parameters, which allows the network to directly learn the nonlinear transformations. Since their introduction, KANs have gained significant attention and have been extended to solving forward and inverse problems as well as operator learning for partial differential equations [42–46]. In this paper, we first address the question of *how KANs can be adapted for constitutive modeling, focusing on hyperelasticity?*

A major challenge in adapting NNs, including KANs, for constitutive modeling is their high expressivity — an advantage that enables them to capture complex material behavior but, if left unchecked, can lead to unphysical predictions of constitutive behavior. To address this, significant efforts have been made to incorporate physical knowledge into NN architectures, constraining them to the subspace of physically admissible constitutive models and mitigating their tendency to overfit or produce non-physical responses. In the context of hyperelasticity, physical admissibility conditions include, for example, (poly)convexity of the strain energy density, objectivity, and zero stress at zero deformation [28]. Several recent studies have tackled this challenge using various adaptations of the *input-convex neural network* (ICNN) architecture [47] for modeling hyperelastic strain energy densities [48–53]. The ICNN architecture is based on adapting a vanilla MLP architecture to ensure that the output is convex with respect to the input values, which aligns well with the (poly)convexity requirements of a hyperelastic strain energy density. However, the ICNN architecture is not directly applicable to KANs due to fundamental differences in their formulation. We note that an independent study by Abdolazizi et al. [54] has also proposed using KANs as hyperelastic constitutive NNs. In contrast to our approach, the KAN models presented by Abdolazizi et al. [54] are not input-convex and thus do not satisfy the (poly-)convexity requirements of hyperelasticity. Additionally, their approach was limited to supervised training using only stress–strain pairs. As discussed below, unsupervised training and physically admissible generalization is not feasible without enforcing (poly-)convexity. In this study, we explore *how to build a monotonic Input-Convex KAN (ICKAN) architecture, thereby enabling physically admissible KANs-based hyperelastic constitutive models for balanced interpretability and expressivity.*

Another challenging aspect of hyperelastic constitutive NNs is their training. While training on labeled stress–strain pairs from multiscale simulations is straightforward (commonly done using Sobolev training; see e.g., [18]), most common experimental setups (e.g., uniaxial or biaxial tension and torsion tests) fail to sufficiently probe the high-dimensional stress–strain space required to train highly parameterized NNs. Additionally, full-field experimental tests using digital image correlation (DIC) provide strain fields but only boundary-averaged projections of stress tensors (i.e., reaction forces). Training constitutive NNs on full-field strain fields and reaction forces without explicit stress labels presents an unsupervised learning challenge.² Here, we address the question of *how to train ICKAN-based hyperelastic constitutive models in an unsupervised manner.*

² In a typical DIC setup, although the strain field is discretized at thousands-to-millions of points, only one or two reaction forces can be measured. As a result, reaction forces act as a regularization rather than serving as direct labels for supervised training.

To tackle this challenge, we previously introduced the NN-EUCLID framework [48], demonstrating that NN-based hyperelastic constitutive NNs can be trained without stress data, relying solely on strain fields and global force measurements—quantities readily available through mechanical testing and DIC. Without stress labels, the learning process is guided by a physics-motivated loss function based on the conservation of linear momentum. NN-EUCLID builds upon the earlier EUCLID framework [11–16] for unsupervised model discovery via sparse regression, as well as the Virtual Fields Method [55,56] for unsupervised calibration of parametric constitutive models. We note that the Equilibrium Gap Method (EGM) [57] and NN-EUCLID are closely related to the Virtual Fields Method (VFM), with EGM being a special case where local ansatz functions are used for the virtual fields. NN-EUCLID employs neural networks to model the constitutive behavior, while using the same ansatz functions for the virtual fields as in EGM, thereby offering a more flexible and modeling-bias-free approach to constitutive modeling. The validity of NN-EUCLID has been further supported by experimental studies conducted by Jailin et al. [53] and Meng et al. [51]. Here, we demonstrate that ICKAN-based hyperelastic constitutive models are compatible with NN-EUCLID for unsupervised training, thereby paving the way for their application in realistic experimental settings. Notably, we demonstrate that monotonicity and input-convexity in ICKANs are essential for addressing the ill-posedness of unsupervised training—without them, the method fails.

2. Modeling hyperelasticity using ICKANs

2.1. Hyperelasticity preliminaries

For hyperelastic materials, the constitutive model is characterized by a strain energy density function $W(\mathbf{F})$, from which both the first Piola–Kirchhoff stress $\mathbf{P}(\mathbf{F})$, and the incremental tangent modulus $\mathbb{C}(\mathbf{F})$, are derived:

$$\mathbf{P}(\mathbf{F}) = \frac{\partial W(\mathbf{F})}{\partial \mathbf{F}}, \quad \mathbb{C}(\mathbf{F}) = \frac{\partial \mathbf{P}(\mathbf{F})}{\partial \mathbf{F}}, \quad \forall \mathbf{F} \in \text{GL}_+(3). \quad (1)$$

Here, $\text{GL}_+(3)$ denotes the set of all invertible second-order tensors with positive determinants. The constitutive modeling task is to choose an appropriate form of $W(\mathbf{F})$ that not only captures the material response accurately but also satisfies key physical and thermodynamic requirements. These requirements include:

- *Stress-free undeformed configuration:* In the undeformed state, the stress must vanish:

$$\mathbf{P}(\mathbf{F} = \mathbf{I}) = \mathbf{0}. \quad (2)$$

- *Objectivity:* The strain energy density must be objective, i.e.,

$$W(\mathbf{R}\mathbf{F}) = W(\mathbf{F}), \quad \forall \mathbf{F} \in \text{GL}_+(3), \quad \mathbf{R} \in \text{SO}(3), \quad (3)$$

where $\text{SO}(3)$ is the group of all 3D rotation matrices. Note that objectivity implies compatibility with the balance of angular momentum.

- *Polyconvexity:* Material stability is ensured if $W(\mathbf{F})$ satisfies the quasiconvexity condition [58–60]:

$$\int_B W(\bar{\mathbf{F}} + \nabla \mathbf{w}) dV \geq \int_B W(\bar{\mathbf{F}}) dV, \quad \forall B \subset \mathbb{R}^3, \quad \bar{\mathbf{F}} \in \text{GL}_+(3), \quad \mathbf{w} \in C_0^\infty(B), \quad (4)$$

with \mathbf{w} vanishing on ∂B . However, direct enforcement of quasiconvexity is generally impractical [61]. Instead, polyconvexity is preferred over quasiconvexity because it offers a more analytically tractable condition and given that polyconvexity inherently implies quasiconvexity. The strain energy density is polyconvex if and only if there exists a convex function $\mathcal{P} : \mathbb{R}^{3 \times 3} \times \mathbb{R}^{3 \times 3} \times \mathbb{R} \rightarrow \mathbb{R}$ such that

$$W(\mathbf{F}) = \mathcal{P}(\mathbf{F}, \text{Cof } \mathbf{F}, \det \mathbf{F}). \quad (5)$$

2.2. Hyperelastic model ansatz

For the scope of this study, we consider isotropic compressible hyperelasticity. Incorporating the aforementioned physical constraints, we propose the following ansatz for the strain energy density function:

$$W(\mathbf{F}) = \underbrace{W_Q^{\text{ICKAN}}(K_1, K_2, K_3)}_{\substack{\text{monotonic input-convex} \\ \text{KAN-based model}}} + \underbrace{W^0}_{\text{Energy correction}}, \quad (6)$$

$$\text{with } K_1 = (\bar{I}_1 - 3), \quad K_2 = (I_2^* - 3\sqrt{3}), \quad K_3 = (J - 1)^2.$$

Here, W_Q^{ICKAN} represents a monotonic Input-Convex Kolmogorov-Arnold Network (ICKAN) with the trainable parameter set \mathcal{Q} . The architecture of ICKAN is discussed in detail in Section 2.3. To model isotropic compressible hyperelasticity, we use deviatoric and volumetric invariants, i.e.,

$$\begin{aligned} \text{deviatoric invariants : } & \quad \bar{I}_1 = J^{-2/3} I_1, \quad I_2^* = (\bar{I}_2)^{3/2} = (J^{-4/3} I_2)^{3/2}, \\ \text{volumetric invariant : } & \quad J = \det(\mathbf{F}) = I_3^{1/2}, \end{aligned} \quad (7)$$

where

$$I_1 = \text{tr}(\mathbf{C}), \quad I_2 = \frac{1}{2} [\text{tr}(\mathbf{C})^2 - \text{tr}(\mathbf{C}^2)], \quad I_3 = \det(\mathbf{C}) \quad (8)$$

represent the principal invariants of the right Cauchy–Green deformation tensor $\mathbf{C} = \mathbf{F}^T \mathbf{F}$. W^0 is a constant scalar correction such that the energy density vanishes at zero deformation ($\mathbf{F} = \mathbf{I}$), i.e.,

$$W(\mathbf{I}) = 0 \quad \implies \quad W^0 = -W_Q^{\text{ICKAN}}(0, 0, 0). \quad (9)$$

Note that the energy correction W^0 is updated during each iteration of the training (discussed later in Section 3). While we consider hyperelastic models with invariants-based inputs here, models based on other inputs such as principal stretches are also possible; see [30].

The first Piola–Kirchhoff stress and tangent modulus are given by (from (1))

$$P_{ij}(\mathbf{F}) = \frac{\partial W_Q^{\text{ICKAN}}(K_1, K_2, K_3)}{\partial F_{ij}} \quad (10)$$

and

$$\mathbb{C}_{ijkl} = \frac{\partial P_{ij}(\mathbf{F})}{\partial F_{kl}} = \frac{\partial^2 W_Q^{\text{ICKAN}}(K_1, K_2, K_3)}{\partial F_{ij} \partial F_{kl}}, \quad (11)$$

respectively. Here, we use the Einstein index notation over the subscripts.

We now discuss how this ansatz satisfies the previously mentioned physical constraints.

- *Stress-free undeformed configuration:* The inputs (K_1, K_2, K_3) in (6) are the modified version of the invariants obtained by appropriately shifting – and in some cases squaring – such that both their values and their derivatives with respect to \mathbf{F} vanish in the undeformed state ($\mathbf{F} = \mathbf{I}$), i.e.,

$$\left. \frac{\partial K_1}{\partial \mathbf{F}} \right|_{\mathbf{F}=\mathbf{I}} = \mathbf{0}, \quad \left. \frac{\partial K_2}{\partial \mathbf{F}} \right|_{\mathbf{F}=\mathbf{I}} = \mathbf{0}, \quad \left. \frac{\partial K_3}{\partial \mathbf{F}} \right|_{\mathbf{F}=\mathbf{I}} = \mathbf{0}. \quad (12)$$

Consequently, the first Piola–Kirchhoff stress vanishes identically at undeformed state:

$$\mathbf{P}(\mathbf{I}) = \frac{\partial W_Q^{\text{ICKAN}}(0, 0, 0)}{\partial K_1} \left. \frac{\partial K_1}{\partial \mathbf{F}} \right|_{\mathbf{F}=\mathbf{I}} + \frac{\partial W_Q^{\text{ICKAN}}(0, 0, 0)}{\partial K_2} \left. \frac{\partial K_2}{\partial \mathbf{F}} \right|_{\mathbf{F}=\mathbf{I}} + \frac{\partial W_Q^{\text{ICKAN}}(0, 0, 0)}{\partial K_3} \left. \frac{\partial K_3}{\partial \mathbf{F}} \right|_{\mathbf{F}=\mathbf{I}} = \mathbf{0} \quad (13)$$

- *Objectivity:* Since the model ansatz (6) is a function of K_1 , K_2 , and K_3 which are in turn function of invariants of \mathbf{C} , it is objective by construction.
- *Polyconvexity:* The invariant \tilde{I}_1 and equivalently, K_1 are polyconvex in \mathbf{F} . $K_3 = (J - 1)^2$ is polyconvex in \mathbf{F} . We note that \tilde{I}_2 is not polyconvex in \mathbf{F} , as previously shown by Hartmann and Neff [62] and Klein et al. [63]. To this end, we use the modified second invariant I_2^* and equivalently K_2 , which is polyconvex in \mathbf{F} (see [62] for proof). In summary, the inputs (K_1, K_2, K_3) are polyconvex in \mathbf{F} . If W_Q^{ICKAN} is convex and monotonically non-decreasing in (K_1, K_2, K_3) , then the overall strain energy density $W(\mathbf{F})$ is polyconvex in \mathbf{F} [64]. In the subsequent section, we present the monotonic input-convex architecture for KANs to satisfy this physical constraint.

2.3. Monotonic input-convex Kolmogorov-Arnold network (ICKAN)

The model architecture of W_Q^{ICKAN} is based on the Kolmogorov-Arnold representation theorem, which states that any multivariate continuous function on a bounded domain can be expressed as a sum of univariate continuous functions. In its simplest form, a multivariate function $f(x_1, \dots, x_n)$ can be written as a sum of activations of univariate functions. KANs extend this principle to deeper and wider architectures, enabling the approximation of more complex multivariate functions. Fig. 1 presents a schematic of the ICKAN architecture, which is discussed in the following sections.

We consider a multi-layer KAN with R layers. For the r th layer of the KAN, let $\mathbf{z}^{(r)} \in \mathbb{R}^{n_r}$ be the (n_r) -dimensional output. For each layer, the output is obtained as [40]

$$\text{KAN layer:} \quad \mathbf{z}^{(r)} = \left[\sum_{j=1}^{n_{r-1}} \phi_{r-1,1,j} \left(z_j^{(r-1)} \right), \dots, \sum_{j=1}^{n_{r-1}} \phi_{r-1,n_r,j} \left(z_j^{(r-1)} \right) \right]^T, \quad (14)$$

where $\mathbf{z}^{(r-1)}$ is the output of the previous layer. Here, $\{\phi_{r-1,i,j}\}_{i=1}^{n_r}$ are n_r trainable univariate functions of the r th layer transforming the j th-dimension of the layer input, i.e., $z_j^{(r-1)}$. Summing all the activations $\phi_{r-1,i,j}(\cdot)$ over $j \in \{1, \dots, n_{r-1}\}$ yields the i th-dimension of the layer output, i.e., $z_i^{(r)}$. The choice of the univariate activations $\phi_{r-1,i,j}(\cdot)$ can be arbitrary. However, to ensure input-convexity, special considerations apply on the choice of $\phi_{r-1,i,j}(\cdot)$. For the following discussion, we omit the subscripts on $\phi(\cdot)$ for brevity, while implicitly assuming that all ϕ functions are independent and have their own trainable parameters.

To ensure monotonic input-convexity, we follow the principles that (i) non-negative sum of convex functions are convex, (ii) convex non-decreasing composition of a convex function is convex [47,65], (iii) non-negative sum of monotonically non-decreasing

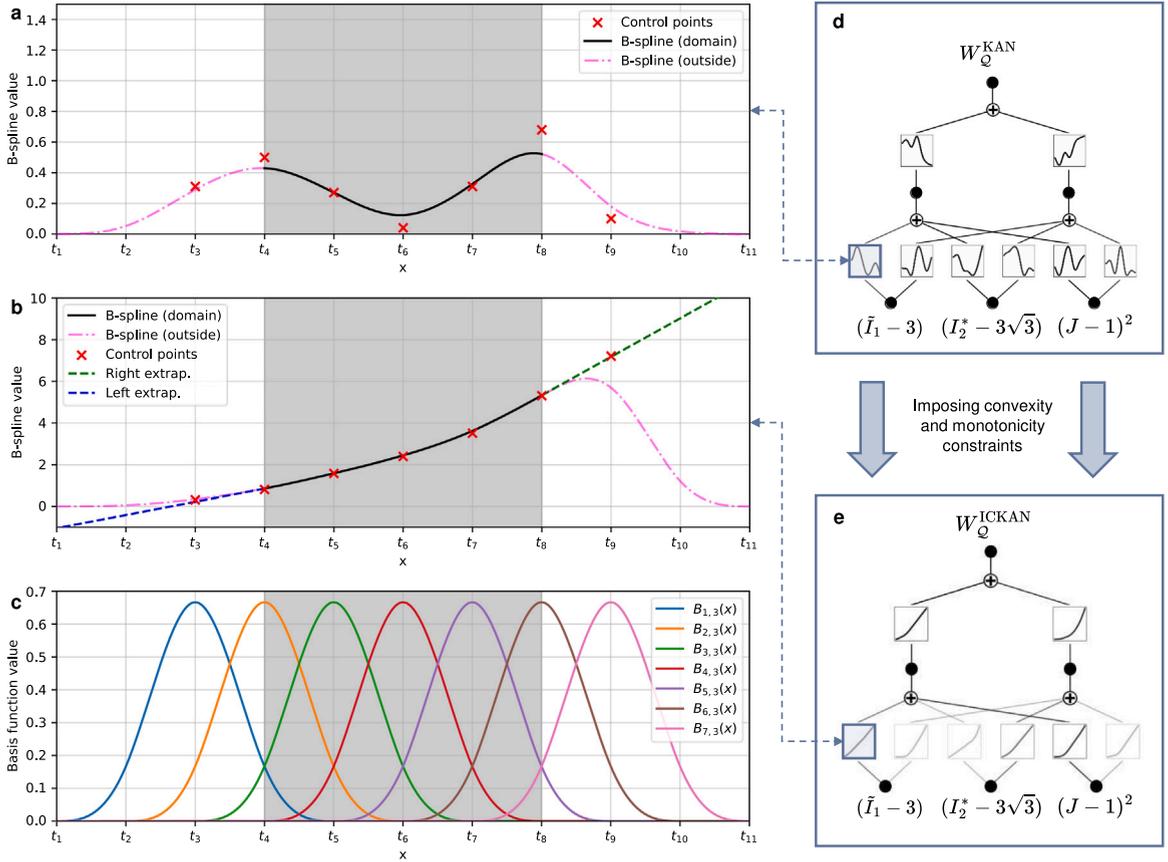


Fig. 1. Imposing monotonicity and convexity constraints on the trainable B-splines of W_Q^{ICKAN} . (a) A B-spline with unconstrained control points (red), illustrating how the spline curve is constructed. The natural domain of definition for the B-spline is indicated by the shaded gray area. (b) The resulting B-spline curve after enforcing both monotonicity and convexity constraints, with linear extrapolation applied at both endpoints to ensure consistent behavior beyond the spline domain. (c) Uniform B-spline basis functions of order $k = 3$. (d) Schematic representation of a vanilla KAN model using unconstrained trainable B-splines, where activation functions remain unrestricted. (e) The ICKAN model, incorporating convexity and monotonicity constraints on the trainable B-spline-based activation functions. These constraints ensure a physically consistent strain energy density. The insets at each activation node illustrate the univariate activation functions used within the respective networks.

functions is monotonically non-decreasing, and (iv) the composition of monotonically non-decreasing functions is monotonically non-decreasing. A sufficient condition to ensure monotonic input-convexity is that each KAN layer is convex and non-decreasing. Consequently, from (14), each univariate function ϕ must be convex and non-decreasing.

