Material-embedding Physics-Augmented Neural Networks: A first application to constitutive law parameterization

Clément Jailin, E. Baranger

Computer Methods in Applied Mechanics and Engineering 445 (2025) 118188

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

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

Material-embedding Physics-Augmented Neural Networks: A first application to constitutive law parameterization

C. Jailin *, E. Baranger

Université Paris-Saclay, CentraleSupélec, ENS Paris-Saclay, CNRS, LMPS – Laboratoire de Mécanique Paris-Saclay, Gif-sur-Yvette, 91190, France

ARTICLE INFO

Keywords:
Machine learning
Physics-informed AI
Constitutive modeling
PANN
Latent space

ABSTRACT

A novel Material-Embedding Physics-Augmented Neural Network (ME-PANN) framework is presented for constitutive modeling in isotropic elasticity. A self-supervised trainable embedding vector is introduced to capture material-specific information. The embedding vector is concatenated with the inputs of each fully connected layer, allowing the network to adapt its response based on the material behavior. Synthetic datasets of various hyperelastic laws are used to assess accuracy and the ME-PANN capacity to generalize across diverse materials. It is demonstrated that the proposed framework effectively encodes model shape and parameter variations, thereby achieving improved generalization compared to standard neural networks for a negligible extra computational cost. A single-trained generic model can handle multiple materials by maintaining a shared set of weights while adapting only the low-dimensional embedding for each new scenario. This approach also yields significant gains in data efficiency since fewer parameters must be updated when transfer learning is performed on new materials. Although this study primarily relies on synthetic datasets, future work will focus on validating the proposed framework using real experimental data, which introduces additional challenges, such as accurate stress measurement.

1. Introduction

Constitutive models are the backbone of material behavior prediction in computational mechanics. Traditionally, these models are built upon a fixed mathematical form, the *model shape*, and are subsequently calibrated by adjusting a set of material parameters to fit experimental data. Techniques such as Finite Element Model Updating (FEMU) [1], Modified Constitutive Relation Error (MCRE) [2], the Virtual Fields Method (VFM) [3,4], and equilibrium gap/EUCLID approaches [5,6] focus on the identification of these parameters, while the model shape is assumed to be known and remains unchanged. Therefore, relaxing the model shape by leveraging data-driven learning is increasingly recognized as essential for improving the accuracy and adaptability of constitutive models. However, by doing this, the model shape and the model parameters become intrinsically mixed.

Recent advances in artificial intelligence have demonstrated that machine learning-based approaches, particularly neural networks, can be used to learn constitutive laws directly from data [7,8]. These approaches leverage high-dimensional function approximation and data-driven feature extraction to model complex material behaviors. Challenges in understanding constitutive behavior, such as physical inconsistencies and limited data access, can be addressed by integrating machine learning models

* Corresponding author.
E-mail address: clement.jailin@ens-paris-saclay.fr (C. Jailin).

https://doi.org/10.1016/j.cma.2025.118188
Received 18 March 2025; Received in revised form 3 June 2025; Accepted 24 June 2025
Available online 3 July 2025

with physical information [9] through various strategies. Some methods incorporate physical knowledge into the neural network's loss function, combining a discrepancy measure and penalties for deviations from physical laws [10,11], for example with the Thermodynamics-based Artificial Neural Networks (TANN) [12,13]. The "Physics-Augmented Neural Network" (PANN) strategy integrates physical knowledge into the model's architecture [14–18], often using Input-Convex Neural Networks (ICNN) to maintain the convexity of free energy [19].

All those techniques train a machine learning (ML) model to learn a specific material behavior [20]. Those approaches thus require a large probing of deformation states to properly identify the material across its full response range for each new training and may lose genericity. In contrast, classical constitutive models, based on fixed analytical forms, are often designed to describe entire families of materials or material grades using a small set of interpretable parameters. This distinction highlights a key limitation of conventional ML-based models: while they offer flexibility, they often lack the transferability that is inherent to classical formulations.

For real experimental applications (or synthetic works mimicking experimental data availability), data are coming from rich full-field measurement techniques, such as digital image [21] or volume correlation [22], or infra-red measurements, providing displacement and thermal fields, and global force sensors. Those approaches allow extracting the tested material states (*e.g.,* deformation states, thermal states) in a so-called material phase space [23].

Data availability poses an additional challenge: while the model should ideally be trained on a large portion of the material's phase space to ensure broad coverage, in practice, the probed domain is often restricted by experimental constraints. As a result, certain deformation or loading regimes may be underrepresented, thereby limiting the training data variability or statistically driving the model to a biased identification region. When the model is evaluated in the test beyond these probed regions, generalization is not guaranteed, and extrapolations may be unreliable.

Real-world applications reveal that both the model shape and its parameters can vary significantly between material batches. This variability makes retraining or re-identifying parameters prohibitively expensive—especially when data are limited (only a limited volume of coupons may be available due to cost or manufacturing processes) or when many materials require simultaneous analysis. Recent studies in PANN have reported that fine-tuning a model without having to retrain all weights can be sensitive to initialization and prone to convergence toward local minima, even in purely elastic settings (see the thesis [24], Chapter 4.1). This provides strong motivation for an approach that learns both the general shape of the constitutive model and the material-specific modifications needed for any particular test.

For simplicity, our approach uses supervised training, where both full-field displacement and stress data are available through synthetic data generation. Although stress states are not measurable experimentally, unsupervised learning methods such as EUCLID [6,25], NN-EUCLID [26], modified Constitutive Relation Error [27,28], equilibrium gap [5] and Equilibrium-based Convolution Neural Network (ECNN) methods [29] have been developed. Their added complexity, evaluated also experimentally [29–31], is not the focus of this study. Instead, by adopting a supervised framework, we concentrate on integrating a material-specific latent embedding for adaptive constitutive modeling while recognizing that unsupervised techniques remain a viable solution for more realistic experimental situations.

In this work, a *material-embedding PANN (ME-PANN)* that incorporates a low-dimensional latent vector to represent material-specific behavior information is proposed. During a simultaneous training phase on multiple datasets, each material is associated with a distinct embedding vector. This low dimensional vector is trained along with the main network weights so that the global part of the network captures generic constitutive laws while the small embedding encodes the material-specific parameters or shape refinements [32,33], ensuring the classical split between the model shape and the material parameters. In this framework, the constitutive model is viewed as being composed of two parts: (i) the model shape description, which represents general forms of the constitutive laws, and (ii) the model parameters, which adapt this shape and amplitudes to the current material. By introducing a material-specific latent embedding, additional information is encoded, capturing, in a very tiny space, both the model shape and material parameter. Consequently, when adapting the model to a new material, only the low-dimensional embedding could be fine-tuned, while the majority of the network (i.e., the learned generic model shape) remains fixed. Outside mechanics, this strategy has already been used to learn a compact user-embedding for personalized movie recommendations [34] or to learn radiologist bias when analyzing mammography images [35]. An important question arises about how well these models extrapolate beyond the range of experimentally accessible loading states.

The model introduces a slight increase of trainable parameters compared to standard PANN while providing significantly improved generalization across materials. This is also due to the addition of low-dimensional embedding vectors, which represent only a small fraction of the total trainable parameters. The marginal increase in computation is justified by the substantial gain in adaptability and transfer learning efficiency. The approach offers several advantages over state-of-the-art models: (i) a flexible calibration with minimal accessible data, and (ii) shared training and generalization from multiple material datasets.

The paper is organized as follows. Section 2 details the method, including the PANN formulation for hyperelasticity and our embedding strategy. Section 3 describes the four synthetic datasets, along with the generation of full-field displacement and stress fields for various materials. Those four datasets allow challenging the approach and provide a clear understanding of each ME-PANN advantage. Section 4 presents numerical results for different test scenarios, highlighting how the embedding space effectively learns both material parameters and model shapes. Finally, Section 5 provides a discussion and concluding remarks on the advantages and future prospects of material-embedding PANN architectures.

2. Method

2.1. Constitutive models

2.1.1. PANN architecture model

The neural network model used in this application is a PANN designed for isotropic hyperelastic applications. Its architecture is based on a neural network composed of dense layers that take as inputs the deformation tensor $\boldsymbol{F}$ and outputs the first Piola–Kirchhoff stresses $\boldsymbol{P}$.

The strain energy density is written as a function of the Green–Lagrange strain tensor $\boldsymbol{E} = (\boldsymbol{C} - \boldsymbol{1})/2$, which represents a sufficient condition for objectivity ($\boldsymbol{C} = \boldsymbol{F}^T \cdot \boldsymbol{F}$ being the right Cauchy–Green deformation tensor). Following [17], the strain energy density is written as:

$$W(\boldsymbol{F}) = W_{\text{ICNN}}(\boldsymbol{F}) - W_{\boldsymbol{F}=\boldsymbol{1}} - \eta\,(J-1) \quad . \tag{1}$$

The first term corresponds to the energy output of an input convex neural network $W_{\text{ICNN}}(\boldsymbol{F})$ [19], and the last two terms correspond to energy and stress normalization. Various physical constraints are implemented following reference papers in the literature [15,17]:

- Manual definition of the deformation invariants before the first layer to enforce isotropy. The deformation invariants are defined as

$$I_1 = \text{tr}(\boldsymbol{C}), \quad I_2 = \text{tr}\,(\text{cof}(\boldsymbol{C}))\,, \quad I_3 = \det(\boldsymbol{C}) = J^2 \quad . \tag{2}$$

  Note that the invariant I_1 is convex in $\boldsymbol{F}$, the invariant I_2 is convex in $\text{cof}(\boldsymbol{F})$, and the invariant I_3 is convex in $\det(\boldsymbol{F})$. The invariants layer receives the deformation gradient $\boldsymbol{F}$ from the input layer and computes strain invariants.
- Thermodynamic consistency: the first Piola–Kirchhoff stress is obtained by derivation of the strain energy density function W: $\boldsymbol{P}(\boldsymbol{F}) = \dfrac{\partial W(\boldsymbol{F})}{\partial \boldsymbol{F}}$. An intermediate output of the Neural Networks is hence a scalar value W.
- Stability of the potential is obtained by the polyconvexity with respect to $\boldsymbol{F}$, $\det(\boldsymbol{F})$ and $\text{cof}(\boldsymbol{F})$ by using the convex and non-decreasing *softplus* activation functions for the hidden layers [19]: $\mathcal{A}(y) = \log(1 + e^y)$ and non-negative model weights $\boldsymbol{\omega} \geq 0$. In order to satisfy the convexity and monotonicity constraints of the strain energy, all the invariants I_1, I_2, and I_3 are typically modeled as convex and monotonically non-decreasing. However, due to this overconstraint (the monotonically increasing constraint is not required in J), these models are unable to represent negative stresses [15,17], which leads to the addition of an extra invariant $I_1^* = -2J$ to the input set: (I_1, I_2, I_3, I_1^*). The relationship between two hidden layers, composed of N_n neurons can hence be written, with $\boldsymbol{\omega}^i$ and b^i the weights and bias of layer i, and $\boldsymbol{z}^{i-1}$ and $\boldsymbol{z}^i$ the neuron values of layers $i-1$ and i

$$\boldsymbol{z}^{(i)} = \mathcal{A} \circ \left[\sum_{k=1}^{N_n} \boldsymbol{\omega}_{jk}^{(i)} z_k^{(i-1)} + b_j^{(i)} \right] \quad . \tag{3}$$

- Normalization of the energy and stresses: $\boldsymbol{P}(\boldsymbol{F} = \boldsymbol{1}) = \boldsymbol{0}$, and $W(\boldsymbol{F} = \boldsymbol{1}) = 0$. This normalization is obtained with the two last terms in the energy formulation, with $W_{\boldsymbol{F}=\boldsymbol{1}} = W_{\text{ICNN}}(\boldsymbol{E}(\boldsymbol{1}))$ being the energy at $\boldsymbol{F} = \boldsymbol{1}$, and η the scalar

$$\eta = 2\left(\frac{\partial W_{\text{ICNN}}}{\partial I_1} + 2\frac{\partial W_{\text{ICNN}}}{\partial I_2} + \frac{\partial W_{\text{ICNN}}}{\partial I_3} + \frac{\partial W_{\text{ICNN}}}{\partial I_1^*}\frac{\partial I_1^*}{\partial I_3} \right)\Bigg|_{\boldsymbol{F}=\boldsymbol{1}} \quad . \tag{4}$$

  Since this term is combined with a linear function of J, it preserves the polyconvexity of the potential.

Unlike the work of Linka et al. [18] introducing an incompressibility constraint, here, compressible behaviors will be studied. In this study, no explicit volumetric growth condition (*e.g.,* $W \to \infty \quad \text{as} \quad J \to 0^+$) is enforced [17].

The core reference PANN architecture comprises the first hard-coded layer that computes the deformation invariants. This feature computation is followed by $N_{hl} = 2$ hidden Fully Connected (FC) layers, each composed of $N_n = 16$ neurons and *Softplus* activation functions. The last top layer links the last layer neurons to the scalar energy value with a *Softplus* activation function. This energy is finally differentiated with respect to the deformation tensor to extract the stresses. A simple number-of-layer-sensitivity study has been performed on this architecture, which gives good results on the presented reference architecture.

The classical supervised training loss is written as a Frobenius norm (*i.e.,* the square root of the sum of the absolute squares of each tensor element), being written with the ground truth $\boldsymbol{P}_{\text{GT}}$ and with $\mathcal{M}_{\text{PANN}}$ the application that output stresses as: $\|P_{\text{ME-PANN}} - P_{\text{GT}}\|_F = \sqrt{\sum_{i,j}(P_{\text{ME-PANN}}^{ij} - P_{\text{GT}}^{ij})^2}$. The weights and biases of the model are estimated by minimizing

$$\{\hat{\boldsymbol{w}}, \hat{\boldsymbol{b}}\} = \underset{\boldsymbol{w}>\boldsymbol{0}, \boldsymbol{b}}{\text{Argmin}} \; \|\mathcal{M}_{\text{PANN}}(\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{F}) - \boldsymbol{P}_{\text{GT}}\|_F \quad . \tag{5}$$

2.2. Material embedding PANN

In addition to the main PANN structure, a low-dimensional self-supervised trainable latent vector, $\boldsymbol{e}_m \in \mathbb{R}^d$, is introduced for each distinct material dataset m. This embedding acts as a compact representation of material-specific variations, capturing

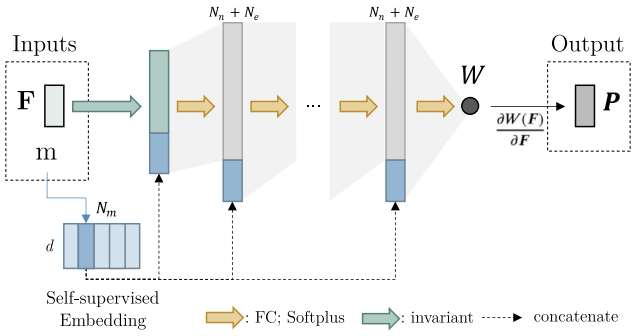


Fig. 1. Architecture of the ME-PANN, in which a low-dimensional latent vector is concatenated with the input of each fully connected layer.

key constitutive characteristics such as elasticity, nonlinearity, and anisotropy. With N_m, the number of different materials, the embedding is of size $(d \times N_m)$. We treat $\{\boldsymbol{e}_m\}$ as extra parameters to be learned jointly with the main network weights. This model is referred to in this paper as material-embedding PANN, ME-PANN.

As illustrated by Fig. 1, at each FC layer, the current embedding vector $\boldsymbol{e}_m$ is concatenated with the layer inputs. This design, inspired by Densely Connected architectures [36], ensures that each layer can adapt its response on the material-specific embedding $\boldsymbol{e}_m$. A summary of the model architecture is presented in Fig. 1.

The ME-PANN dense layers can be written with $\tilde{\boldsymbol{z}}^{(i)} = \{\boldsymbol{z}^{(i)}, \boldsymbol{e}_m\}$ the concatenated neurons at layer i, for the material m

$$\boldsymbol{z}_j^{(i)} = \mathcal{A} \circ \left[\sum_{k=1}^{N_n+d} \boldsymbol{\omega}_{jk}^{(i)} \tilde{\boldsymbol{z}}_k^{(i-1)} + b_j^{(i)} \right] \quad . \tag{6}$$

Thus, being connected to all FC layers, the extra number of trainable parameters due to the embedding, N_{extra}, in the networks is

$$N_{\text{extra}} = N_{\text{hl}} \cdot N_{\text{n}} \cdot d + (d+1) N_m \quad . \tag{7}$$

With this formulation, the ME-PANN is convex and monotonic with respect to the embedding vector. Note that this current choice could be challenged with recent input-specific neural network architectures [37]. It can be noted that ME-PANN adds a small number of trainable parameters (for a negligible computation cost) while enabling each fully connected layer to adapt efficiently to material-specific characteristics.

The model is trained in a supervised manner, assuming for each data point both the deformation $\boldsymbol{F}$ and stress $\boldsymbol{P}$ are known (*e.g.,* from synthetic simulations) and tuning the network weights and biases (common for all tests) as well as the embedding vector (different for each material). Although measuring stress experimentally is not possible, this assumption simplifies initial development. Future extensions or alternative frameworks (*e.g.,* EUCLID [6,25], NN-EUCLID [26], modified Constitutive Relation Error [27,28] and Equilibrium-based Convolution Neural Network (ECNN) methods [29]) can infer stress data by leveraging equilibrium constraints.

The ME-PANN loss is written as a main mean squared error loss term over the stresses compared with the ground truth $\boldsymbol{P}_{\text{GT}}$

$$\mathcal{L}_{\text{ME-PANN}} = \| \mathcal{M}_{\text{ME-PANN}}(m, \boldsymbol{w}, \boldsymbol{b}, \boldsymbol{e}, \boldsymbol{F}) - \boldsymbol{P}_{\text{GT}}(m) \|_F \quad , \tag{8}$$

with no loss term applied to the energy term, which is also an output of the NN. Here, m denotes the material index corresponding to the current data sample. It is not a trainable parameter, but rather an identifier used to select the appropriate embedding vector during training.

2.3. Regularizers for material embeddings

In many applications, it is crucial that the learned embedding vectors capture distinct and disentangled features so that different physical parameters (e.g., Young's modulus, Poisson's ratio, and model type) are represented in a meaningful way. Regularization terms guide the learning process by imposing additional constraints on the latent space, helping to enforce a structured representation [38,39]. Two regularizers designed for material embeddings are described with $e_m \in \mathbb{R}^{d \times N_m}$ and therefore can act globally on all the material samples.

Orthogonalization. This regularizer is designed to reduce the scalar product between all embeddings penalizing orthogonality between the different embedding modes. The cosine similarity between two normalized embeddings is defined as

$$S_{i,j} = \frac{\mathbf{e}^i \cdot \mathbf{e}^j}{\|\mathbf{e}^i\|_2 \|\mathbf{e}^j\|_2} \quad . \tag{9}$$

The disentanglement regularization loss is given by the mean squared off-diagonal elements of the similarity matrix:

$$\mathcal{L}_{\text{ortho}} = \frac{1}{d(d-1)} \sum_{\substack{i,j \\ i \neq j}}^{d} S_{i,j}^2 \quad . \tag{10}$$

Kullback–Leibler (KL) divergence. This regularizer is designed to match the embedding to statistically independent normal distributions [40]. This type of regularizer works when enough material N_m allows for statistical quantity to be extracted. With μ and σ the mean and standard deviation

$$\mu_j = \frac{1}{N_m}\sum_{m=1}^{N_m} e_m^j, \quad \sigma_j^2 = \frac{1}{N_m}\sum_{m=1}^{N_m}\left(e_m^j - \mu^j\right)^2 \quad , \tag{11}$$

the KL divergence between the empirical distribution in dimension j (assumed Gaussian) and a standard normal distribution is

$$\mathcal{L}_{\text{KL}} = \sum_{j=1}^{d} -\frac{1}{2}\left(1 + \log\left(\sigma_j^2 + \epsilon\right) - \mu_j^2 - \sigma_j^2\right) \quad . \tag{12}$$

The KL regularization enforces a structured Gaussian prior, which helps in generating embeddings that remain smoothly distributed across different materials, reducing abrupt discontinuities in the latent space. The orthogonalization term prevents overlap between embedding dimensions, ensuring that variations in one latent component do not interfere with others.