We use uniform B-splines as a choice for univariate activation function, i.e.,

$$\phi(x) = g(w_s)\psi(x) \tag{15}$$

for a scalar input x . Here, $w_s \in \mathbb{R}$ is a trainable scalar weight and $g : \mathbb{R} \rightarrow \mathbb{R}^+$ is a non-negative function. This ensures that as long as ψ is convex and non-decreasing, ϕ remains convex and non-decreasing, while allowing for a trainable scaling parameter to alter the magnitude of the activations. Here, we choose $g(\cdot) = \log(1 + \exp(\cdot))$, i.e., the softplus function. The trainable univariate function $\psi(x)$ is a convex and non-decreasing uniform B-spline of order k and a linear combination of n_b piecewise polynomial basis functions $B_{i,k}(x)$:

$$\psi(x) = \sum_{i=1}^{n_b} c_i B_{i,k}(x), \quad \text{with} \quad \sum_{i=1}^{n_b} B_{i,k}(x) = 1 \quad \text{for} \quad x \in [x_{\min}, x_{\max}]. \tag{16}$$

Here, c_i is the control point associated with the corresponding basis functions $B_{i,k}$. $[x_{\min}, x_{\max}]$ is natural definition domain of the B-spline curve where zeroth-order consistency is satisfied, i.e., constant functions can be represented exactly. The learning process involves training the control points $\{c_i\}_{i=1}^{n_b}$ (along with the scaling weight w_s), for each univariate function $\phi(\cdot)$ (recall we dropped the subscripts for brevity).

To define the k th-order B-spline basis functions (illustrated in Fig. 1c) for each ϕ , we consider a set of $m_b = (k + n_b + 1)$ knots: $\{t_i\}_{i=1}^{m_b}$ and recursively use the algorithm introduced by De Boor [66] as follows:

Zero-order basis function ($k = 0$):

$$B_{i,0}(x) = \begin{cases} 1, & \text{if } t_i \leq x < t_{i+1}, \\ 0, & \text{otherwise.} \end{cases} \quad (17)$$

Recursive definition for higher orders ($k > 0$):

$$B_{i,k}(x) = \frac{x - t_i}{t_{i+k} - t_i} B_{i,k-1}(x) + \frac{t_{i+k+1} - x}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(x). \quad (18)$$

For the special case of a uniform B-spline, the knots are equally spaced, i.e.,

$$t_{i+2} - t_{i+1} = t_{i+1} - t_i, \quad \forall i \in [1, m_b - 2]. \quad (19)$$

We choose the knots by uniformly discretizing a range of pre-defined (as a hyperparameter) univariate inputs of $\phi(x)$.

For a uniform B-spline to be convex and monotonically non-decreasing, the control points must satisfy the following condition:

$$c_{i+2} - c_{i+1} \geq c_{i+1} - c_i \geq 0, \quad \forall i \in [1, n_b - 2]. \quad (20)$$

The proof is provided in [Appendix A.1](#). We note that the convexity and non-decreasing monotonicity of a uniform B-spline curve depends entirely on the control points and is independent of the knot spacing. This result follows from the convex hull property of B-splines: since the B-spline is contained in the convex hull of its control points, the control points must form a convex set and be non-decreasing for the B-spline curve. The implementation of the convexity and non-decreasing constraints are outlined in [Appendix A.2](#).

The construction in (20) only guarantees convexity and non-decreasing monotonicity in the natural definition domain of the B-spline. The range of the knots are chosen to be sufficiently large for any reasonable data distribution relevant to hyperelasticity. Nevertheless, for extreme input data (e.g., exceptionally large strains), we extend the B-spline beyond its natural domain by maintaining a constant slope at its endpoints to preserve these properties; see [Fig. 1b](#) for a schematic and [Appendix A.3](#) for implementation details. To ensure that the spline domain encompasses the expected input range, we set the knots' range during initialization. This update is done only once and remains fixed throughout training and inference (more details in [Appendix B](#)).

In summary, the ICKAN architecture for isotropic compressible hyperelasticity can be described as follows:

$$\text{Input layer : } F, \quad (21a)$$

$$\text{Invariants layer : } \mathbf{z}^{(0)} = [K_1 = J^{-2/3} I_1 - 3, K_2 = (J^{-4/3} I_2)^{3/2} - 3\sqrt{3}, K_3 = (J - 1)^2]^T, \quad (21b)$$

$$\text{First KAN layer : } \mathbf{z}^{(1)} = \left[\sum_{j=1}^{n_0} \phi_{0,1,j}(z_j^{(0)}), \dots, \sum_{j=1}^{n_0} \phi_{0,n_1,j}(z_j^{(0)}) \right]^T, \quad (21c)$$

⋮

$$r\text{th KAN layer : } \mathbf{z}^{(r)} = \left[\sum_{j=1}^{n_{r-1}} \phi_{r-1,1,j}(z_j^{(r-1)}), \dots, \sum_{j=1}^{n_{r-1}} \phi_{r-1,n_r,j}(z_j^{(r-1)}) \right]^T, \quad (21d)$$

⋮

$$\text{Output layer : } W_Q^{\text{ICKAN}} = \mathbf{z}^{(R)} = \sum_{j=1}^{n_{R-1}} \phi_{R-1,1,j}(z_j^{(R-1)}) \quad (21e)$$

$$\text{Trainable parameters : } Q = \left\{ \{c_k^{(r,i,j)} : k = 1, \dots, n_b\}, \{w_s^{(r,i,j)} : r = 1, \dots, R; i = 1, \dots, n_r; j = 1, \dots, n_{r-1}\} \right\}. \quad (21f)$$

[Table C.2](#) provides a summary of all the architectural hyperparameters.

3. Unsupervised training of ICKANs

Instead of training the ICKAN model on labeled stress–strain pairs, we train the model on full-field displacement/strain-fields and global force data which are realistically available through mechanical testing and DIC techniques. To achieve this, we extend our previously introduced NN-based constitutive modeling framework, NN-EUCLID [48], to ICKANs. Here, we provide a brief review of NN-EUCLID in the context of ICKANs. [Fig. 2](#) illustrates a schematic overview of the ICKAN and its unsupervised training framework.

Consider a hyperelastic specimen undergoing quasi-static deformation in a two-dimensional reference domain $\Omega \subset \mathbb{R}^2$. The specimen is designed with a complex geometry (for example, a plate with a hole) to induce diverse and heterogeneous strain states, while the material itself is assumed to be homogeneous and isotropic. Without loss of generality, we assume plane strain conditions.

Boundary conditions are applied such that Dirichlet conditions are enforced on $\partial\Omega_u \subset \partial\Omega$ and Neumann conditions on the remainder, $\partial\Omega_t = \partial\Omega \setminus \partial\Omega_u$. For simplicity, our analysis focuses on displacement-controlled loading (i.e., Dirichlet boundary conditions), while noting that applied forces in load-controlled scenarios are equivalent to reaction forces under displacement control. The dataset comprises of n_t snapshots of displacement measurements,

$$\mathcal{U} = \{u^{a,t} \in \mathbb{R}^2 : a = 1, \dots, n_n; t = 1, \dots, n_t\}, \quad (22)$$

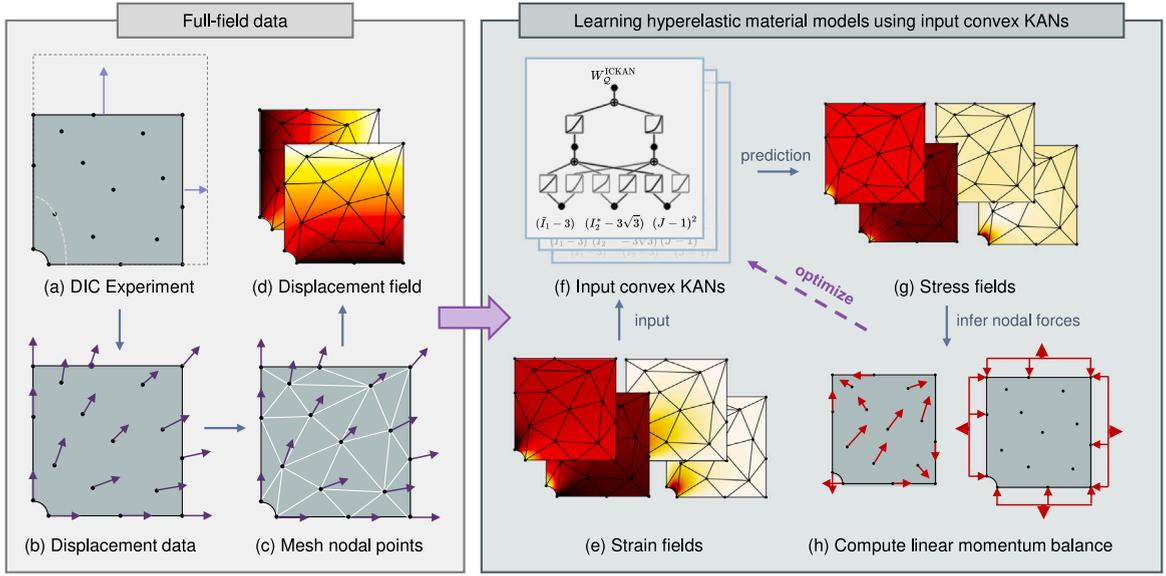


Fig. 2. Schematic overview of the unsupervised deep learning approach for hyperelastic constitutive modeling using ICKANs. In (a) and (b), pointwise measurements of displacements and reaction forces are acquired from a hyperelastic specimen under quasi-static deformation using a DIC setup. These data, together with a finite element mesh of the domain (c), are used to reconstruct continuous displacement and strain fields (d,e). The physics-consistent ICKAN-based constitutive model (f) then predicts the corresponding stress fields at the quadrature points of each element (g) given the strains. The computed stress fields serve to calculate both the internal and external nodal forces (h). By employing the weak form of the conservation of linear momentum, the residual forces are minimized—applied pointwise for free degrees of freedom and in an aggregated manner for fixed degrees of freedom corresponding to each set of Dirichlet constraints with measured reaction forces. This iterative optimization process refines the parameters of the ICKAN-based constitutive models.

recorded at n_n reference points

$$\mathcal{X} = \{ \mathbf{X}^a \in \Omega : a = 1, \dots, n_n \}. \quad (23)$$

Additionally, for each snapshot, n_β reaction forces

$$\{ \mathbf{R}^{\beta,t} : \beta = 1, \dots, n_\beta; t = 1, \dots, n_t \} \quad (24)$$

are measured at selected Dirichlet boundaries (e.g., using load cells). For brevity, the superscript $(\cdot)^t$ is omitted in the subsequent discussion, although the numerical procedure is applied independently to every snapshot.

Given these limited measurements, the primary goal is to infer the constitutive model $W(\mathbf{F})$ governing the stress–strain response.

To reconstruct the displacement field, the reference domain \mathcal{X} is discretized using linear triangular elements (each with a single quadrature point at its barycenter), yielding the approximation

$$\mathbf{u}(\mathbf{X}) = \sum_{a=1}^{n_n} N^a(\mathbf{X}) \mathbf{u}^a. \quad (25)$$

Here, $N^a : \Omega \rightarrow \mathbb{R}$ represents the shape function associated with node \mathbf{X}^a . The corresponding deformation gradient field is approximated as

$$\mathbf{F}(\mathbf{X}) = \mathbf{I} + \sum_{a=1}^{n_n} \mathbf{u}^a \otimes \nabla N^a(\mathbf{X}), \quad (26)$$

where \mathbf{I} is the identity matrix, and ∇ denotes the gradient operator with respect to the reference coordinates.

As outlined in [11,48], we leverage the conservation of linear momentum to guide the learning of the constitutive model, which eliminates the need of stress labels. Assuming quasi-static loading and negligible body forces, the weak form of the linear momentum balance in the reference domain Ω is given by

$$\int_{\Omega} \mathbf{P} : \nabla \mathbf{v} \, dV - \int_{\partial\Omega_t} \hat{\mathbf{t}} \cdot \mathbf{v} \, dS = 0 \quad \forall \text{admissible } \mathbf{v}, \quad (27)$$

where $\hat{\mathbf{t}}$ denotes the prescribed tractions and \mathbf{v} is a test function that vanishes on the Dirichlet boundary $\partial\Omega_d$. The weak formulation is preferred over the strong form, as it avoids the need for second derivatives, which are sensitive to noise.

Let $\mathcal{D} = \{(a, i) : a = 1, \dots, n_n; i = 1, 2\}$ denote the set of displacement degrees of freedom, which we partition into:

- $\mathcal{D}^{\text{free}}$: unconstrained degrees of freedom,

- $\mathcal{D}_\beta^{\text{fix}}$ (with $\beta = 1, \dots, n_\beta$): degrees of freedom under Dirichlet constraints that contribute to the observed reaction force R^β .

Approximating the test function by

$$\mathbf{v}(\mathbf{X}) = \sum_{a=1}^{n_n} N^a(\mathbf{X}) \mathbf{v}^a, \quad \text{with} \quad v_i^a = 0 \quad \forall (a, i) \in \bigcup_{\beta=1}^{n_\beta} \mathcal{D}_\beta^{\text{fix}}, \quad (28)$$

the weak form (27) reduces to

$$\sum_{a=1}^{n_n} v_i^a f_i^a = 0, \quad \text{with} \quad f_i^a = \underbrace{\int_{\Omega} P_{ij} \nabla_j N^a \, dV}_{\text{internal force}} - \underbrace{\int_{\partial\Omega_t} \hat{t}_i N^a \, dS}_{\text{external force}}. \quad (29)$$

Here, the integrals are computed using numerical quadrature over the same discretization and mesh as (25). Since the test functions are arbitrary, the force residual must vanish at every free degree of freedom,

$$f_i^a = 0 \quad \forall (a, i) \in \mathcal{D}^{\text{free}}. \quad (30)$$

At the fixed degrees of freedom, the internal and external forces are counteracted by the reaction force imposed by the Dirichlet constraints. Since point-wise reaction forces cannot be measured experimentally, they are assumed to be unavailable. Instead, only the total reaction forces integrated over the boundary segments are known. As a result, the global force balance for each measured reaction force is expressed as

$$\sum_{(a,i) \in \mathcal{D}_\beta^{\text{fix}}} f_i^a = R^\beta \quad \forall \quad \beta = 1, \dots, n_\beta, \quad (31)$$

where the summation is carried out over all point-wise forces associated with the degrees of freedom in the β th Dirichlet constraint, denoted as $\mathcal{D}_\beta^{\text{fix}}$. As noted earlier, the superscript $(\cdot)^t$ has been omitted for brevity, but the above force balance conditions apply to all data snapshots at $t = 1, \dots, n_t$.

Our goal is then to learn the constitutive model $W(\mathbf{F})$ in (6) (with $\mathbf{P}(\mathbf{F})$ given by (10)), now parameterized by the monotonic input-convex Kolmogorov–Arnold network W_Q^{ICKAN} , such that the displacement and reaction force data satisfy the physics-based constraints (30) and (31). We formulate the inverse problem as the minimization of the force balance residuals [48] with respect to the trainable parameters of the ICKAN:

$$Q \leftarrow \arg \min_Q \sum_{t=1}^{n_t} \left[\underbrace{\sum_{(a,i) \in \mathcal{D}^{\text{free}}} (f_i^{a,t})^2}_{\text{free degrees of freedom}} + \sum_{\beta=1}^{n_\beta} \underbrace{\left(R^{\beta,t} - \sum_{(a,i) \in \mathcal{D}_\beta^{\text{fix}}} f_i^{a,t} \right)^2}_{\text{fixed degrees of freedom}} \right]. \quad (32)$$

This optimization is performed via gradient-based minimization, enabled by automatic differentiation (see Table C.2 for details on the optimizer and related hyperparameters).

4. Results

4.1. Numerical benchmarks

To evaluate the ICKAN-based constitutive models, we use the dataset provided by Thakolkaran et al. [48], which emulates a digital image correlation (DIC) experiment by simulating a hyperelastic square plate with a hole under plane-strain conditions using the finite element method (FEM); see Fig. 3a for schematic. The specimen is subjected to displacement-controlled asymmetric biaxial tension, with symmetry enforced along the bottom and left boundaries and loading parameter δ . This configuration yields diverse and heterogeneous strain states, making the data ideal for training a generalizable constitutive model from a single experiment; hence, the specimen is termed the *training specimen*.

The dataset includes synthetic data corresponding to several well-established material models:

1. Neo-Hookean (NH) model:

$$W(\mathbf{F}) = 0.5(\tilde{I}_1 - 3) + 1.5(J - 1)^2. \quad (33)$$

2. Isihara (IH) model [67]:

$$W(\mathbf{F}) = 0.5(\tilde{I}_1 - 3) + (\tilde{I}_2 - 3) + (\tilde{I}_1 - 3)^2 + 1.5(J - 1)^2. \quad (34)$$

3. Haines-Wilson (HW) model [68]:

$$W(\mathbf{F}) = 0.5(\tilde{I}_1 - 3) + (\tilde{I}_2 - 3) + 0.7(\tilde{I}_1 - 3)(\tilde{I}_2 - 3) + 0.2(\tilde{I}_1 - 3)^3 + 1.5(J - 1)^2. \quad (35)$$

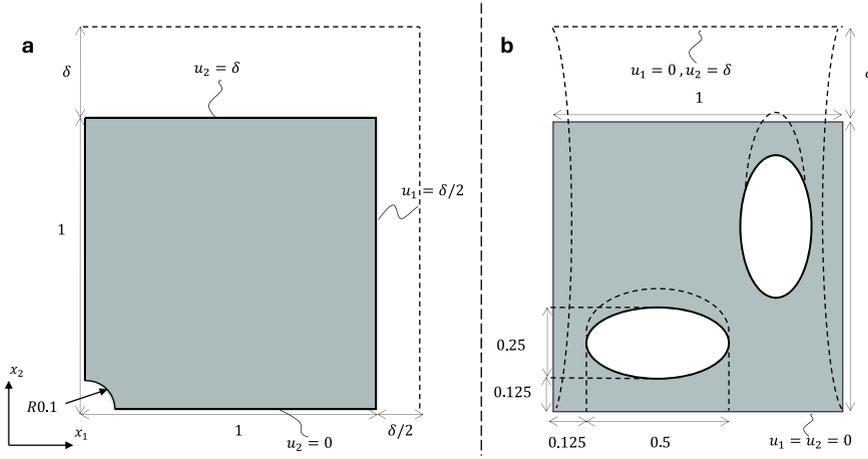


Fig. 3. (a) *Training specimen*: A square plate with a hole in the bottom-left corner is subjected to displacement-controlled asymmetric biaxial tension. The resulting dataset (noisy full-field displacements and reaction forces) is used to train the ICKAN-based constitutive models. (b) *Validation specimen*: A square plate containing two asymmetric elliptical holes is subjected to displacement-controlled uniaxial tension. This geometry is employed solely for validation, and no data from these experiments enter the training process. All lengths and displacements are normalized by the side length of the undeformed specimen. The parameters used to generate the data can be found in [Table C.2](#).

Source: Figures adapted from [48].

4. Gent-Thomas (GT) model [69]:

$$W(\mathbf{F}) = 0.5(\bar{I}_1 - 3) + \log(\bar{I}_2/3) + 1.5(J - 1)^2. \quad (36)$$

5. Arruda-Boyce (AB) model [70]:

$$W(\mathbf{F}) = 2.5\sqrt{N_c} \left[\beta_c \lambda_c - \sqrt{N_c} \log \left(\frac{\sinh \beta_c}{\beta_c} \right) \right] - c_{AB} + 1.5(J - 1)^2, \quad (37)$$

where $\lambda_c = \sqrt{\bar{I}_1/3}$, $\beta_c = \mathcal{L}^{-1}(\lambda_c/\sqrt{N_c})$, and \mathcal{L}^{-1} denotes the inverse Langevin function. Here, $N_c = 28$ represents the number of polymeric chain segments, and $c_{AB} \approx 3.7910$ offsets the energy density to zero at $\mathbf{F} = \mathbf{I}$, since the Arruda-Boyce formulation does not inherently satisfy this condition.

6. Ogden (OG) model [71]:

$$W(\mathbf{F}) = \frac{\mu}{\eta} (\bar{\lambda}_1^\eta + \bar{\lambda}_2^\eta + \bar{\lambda}_3^\eta - 3) + 1.5(J - 1)^2, \quad \text{with } \bar{\lambda}_k = J^{-1/3} \lambda_k, \quad k = 1, 2, 3, \quad (38)$$

where $\lambda_1, \lambda_2, \lambda_3$ are the principal stretches and $\mu = \eta = 1.3$.

Note that the models given in Eqs. (34)–(36) are not polyconvex, as they involve the modified invariant \bar{I}_2 , which does not fulfill ellipticity. For discussions on the polyconvexity of the Ogden model, we refer the reader to the work of Hartmann and Neff [62]. These models are included as they are commonly used in literature and provide a representative benchmark for evaluating constitutive modeling approaches. Additionally, without loss of generality, all physical quantities in the material models are considered as non-dimensionalized.

For benchmarking purposes, the ICKAN-based model is trained to approximate these material laws using the NN-EUCLID loss described in (32). Additional details regarding hyperparameters and the training algorithm are provided in [Appendix C](#).

Considering that real-world DIC measurements contain noise, artificial noise has been added to the displacement data. The noise level, determined by the pixel resolution of the imaging system, remains constant for each degree of freedom across all load steps. Specifically, the displacement data is modeled as

$$u_i^{a,t} = u_i^{\text{fem},a,t} + \varepsilon_i^{a,t}, \quad \text{with } \varepsilon_i^{a,t} \sim \mathcal{N}(0, \sigma_u^2) \quad \forall (a, i) \in D, \quad t \in 1, \dots, n_t. \quad (39)$$

Here, $u_i^{\text{fem},a,t}$ represents the FEM-computed displacement, while $\varepsilon_i^{a,t}$ is sampled from a normal distribution with zero mean and standard deviation σ_u . Following [11,12,48], two noise levels are considered (normalized by specimen length): $\sigma_u = 10^{-4}$ (low noise) and $\sigma_u = 10^{-3}$ (high noise), which emulate the noise levels representative of modern DIC setups.

We use pre-processed synthetic datasets from Thakolkaran et al. [48], where the displacement fields were already denoised following standard DIC post-processing routines. As described in that work, kernel ridge regression (KRR) was used for spatial denoising. To enhance data efficiency, the denoised displacements on the high-resolution mesh with 63,601 nodes are projected onto a coarser mesh with $n_n = 1441$ nodes, which serve as the training dataset for the ICKAN models.

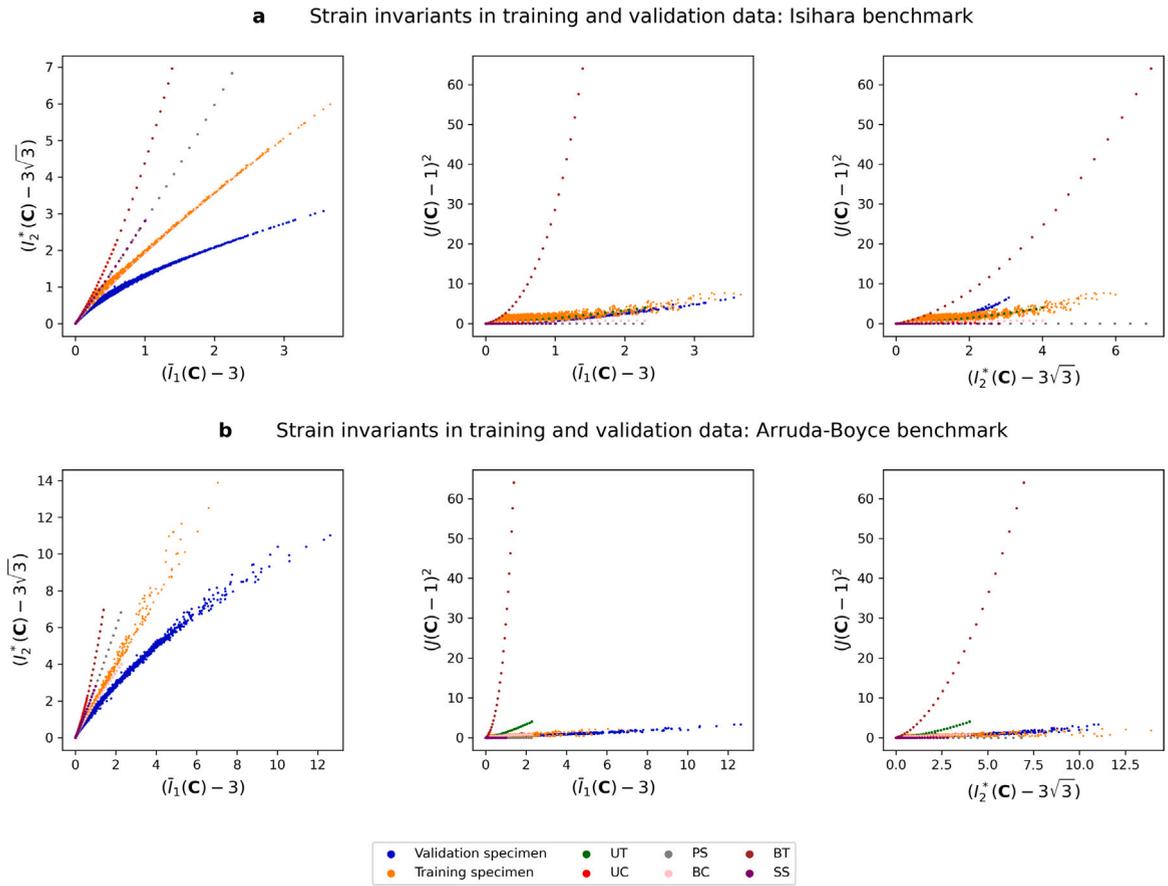


Fig. 4. Two-dimensional projections of the strain invariants $(\bar{I}_1 - 3)$, $(I_2^* - 3\sqrt{3})$, and $(J - 1)^2$ for all elements in the training specimen (Fig. 3a) across all load steps, shown for (a) Arruda-Boyce (37) and (b) Isihara (34) benchmarks. The figure also includes the strain invariants of the validation specimen (Fig. 3b) across all elements and load steps, along with those corresponding to the six evaluation deformation paths (40). All data presented are derived from the ground-truth model without noise, including the strain invariants of the validation specimen.

4.2. Accuracy and generalizability beyond the training data

We train n_e ICKAN models with identical architectures, each trained independently with different random initializations to mitigate bad local minima in the optimized model parameters. We choose the model whose final loss (as defined in (32)) is the lowest loss in the ensemble; the others are discarded.

For each benchmark problem (Eqs. (33)–(38)), the performance of the ICKAN-based constitutive models is evaluated against the corresponding ground truth along six deformation paths:

$$\begin{aligned}
 \mathbf{F}^{\text{UT}}(\gamma) &= \begin{bmatrix} 1 + \gamma & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, & \mathbf{F}^{\text{UC}}(\gamma) &= \begin{bmatrix} \frac{1}{1 + \gamma} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, & \mathbf{F}^{\text{BT}}(\gamma) &= \begin{bmatrix} 1 + \gamma & 0 & 0 \\ 0 & 1 + \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\
 \mathbf{F}^{\text{BC}}(\gamma) &= \begin{bmatrix} \frac{1}{1 + \gamma} & 0 & 0 \\ 0 & \frac{1}{1 + \gamma} & 0 \\ 0 & 0 & 1 \end{bmatrix}, & \mathbf{F}^{\text{SS}}(\gamma) &= \begin{bmatrix} 1 & \gamma & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, & \mathbf{F}^{\text{PS}}(\gamma) &= \begin{bmatrix} 1 + \gamma & 0 & 0 \\ 0 & \frac{1}{1 + \gamma} & 0 \\ 0 & 0 & 1 \end{bmatrix},
 \end{aligned} \tag{40}$$

where $\gamma \in [0, 2]$ denotes the loading parameter for UT, BT, and BC, and $\gamma \in [0, 1]$ for UC, SS, and PS (as these three cases already involve significant extrapolation; as shown in Fig. 4). These deformation paths are used solely for evaluation and not for training. The abbreviations used are as follows — UT: uniaxial tension, UC: uniaxial compression, BT: biaxial tension, BC: biaxial compression, SS: simple shear, PS: pure shear. Note that, while the uniaxial tension load case represents a uniaxial strain state, it induces a multiaxial stress state.

Figs. 5–6 and 9–10 (low noise) and Figs. 7–8 and 11–12 (high noise) confirm that both the ICKAN-based strain energy density as well as corresponding stress predictions closely match the ground truth across all benchmarks, deformation paths, and noise levels. This verifies the generalizability of the trained constitutive models. We observe that in some cases the predictions start to deteriorate

at larger strains. However, the issue can be addressed by conducting additional experiments (and include them as training data) or using specimens with a larger diversity in strain states to improve generalizability.

We highlight that the monotonic input-convex architecture of KANs guarantees learning physically admissible models. In [Appendix D](#), we demonstrate the performance of vanilla KANs without monotonicity and input-convexity. We observe poor prediction accuracies for both strain energy densities and stresses relative to their monotonic input-convex variants. In addition, we observe non-physical behaviors such as negative stiffness and fictitious material instabilities without monotonicity and input-convexity in the KAN architecture.

We further validate the ICKAN-based constitutive models by deploying them within a finite element simulation framework. Linear triangular elements and a Newton–Raphson nonlinear solver are employed, with element-level stress and tangent modulus computed via automatic differentiation from Eqs. (10) and (11). Since tangent stiffness calculations require the energy density to be at least twice differentiable, the splines must have a minimum order of $k = 3$ to ensure adequate smoothness. In this work, we use splines of order $k = 5$. These simulations were carried out using a custom in-house finite element solver implemented in Python. To assess generalization beyond the training domain, a validation specimen with a more complex geometry—featuring two asymmetric elliptical holes and subjected to quasi-static uniaxial loading—is considered (see [Fig. 3b](#)). The lowest-loss ICKAN models (for each material model, respectively), selected from the ensemble as described above and trained on high-noise data, are used in these simulations. As shown in [Fig. 13](#), the finite element solutions for two representative ground-truth models (Isihara (34) and Arruda-Boyce (37), respectively) exhibit excellent agreement with those obtained using the ICKAN-based models. This is a strong indication that the ICKAN approach is not overfitting to the training data but instead captures the underlying constitutive behavior in a generalizable form. Quantitatively, this is supported by high goodness-of-fit R^2 scores: greater than 0.99 for Isihara and greater than 0.95 for Arruda-Boyce in the high-noise case, for element-wise strain invariants and close matching of reaction forces. Remarkably, the strain states of this validation specimen are completely different from that of the training specimen — as shown in [Fig. 4](#); yet the ICKANs show excellent generalization capability to these unseen strain states. In [Appendix E](#), we also provide a comparative analysis of the ICKAN framework with the EUCLID approach of Flaschel et al. [11].

4.3. Interpretability

One advantage of ICKANs over ICNNs and other black-box neural networks is the enhanced interpretability they offer in constitutive modeling. This interpretability manifests in two key aspects.

(i) Due to the highly nonlinear and expressive nature of the activations – enabled by trainable splines – only a few layers with a small number of dimensions are required in KANs. Consequently, each activation function in ICKANs can be plotted and interpreted individually; see the insets of [Fig. 1d](#) and [e](#). In contrast, ICNNs and similar classical models rely on less expressive nonlinear activations (e.g., rectified linear unit or softplus), which necessitate many more linear layers with higher dimensionality to achieve comparable performance. This results in a significantly larger number of parameters and reduced interpretability.

(ii) The Kolmogorov–Arnold representation allows the network architecture to be decomposed into univariate activation functions. This decomposition enables the extraction of analytical expressions at each activation node using symbolic regression, which can then be sequentially assembled across the entire network to form an analytical expression for the entire constitutive model. Such a univariate decomposition is not possible with ICNNs and similar models, thereby limiting their capacity for symbolic interpretation.

To quantitatively demonstrate the latter aspect, we extract the symbolic expression of the strain energy density from the trained ICKAN model via symbolic regression at each activation node of the network independently. These expressions are derived using a small library of univariate symbolic functions: $\{x, \exp(x), \log(1 + \exp(x)), \log(1 + \exp(x))^2, \log(1 + \exp(x))^3, \log(1 + \exp(x))^4\}$. Each candidate function in the library is purposefully chosen to be convex and non-decreasing. In [Appendix F](#), we introduce a monotonic input-convex symbolic regression strategy that ensures that the symbolic approximation of the general ICKAN-based hyperelastic model remains polyconvex.

[Table 1](#) presents the symbolic expressions of the strain energy densities obtained from the ICKANs for both low- and high-noise cases. We overlay the predictions of the symbolic models in each plot of [Figs. 5–12](#) for comparison with the true and ICKAN-based strain energy density and stress. In general, the symbolic regression models closely match the ICKAN predictions, and in some cases, they even provide a slightly better fit at higher strains. Implementation details of the monotonic input-convex symbolic regression are provided in [Appendix F](#). *Limitations of symbolic interpretation:* Expectedly, in a few cases, the symbolic models show a deterioration in the prediction of strain energy densities and stresses relative to the trained ICKAN models. This can be attributed to the limited expressivity of the chosen function library for symbolic regression and the compounding errors from fitting a symbolic expression to already fitted splines of the ICKAN.

5. Conclusion

We introduced monotonic Input-Convex Kolmogorov–Arnold Networks (ICKANs) for learning hyperelastic constitutive laws. By integrating convexity and monotonicity constraints directly into the KAN architecture, our approach guarantees that the predicted strain energy density satisfies polyconvexity. At the same time, the use of trainable spline-based activation functions allows for a compact and interpretable representation of the constitutive model. We further demonstrated that ICKANs can be trained in an unsupervised manner on realistically measurable data in the form of strain fields and global reaction forces.

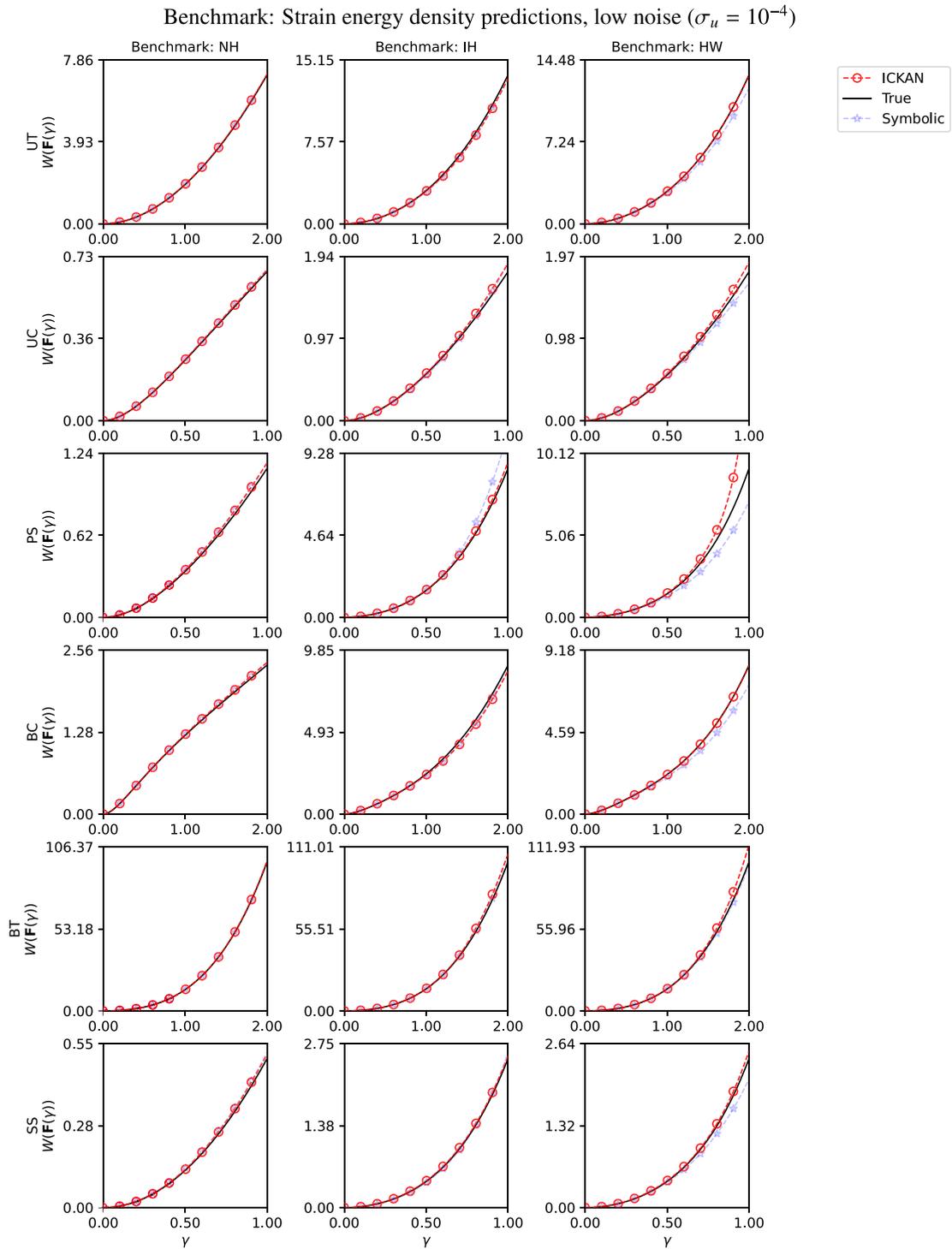


Fig. 5. Strain energy density $W(F(\gamma))$ predictions for the deformation paths in (40), evaluated under low noise ($\sigma_u = 10^{-4}$). Results are shown for the best ICKAN-based constitutive model in the ensemble, the model obtained through symbolic regression of the ICKAN, as well as for the hidden true model, for the NH (33), IH (34), and HW (35) benchmarks.