The optimization consists of minimizing:

$$\{\hat{\boldsymbol{w}}, \hat{\boldsymbol{b}}, \hat{\boldsymbol{e}}\} = \underset{\boldsymbol{w}>\boldsymbol{0},\boldsymbol{b},\boldsymbol{e}}{\text{Argmin}}\ \mathcal{L}_{\text{ME-PANN}} + \alpha\mathcal{L}_{\text{Reg}} \quad , \tag{13}$$

with $\mathcal{L}_{\text{Reg}}$ being one or a combination of the previously described regularizer and α, a hyperparameter weight that controls the disentanglement of the embedding. In the following application, $\alpha = 0.1$ and the regularizer is

$$\mathcal{L}_{\text{Reg}} = \mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{ortho}} \quad . \tag{14}$$

The weighting coefficient was selected empirically to ensure that the contributions of the ME-PANN loss (data fitting term) and the regularization terms (KL divergence and orthogonality) are of comparable magnitude during training. This balance prevents either component from dominating the optimization process. The choice of equal regularizer weightings is a first step and may lead to further work. The optimization is performed with ADAM [41], with a piece-wise constant learning rate scheduler starting at $l_r = 10^{-2}$ and dividing by 10 every 10k epochs. The model is trained during 40k epochs, with a batch size of 1000 (note that in the supervised approach, one forward pass refers to one deformation-stress couple) such that each step achieves convergence before the learning rate is reduced. No early stop was used as the model did not show overfitting and converged smoothly due to the highly regularized nature of PANN. The neural network architecture is developed using the TensorFlow library in Python and trained with an 8 Gb GPU unit (Nvidia RTX-A2000 Mobile). Each training took approximately 30 min.

It should be emphasized that the chosen embedding dimension d influences both model capacity and interpretability. Although higher d can capture finer variations among materials, it may also obscure the physical meaning of each latent dimension. Future work may apply more advanced regularization or manifold learning techniques to encourage structured embeddings.

2.4. Fine-tuning embedding

In addition to the main training scenarios, a targeted fine-tuning experiment was performed to isolate the learning of the embedding vector in a new material identification. In this setting, a new synthetic dataset was generated with the same type of constitutive models and different material parameters, leading to different stress–strain data. The ME-PANN was initialized using the fully pre-trained weights, and only the embedding vector was re-optimized to match the new data, while all other network weights were kept fixed. This focuses adaptation exclusively on the low-dimensional embedding, thereby emulating transfer learning to a new material instance.

This setup allows assessing the ability of the embedding vector alone to capture new material-specific variations when the underlying constitutive structure is presumed to be already well modeled.

2.5. Evaluation metrics

To quantitatively assess the performance of the proposed ME-PANN, we employ several metrics computed on the validation and test datasets. Since the synthetic test cases provide the ground-truth free energy function W_{GT}, the following metrics are defined:

- **Energy error:** The error in the predicted free energy is measured using an Euclidean norm $\|\bullet\|_2$:

$$E_W = \frac{\left\|W_{\text{ME-PANN}} - W_{\text{GT}}\right\|_2}{\left\|W_{\text{GT}}\right\|_2} \quad . \tag{15}$$

- **Relative maximal energy error:** The relative error in energy is defined as:

$$E_{\text{rel}} = \frac{\left\|W_{\text{ME-PANN}} - W_{\text{GT}}\right\|_2}{\max(W_{\text{GT}})} \quad . \tag{16}$$

  This metric gives an easier visual evaluation of the error amplitudes.

- **Stress error:** The error in the predicted stress is measured using an Frobenius norm $\|\bullet\|_F$:

$$E_P = \frac{\|P_{\text{ME-PANN}} - P_{\text{GT}}\|_F}{\|P_{\text{GT}}\|_F} \quad . \tag{17}$$

  Note that this metric is proportional to the ME-PANN loss term.

3. Data

To validate the proposed ME-PANN framework, we generate a synthetic dataset covering a wide range of material behaviors and parameter variations. The dataset is structured into three distinct cases to systematically assess the model's ability to capture (i) continuous parameter variations within a single model, (ii) both discrete model shape changes and parameter variations, and (iii) the benefits of training on multiple datasets that probe different regions of the strain phase space.

3.1. Traditional hyperelastic models

Standard isotropic hyperelastic models are used as a reference. In these models, the strain energy density function is written with the deformation invariants and μ and λ the Lamé coefficients. Synthetic data are generated using five constitutive models:

- Model 0 (Neo-Hookean):

$$W_{\text{NH}} = \frac{\mu}{2}(I_1 - 3) - \mu \ln J + \frac{\lambda}{2}(\ln J)^2 \quad . \tag{18}$$

- Model 1 (Neo-Hookean variant):

$$W_{\text{NHv}} = \frac{\mu}{2}(I_1 - 3) - \mu \ln J + \frac{\lambda}{2}(J - 1)^2 \quad . \tag{19}$$

- Model 2 (Arruda–Boyce, series expansion):

$$W_{\text{AB}} = \mu\left[\frac{1}{2}(I_1 - 3) + \frac{1}{20}(I_1 - 3)^2 + \frac{11}{1050}(I_1 - 3)^3\right] - \mu \ln J + \frac{\lambda}{2}(\ln J)^2 \quad . \tag{20}$$

- Model 3 (Gent, with $J_m = 10$):

$$W_{\text{Gent}} = -\frac{\mu J_m}{2} \ln\left(1 - \frac{I_1 - 3}{J_m}\right) - \mu \ln J + \frac{\lambda}{2}(\ln J)^2 \quad , \tag{21}$$

- Model 4 (Mooney–Rivlin, with I_2 dependency):

$$W_{\text{MR}} = \frac{\mu}{2}(\tilde{I}_1 - 3) + \frac{\mu}{4}(\tilde{I}_2 - 3) + \frac{\lambda}{2}(\ln J)^2 \quad . \tag{22}$$

  with $\tilde{I}_1 = I_1 J^{-2/3}$ and $\tilde{I}_2 = I_2 J^{-4/3}$ considering a deviatoric/volumetric partition.

The responses of the models for a simple uniaxial synthetic loading are presented in Fig. 2. For this computation, a deformation tensor $\mathbf{F}_{\text{uniaxial}}$, composed of α and β amplitudes, was optimized so that the radial components of the stresses vanish.

$$\mathbf{F}_{\text{uniaxial}} = \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & \beta \end{bmatrix} \tag{23}$$

The optimization was performed for each loading amplitude β and each model.

3.2. Data generation and cases

Data were generated to explore three main scenarios, here referred to as *Case 1* through *Case 3*. The following paragraphs summarize the procedures for each case, including parameter ranges, sample sizes, and the types of experiments considered.

While the current study is based on synthetic datasets, we acknowledge that real experimental data may introduce additional complexities, such as noise and measurement uncertainties [42,43]. Future extensions of this work will incorporate real-world data to further validate the model's robustness to noise.

Case-1 — Encoding material parameters. A single Neo-Hookean law (Model-0) is adopted, but parameter variations (E, ν) are sampled. This tests whether the embedding vector can capture continuous parameter changes within the same fundamental model shape.

Normal random samples of Young's modulus E and Poisson's ratio ν were drawn from distributions with means of 15 and 0.25 and standard deviations of 4 and 0.1, respectively. They were subsequently clipped to the ranges $E \in [0.1, \infty)$ and $\nu \in [0, 0.45]$ to keep physical consistency. Note that this Gaussian sampling is done on E and ν and not on the Lamé parameters that would give an easier control on the hyperelastic models.

With $\mathcal{R}(F_{\min}, F_{\max})$ a uniformly random sampling function, in the interval $[F_{\min}, F_{\max}]$, the generated set of deformation tensors were

$$\{\boldsymbol{F}\} \sim \begin{bmatrix} \mathcal{R}(0.7, 1.5) & \mathcal{R}(-0.15, 0.15) & \mathcal{R}(-0.15, 0.15) \\ \mathcal{R}(-0.15, 0.15) & \mathcal{R}(0.7, 1.5) & \mathcal{R}(-0.15, 0.15) \\ \mathcal{R}(-0.15, 0.15) & \mathcal{R}(-0.15, 0.15) & \mathcal{R}(0.7, 1.5) \end{bmatrix} \quad , \tag{24}$$

A total of 1250 data points were included in the training set, of which 20% were used for validation, and 500 additional data points were allocated for testing. This sampling strategy was designed to investigate whether a single constitutive model could capture

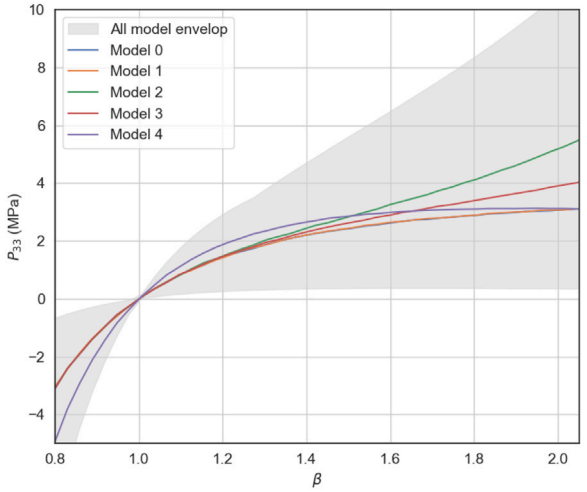


Fig. 2. Uniaxial response of the 5 hyperelastic models, with the same model parameters ($E = 10$ MPa and $\nu = 0.4$). This figure highlights the different model shapes in 1D. The light gray beam shows the whole diversity of model responses when sampling the two model parameters.

continuous variations in material parameters through an embedding-based approach. The embedding dimension was evaluated, $d \in \{0, 1, 2, 3, 4\}$, using the same training, validation, and test sets.