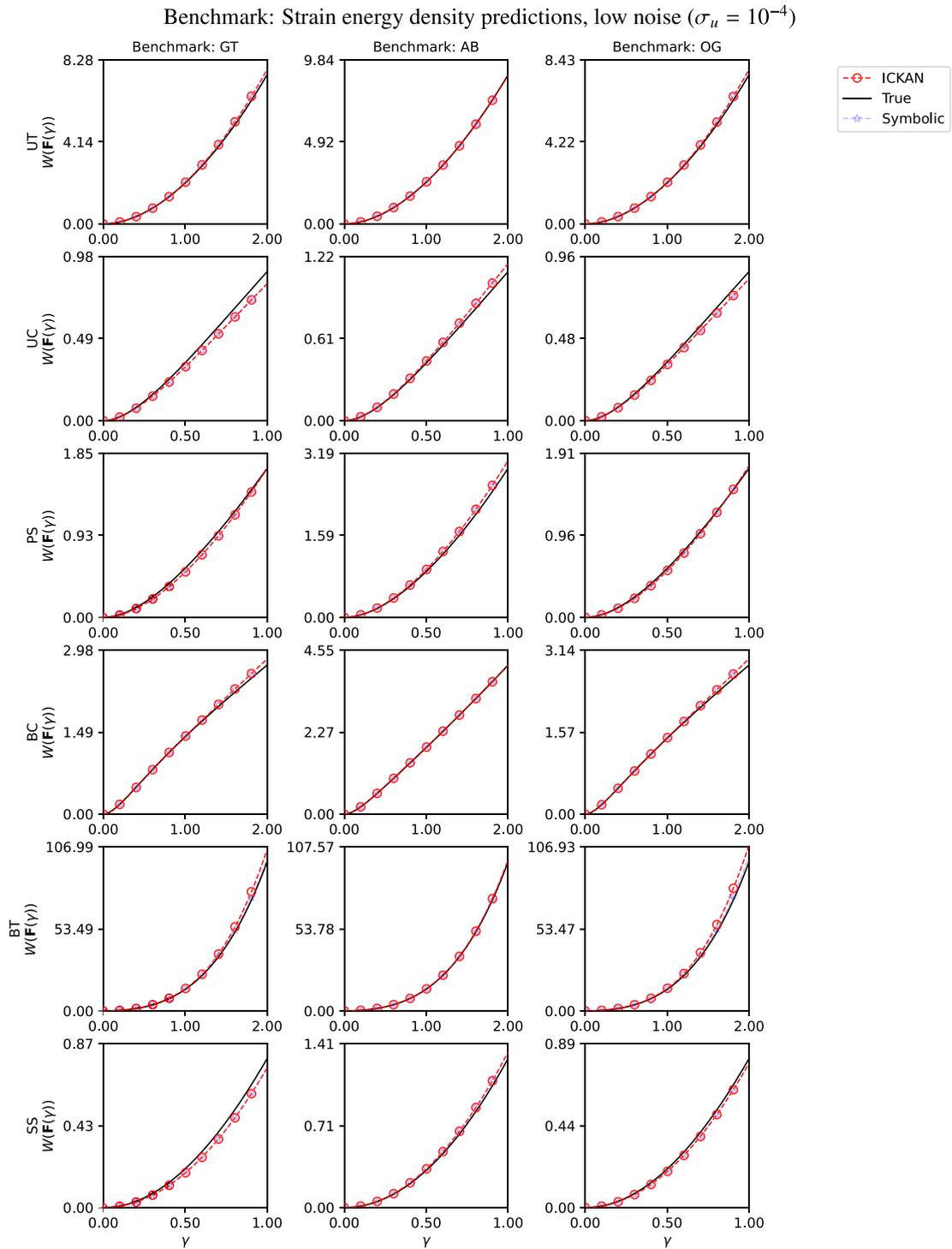


Fig. 6. Strain energy density $W(F(\gamma))$ predictions for the deformation paths in (40), evaluated under low noise ($\sigma_u = 10^{-4}$). Results are shown for the best ICKAN-based constitutive model in the ensemble, the model obtained through symbolic regression of the ICKAN, as well as for the hidden true model, for the GT (36), AB (37), and OG (38) benchmarks.

Benchmark: Strain energy density predictions, high noise ($\sigma_u = 10^{-3}$)

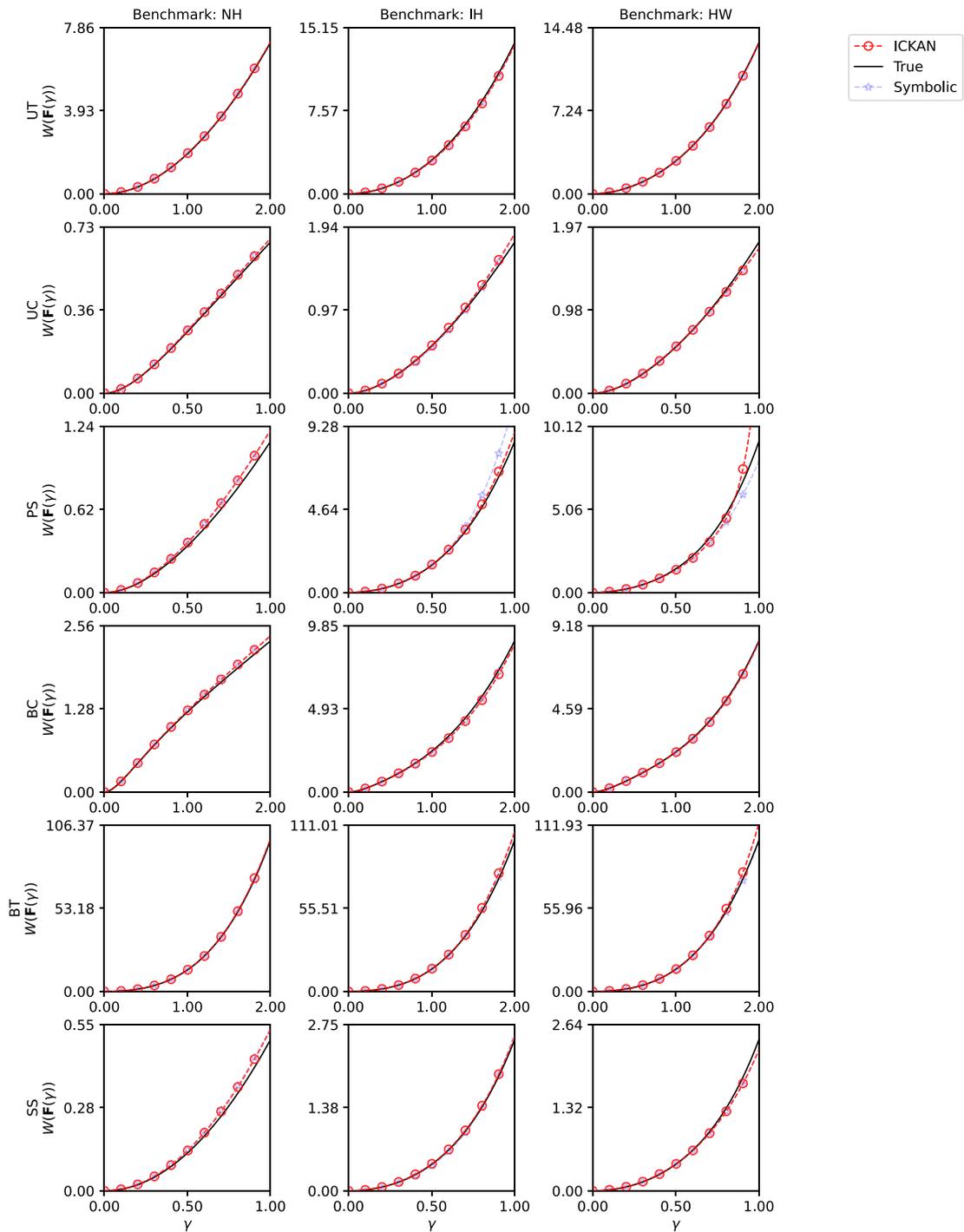


Fig. 7. Strain energy density $W(F(\gamma))$ predictions for the deformation paths in (40), evaluated under high noise ($\sigma_u = 10^{-3}$). Results are shown for the best ICKAN-based constitutive model in the ensemble, the model obtained through symbolic regression of the ICKAN, as well as for the hidden true model, for the NH (33), IH (34), and HW (35) benchmarks.

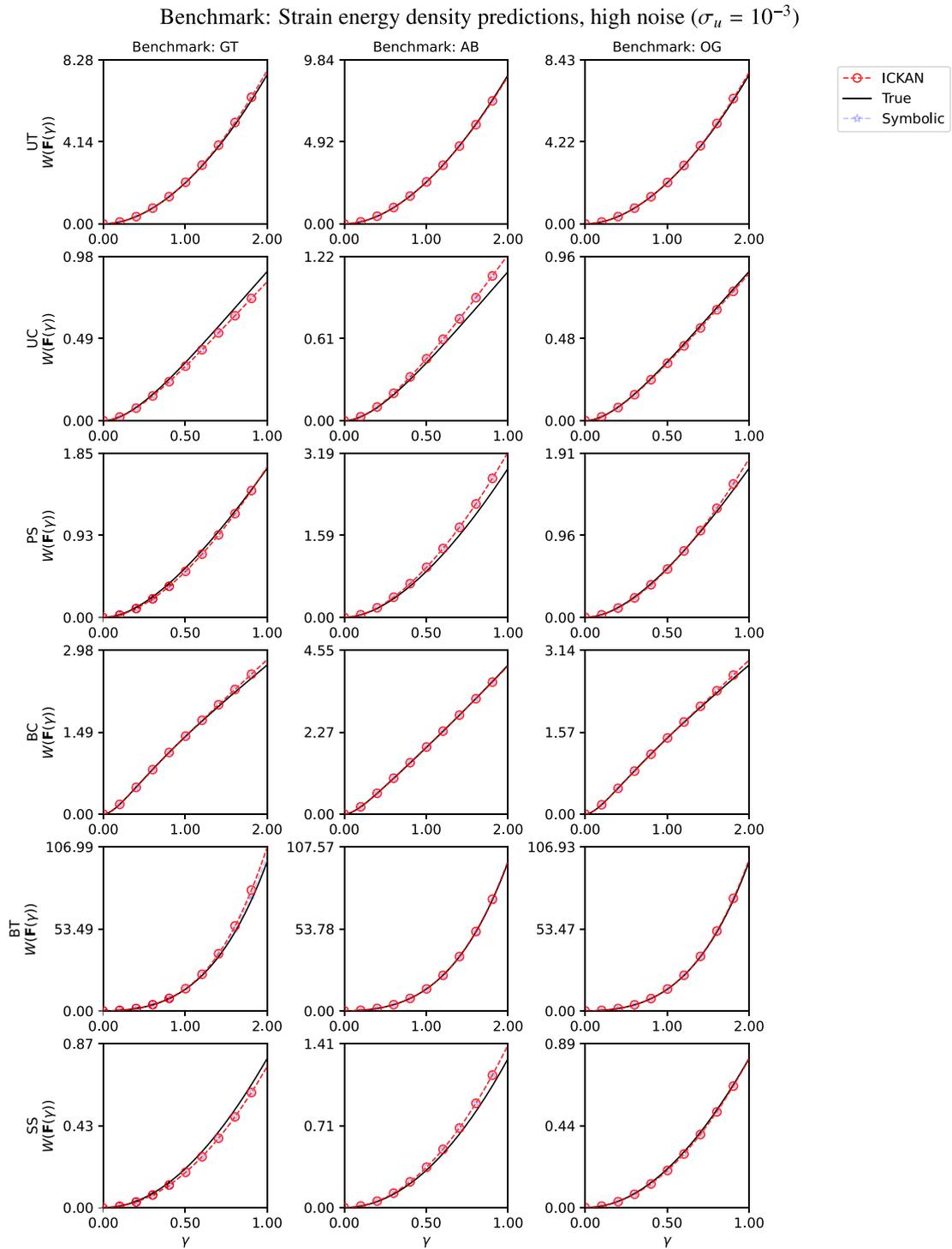


Fig. 8. Strain energy density $W(F(\gamma))$ predictions for the deformation paths in (40), evaluated under high noise ($\sigma_u = 10^{-3}$). Results are shown for the best ICKAN-based constitutive model in the ensemble, the model obtained through symbolic regression of the ICKAN, as well as for the hidden true model, for the GT (36), AB (37), and OG (38) benchmarks.

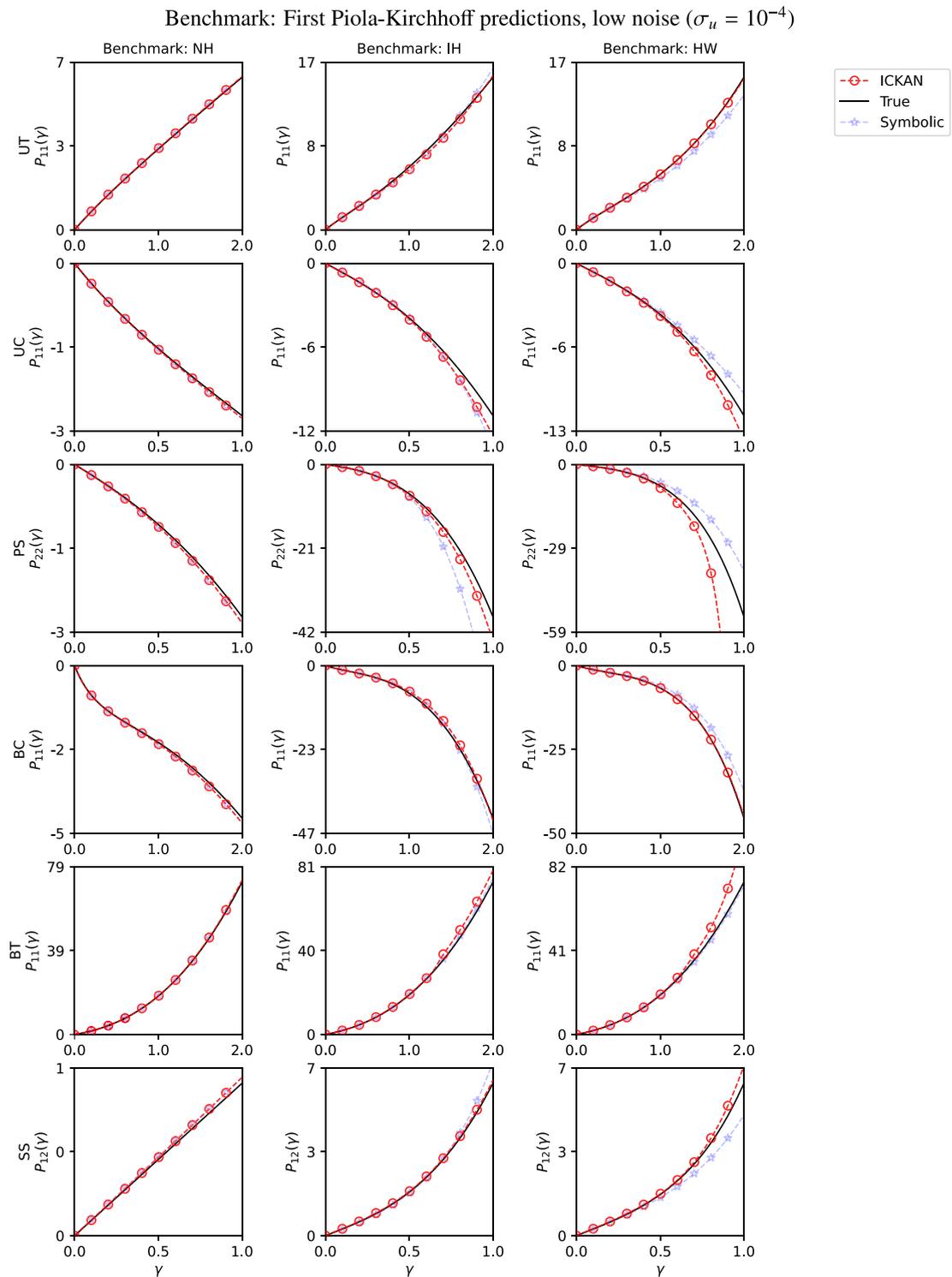


Fig. 9. Predicted first Piola-Kirchhoff stress $P(F(\gamma))$ components along the deformation paths specified in (40) for the low noise case ($\sigma_u = 10^{-4}$). The best ICKAN-based constitutive model is shown for the NH (33), IH (34), and HW (35) benchmarks, the model obtained through symbolic regression of the ICKAN, alongside the response of the true (hidden) model for comparison.