Fig. 3(a) illustrates the energy distributions in terms of a selected invariant for all data points. Note that those figures are only shown for visualization and restricted to the I_1, J plane.

Case-2a — Encoding model shapes and material parameters — 25 materials. All five hyperelastic models are sampled, including variations in (E, ν). This setup tests whether the embedding distinguishes fundamentally different strain-energy forms in addition to parameter changes.

Specifically, we use five hyperelastic models: two Neo-Hookean model variants (NH), Arruda–Boyce, Gent ($J_m = 10$) and Mooney–Rivlin. Each is discretized over a range of parameters (E, ν), where E is the Young's modulus and ν is the Poisson ratio. The same ranges and sampling rules of $E \in [5, 15]$ and $\nu \in [0.2, 0.4]$ were used, but five different hyperelastic models were also introduced. For each model, five distinct materials were defined by selecting five parameter pairs (E, ν). This led to $m = 25$ total parameter-model combinations. The embedding dimension was evaluated, $d \in \{0, 2, 4, 6, 8, 10, 12\}$ using the same training, validation, and test sets.

For Case-2a, from each of these 25 combinations, 50 data points (stress–strain couples) were generated for training (for a total of 1250 training data points overall, with 20% selected for validation). Another 1000 data points in total were prepared for final testing. Data were generated with the same functions as in Case-1.

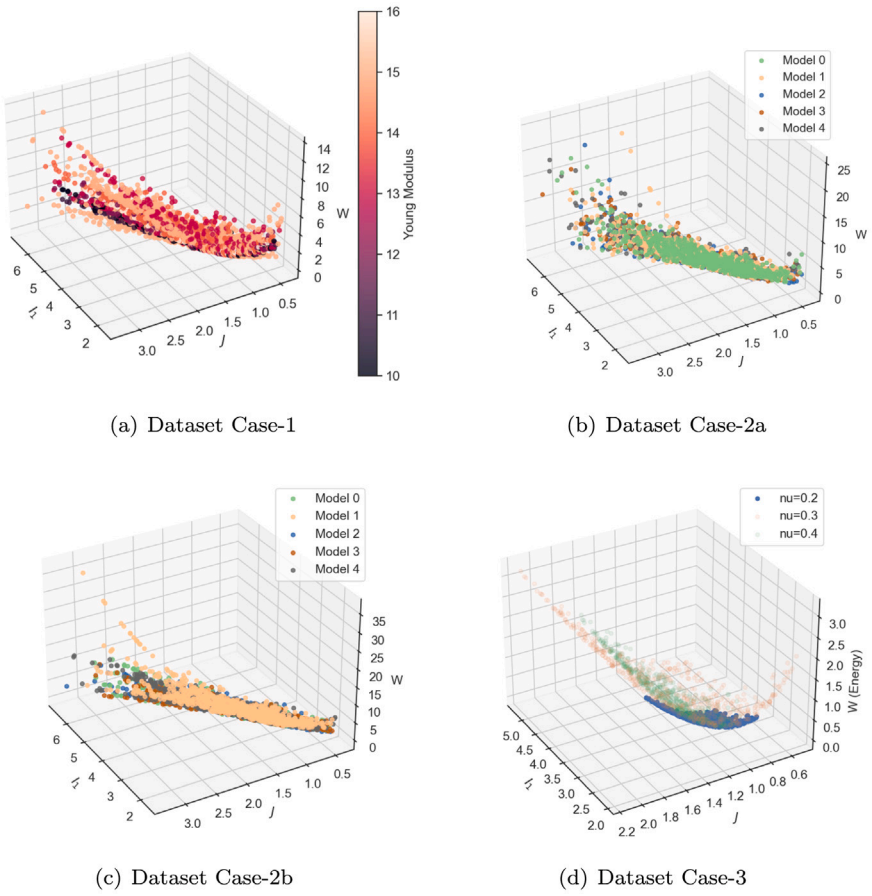
Case-2b — Encoding model shapes and material parameters — 200 materials. A second dataset with 200 materials from the 5 models, each comprising only 10 data points, is generated and referred to as *Case-2b*. Data were generated using the same deformation sampling as in Case-1. An embedding of $d = 6$ was chosen, from the results of Case-2a. Fig. 3(b–c) illustrates the energy distributions in terms of a selected invariant for all data points across the five models for cases 2a, 2b and 2c.

Case-2c — Embedding adaptation. To further evaluate the embedding's role in material identification and model adaptability, a complementary experiment was conducted based on the fine-tuning of Case-2b model. A new dataset was generated using different random parameters and the same five hyperelastic models. Using the ME-PANN trained on the 200 materials in Case-2b, only the embedding vector for the new material was fine-tuned using the new stress–strain data, with all network weights frozen.

This setup isolates the embedding as the only trainable component, allowing material-specific identification without altering the shared constitutive model. Each material's embedding is thus trained independently, with only 10 datapoints, with no cross-material coupling during optimization. This experiment demonstrates the potential of the ME-PANN framework for efficient transfer learning, enabling rapid adaptation to new materials with minimal computational overhead.

Case-3 — Generic dataset benefit. In this case, three small and distinct datasets are used, each of which probes a different region of the invariant space, the latter called *phase space*. It is generally recognized that a neural network remains valid mainly within the area where training data are available. Therefore, the objective is to determine whether a common training process with an embedding-enhanced architecture can learn effectively from all experiments at once, rather than learning only one experiment at a time.

Three materials were considered in this scenario, each corresponding to a fixed Young's modulus E and distinct Poisson ratios $\nu \in \{0.2, 0.3, 0.4\}$. The same Neo Hookean hyperelastic model (model 0) was used in all three materials, but different loading paths were introduced. For clarity and simplicity in visualization, we adopt a plane strain hypothesis. Under this assumption, the strain energy density function depends primarily on the first invariant I_1 and the Jacobian J (with I_2 being a linear combination of these invariants). With $\mathcal{R}(F_{\min}, F_{\max})$ a uniformly random sampling function:



(a) Dataset Case-1 (b) Dataset Case-2a

(c) Dataset Case-2b (d) Dataset Case-3

Fig. 3. Energy values of the different material datasets plotted against a representative strain invariant showing non-univocal surfaces. 4 datasets are generated to evaluate the capacity of the model to learn and generalize across different materials.

- Dataset 1: 500 datapoints, with $\nu = 0.2$, sampled in:

$$\{\boldsymbol{F}_1\} \sim \begin{bmatrix} \mathcal{R}(0.8, 1.2) & \mathcal{R}(-0.1, 0.1) & 0 \\ \mathcal{R}(-0.1, 0.1) & \mathcal{R}(0.8, 1.2) & 0 \\ 0 & 0 & 1 \end{bmatrix} . \tag{25}$$

- Dataset 2, represented by a strong compressive state, 500 datapoints, with $\nu = 0.3$, sampled in:

$$\{\boldsymbol{F}_2\} \sim \begin{bmatrix} \mathcal{R}(0.7, 1.5) & \mathcal{R}(-0.2, 0.2) & 0 \\ \mathcal{R}(-0.2, 0.2) & \mathcal{R}(0.7, 1.5) & 0 \\ 0 & 0 & 1 \end{bmatrix} . \tag{26}$$

- Dataset 3, represented by a strong tension state, 500 datapoints, with $\nu = 0.4$, sampled in:

$$\{\boldsymbol{F}_3\} \sim \begin{bmatrix} \mathcal{R}(0.9, 1.35) & \mathcal{R}(-0.3, 0.3) & 0 \\ \mathcal{R}(-0.3, 0.3) & \mathcal{R}(0.9, 1.35) & 0 \\ 0 & 0 & 1 \end{bmatrix} . \tag{27}$$

The deformation amplitudes in dataset 1 are much smaller than in the other datasets. Thus, dataset 1 probes only a narrow region of phase space, whereas the other two datasets cover a much broader region, despite different materials.

Each dataset contributed 500 data points, resulting in 1500 training data points in total. This design was chosen to examine whether a joint training process could be applied when each experiment probes a distinct region of the strain phase space. To identify a physical univocal solution, an overlap was chosen between those 3 datasets. It can be seen that the Dataset 1 covers a very small phase space area. It is in this overlap that forces the ML model to learn an univocal representation into the embedding. In addition, 3 materials were generated to identify a single dimension embedding $d = 1$. Fig. 3(d) illustrates the energy distributions in terms of a selected invariant for all data points for the 3 materials. In addition to the different probed spaces, the slope of the energy surface is different due to the Poisson ratio change.

Table 1
α amplitudes of the generic model described in Eq. (28) to retrieve the 5 traditional hyperelastic behaviors used for the synthetic data generation.

Model	α_1	α_2	α_3	α_4	α_5	α_6
0	μ	0	λ	0	0	0
1	μ	0	0	λ	0	0
2	μ	μ	λ	0	0	0
3	0	0	λ	0	μ	0
4	0	0	λ	0	0	μ

Table 2
Summary of the application datasets.

Name	Models	# variable elastic parameter	d	N_m	# data per material	Total data
Case-1	0	2	{0,1,2,3,4}	25	50	1250
Case-2a	0,1,2,3,4	2	{0,2,4,6,8,10,12}	25	50	1250
Case-2b	0,1,2,3,4	2	6	200	10	2000
Case-2c	0,1,2,3,4	2	6	200	10	2000
Case-3	0	1	1	3	500	1500

Synthetic data space. In terms of dimensionality, the generated datasets present spaces of different intrinsic dimensions based on the defined cases. For Case 1, the space is composed of two dimensions corresponding directly to the two Lamé parameters. For Case-3, the dimensionality reduces to one, representing variations solely in the Poisson ratio.

The more complex Case-2 involves data generated from all five hyperelastic models and their respective two model parameters, resulting in a general representation requiring six dimensions. Explicitly, this six-dimensional space can be represented as follows:

$$\begin{aligned} W = {} & \alpha_1 \left[\frac{I-3}{2} - \ln J \right] + \alpha_2 \left[\frac{(I-3)^2}{20} + \frac{11(I-3)^3}{1050} \right] + \alpha_3 \left[\frac{(\ln J)^2}{2} \right] \\ & + \alpha_4 \left[\frac{(J-1)^2}{2} \right] + \alpha_5 \left[\frac{J_m}{2} \ln \left(1 - \frac{I-3}{J_m} \right) + \ln J \right] + \alpha_6 \left[\frac{1}{2}(\tilde{I}_1 - 3) + \frac{1}{4}(\tilde{I}_2 - 3) \right] \quad , \end{aligned} \tag{28}$$

where each α term encodes variations and interactions between the model parameters and the specific forms of the hyperelastic models considered, as shown in Table 1,

Note that the decomposition into distinct modes used here represents one possible basis for describing the complex structure of the constitutive model data space. Alternative bases might be selected, potentially incorporating additional constraints such as orthogonality to enforce the separation and/or interpretability of the different model components. Due to the inherent complexity of coupling between embedding vectors, which encode material-specific information, and the deformation invariants, the dense concatenation approach was adopted within the ME-PANN architecture. This architectural choice ensures each layer of the neural network fully exploits the joint information encoded in both embedding vectors and invariants, enabling accurate and efficient modeling of diverse constitutive behaviors across multiple materials.

Data summary. A summary of the 4 generated datasets and the chosen ME-PANN models is presented in Table 2

4. Results

The PANN model (without embedding) was initially trained independently on all model shapes and parameters to ensure that it has the capacity to learn the behavior. This shows that the NN has the right architecture to capture the constitutive laws.

Results – Case-1 — Encoding material parameters. The model was trained across multiple material datasets, synthetically composed of the same material model (Model-0) and probing different parameter variations E and ν. Convergence analysis demonstrated excellent agreement with ground truth data, yielding an energy norm comparable to independently trained models. These results confirm the ME-PANN's ability to learn and differentiate both material shape functions and parameter-dependent variations on a single constitutive law.

After training on Case-1, the extracted embedding vectors displayed clear separations when visualized against the known elastic parameters. The use of both KL divergence and orthogonalization regularizers guided the network to develop an embedding space in which distinct latent dimensions correlate with specific material properties. As illustrated in Fig. 4, a correlation is observed between the learned embedding components and the elastic parameters (e.g., Young's modulus and Poisson's ratio), confirming that the network effectively disentangles and encodes the key aspects of material behavior. These results not only validate the ME-PANN approach but also provide insight into how different dimensions of the embedding relate to physical properties.

This is further validated by the orthogonality constraint, which ensures independent embedding dimensions capture distinct material attributes. This structured embedding space improves generalization and interpretability. Although no linear relationships are expected and imposed, an univocal correlation is observed between the obtained embeddings and the reference model parameters. It can be noted that this solution is not unique and depends on the model architecture and non-linear transformations.

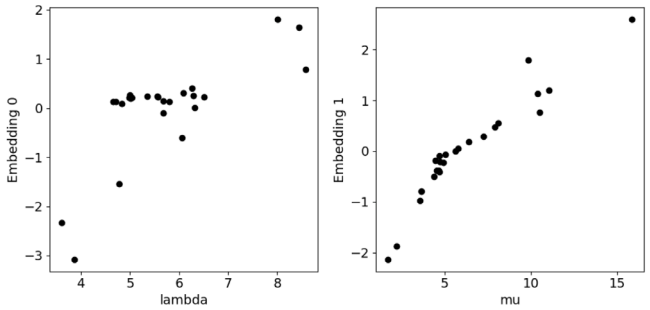


Fig. 4. Comparison of the first two embedding modes of the learned embedding with the two elastic parameters for the 25 different materials.

Table 3
Converged errors for Case-1 with different embedding sizes, estimated on the test set.

Embedding dimension d	Energy error E_W in %	Stress error E_P in %
0 (standard PANN)	41.1	39.6
1	3.0	2.2
2	**0.58**	**0.86**
3	0.59	1.01
4	0.61	0.96

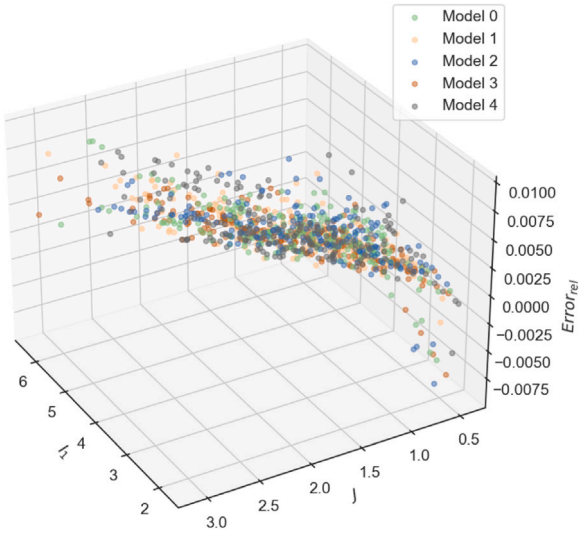


Fig. 5. Energy error, relative to the highest scalar value. Less than 1.3% of maximal error regarding the model and model parameters using an embedding of 6.

When changing the size of the embedding space, the energy error is presented in Table 3. A good reproduction of the energy is observed with an embedding of 2, corresponding to the true dimension of the problem. Adding more embedding dimensions does not improve. An embedding dimension of $d = 1$ only allows the capture of a partial behavior.

Results – Case-2a — Encoding model shapes. The model was trained on the different material datasets (with varying model shapes and model parameters) and converged toward almost perfect results (with an energy norm similar to the model trained independently on each dataset), highlighting the model's capacity to learn model shapes and parameter changes. The results in terms of error are presented in Fig. 5.

Multiple embedding sizes were tested and are presented in Table 4. A huge error drop appears when switching from PANN to ME-PANN as the error in energy decreases from 44.1% to 0.65%. It can be seen that the embedding already captures the 25 material behaviors with a size of $d = 4$. However, the best results were obtained with an embedding of 6.

Training data are combined from all five hyperelastic models, each sampled at different parameter values. Fig. 6 shows the embedding, with 25 materials and 6 dimensions in the embedding space. The embedding space encodes parameter changes and the underlying functional form of each model.

Table 4
Converged errors for Case-2a with different embedding sizes estimated on the test set.

Embedding dimension d	Energy error E_W in %	Stress error E_P in %
0 (standard PANN)	44.1	65.1
2	4.3	5.2
4	1.1	1.2
6	**0.65**	**0.85**
8	0.75	0.91
10	0.72	0.84
12	0.66	0.84

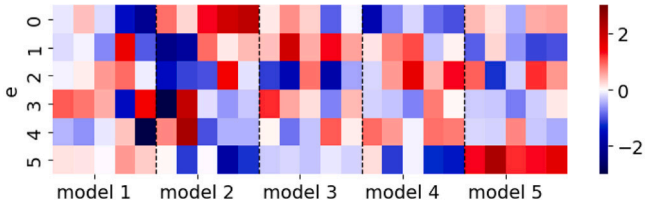


Fig. 6. Visual representation of the obtained embedding values $\{e\}$, where each column corresponds to a material and each row to an embedding component. This serves as a barcode-like identifier for each material. Distinct blocks are visible, reflecting clustering by model type. For example, embedding dimension 5 shows consistently positive values for materials from Model 5 and around 0 values for Model 3.

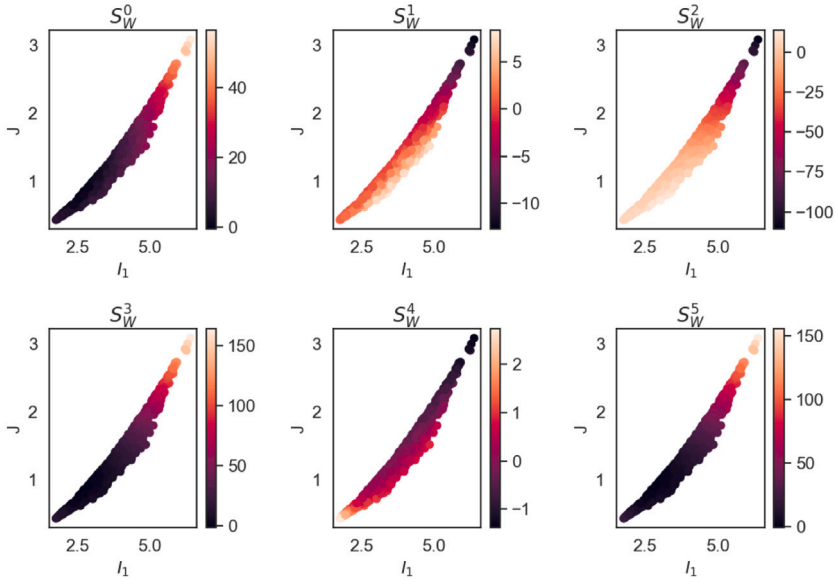


Fig. 7. Sensitivities S_W^i of the energy with respect to the embedding values..

To visually assess the impact of the embedding on the obtained energy, the sensitivities, *i.e.,* the impact of an embedding value change, can be computed from reference embedding values e_r using finite differences:

$$S_W^i = \frac{\partial W}{\partial e^i} \approx \frac{W(\boldsymbol{e}_r) - W(\boldsymbol{e}_r + 0.01\boldsymbol{\delta}_i)}{0.01} \quad , \tag{29}$$

with $\boldsymbol{\delta}_i \in \mathbb{R}^d$ the unit vector in the ith direction, defined component-wise by $(\boldsymbol{\delta}_i)_k = \delta_{ik}$. In this work, sensitivities are computed using finite differences, in line with classical FEMU procedures where analytical gradients are not readily available [1]. However, since the full finite element model is implemented in TensorFlow, automatic differentiation could also be used to compute sensitivities more efficiently. This feature opens the door to future extensions involving gradient-based identification or optimization. Fig. 7 shows the sensitivities of the energy with respect to all dimensions of the embedding space. With the orthogonality and KL divergence regularizers, the contributions are quite separated, with different impacts of an embedding change in the output. Note that the function is non-linear, and this sensitivity only reflects a linearization around a reference embedding.