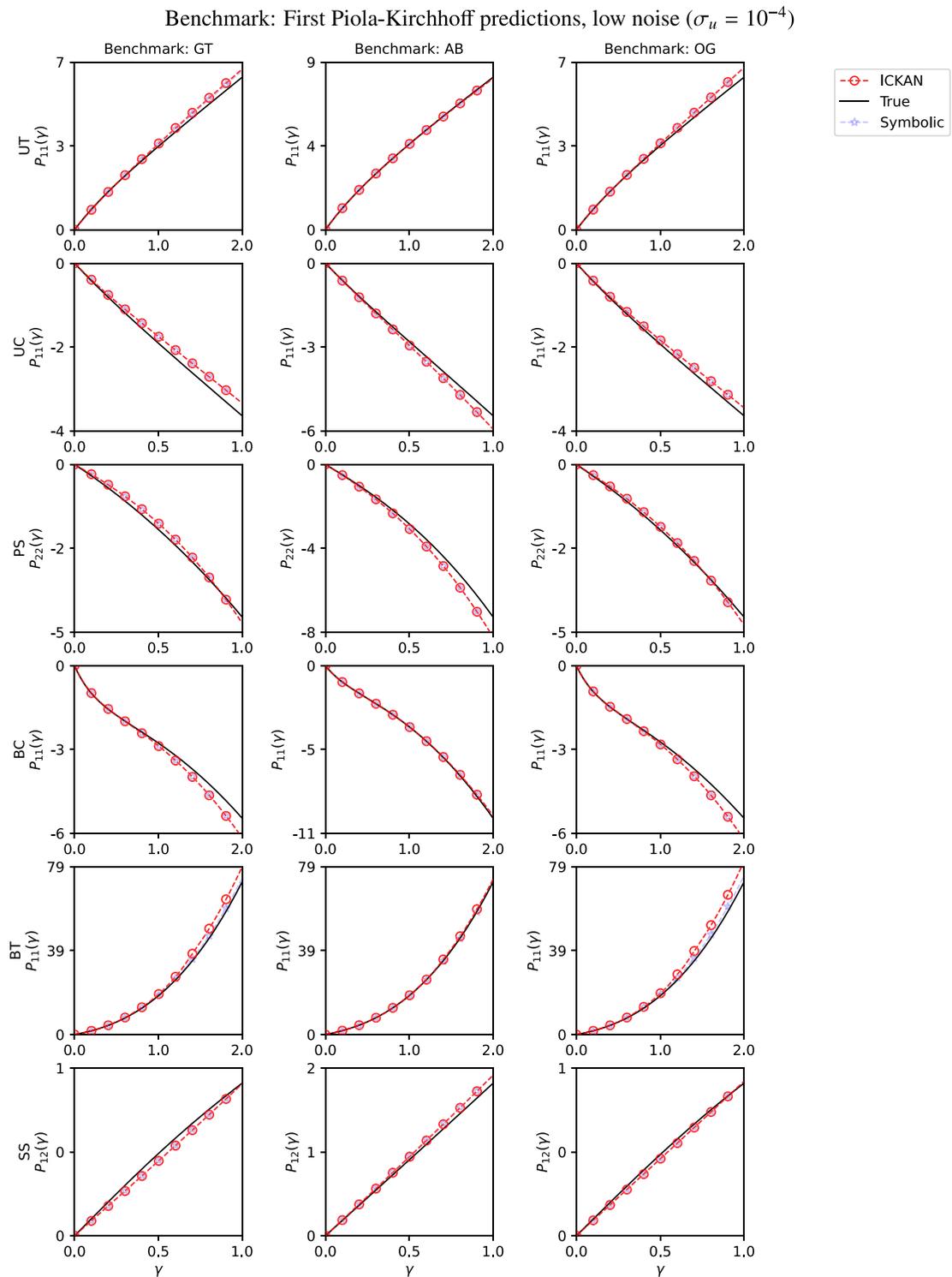


Fig. 10. Predicted first Piola-Kirchhoff stress $P(F(\gamma))$ components along the deformation paths specified in (40) for the low noise case ($\sigma_u = 10^{-4}$). The best ICKAN-based constitutive model is shown for the GT (36), AB (37), and OG (38) benchmarks, the model obtained through symbolic regression of the ICKAN, alongside the response of the true (hidden) model for comparison.

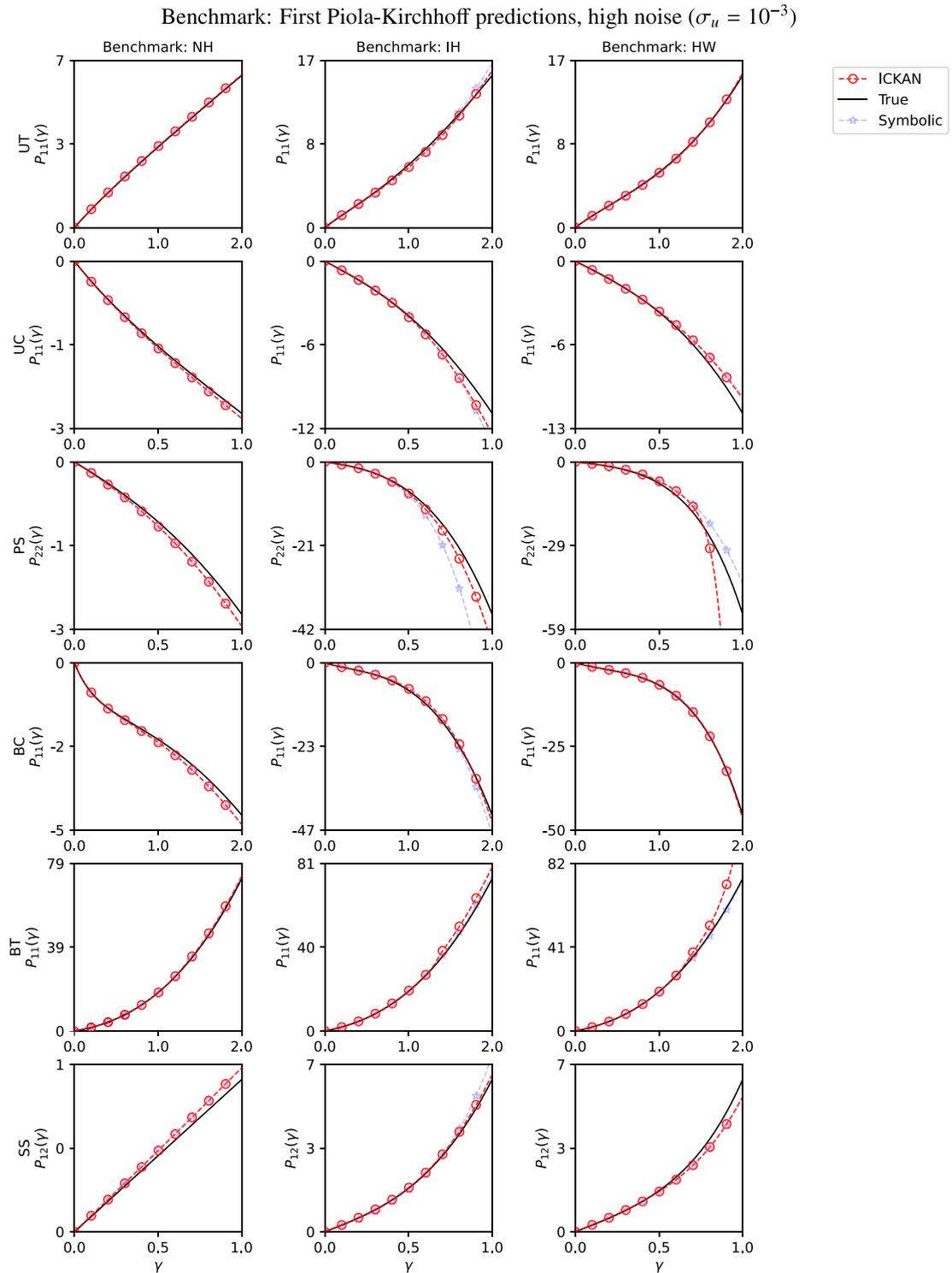


Fig. 11. Predicted first Piola-Kirchhoff stress $P(F(\gamma))$ components along the deformation paths specified in (40) for the high noise case ($\sigma_u = 10^{-3}$). The best ICKAN-based constitutive model is shown for the NH (33), IH (34), and HW (35) benchmarks, the model obtained through symbolic regression of the ICKAN, alongside the response of the true (hidden) model for comparison.

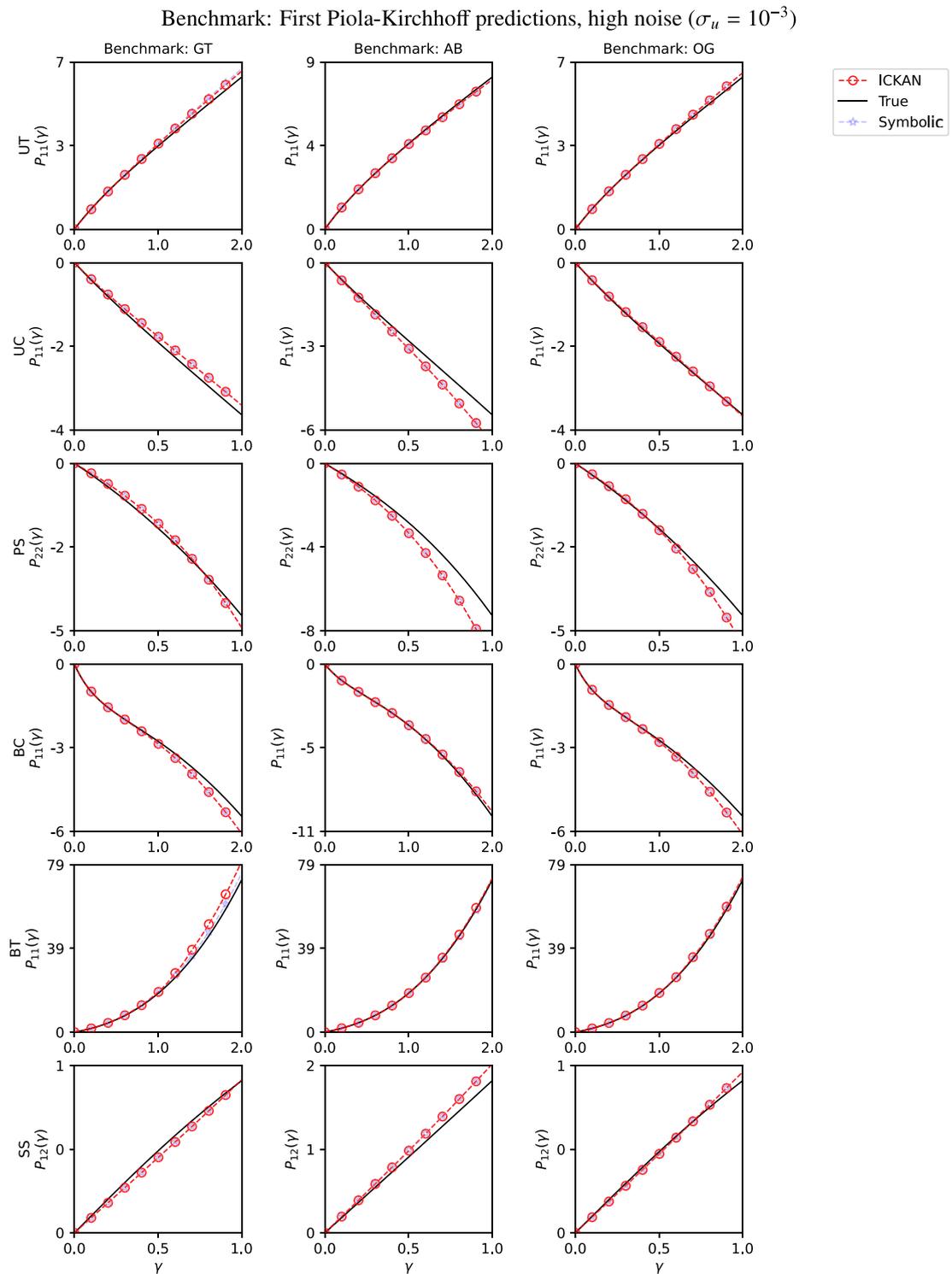


Fig. 12. Predicted first Piola-Kirchhoff stress $P(F(\gamma))$ components along the deformation paths specified in (40) for the high noise case ($\sigma_u = 10^{-3}$). The best ICKAN-based constitutive model is shown for the GT (36), AB (37), and OG (38) benchmarks, the model obtained through symbolic regression of the ICKAN, alongside the response of the true (hidden) model for comparison.

Table 1

Strain energy density of the (true) hidden and discovered symbolic expressions by the ICKAN-based material models for different noise levels σ_u . Note that the symbolic expressions do not satisfy $W(F = I) = 0$; this can be manually fixed by re-introducing the energy correction term W^0 (see (9)). However, they do satisfy the condition $P(F = I) = 0$ identically (see (13)).

Model	Noise	Strain energy density (W)
NH2 (33)	Truth	$0.5(\bar{I}_1 - 3) + 1.5(J - 1)^2$
	$\sigma = 10^{-4}$	$0.5201K_1 + 1.4969K_3$
	$\sigma = 10^{-3}$	$0.5395K_1 + 1.4986K_3$
IH (34)	Truth	$0.5(\bar{I}_1 - 3) + (\bar{I}_2 - 3) + (\bar{I}_1 - 3)^2 + 1.5(J - 1)^2$
	$\sigma = 10^{-4}$	$0.35K_1 + 1.49K_3 + 0.004 \log(3568 \exp(4.23K_2) + 1)^2 + 0.004 \log(22.43 \exp(3.3K_1) + 1)^3$
	$\sigma = 10^{-3}$	$0.31K_1 + 1.48K_3 + 0.004 \log(3568 \exp(4.30K_2) + 1)^2 + 0.009 \log(10.16 \exp(2.5K_1) + 1)^3$
HW (35)	Truth	$0.5(\bar{I}_1 - 3) + (\bar{I}_2 - 3) + 0.7(\bar{I}_1 - 3)(\bar{I}_2 - 3) + 0.2(\bar{I}_1 - 3)^3 + 1.5(J - 1)^2$
	$\sigma = 10^{-4}$	$0.43K_2 + 1.48K_3 + 0.69 \log(0.57 \exp(1.04K_1) + 1)^2 + 0.008 \log(6.64 \exp(1.90K_1) + 1)^3$
	$\sigma = 10^{-3}$	$0.42K_2 + 1.49K_3 + 0.88 \log(0.57 \exp(1.04K_1) + 1)^2 + 0.591 \log(0.57 \exp(1.04K_1) + 1)^2$
GT (36)	Truth	$0.5(\bar{I}_1 - 3) + \log(\bar{I}_2/3) + 1.5(J - 1)^2$
	$\sigma = 10^{-4}$	$0.65K_1 + 0.03K_2 + 0.44K_3 + 1.58 \log(35.20 \exp(0.69K_3) + 1)$
	$\sigma = 10^{-3}$	$0.62K_1 + 0.04K_2 + 0.78K_3 + 1.09 \log(16.45 \exp(0.72K_3) + 1)$
AB (37)	Truth	Refer to (37)
	$\sigma = 10^{-4}$	$1.1959K_1 + 0.0533K_2 + 1.4905K_3$
	$\sigma = 10^{-3}$	$1.0861K_1 + 0.1140K_2 + 1.4790K_3$
OG (38)	Truth	Refer to (38)
	$\sigma = 10^{-4}$	$0.76K_1 + 0.001K_2 + 0.84K_3 + 1.09 \log(13.46 \exp(0.66K_3) + 1)$
	$\sigma = 10^{-3}$	$0.69K_1 + 0.042K_2 + 1.35K_3 + 0.23 \log(17.64 \exp(0.73K_3) + 1)$

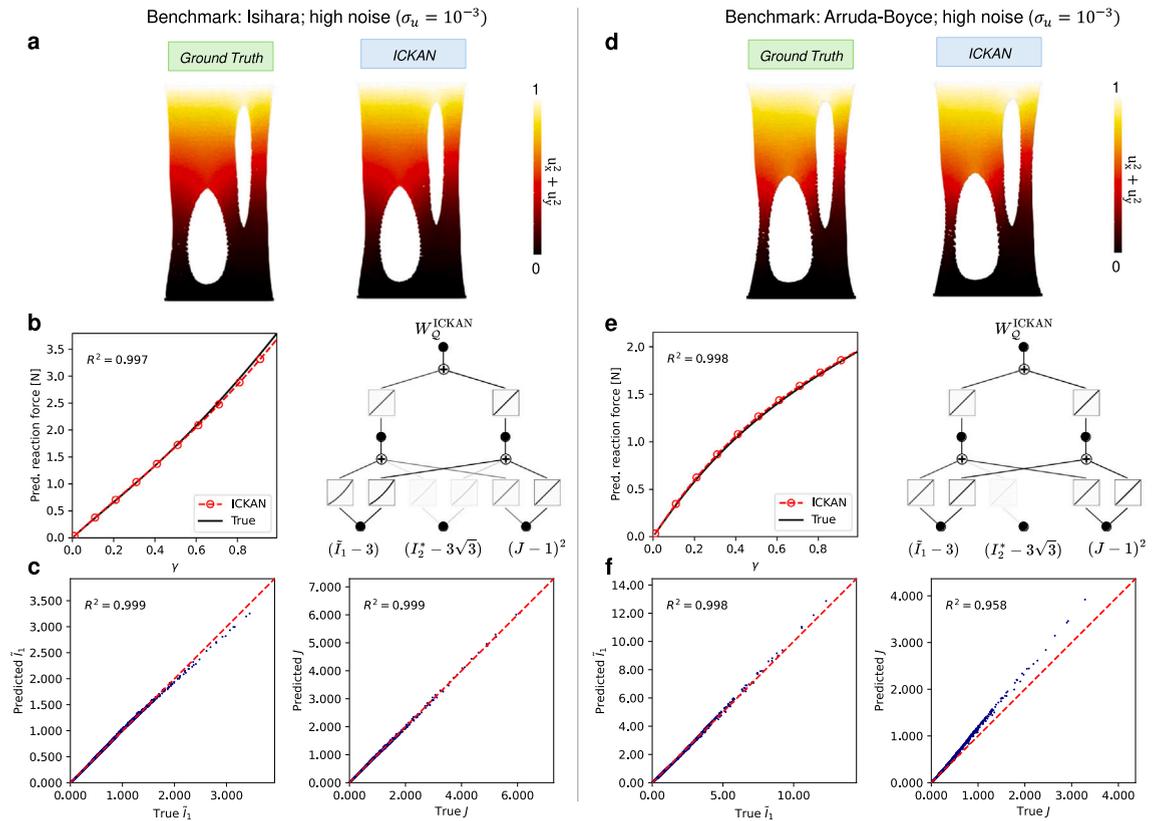


Fig. 13. Comparison of FEM for the validation specimen (see Fig. 3b) between the ground-truth and ICKAN-based constitutive model for the Isihara benchmark (34) (a-c) and the Arruda-Boyce benchmark (37) (d-f) (trained on high noise data). (a & d) Deformed shape of the validation specimen, with colors indicating displacement magnitude, as obtained from both the ground-truth (left) and ICKAN-based (right) simulations. (b & e) Comparison of the reaction force on the top surface plotted against the loading parameter γ (left) and the trained ICKAN-model illustrated with the in-built plotting functionality (right) (c & f) Parity plot of the first strain invariant \bar{I}_1 (left) and the determinant J (right) predicted by the ICKAN model versus the ground-truth values, with the gray dashed line representing perfect agreement (zero intercept, unit slope) and the R^2 value indicating accuracy.

Extensive numerical experiments demonstrated that the ICKAN-based models accurately capture the nonlinear stress–strain behavior of hyperelastic materials, even under varying levels of noise in the data. The predicted strain energy densities and first Piola–Kirchhoff stresses closely match the ground truth for numerous material models across multiple deformation paths. Furthermore, finite element simulations on a validation specimen with a complex geometry confirmed the generalization capability of our method, as the ICKAN-based constitutive models produced robust and physically consistent results.

Our findings indicate that the ICKAN framework successfully overcomes key limitations of ICNN-based models by combining physical consistency with enhanced interpretability and efficiency. Future work will explore extending the framework to model additional material behaviors such as viscoelasticity and plasticity, as well as integrating uncertainty quantification to further bolster model reliability in real-world applications. Additionally, the ICKAN approach can be applied to any machine learning task that ICNNs, as introduced by Amos et al. [47], can address, thereby benefiting the broader machine learning community.

CRediT authorship contribution statement

Prakash Thakolkaran: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Yaqi Guo:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Shivam Saini:** Validation, Software, Methodology, Investigation. **Mathias Peirlinck:** Supervision. **Benjamin Alheit:** Methodology. **Siddhant Kumar:** Writing – review & editing, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization.

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.

Acknowledgment

Y. Guo sincerely thanks Dr. Ye Ji and Chenghuai Lin (MSc.) for their insightful discussions on convex B-splines.

Appendix A. Details on enforcing convexity and monotonicity

A.1. Proof of the non-decreasing convexity condition

The derivative of a spline basis function can be computed as [72]:

$$\frac{dB_{i,k}(x)}{dx} = \frac{k}{t_{i+k} - t_i} B_{i,k-1}(x) - \frac{k}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(x). \tag{A.1}$$

This implies that

$$\begin{aligned} \frac{d\psi(x)}{dx} &= \sum_{i=1}^{n_b} c_i \left(\frac{k}{t_{i+k} - t_i} B_{i,k-1}(x) - \frac{k}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(x) \right) \\ &= k \sum_{i=0}^{n_b-1} \frac{c_{i+1}}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(x) - k \sum_{i=1}^{n_b} \frac{c_i}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(x) \\ &= \frac{kc_1}{t_{k+1} - t_1} B_{1,k-1}(x) + k \sum_{i=1}^{n_b-1} \frac{c_{i+1} - c_i}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(x) - \frac{kc_{n_b}}{t_{n_b+k+1} - t_{n_b+1}} B_{n_b+1,k-1}(x) \end{aligned} \tag{A.2}$$

Note that by definition of the basis functions, the first and last terms are zero. For uniform B-splines, the knots are equally spaced by interval $s = t_{i+1} - t_i > 0$, for $i \in [1, n_b - 1]$. Hence the first order derivative can be simplified as

$$\frac{d\psi(x)}{dx} = \frac{1}{s} \sum_{i=1}^{n_b-1} (c_{i+1} - c_i) B_{i+1,k-1}(x). \tag{A.3}$$

Since basis functions are by definition non-negative, the non-decreasing requirement

$$\frac{d\psi(x)}{dx} \geq 0 \tag{A.4}$$

leads to

$$c_{i+1} - c_i \geq 0, \quad \forall i \in [1, n_b - 1]. \tag{A.5}$$

The second-order derivative of the basis function is:

$$\begin{aligned} \frac{d^2 B_{i,k}(x)}{dx^2} &= \frac{k}{t_{i+k} - t_i} \left(\frac{k-1}{t_{i+k-1} - t_i} B_{i,k-2}(x) - \frac{k-1}{t_{i+k} - t_{i+1}} B_{i+1,k-2}(x) \right) - \\ &\frac{k}{t_{i+k+1} - t_{i+1}} \left(\frac{k-1}{t_{i+k} - t_{i+1}} B_{i+1,k-2}(x) - \frac{k-1}{t_{i+k+1} - t_{i+2}} B_{i+2,k-2}(x) \right) \\ &= \frac{1}{s^2} (B_{i,k-2}(x) - 2B_{i+1,k-2}(x) + B_{i+2,k-2}(x)). \end{aligned} \quad (A.6)$$

Substituting this into $\psi(x)$ and considering the definition of the basis functions, we have:

$$\begin{aligned} \frac{d^2 \psi(x)}{dx^2} &= \sum_{i=1}^{n_b} c_i \frac{d^2 B_{i,k}(x)}{dx^2} \\ &= \frac{1}{s^2} \left(\sum_{i=1}^{n_b-2} c_{i+2} B_{i+2,k-2}(x) - 2 \sum_{i=0}^{n_b-1} c_{i+1} B_{i+2,k-2}(x) + \sum_{i=1}^{n_b} c_i B_{i+2,k-2}(x) \right) \\ &= \frac{1}{s^2} \left(\sum_{i=1}^{n_b-1} (c_{i+2} - 2c_{i+1} + c_i) B_{i+2,k-2}(x) \right). \end{aligned} \quad (A.7)$$

Convexity condition requires

$$\frac{d^2 \psi(x)}{dx^2} \geq 0. \quad (A.8)$$

Since $B_{i+2,k-2}(x) \geq 0$, convexity is ensured if:

$$c_{i+2} - 2c_{i+1} + c_i \geq 0, \quad \forall i \in [1, n_b - 2]. \quad (A.9)$$

A.2. Implementation of convexity and monotonicity constraints

Let $\mathbf{p} = \{p_i\}_{i=1}^{n_b}$ be a sequence of randomly initialized coefficients, A sequence of convex coefficients $\mathbf{c} = \{c_i\}_{i=1}^{n_b}$ can be derived from \mathbf{p} through the following steps:

$$\begin{aligned} \mathbf{h} &= \{h_i\}_{i=1}^{n_b}, \quad \text{with } h_i = \begin{cases} p_1, & \text{if } i = 1, \\ \max(0, p_i), & \text{otherwise,} \end{cases} \\ \mathbf{d} &= \{d_i\}_{i=1}^{n_b}, \quad \text{with } d_i = \begin{cases} h_1, & \text{if } i = 1, \\ \sum_{j=2}^i h_j, & \text{otherwise,} \end{cases} \\ \mathbf{c} &= \{c_i\}_{i=1}^{n_b}, \quad \text{with } c_i = \sum_{j=1}^i d_j. \end{aligned} \quad (A.10)$$

Since $\mathbf{d} = \{d_i\}_{i=2}^{n_b}$ is the cumulative sum of a sequence of non-negative values $\mathbf{h} = \{h_i\}_{i=2}^{n_b}$, it follows that:

$$c_{i+1} - c_i = d_{i+1} \geq d_i = c_i - c_{i-1} \geq 0, \quad \forall i \in [2, n_b - 1]. \quad (A.11)$$

Thus, the sequence $\mathbf{c} = \{c_i\}_{i=1}^{n_b}$ is non-decreasing and convex.

This algorithm enforces a hard constraint on the coefficients, ensuring convexity by construction. Consequently, backpropagation can be performed efficiently during training without introducing additional computational overhead.

A.3. Linear extrapolation at the endpoints

As illustrated in Fig. 1b, convexity cannot be guaranteed outside the B-spline domain $[t_{k+1}, t_{m_b-k}]$. To smoothly extend the curve $\psi(x)$ beyond these boundaries, we apply a linear extrapolation that preserves the local behavior at the endpoints.

For the left endpoint at $x = t_{k+1}$, choose two sample points using a small offset $1 \gg \varepsilon > 0$:

$$x_0 = t_{k+1}, \quad x_1 = t_{k+1} + \varepsilon. \quad (A.12)$$

Evaluating the curve at these points gives:

$$y_0 = \psi(x_0), \quad y_1 = \psi(x_1). \quad (A.13)$$

We fit a linear function of the form:

$$\psi_{\text{left}}(x) = a_{\text{left}}(x - t_{k+1}) + b_{\text{left}}. \quad (A.14)$$

Using the finite difference approximation, the coefficients are determined as:

$$a_{\text{left}} = \frac{y_1 - y_0}{\varepsilon} = \frac{\psi(t_{k+1} + \varepsilon) - \psi(t_{k+1})}{\varepsilon}, \quad b_{\text{left}} = \psi(t_{k+1}). \quad (A.15)$$

Thus, for $x < t_{k+1}$, the extrapolated curve is given by

$$\psi_{\text{left}}(x) = \frac{\psi(t_{k+1} + \varepsilon) - \psi(t_{k+1})}{\varepsilon}(x - t_{k+1}) + \psi(t_{k+1}). \quad (\text{A.16})$$

Similarly, for the right endpoint at $x = t_{m_b-k}$, extrapolation function can be evaluated as:

$$\psi_{\text{right}}(x) = \frac{\psi(t_{m_b-k}) - \psi(t_{m_b-k} - \varepsilon)}{\varepsilon}(x - t_{m_b-k}) + \psi(t_{m_b-k}), \quad x > t_{m_b-k}. \quad (\text{A.17})$$

By applying the above linear extrapolation at both endpoints, the extended B-spline curve $\bar{\psi}(x)$ is defined as

$$\bar{\psi}(x) = \begin{cases} \frac{\psi(t_{k+1} + \varepsilon) - \psi(t_{k+1})}{\varepsilon}(x - t_{k+1}) + \psi(t_{k+1}), & x < t_{k+1}, \\ \psi(x), & t_{k+1} \leq x \leq t_{m_b-k}, \\ \frac{\psi(t_{m_b-k}) - \psi(t_{m_b-k} - \varepsilon)}{\varepsilon}(x - t_{m_b-k}) + \psi(t_{m_b-k}), & x > t_{m_b-k}. \end{cases} \quad (\text{A.18})$$

This approach ensures a smooth extension of the B-spline, preserving both function continuity (C^0) and first-order smoothness (C^1) at the boundaries, thereby maintaining local shape consistency. Note that while we can ensure physical consistency for W outside the spline domain using linear extrapolation, it may not be the most ideal solution and could hinder generalization.

Appendix B. ICKAN spline grid initialization

In our ICKAN implementation, the natural definition domain bounds of the splines are set during the initialization phase to ensure they cover the expected range of input values.

This process begins by constructing dummy inputs in the form of $z^{(0)}$ —with each dimension's values set in the sufficiently large range of $[-5, 25]$ discretized along 100 uniformly spaced points. These dummy inputs are passed through the first KAN layer ($r = 1$) and the outputs, i.e., $z^{(1)}$, are recorded. The range of values for $z_j^{(1)}$ (with $j \in \{1, \dots, n_0\}$) is computed, which is then set as the natural definition domain for $\phi_{0,i,j}$ for all $i \in \{1, \dots, n_1\}$.

Similar to the previous step, the new range of values for each dimension of $z^{(1)}$ is set to the respective natural definition domains and new dummy inputs are created for the second KAN layer ($r = 2$). The process is then subsequently repeated for the following layers in a similar fashion.

Once this initialization step is complete for all KAN layers, the bounds remains fixed and are not updated during training or inference.

Appendix C. Training details

The Algorithm 1 (adapted from NN-EUCLID [48]) summarizes the unsupervised training of the ICKAN-based constitutive models. All parameters used for data generation and training are found in Table C.2.

Appendix D. Learning material models without convexity constraints

Figs. D.14–D.15 illustrate the constitutive response for three representative benchmarks when the ICKAN model in (6) is replaced with a standard KAN without the convexity and monotonicity constraints while maintaining the same network architecture. We kept the initial update of the knot range (outlined in Appendix B). Here, we included the bias function in the construction of the univariate function to improve the learning (as originally suggested for KANs by Liu et al. [40]), i.e.,

$$\phi(x) = w_b b(x) + w_s \psi(x), \quad (\text{D.1})$$

where $b(x) = x(1 + \exp(-x))^{-1}$ is the SiLU activation [73] and w_b is a trainable scalar weight. Note that we did not perform any other hyperparameter tuning in this case.

Appendix E. Comparison with EUCLID

Flaschel et al. [11] have previously introduced the EUCLID framework for discovering interpretable and analytical hyperelastic models via sparse (linear) regression on a large catalog of candidate functions in an unsupervised setting. Since the candidate functions are predefined, EUCLID inherently provides interpretable models. However, its expressivity may be limited by the chosen set of functions used for sparse regression.

Here, we compare the vanilla EUCLID framework with ICKANs. The EUCLID implementation used in this work is directly adapted from [11] (code available at: github.com/EUCLID-code/EUCLID-hyperelasticity). In consultation with the authors of Flaschel et al. [11], we changed the penalty parameter λ_p therein from 0.0001 to 0.0125 to adapt the implementation to our data. For details such as the methodology, hyperparameters, and the catalog of candidate functions, please refer to Flaschel et al. [11].

The results in Fig. E.16 demonstrate that both the EUCLID framework [11] and ICKAN perform well in capturing the underlying constitutive behavior for the IH (34) and AB (37) benchmarks under high noise conditions. EUCLID discovers physically meaningful and compact strain energy formulations by operating within a predefined functional basis, which promotes interpretability but can

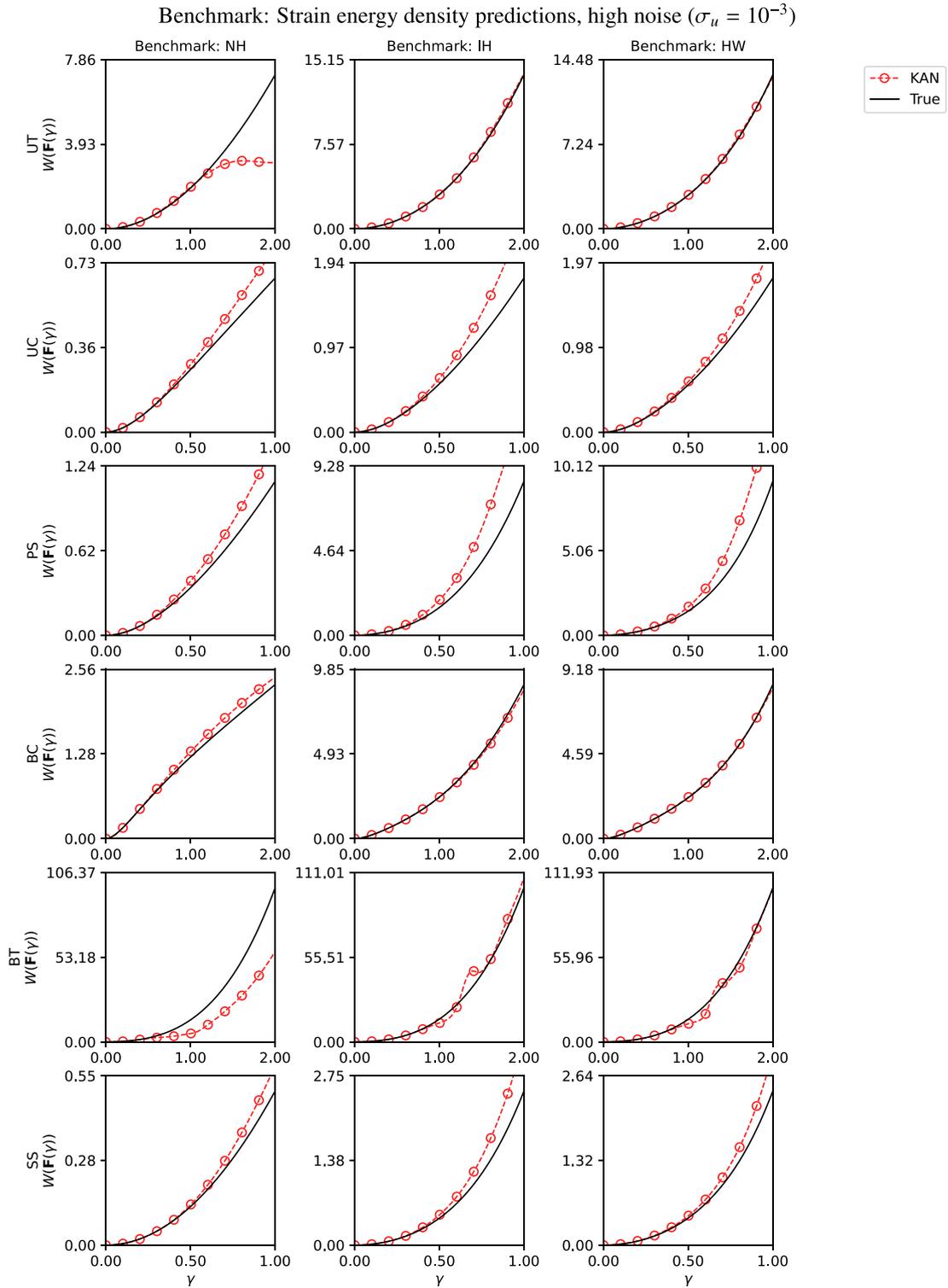


Fig. D.14. Strain energy density $W(F(\gamma))$ predictions for the deformation paths in (40), evaluated under high noise ($\sigma_u = 10^{-3}$). Results are shown for a vanilla KAN-based model, as well as for the hidden true model, for the NH (33), IH (34), and HW (35) benchmarks.