To visualize and assess the impact of embedding values on the mechanical response predicted by the ME-PANN model, a simple numerical simulation was specifically developed. This simulation aims to highlight the distinct influence of each embedding

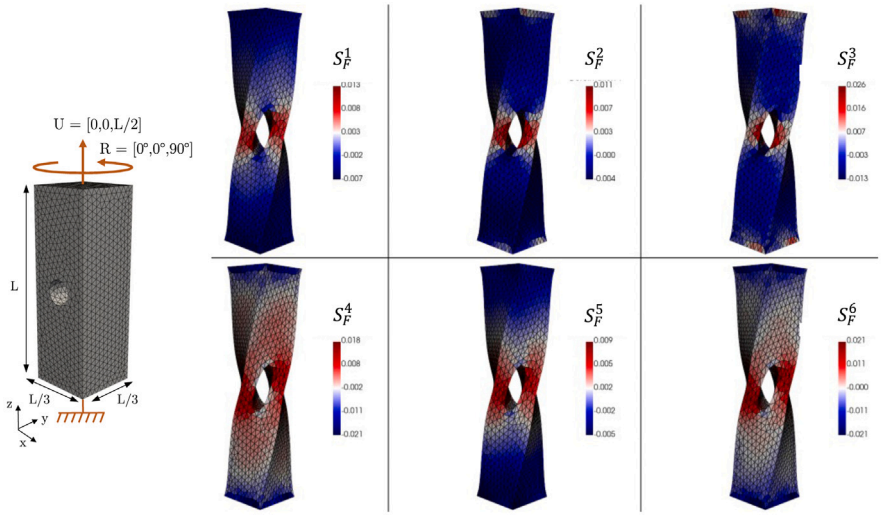


Fig. 8. Sensitivities S_F^i of the vertical deformation F_{zz} with respect to the embedding values, represented in the deformed mesh configuration.

dimension on the deformation field computed by the model. For this purpose, homemade finite element analysis (FEA) software was implemented within the TensorFlow framework, directly integrating the ME-PANN constitutive behavior.

The chosen test geometry is a simple beam with a squared cross-section, with dimensions set to $L/3 \times L/3 \times L$. A central circular hole, whose diameter is $L/6$, was included in order to introduce stress concentration and thus amplify the effect of constitutive variations. The beam geometry was discretized into a mesh consisting of linear tetrahedral elements (T4), comprising a total of 3241 nodes and 14,092 elements. Boundary conditions were imposed as follows: the bottom face of the beam was fully constrained, preventing any displacement, thus serving as a fixed support. On the opposite side, the top face was subjected to prescribed displacement conditions, combining a vertical tension displacement of magnitude $L/2$ along the vertical axis with a rotation of 90°. These loading conditions were intentionally selected to cover a significant portion of the deformation space, thereby accentuating the impact of variations introduced by the embedding vector.

For analysis, a reference computation was initially performed using a baseline embedding vector, yielding a baseline deformation state. Subsequently, this reference simulation was systematically re-run with incremental variations introduced into each dimension of the embedding vector. The deformation field of interest, specifically the vertical deformation component F_{zz}, was extracted from each simulation and analyzed comparatively and used to compute the sensitivites

$$S_F^i = 100\frac{\partial F_{zz}}{\partial e^i} \approx F_{zz}(\boldsymbol{e}_r) - F_{zz}(\boldsymbol{e}_r + 0.01\boldsymbol{\delta}_i) \quad . \tag{30}$$

The resulting variations in vertical deformation F_{zz}, depicted in Fig. 8, show distinct modes of deformation influenced directly by individual embedding dimensions. Such distinct deformation modes underscore the effectiveness of the embedding strategy. Indeed, the imposed orthogonality and KL regularization constraints during ME-PANN training are intended to encourage disentanglement within the embedding space. This disentanglement facilitates interpretability by linking specific deformation features to individual embedding dimensions, as is clearly observed in the deformation patterns. This example illustrates that, under the chosen loading and geometry, the ME-PANN embedding exhibits a visually interpretable decoupling, with different components affecting distinct regions of the mechanical response, a property that may benefit inverse identification frameworks such as FEMU [1].

Finally, we conducted a sensitivity study to evaluate the impact of network depth on model performance. Specifically, we varied the number of hidden layers while keeping the number of neurons per layer fixed at 16. All models were trained using the same datasets and identical training procedures to ensure fair comparison.

Table 5 summarizes the resulting energy and stress errors. The model with two hidden layers yields the best performance. A single hidden layer appears insufficient to capture the underlying complexity of the data, while adding more than two layers does not lead to significant improvement. It is important to note that optimal network architecture depends on various factors, including the degree of material nonlinearity, the type and magnitude of deformation, the number of embedded models, and the richness of the dataset. Therefore, the presented architecture is optimized for the specific case studied here.

Results – Case-2b — Large data encoding. Training with 200 materials from the 5 models was generated with only 10 data points per material. Similar results were obtained as in case-2a in terms of convergence and final metrics.

Fig. 9 shows the correlation between the 6 embedding vectors, separated and colored by the 5 hyperelastic models used to generate the data. The second vector e_m^1, for example, shows a clear separation of the embedding values with respect to the models as the separated distributions display different values. A specific visual highlight of the (e_m^1, e_m^2) coupling, on the top right corner, shows visible model clusters. With this self-supervised disentangled training, it may be feasible to retrieve material classes from the embedding space using clustering techniques.

Table 5
Effect of network depth on model performance (fixed width: 16 neurons per layer).

# hidden layers	Energy error E_W %	Stress error E_p in %
1	1.60	1.90
2	**0.65**	**0.85**
3	0.85	0.94
4	0.90	1.10
5	0.84	1.15

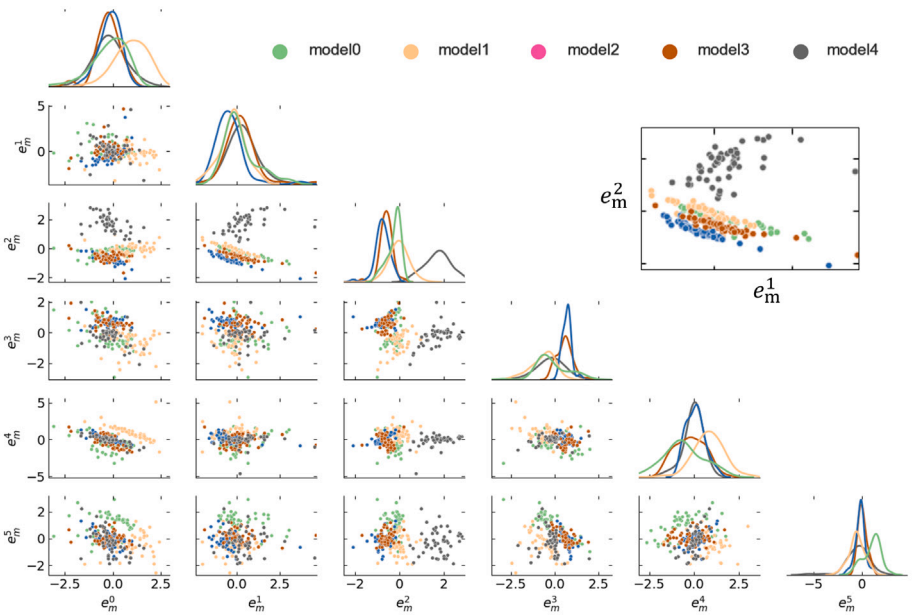


Fig. 9. Case-2b embedding space composed of 6 dimensions. The plot separates the amplitudes for the 5 model types, highlighting different distributions and clusters. A specific focus on the (e_m^1, e_m^2) coupling allows seeing clear self-trained model clusters (with model 0 and model 1, both NH models, being coupled). In the lower triangle of the figure, plots correspond to the coordinates for a given couple of embedding directions, and their color to the associated model. On the diagonal, probability density functions for each embedding direction are presented. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

When colored with respect to the model parameters (here, the Poisson ratio in Fig. 10), a data structure is also visible. Despite some outliers, a Poisson ratio gradient can be seen, from low to high values. This highlights that the material information is encoded in the embedding and could reappear when applying a non-linear transformation in the embedding space.

This analysis goes in the direction of interpretable embedding, permitted by the previously introduced regularizations.

Results – Case-2c — Large data encoding. The adapted embedding successfully captured the material behavior, yielding an energy prediction error of 1% and a stress prediction error of 0.9%. Remarkably, this fine-tuning was performed independently using only 10 stress–strain datapoints, which would be insufficient to calibrate a standard PANN from scratch. This highlights the strength of the ME-PANN framework: a data-learned, parametric PANN can generalize and adapt to unseen material behaviors by optimizing only a low-dimensional embedding, without requiring retraining of the entire model. Such capability enables efficient transfer learning and real-time material identification, even in data-scarce scenarios, significantly improving practicality for experimental or industrial applications.

The fine-tuning was performed with 2k epochs, representing a 5% of the initial number of epochs, with a number of trainable parameters being: $d \times N_m$ and a weight initialization of 0.

Results – Case-3 — Generic dataset benefit. The capacity of the model to generalize to unseen data is evaluated. The capacity of the model to extrapolate is evaluated between two situations:

- a standard PANN (without embeddings) trained only on one single material dataset (Dataset 1), referred to as the reference model. In this situation, the PANN is trained only on the phase space area probed in the single-material dataset.
- the ME-PANN trained on the same reference dataset, enriched with additional material data, with an embedding size of 1. In this situation, the ME-PANN benefits from learning from all available datasets.

The model is finally evaluated on all probed data, which were all recomputed with the reference model parameters. This allows verifying if the trained model, which performs well on its training data, is also able to extend beyond them and to "extrapolate".