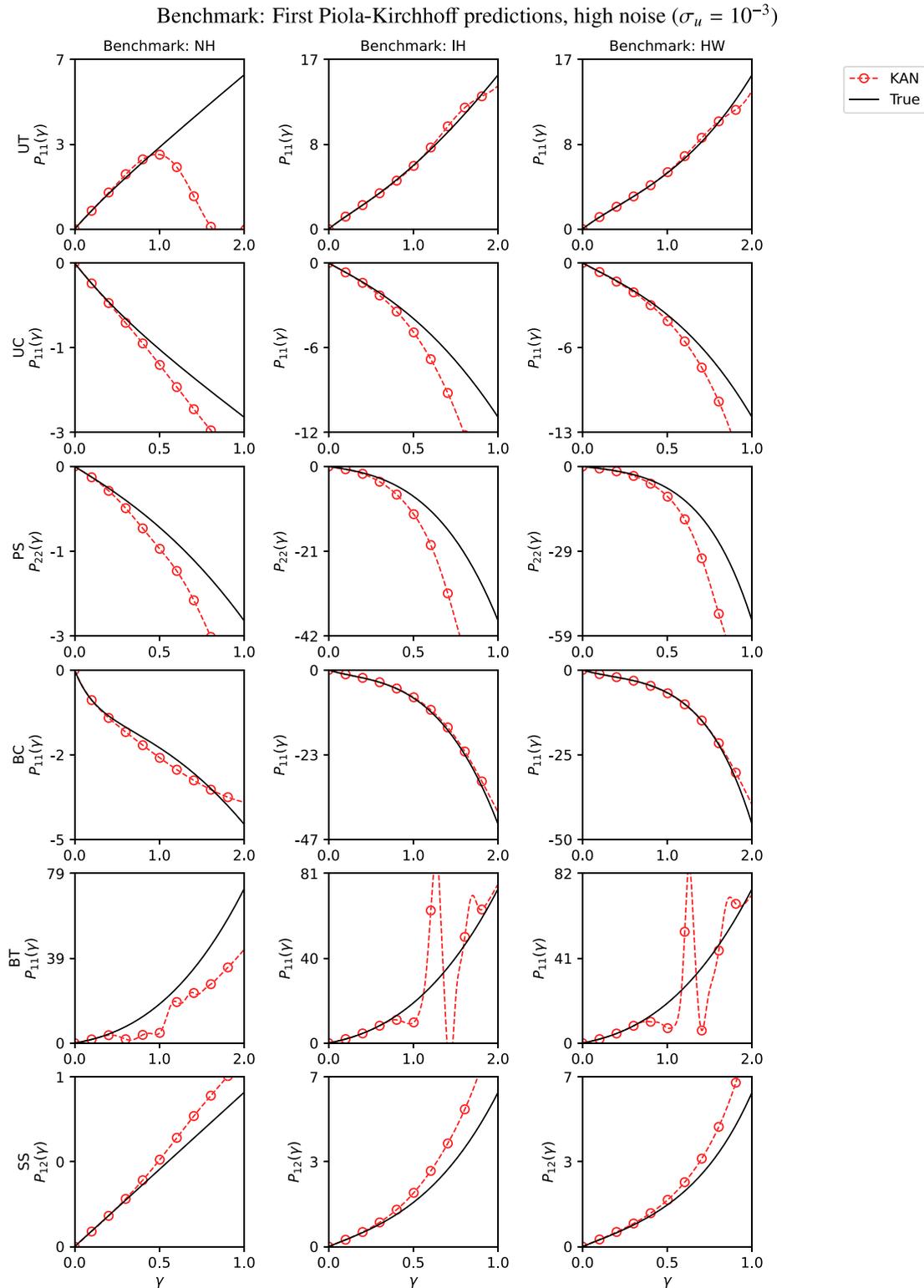


Fig. D.15. Predicted first Piola-Kirchhoff stress $P(F(\gamma))$ components along the deformation paths specified in (40) for the high noise case ($\sigma_u = 10^{-3}$). The response of a trained vanilla KAN-based model is shown for the NH (33), IH (34), and HW (35) benchmarks, alongside the response of the true (hidden) model for comparison.

Algorithm 1 Unsupervised training of the ICKAN-based constitutive models

```

1: Input: Point-wise displacement data  $\mathcal{U} = \{\mathbf{u}^{a,t} \in \mathbb{R}^2 : a = 1, \dots, n_n; t = 1, \dots, n_t\}$ 
2: Input: Global reaction forces  $\{R^{\beta,t} : \beta = 1, \dots, n_\beta; t = 1, \dots, n_t\}$ 
3: Randomly initialize ICKAN parameters:  $\mathcal{Q}$ .
4: Initialize learning rate scheduler
5: Initialize Adam optimizer with parameters  $\mathcal{Q}$  and learning rate scheduler
6: for  $e = 1, \dots, n_e$  do ▷ Training epochs
7:   Loss:  $\ell \leftarrow 0$  ▷ Initialize loss for current epoch
8:   for  $t = 1, \dots, n_t$  do ▷ Iterate over snapshots
9:     for each element in mesh do
10:       $W^0 \leftarrow W_Q^{\text{ICKAN}}|_{F=t}$  ▷ see (9)
11:       $W \leftarrow W_Q^{\text{ICKAN}} + W^0$  ▷ see (6)
12:       $P \leftarrow \frac{\partial W}{\partial F}$  ▷ see (10)
13:    end for
14:    for  $a = 1, \dots, n_a$  do
15:      for  $i = 1, 2$  do
16:        Compute force  $f_i^{a,t}$  using (29)
17:      end for
18:    end for
19:    for  $(a, i) \in \mathcal{D}^{\text{free}}$  do
20:       $\ell \leftarrow \ell + (f_i^{a,t})^2$  ▷ force balance at free degrees of freedom; see (32)
21:    end for
22:    for  $\beta = 1, \dots, n_\beta$  do
23:       $r^{\beta,t} \leftarrow 0$ 
24:      for  $(a, i) \in \mathcal{D}_\beta^{\text{fix}}$  do
25:         $r^{\beta,t} \leftarrow r^{\beta,t} + f_i^{a,t}$  ▷ see (32)
26:      end for
27:       $\ell \leftarrow \ell + (R^{\beta,t} - r^{\beta,t})^2$  ▷ force balance at fixed degrees of freedom; see (32)
28:    end for
29:  end for
30:  Compute gradients  $\partial \ell / \partial \mathcal{Q}$  using automatic differentiation
31:  Update  $\mathcal{Q}$  with Adam optimizer using gradients  $\partial \ell / \partial \mathcal{Q}$ .
32:  Update learning rate with learning rate scheduler based on epoch number  $e$ 
33: end for
34: Output: Trained ICKAN model  $W_Q^{\text{ICKAN}}$ 

```

limit expressivity in more complex settings. In contrast, ICKANs begin with a highly expressive KAN-based architecture capable of fitting intricate stress–strain responses, and subsequently distills interpretable expressions through symbolic regression. For both benchmarks, the models obtained from EUCLID and ICKAN show strong agreement with the ground truth responses. These results highlight the complementary nature of the two approaches — EUCLID provides interpretable models by design, while ICKAN achieves accuracy-first learning with interpretability introduced in a post-processing stage.

Appendix F. Monotonic input-convex symbolic regression

To improve the interpretability of the learned ICKAN constitutive models, symbolic regression is applied to analytically approximate the trained univariate activation functions individually. The symbolic expressions of each activation are then assembled across the entire network to obtain a symbolic expression of the whole constitutive model. Here, we adapt the symbolic framework of Liu et al. [40] to enable monotonic input-convex symbolic regression and, thereby, obtain symbolic polyconvex hyperelastic models.

Given a trained univariate function mapping from layer r to layer $r+1$, the activation function $\phi_{r,i,j}$ is approximated by searching for the best symbolic representation $\hat{\phi}_{r,i,j}$ from a candidate function library \mathcal{W} . In the subsequent discussion, we drop the subscripts on ϕ for the sake of brevity.

For a given candidate function $f \in \mathcal{W}$ that is convex and non-decreasing, we construct an approximation ansatz of the form:

$$\hat{\phi}(x) = cf(ax + b) + d, \quad \text{with } a \geq 0 \text{ and } c \geq 0, \quad (\text{F.1})$$

where a, b, c, d are scalar fitting parameters. The convex and non-decreasing constraint on f and along with $a \geq 0$ and $c \geq 0$ ensure that $\hat{\phi}$ is convex and non-decreasing. Consequently, this guarantees that the ICKAN's symbolic model is monotonic and input-convex and the resulting hyperelastic strain energy density is polyconvex.

Table C.2

List of parameters and hyperparameters used for the data generation and benchmarks.

Parameter	Notation	Value
<i>Training specimen:</i>		
Number of nodes in mesh for FEM-based data generation	–	63,601
Number of nodes in data available for learning	n_n	1441
Number of reaction force constraints	n_β	4
Number of data snapshots for NH, GT	n_t	3
Number of data snapshots for IH, HW	n_t	8
Number of data snapshots for AB	n_t	10
Number of data snapshots for OG	n_t	6
Loading parameter for NH, GT, IH, HW	δ	$\{0.1 \times t : t = 1, \dots, n_t\}$
Loading parameter for AB, OG	δ	$\{0.05 \times t : t = 1, \dots, n_t\}$
<i>Validation specimen:</i>		
Number of nodes in the mesh	–	4908
Loading parameter	δ	$\{0.01 \times t : t = 1, \dots, 100\}$
<i>ICKAN hyperparameters:</i>		
Order of B-splines	k	5
Number of coefficients per trainable activations	n_b	17
Number of knots	m_b	23
Number of hidden layers	$R - 1$	1
Number of trainable activations in hidden layer ($R - 1$)	$n_0 \times n_{R-1}$	$3 \times 2 = 6$
Number of trainable activations in output layer (R)	$n_{R-1} \times n_R$	$2 \times 1 = 2$
Number of trained ICKAN models	n_e	10
Optimizer	–	Adam
Epochs	–	1000
Learning rate scheduler	–	Cyclic
Base learning rate	–	0.001
Maximum learning rate	–	0.1
Learning rate scheduler steps	–	50
Symbolic regression balance parameter	λ_{sym}	0.8

The parameters a, b, c, d are obtained by numerically fitting $\hat{\phi}$ to the splines-based ϕ as

$$a, b, c, d \leftarrow \arg \min_{a, b, c, d} \sum_{i=1}^n \|\hat{\phi}(x_i) - \phi(x_i)\|^2 \quad \text{s.t.} \quad a \geq 0 \quad \text{and} \quad c \geq 0. \quad (\text{F.2})$$

Here, the functions $\hat{\phi}$ and ϕ are evaluated at $n = 100$ points for the purpose of fitting. The parameters a and b are determined via iterative grid search in the range $[0, 10]$ and $[-10, 10]$ respectively, while c and d are obtained through linear regression in each iteration.

In this study, we choose the function library as

$$\mathcal{W} = \{x, \exp(x), \log(1 + \exp(x)), \log(1 + \exp(x))^2, \log(1 + \exp(x))^3, \log(1 + \exp(x))^4\} \quad \text{with} \quad x \in \mathbb{R}, \quad (\text{F.3})$$

where all the candidate functions are convex and non-decreasing.

To select the optimal symbolic function f from the library \mathcal{W} , the following loss function and selection criterion is used:

$$f \leftarrow \arg \min_{f \in \mathcal{W}} \lambda_{\text{sym}} \mathcal{L}_{\text{complexity}} + (1 - \lambda_{\text{sym}}) \mathcal{L}_{\text{fit}}, \quad (\text{F.4})$$

where $\lambda_{\text{sym}} \in [0, 1]$ is a scalar hyperparameter. \mathcal{L}_{fit} quantifies the approximation error due to f using the goodness-of-fit R^2 score, defined as

$$\mathcal{L}_{\text{fit}} = \log_2(1 + 10^{-5} - R^2(\hat{\phi}, \phi)), \quad (\text{F.5})$$

while $\mathcal{L}_{\text{complexity}}$ penalizes the complexity of f (i.e., promotes parsimony) by assigning numerical weights to different expressions:

$$\mathcal{L}_{\text{complexity}} = \begin{cases} 1 & \text{if } f(x) = x, \\ 2 & \text{if } f(x) = \exp(x), \\ 2 & \text{if } f(x) = \log(1 + \exp(x))^n, \quad \text{where } n \in 1, 2, 3, 4. \end{cases} \quad (\text{F.6})$$

The parameter λ_{sym} is tunable (see Table C.2 for chosen value), which allows for a balance between simplicity and accuracy.

Data and code availability

The data and codes generated in the current study are freely available at <https://github.com/mmc-group/ICKANs/>.

Benchmark: First Piola-Kirchhoff predictions, high noise ($\sigma_u = 10^{-3}$)

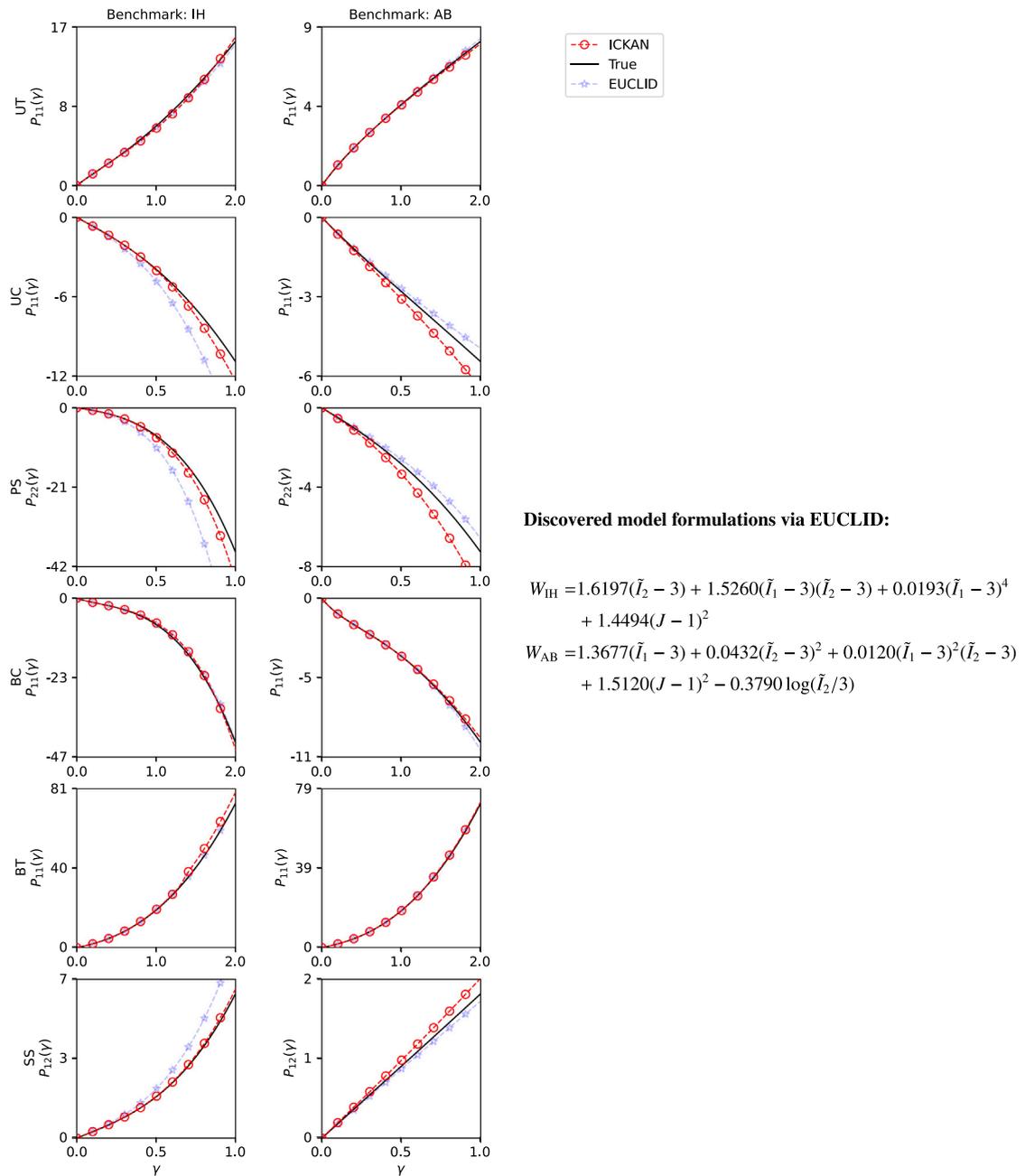


Fig. E.16. Predicted first Piola-Kirchhoff stress $\mathbf{P}(\mathbf{F}(\gamma))$ components along the deformation paths specified in (40) for the high noise case ($\sigma_u = 10^{-3}$). The best ICKAN-based constitutive model is shown for the IH (34) and AB (37) benchmarks, and the model obtained through EUCLID, alongside the response of the true (hidden) model for comparison.

References

- [1] J.N. Fuhg, G. Anantha Padmanabha, N. Bouklas, B. Bahmani, W. Sun, N.N. Vlassis, M. Flaschel, P. Carrara, L. De Lorenzis, A review on data-driven constitutive laws for solids, Arch. Comput. Methods Eng. (2024) <http://dx.doi.org/10.1007/s11831-024-10196-2>.
- [2] S. Kumar, D.M. Kochmann, What machine learning can do for computational solid mechanics, in: Current Trends and Open Problems in Computational Mechanics, Springer, 2022, pp. 275–285, http://dx.doi.org/10.1007/978-3-030-87312-7_27.