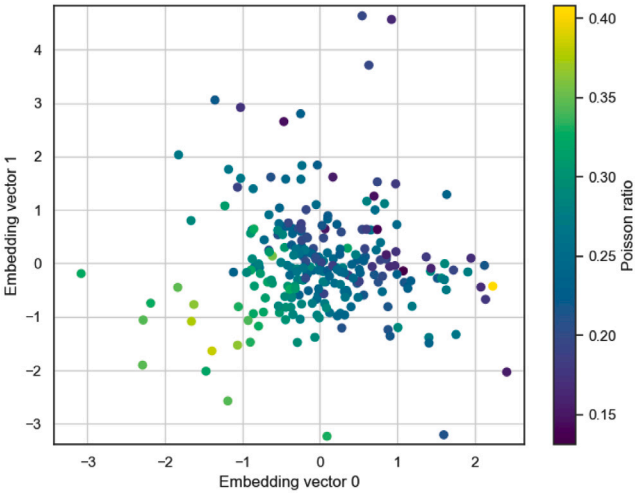
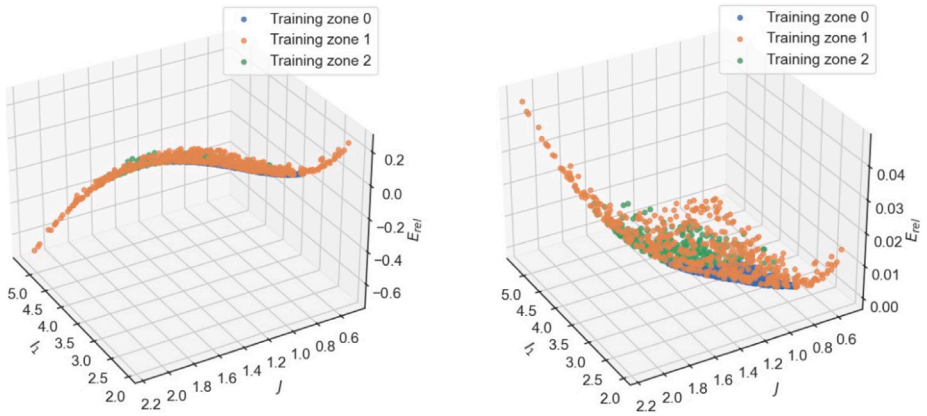


Fig. 10. Joint plot of two embeddings (e_m^0, e_m^1) colored by the Poisson ratio. An ordering of the data points with the Poisson ratio can be observed, though polluted by some outliers. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) PANN - local training (b) ME-PANN

Fig. 11. Energy error with respect to the ground truth. All validation data are computed from the Dataset-1 model. with (a) a PANN trained on a specific dataset trying to generalize outside its training space and (a) ME-PANN trained on all data simultaneously.

It is observed, in Fig. 11, that a standard PANN trained on a specific, local dataset does not generalize well to unseen data from other regions of the phase space. The energy slope captured by the PANN is not able to extrapolate beyond the trained area (maximal error of 60 %). By contrast, the embedding-enhanced approach achieves significantly better generalization (maximal error under 4%) even far from the initial true training data.

5. Discussion

5.1. Discussion on the cases

The presented results indicate that material-embedding PANNs are capable of learning both the model shape and model parameters from synthetic data, a significant departure from traditional identification methods in which only parameters are adjusted within a fixed model shape. Unlike approaches such as NN-EUCLID [6] that rely on unsupervised, equilibrium-based loss functions, the proposed method employs supervised training, thereby enabling a direct measure of prediction accuracy. However, the proposed method is compatible with the NN-EUCLID framework, among others. Observations that are made for each case are summarized as follows:

- (Case-1) Variations in material parameters (including non-linear responses with the Poisson ratio) are captured successfully by the embedding-based model. The orthogonalization and KL regularizer allow the embedding to be connected directly to

independent model parameters. The network was able to capture a relationship between the Young modulus and the Poisson ratio as those parameters were generated through a Gaussian distribution. Lamé parameters, in contrast, do not fit, in this specific material sampling, a normal distribution.

- (Case-2) Multiple model shapes are encoded in the learned embedding. The ME-PANN is thereby viewed as capturing the "modal" shape, while the embedding drives amplitude and shape specificity in a non-linear reduced-order manner. With the KL divergence loss, it becomes possible to continuously interpolate from one model to another.
- (Case-3) Improved overall generalization is achieved by combining disparate datasets, each representing a distinct region of the invariant phase space. Inconsistencies among these datasets are reconciled by the low-dimensional embedding, thus enabling a unified ME-PANN that generalizes to all probed domains. This indicates that it becomes feasible to combine all relevant datasets (possibly involving different materials and experimental conditions) into a single training workflow. As a result, benefits are gained from collectively leveraging the broader pool of information, while a material-specific embedding vector preserves each distinct material or loading path's individual characteristics.

This architecture allows for the generation of a univocal representation of different material responses through an embedding vector that encodes the parameters and shapes of the constitutive law. Non-negativity of the energy has to be evaluated numerically after training on a set of admissible kinematic fields [17]. In this study, the energy non-negativity was numerically verified on the validation data for each training. As this evaluation is performed only on a discrete sampling of data, for an application, one could perform online posterior tests to evaluate whether the model shows positive energy in a specific loading scenario [44].

From the initial analytical representation of the constitutive laws, the minimal dimension of d for the embedding vector for the multi-hyperelastic model datasets can be deduced. Specifically, for general nonlinearities without restriction, the embedding space has minimal dimension $d = 6$ for Case-2. Note that the solution of the embedding is not unique, but it is here emphasized the space generated by this basis. These analytical considerations align well with the embedding dimensions empirically tested and validated in this study. The fact that no perfect results are obtained may come from (i) an insufficient capacity of the NN to model strong non-linearities and (ii) the regularizers that complicate the computation of the most reduced space. Note that adding a first set of FC layers after the embedding, before the first concatenation, acting as pre-processing, did not provide better results. In addition to the optimization of the ME-PANN layer types (*e.g.,* Feature-wise Linear Modulation layers (FiLM) [45] allowing for multiplicative embedding/invariant features, or attention mechanisms [46]), using recent Constitutive Kolmogorov–Arnold Networks [47,48], Parametrized polyconvex hyperelasticity [49] or Generalized Standard Material Networks [50] could be an interesting opportunity. Those state-of-the-art approaches allow for a better representation of each material while the coupling with embedding promotes generalization across diverse material-specific characteristics.

Compared to conventional methods that focus solely on parameter identification (*e.g.,* FEMU, MCRE, VFM, equilibrium gap), the proposed approach learns an adaptable model shape. This flexibility is particularly beneficial when new materials are introduced. The significant performance improvements observed in transfer learning experiments further underscore the potential of material-embedding PANN frameworks as agile and robust tools for constitutive modeling.

Visualization of the embedding confirms that the learned embeddings cluster in a physically meaningful way, with distinct dimensions correlating strongly with parameters like model type, Young's modulus, and Poisson's ratio. The use of KL divergence and orthogonalization regularizers has been instrumental in structuring the latent space so that each dimension reflects a unique aspect of the material response. This enhanced interpretability opens promising avenues for future work, including the development of quantitative material similarity metrics and automated classification schemes. Such advancements could ultimately streamline the characterization process and improve the transferability of models to new material systems. These enhancements not only improve interpretability but also enhance generalization to unseen material cases [39,51], as validated by the improved correlation of the embedding with physical parameters in Fig. 4.

Other regularizers, such as L1 to promote sparsity, entropy to focus on discrete variables, and contrastive loss [52] to supervise the embedding disentanglement, are also possible depending on the type of data to be disentangled and the available labels. For real applications, specifying information on the tested material could be used to enforce separated embeddings. This should lead to further work.

5.2. ME-PANN interpretation

The embedding effectively encodes a low-dimensional latent descriptor of the material in the embedding, driving learned model shapes. During training across multiple materials, the shared ME-PANN layers learn universal aspects of constitutive behavior, while the embedding vectors concentrate on material-specific details. A few practical benefits of this setup are:

- *Adaptability.* Once the main ME-PANN is trained on various materials, fine-tuning to a new material may only require optimizing the new embedding vector $\mathbf{e}_{\text{new}}$, keeping the main weights fixed. This accessible fine-tuning thus requires much less experimental data.
- *Compactness.* The dimension d of the embedding can be very small (*e.g.,* d = 5–20), thus representing a highly compressed parameterization of the differences among experiments while keeping the model shape adaptivity high. The embedding size is chosen as a tunable hyperparameter, enabling the capture of varying material complexities.
- *Generalization.* The model can be trained from complementary experiments, thus sharing the benefits of probed areas from one experiment to another.

A key advantage of the material-embedding PANN is the potential to extract physically meaningful features from the learned latent space.

5.3. Limits and perspectives

The study was performed with supervised training. An extension of this model in an EUCLID framework and for real experimental tests is appealing. General training on all elastic materials, or more complex behavior, in the laboratory could be performed to extract a real experimental embedding.

One limitation arises from the non-uniqueness of the embedding solution. Different local minima in the embedding space may yield equally low training errors, especially if the coverage of materials and loading paths is incomplete. Although this does not necessarily degrade predictive accuracy, it can complicate attempts to assign direct physical meaning to each latent dimension. This is a reason why the 3 different datasets used in Case-3 were chosen with overlaps.

With such a low-dimensional embedding, it also becomes possible to train the model, *i.e.,* its embedding space, while the rest of the model is frozen, from traditional identification such as FEMU. With the ME-PANN model, noise impact is reduced as data have to fit in a low dimension mechanically suited space [53], allowing the encoding of the mechanical content rather than the noise [54].

This framework is well-suited for applications involving families of materials with similar elastic responses, such as metals, polymers, or composites, where the shared model captures general constitutive behavior and the embedding vector encodes material-specific variations. In addition, this embedding could also encode experimental biases across different experiments performed on the same material.

6. Conclusion

In this work, a material-embedding, physics-augmented neural network (ME-PANN) framework was proposed for the constitutive modeling of isotropic elasticity. The low-dimensional latent vector, concatenated at each network layer, was shown to introduce a clear splitting between both the model shape and the material parameters, thereby extending the capacity of traditional neural network-based approaches. Unlike classical identification methods, which restrict the model form and only calibrate parameters, this self-supervised approach integrates a flexible architecture that adapts to different material behaviors through learned embeddings.

Results on synthetic hyperelastic datasets, with a supervised loss, indicated that the embedding vector effectively represented multiple constitutive laws and parameter variations, leading to improved generalization performance. This conditional learning capability underscores the utility of the method in practical scenarios where multiple experiments or limited data sets are present.

It is anticipated that the material-embedding PANN will serve as a generic platform for constitutive modeling, particularly in cases where new materials or loading paths must be characterized with minimal additional experimental data. Future investigations may examine the extension of the approach to anisotropic or rate-dependent behaviors, as well as the development of a distance between constitutive relations. Additionally, efforts to integrate partially supervised or purely unsupervised PANN learning techniques (such as EUCLID) could broaden applicability in real experimental settings where direct stress data are not readily available.

CRediT authorship contribution statement

C. Jailin: Writing – original draft, Methodology, Investigation, Funding acquisition, Conceptualization. **E. Baranger:** Writing – original draft, Methodology, Investigation, 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.

Acknowledgments

The authors would like to warmly acknowledge Elodie Ripaud (GE HealthCare), France for fruitful discussions on the ML model. This project was made possible by the *Agence Nationale de la Recherche* (ANR), France grant No. ANR-22-CPJ2-0046-01.

Data availability

Data will be made available on request.

References

[1] F. Mathieu, H. Leclerc, F. Hild, S. Roux, Estimation of elastoplastic parameters via weighted FEMU and integrated-DIC, Exp. Mech. 55 (2015) 105–119.
[2] P. Feissel, O. Allix, Modified constitutive relation error identification strategy for transient dynamics with corrupted data: The elastic case, Comput. Methods Appl. Mech. Engrg. 196 (13–16) (2007) 1968–1983.

[3] S. Conti, S. Müller, M. Ortiz, Data-driven finite elasticity, Arch. Ration. Mech. Anal. 237 (1) (2020) 1–33.
[4] A. Platzer, A. Leygue, L. Stainier, M. Ortiz, Finite element solver for data-driven finite strain elasticity, Comput. Methods Appl. Mech. Engrg. 379 (2021) 113756.
[5] 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.
[6] M. Flaschel, S. Kumar, L. De Lorenzis, Unsupervised discovery of interpretable hyperelastic constitutive laws, Comput. Methods Appl. Mech. Engrg. 381 (2021) 113852.
[7] 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) 1–43.
[8] A. Hussain, A.H. Sakhaei, M. Shafiee, Machine learning-based constitutive modelling for material non-linearity: A review, Mech. Adv. Mater. Struct. (2024) 1–19.
[9] L.A. Herrmann, S. Kollmannsberger, Deep learning in computational mechanics: a review, Comput. Mech. (2024) URL https://api.semanticscholar.org/CorpusID:266980168.
[10] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.
[11] D.W. Abueidda, S. Koric, E. Guleryuz, N.A. Sobh, Enhanced physics-informed neural networks for hyperelasticity, Internat. J. Numer. Methods Engrg. 124 (7) (2023) 1585–1601.
[12] F. Masi, I. Stefanou, P. Vannucci, V. Maffi-Berthier, Thermodynamics-based artificial neural networks for constitutive modeling, J. Mech. Phys. Solids 147 (2021) 104277.
[13] F. Masi, I. Stefanou, Multiscale modeling of inelastic materials with Thermodynamics-based Artificial Neural Networks (TANN), Comput. Methods Appl. Mech. Engrg. 398 (2022) 115190.
[14] M. Fernández, M. Jamshidian, T. Böhlke, K. Kersting, O. Weeger, Anisotropic hyperelastic constitutive models for finite deformations combining material theory and data-driven approaches with application to cubic lattice metamaterials, Comput. Mech. 67 (2) (2021) 653–677.
[15] 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.
[16] 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.
[17] 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 (2023) 105363.
[18] K. Linka, E. Kuhl, A new family of constitutive artificial neural networks towards automated model discovery, Comput. Methods Appl. Mech. Engrg. 403 (2023) 115731.
[19] B. Amos, L. Xu, J.Z. Kolter, Input convex neural networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 146–155.
[20] M. Rosenkranz, K.A. Kalina, J. Brummund, M. Kästner, A comparative study on different neural network architectures to model inelasticity, Internat. J. Numer. Methods Engrg. 124 (21) (2023) 4802–4840.
[21] M.A. Sutton, W. Wolters, W. Peters, W. Ranson, S. McNeill, Determination of displacements using an improved digital correlation method, Image Vis. Comput. 1 (3) (1983) 133–139.
[22] A. Buljac, C. Jailin, A. Mendoza, J. Neggers, T. Taillandier-Thomas, A. Bouterf, B. Smaniotto, F. Hild, S. Roux, Digital volume correlation: review of progress and challenges, Exp. Mech. 58 (2018) 661–708.
[23] T. Kirchdoerfer, M. Ortiz, Data-driven computational mechanics, Comput. Methods Appl. Mech. Engrg. 304 (2016) 81–101.
[24] A. Benady, Physics-Augmented Neural Networks for Constitutive Modeling (Ph.D. thesis), Université Paris-Saclay, 2024.
[25] 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.
[26] 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.
[27] A. Benady, E. Baranger, L. Chamoin, Unsupervised learning of history-dependent constitutive material laws with thermodynamically-consistent neural networks in the modified constitutive relation error framework, Comput. Methods Appl. Mech. Engrg. 425 (2024) 116967, http://dx.doi.org/10.1016/j.cma.2024.116967.
[28] A. Benady, E. Baranger, L. Chamoin, NN-mCRE: A modified constitutive relation error framework for unsupervised learning of nonlinear state laws with physics-augmented neural networks, Internat. J. Numer. Methods Engrg. 125 (8) (2024) e7439.
[29] L. Li, C.Q. Chen, Equilibrium-based convolution neural networks for constitutive modeling of hyperelastic materials, J. Mech. Phys. Solids 164 (2022) 104931.
[30] C. Jailin, A. Benady, R. Legroux, E. Baranger, Experimental learning of a hyperelastic behavior with a physics-augmented neural network, Exp. Mech. (2024) 1–17.
[31] L. Li, S. Li, H. Gao, C. Chen, ENNStressNet-an unsupervised equilibrium-based neural network for end-to-end stress mapping in elastoplastic solids, J. Mech. Phys. Solids (2025) 106117.
[32] Z. Akata, F. Perronnin, Z. Harchaoui, C. Schmid, Label-embedding for image classification, IEEE Trans. Pattern Anal. Mach. Intell. 38 (7) (2015) 1425–1438.
[33] L. Zhang, T. Xiang, S. Gong, Learning a deep embedding model for zero-shot learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2021–2030.
[34] P. Blandfort, T. Karayil, F. Raue, J. Hees, A. Dengel, Fusion strategies for learning user embeddings with neural networks, in: 2019 International Joint Conference on Neural Networks, IJCNN, IEEE, 2019, pp. 1–8.
[35] E. Ripaud, C. Jailin, P. Milioni de Carvalho, L. Vancamberg, I. Bloch, Deep learning framework for managing inter-reader variability in background parenchymal enhancement classification for contrast-enhanced mammography, in: MICCAI-2025, 2025.
[36] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.
[37] J. Asghar, S. D. Thomas, J. Reese E., F. Jan, Input specific neural networks, 2025, Preprint.
[38] D. Bouchacourt, R. Tomioka, S. Nowozin, Multi-level variational autoencoder: Learning disentangled representations from grouped observations, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018.
[39] D.P. Kingma, M. Welling, et al., An introduction to variational autoencoders, Found. Trends® Mach. Learn. 12 (4) (2019) 307–392.
[40] S. Kullback, R.A. Leibler, On information and sufficiency, Ann. Math. Stat. 22 (1) (1951) 79–86.
[41] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.
[42] S. Zschocke, W. Graf, M. Kaliske, Incorporating uncertainty in stress-strain data acquisition: extended model-free data-driven identification, PAMM 23 (2) (2023) e202300008.
[43] S. Roux, F. Hild, Comprehensive full-field measurements via digital image correlation, Compr. Mech. Mater. (2024).
[44] B. Masseron, G. Rastiello, R. Desmorat, Analytical strain localization analysis of isotropic and anisotropic damage models for quasi-brittle materials, Int. J. Solids Struct. 254 (2022) 111869.

[45] E. Perez, F. Strub, H. De Vries, V. Dumoulin, A. Courville, Film: Visual reasoning with a general conditioning layer, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018.
[46] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).
[47] 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, arXiv preprint arXiv:2502.05682.
[48] P. Thakolkaran, Y. Guo, S. Saini, M. Peirlinck, B. Alheit, S. Kumar, Can KAN CANs? Input-convex Kolmogorov-Arnold networks (KANs) as hyperelastic constitutive artificial neural networks (CANs), 2025, arXiv preprint arXiv:2503.05617.
[49] D.K. Klein, F.J. Roth, I. Valizadeh, O. Weeger, Parametrized polyconvex hyperelasticity with physics-augmented neural networks, Data- Centric Eng. 4 (2023) e25.
[50] M. Flaschel, P. Steinmann, L. De Lorenzis, E. Kuhl, Convex neural networks learn generalized standard material models, J. Mech. Phys. Solids (2025) 106103.
[51] J. Hu, D. Liu, N. Fu, R. Dong, Realistic material property prediction using domain adaptation based machine learning, Digit. Discov. 3 (2) (2024) 300–312.
[52] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, D. Krishnan, Supervised contrastive learning, Adv. Neural Inf. Process. Syst. 33 (2020) 18661–18673.
[53] B. Bahmani, W. Sun, Distance-preserving manifold denoising for data-driven mechanics, Comput. Methods Appl. Mech. Engrg. 405 (2023) 115857.
[54] X. He, Q. He, J.-S. Chen, Deep autoencoders for physics-constrained data-driven nonlinear materials modeling, Comput. Methods Appl. Mech. Engrg. 385 (2021) 114034.