- [3] T. Kirchdoerfer, M. Ortiz, Data-driven computational mechanics, *Comput. Methods Appl. Mech. Engrg.* 304 (2016) 81–101, <http://dx.doi.org/10.1016/j.cma.2016.02.001>, URL: <https://www.sciencedirect.com/science/article/pii/S0045782516300238>.
- [4] T. Kirchdoerfer, M. Ortiz, Data-driven computing in dynamics, *Internat. J. Numer. Methods Engrg.* 113 (11) (2018) 1697–1710, <http://dx.doi.org/10.1002/nme.5716>, URL: <https://onlinelibrary.wiley.com/doi/10.1002/nme.5716>.
- [5] R. Eggersmann, T. Kirchdoerfer, S. Reese, L. Stainier, M. Ortiz, Model-free data-driven inelasticity, *Comput. Methods Appl. Mech. Engrg.* 350 (2019) 81–99, <http://dx.doi.org/10.1016/j.cma.2019.02.016>, URL: <https://linkinghub.elsevier.com/retrieve/pii/S0045782519300878>.
- [6] R. Ibañez, D. Borzacchiello, J.V. Aguado, E. Abisset-Chavanne, E. Cueto, P. Ladeveze, F. Chinesta, Data-driven non-linear elasticity: constitutive manifold construction and problem discretization, *Comput. Mech.* 60 (5) (2017) 813–826, <http://dx.doi.org/10.1007/s00466-017-1440-1>, URL: <http://link.springer.com/10.1007/s00466-017-1440-1>.
- [7] S. Conti, S. Müller, M. Ortiz, Data-driven problems in elasticity, *Arch. Ration. Mech. Anal.* 229 (1) (2018) 79–123, <http://dx.doi.org/10.1007/s00205-017-1214-0>, URL: <http://link.springer.com/10.1007/s00205-017-1214-0>.
- [8] L.T.K. Nguyen, M.-A. Keip, A data-driven approach to nonlinear elasticity, *Comput. Struct.* 194 (2018) 97–115, <http://dx.doi.org/10.1016/j.compstruc.2017.07.031>, URL: <https://linkinghub.elsevier.com/retrieve/pii/S0045794917301311>.
- [9] P. Carrara, L. De Lorenzis, L. Stainier, M. Ortiz, Data-driven fracture mechanics, *Comput. Methods Appl. Mech. Engrg.* 372 (2020) 113390, <http://dx.doi.org/10.1016/j.cma.2020.113390>, URL: <https://linkinghub.elsevier.com/retrieve/pii/S0045782520305752>.
- [10] K. Karapiperis, M. Ortiz, J. Andrade, Data-driven nonlocal mechanics: Discovering the internal length scales of materials, *Comput. Methods Appl. Mech. Engrg.* 386 (2021) 114039, <http://dx.doi.org/10.1016/j.cma.2021.114039>, URL: <https://linkinghub.elsevier.com/retrieve/pii/S0045782521003704>.
- [11] M. Flaschel, S. Kumar, L. De Lorenzis, Unsupervised discovery of interpretable hyperelastic constitutive laws, *Comput. Methods Appl. Mech. Engrg.* 381 (2021) 113852, <http://dx.doi.org/10.1016/j.cma.2021.113852>, URL: <https://linkinghub.elsevier.com/retrieve/pii/S0045782521001894>.
- [12] A. Joshi, P. Thakolkaran, Y. Zheng, M. Escande, M. Flaschel, L. De Lorenzis, S. Kumar, Bayesian-EUCLID: Discovering hyperelastic material laws with uncertainties, *Comput. Methods Appl. Mech. Engrg.* 398 (2022) 115225, URL: <http://dx.doi.org/10.1016/j.cma.2022.115225>.
- [13] M. Flaschel, S. Kumar, L. De Lorenzis, Discovering plasticity models without stress data, *Npj Comput. Mater.* 8 (1) (2022) 91, URL: <http://dx.doi.org/10.1038/s41524-022-00752-4>.
- [14] M. Flaschel, S. Kumar, L. De Lorenzis, Automated discovery of generalized standard material models with EUCLID, *Comput. Methods Appl. Mech. Engrg.* 405 (2023) 115867, <http://dx.doi.org/10.1016/j.cma.2022.115867>.
- [15] M. Flaschel, H. Yu, N. Reiter, J. Hinrichsen, S. Budday, P. Steinmann, S. Kumar, L. De Lorenzis, Automated discovery of interpretable hyperelastic material models for human brain tissue with EUCLID, *J. Mech. Phys. Solids* 180 (2023) 105404, <http://dx.doi.org/10.1016/j.jmps.2023.105404>.
- [16] E. Marino, M. Flaschel, S. Kumar, L. De Lorenzis, Automated identification of linear viscoelastic constitutive laws with EUCLID, *Mech. Mater.* 181 (2023) 104643, <http://dx.doi.org/10.1016/j.mechmat.2023.104643>.
- [17] Z. Wang, J. Estrada, E. Arruda, K. Garikipati, Inference of deformation mechanisms and constitutive response of soft material surrogates of biological tissue by full-field characterization and data-driven variational system identification, *J. Mech. Phys. Solids* 153 (2021) 104474, <http://dx.doi.org/10.1016/j.jmps.2021.104474>.
- [18] B. Bahmani, W. Sun, Physics-constrained symbolic model discovery for polyconvex incompressible hyperelastic materials, *Internat. J. Numer. Methods Engrg.* 125 (15) (2024) <http://dx.doi.org/10.1002/nme.7473>.
- [19] B. Bahmani, H.S. Suh, W. Sun, Discovering interpretable elastoplasticity models via the neural polynomial method enabled symbolic regressions, *Comput. Methods Appl. Mech. Engrg.* 422 (2024) 116827, <http://dx.doi.org/10.1016/j.cma.2024.116827>.
- [20] N.N. Phan, W. Sun, J.D. Clayton, HYDRA: Symbolic feature engineering of overparameterized Eulerian hyperelasticity models for fast inference time, *Comput. Methods Appl. Mech. Engrg.* 437 (2025) 117792, <http://dx.doi.org/10.1016/j.cma.2025.117792>.
- [21] J. Hou, X. Chen, T. Wu, E. Kuhl, X. Wang, Automated data-driven discovery of material models based on symbolic regression: A case study on the human brain cortex, *Acta Biomater.* 188 (2024) 276–296, <http://dx.doi.org/10.1016/j.actbio.2024.09.005>.
- [22] R. Abdusalamov, M. Hillgärtner, M. Itskov, Hyperelastic material modelling using symbolic regression, *PAMM* 22 (1) (2023) <http://dx.doi.org/10.1002/pamm.202200263>.
- [23] G. Kissas, S. Mishra, E. Chatzi, L. De Lorenzis, The language of hyperelastic materials, *Comput. Methods Appl. Mech. Engrg.* 428 (2024) 117053, <http://dx.doi.org/10.1016/j.cma.2024.117053>.
- [24] K. Upadhyay, J.N. Fuhg, N. Bouklas, K.T. Ramesh, Physics-informed data-driven discovery of constitutive models with application to strain-rate-sensitive soft materials, *Comput. Mech. Engrg.* (2024) <http://dx.doi.org/10.1007/s00466-024-02497-x>.
- [25] I. Rocha, P. Kerfriden, F. van der Meer, On-the-fly construction of surrogate constitutive models for concurrent multiscale mechanical analysis through probabilistic machine learning, *J. Comput. Phys.: X* 9 (2021) 100083, <http://dx.doi.org/10.1016/j.jcpx.2020.100083>, URL: <https://www.sciencedirect.com/science/article/pii/S2590055220300354>.
- [26] J.N. Fuhg, M. Marino, N. Bouklas, Local approximate Gaussian process regression for data-driven constitutive models: development and comparison with neural networks, *Comput. Methods Appl. Mech. Engrg.* 388 (2022) 114217, <http://dx.doi.org/10.1016/j.cma.2021.114217>, URL: <https://www.sciencedirect.com/science/article/pii/S004578252100548X>.
- [27] A. Aggarwal, B.S. Jensen, S. Pant, C.-H. Lee, Strain energy density as a gaussian process and its utilization in stochastic finite element analysis: application to planar soft tissues, *Comput. Methods Appl. Mech. Engrg.* 404 (2023) 115812, <http://dx.doi.org/10.1016/j.cma.2022.115812>.
- [28] L. Linden, D.K. Klein, K.A. Kalina, J. Brummund, O. Weeger, M. Kästner, Neural networks meet hyperelasticity: A guide to enforcing physics, *J. Mech. Phys. Solids* 179 (2023) 105363, <http://dx.doi.org/10.1016/j.jmps.2023.105363>.
- [29] N.N. Vlassis, R. Ma, W. Sun, Geometric deep learning for computational mechanics part I: anisotropic hyperelasticity, *Comput. Methods Appl. Mech. Engrg.* 371 (2020) 113299, <http://dx.doi.org/10.1016/j.cma.2020.113299>.
- [30] K.P. Abdolazizi, K. Linka, C.J. Cyron, Viscoelastic constitutive artificial neural networks (vcanns) – a framework for data-driven anisotropic nonlinear finite viscoelasticity, *J. Comput. Phys.* 499 (2024) 112704, <http://dx.doi.org/10.1016/j.jcp.2023.112704>.
- [31] C. Bonatti, D. Mohr, One for all: Universal material model based on minimal state-space neural networks, *Sci. Adv.* 7 (26) (2021) <http://dx.doi.org/10.1126/sciadv.abf3658>.
- [32] D.K. Klein, R. Ortigosa, J. Martínez-Frutos, O. Weeger, Nonlinear electro-elastic finite element analysis with neural network constitutive models, *Comput. Methods Appl. Mech. Engrg.* 425 (2024) 116910, <http://dx.doi.org/10.1016/j.cma.2024.116910>.
- [33] K.A. Kalina, P. Gebhart, J. Brummund, L. Linden, W. Sun, M. Kästner, Neural network-based multiscale modeling of finite strain magneto-elasticity with relaxed convexity criteria, *Comput. Methods Appl. Mech. Engrg.* 421 (2024) 116739, <http://dx.doi.org/10.1016/j.cma.2023.116739>.
- [34] M. Mozaffar, R. Bostanabad, W. Chen, K. Ehmann, J. Cao, M.A. Bessa, Deep learning predicts path-dependent plasticity, *Proc. Natl. Acad. Sci.* 116 (52) (2019) 26414–26420, <http://dx.doi.org/10.1073/pnas.1911815116>.
- [35] L. Zheng, D.M. Kochmann, S. Kumar, HyperCAN: Hypernetwork-driven deep parameterized constitutive models for metamaterials, *Extrem. Mech. Lett.* 72 (2024) 102243, <http://dx.doi.org/10.1016/j.eml.2024.102243>.
- [36] V. Taç, F. Sahli Costabal, A.B. Tepole, Data-driven tissue mechanics with polyconvex neural ordinary differential equations, *Comput. Methods Appl. Mech. Engrg.* 398 (2022) 115248, <http://dx.doi.org/10.1016/j.cma.2022.115248>.
- [37] K. Linka, E. Kuhl, A new family of constitutive artificial neural networks towards automated model discovery, *Comput. Methods Appl. Mech. Engrg.* 403 (2023) 115731, <http://dx.doi.org/10.1016/j.cma.2022.115731>.

- [38] K. Linka, G.A. Holzapfel, E. Kuhl, Discovering uncertainty: Bayesian constitutive artificial neural networks, *Comput. Methods Appl. Mech. Engrg.* 433 (2025) 117517, <http://dx.doi.org/10.1016/j.cma.2024.117517>.
- [39] V. Taç, M.K. Rausch, I. Bilonis, F. Sahli Costabal, A.B. Tepole, Generative hyperelasticity with physics-informed probabilistic diffusion fields, *Eng. Comput.* 41 (1) (2024) 51–69, <http://dx.doi.org/10.1007/s00366-024-01984-2>.
- [40] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T.Y. Hou, M. Tegmark, Kan: Kolmogorov–Arnold networks, 2024, arXiv preprint [arXiv:2404.19756](https://arxiv.org/abs/2404.19756). URL: <https://arxiv.org/abs/2404.19756>.
- [41] J. Schmidt-Hieber, The Kolmogorov–Arnold representation theorem revisited, *Neural Netw.* 137 (2021) 119–126.
- [42] S. Patra, S. Panda, B.K. Parida, M. Arya, K. Jacobs, D.I. Bondar, A. Sen, Physics informed Kolmogorov–Arnold neural networks for dynamical analysis via efficient-kan and wav-kan, 2024, arXiv preprint [arXiv:2407.18373](https://arxiv.org/abs/2407.18373).
- [43] D.W. Abueidda, P. Pantidis, M.E. Mobasher, DeepOKAN: Deep operator network based on Kolmogorov Arnold networks for mechanics problems, *Comput. Methods Appl. Mech. Engrg.* 436 (2025) 117699, <http://dx.doi.org/10.1016/j.cma.2024.117699>, URL: <https://www.sciencedirect.com/science/article/pii/S0045782524009538>.
- [44] Y. Wang, J. Sun, J. Bai, C. Anitescu, M.S. Eshaghi, X. Zhuang, T. Rabczuk, Y. Liu, Kolmogorov–Arnold-informed neural network: A physics-informed deep learning framework for solving forward and inverse problems based on Kolmogorov–Arnold networks, *Comput. Methods Appl. Mech. Engrg.* 433 (2025) 117518, <http://dx.doi.org/10.1016/j.cma.2024.117518>, URL: <https://www.sciencedirect.com/science/article/pii/S0045782524007722>.
- [45] Z. Gao, G.E. Karniadakis, Scalable Bayesian physics-informed Kolmogorov–Arnold networks, 2025, <https://arxiv.org/abs/2501.08501>, URL: <https://arxiv.org/abs/2501.08501>.
- [46] J.D. Toscano, L.-L. Wang, G.E. Karniadakis, KKANs: Kurkova–Kolmogorov–Arnold networks and their learning dynamics, 2024, <https://arxiv.org/abs/2412.16738>, URL: <https://arxiv.org/abs/2412.16738>.
- [47] B. Amos, L. Xu, J.Z. Kolter, Input convex neural networks, in: *Proceedings of the 34th International Conference on Machine Learning*, in: *Proceedings of Machine Learning Research*, vol. 70, PMLR, 2017, pp. 146–155, URL: <https://proceedings.mlr.press/v70/amos17b.html>.
- [48] P. Thakolkaran, A. Joshi, Y. Zheng, M. Flaschel, L. De Lorenzis, S. Kumar, NN-EUCLID: Deep-learning hyperelasticity without stress data, *J. Mech. Phys. Solids* 169 (2022) 105076, <http://dx.doi.org/10.1016/j.jmps.2022.105076>.
- [49] F. As'ad, P. Avery, C. Farhat, A mechanics-informed artificial neural network approach in data-driven constitutive modeling, *Internat. J. Numer. Methods Engrg.* 123 (12) (2022) 2738–2759, <http://dx.doi.org/10.1002/nme.6957>, URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.6957>. arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.6957>.
- [50] D.K. Klein, M. Fernández, R.J. Martin, P. Neff, O. Weeger, Polyconvex anisotropic hyperelasticity with neural networks, *J. Mech. Phys. Solids* 159 (2022) 104703, URL: <http://dx.doi.org/10.1016/j.jmps.2021.104703>.
- [51] S. Meng, A.A.K. Yousefi, S. Avril, Machine-learning-based virtual fields method: Application to anisotropic hyperelasticity, *Comput. Methods Appl. Mech. Engrg.* 434 (2025) 117580, <http://dx.doi.org/10.1016/j.cma.2024.117580>.
- [52] R. Shi, H. Yang, J. Chen, K. Hackl, S. Avril, Y. He, Deep learning without stress data on the discovery of multi-regional hyperelastic properties, *Comput. Mech.* (2025) <http://dx.doi.org/10.1007/s00466-024-02591-0>.
- [53] C. Jailin, A. Benady, R. Legroux, E. Baranger, Experimental learning of a hyperelastic behavior with a physics-augmented neural network, *Exp. Mech.* 64 (9) (2024) 1465–1481, <http://dx.doi.org/10.1007/s11340-024-01106-5>.
- [54] K.P. Abdolazizi, R.C. Aydin, C.J. Cyron, K. Linka, Constitutive Kolmogorov–Arnold networks (CKANs): Combining accuracy and interpretability in data-driven material modeling, 2025, <https://arxiv.org/abs/2502.05682>, URL: <https://arxiv.org/abs/2502.05682>.
- [55] M. Grédiac, F. Pierron, S. Avril, E. Toussaint, The virtual fields method for extracting constitutive parameters from full-field measurements: a review, *Strain* 42 (4) (2006) 233–253, <http://dx.doi.org/10.1111/j.1475-1305.2006.tb01504.x>.
- [56] F. Pierron, M. Grédiac, *The Virtual Fields Method: Extracting Constitutive Mechanical Parameters from Full-field Deformation Measurements*, Springer New York, 2012, <http://dx.doi.org/10.1007/978-1-4614-1824-5>.
- [57] D. Claire, F. Hild, S. Roux, A finite element formulation to identify damage fields: the equilibrium gap method, *Internat. J. Numer. Methods Engrg.* 61 (2) (2004) 189–208, URL: <http://dx.doi.org/10.1002/nme.1057>.
- [58] J.M. Ball, Convexity conditions and existence theorems in nonlinear elasticity, *Arch. Ration. Mech. Anal.* 63 (4) (1976) 337–403, <http://dx.doi.org/10.1007/BF00279992>, URL: <http://link.springer.com/10.1007/BF00279992>.
- [59] J. Schröder, Anisotropic polyconvex energies, in: *CISM International Centre for Mechanical Sciences*, Springer Vienna, 2010, pp. 53–105, http://dx.doi.org/10.1007/978-3-7091-0174-2_3.
- [60] C.B. Morrey, Quasi-convexity and the lower semicontinuity of multiple integrals, *Pacific J. Math.* 2 (1) (1952) 25–53, <http://dx.doi.org/10.2140/pjm.1952.2.25>.
- [61] S. Kumar, A. Vidyasagar, D.M. Kochmann, An assessment of numerical techniques to find energy-minimizing microstructures associated with nonconvex potentials, *Internat. J. Numer. Methods Engrg.* 121 (7) (2019) 1595–1628, <http://dx.doi.org/10.1002/nme.6280>.
- [62] S. Hartmann, P. Neff, Polyconvexity of generalized polynomial-type hyperelastic strain energy functions for near-incompressibility, *Int. J. Solids Struct.* 40 (11) (2003) 2767–2791, [http://dx.doi.org/10.1016/S0020-7683\(03\)00086-6](http://dx.doi.org/10.1016/S0020-7683(03)00086-6), URL: <https://www.sciencedirect.com/science/article/pii/S0020768303000866>.
- [63] D.K. Klein, R. Ortigosa, J. Martínez-Frutos, O. Weeger, Finite electro-elasticity with physics-augmented neural networks, *Comput. Methods Appl. Mech. Engrg.* 400 (2022) 115501, URL: <http://dx.doi.org/10.1016/j.cma.2022.115501>.
- [64] D. Balzani, P. Neff, J. Schröder, G.A. Holzapfel, A polyconvex framework for soft biological tissues. Adjustment to experimental data, *Int. J. Solids Struct.* 43 (20) (2006) 6052–6070.
- [65] S. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
- [66] C. De Boor, On calculating with B-splines, *J. Approx. Theory* 6 (1) (1972) 50–62.
- [67] A. Isihara, N. Hashitume, M. Tatibana, Statistical theory of rubber-like elasticity. IV. (Two-dimensional stretching), *J. Chem. Phys.* 19 (12) (1951) 1508–1512, <http://dx.doi.org/10.1063/1.1748111>, URL: <https://aip.scitation.org/doi/10.1063/1.1748111>. Publisher: American Institute of Physics.
- [68] D.W. Haines, W.D. Wilson, Strain-energy density function for rubberlike materials, *J. Mech. Phys. Solids* 27 (4) (1979) 345–360, [http://dx.doi.org/10.1016/0022-5096\(79\)90034-6](http://dx.doi.org/10.1016/0022-5096(79)90034-6), URL: <https://www.sciencedirect.com/science/article/pii/0022509679900346>.
- [69] A.N. Gent, A.G. Thomas, Forms for the stored (strain) energy function for vulcanized rubber, *J. Polym. Sci.* 28 (118) (1958) 625–628, <http://dx.doi.org/10.1002/pol.1958.1202811814>, eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/pol.1958.1202811814>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/pol.1958.1202811814>.
- [70] E.M. Arruda, M.C. Boyce, A three-dimensional constitutive model for the large stretch behavior of rubber elastic materials, *J. Mech. Phys. Solids* 41 (2) (1993) 389–412, [http://dx.doi.org/10.1016/0022-5096\(93\)90013-6](http://dx.doi.org/10.1016/0022-5096(93)90013-6), URL: <https://www.sciencedirect.com/science/article/pii/0022509693900136>.
- [71] R.W. Ogden, R. Hill, Large deformation isotropic elasticity – on the correlation of theory and experiment for incompressible rubberlike solids, *Proc. R. Soc. A* 326 (1567) (1972) 565–584, <http://dx.doi.org/10.1098/rspa.1972.0026>, URL: <https://royalsocietypublishing.org/doi/10.1098/rspa.1972.0026>. Publisher: Royal Society.
- [72] K.R. Butterfield, The computation of all the derivatives of a B-spline basis, *IMA J. Appl. Math.* 17 (1) (1976) 15–25, URL: <http://dx.doi.org/10.1093/imamat/17.1.15>.
- [73] S. Eilfwing, E. Uchibe, K. Doya, Sigmoid-weighted linear units for neural network function approximation in reinforcement learning, *Neural Netw.* 107 (2018) 3–11, URL: <http://dx.doi.org/10.1016/j.neunet.2017.12.012>.