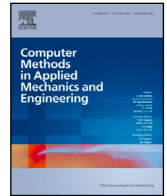


Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

## Comput. Methods Appl. Mech. Engrg.

journal homepage: [www.elsevier.com/locate/cma](http://www.elsevier.com/locate/cma)

# An indirect training approach for implicit constitutive modelling using recurrent neural networks and the virtual fields method

Rúben Lourenço<sup>a,\*</sup>, Petia Georgieva<sup>b</sup>, Elias Cueto<sup>c</sup>, A. Andrade-Campos<sup>a</sup>

<sup>a</sup> Department of Mechanical Engineering; TEMA - Centre for Mechanical Technology and Automation; LASI - Intelligent Systems Associate Laboratory, University of Aveiro, Campus Universitário de Santiago, Aveiro 3810-193, Portugal

<sup>b</sup> Department of Electronics, Telecommunications and Informatics; Institute of Telecommunications; Institute of Electronics and Informatics Engineering of Aveiro, University of Aveiro, Campus Universitário de Santiago, Aveiro 3810-193, Portugal

<sup>c</sup> Aragón Institute of Engineering Research, Universidad de Zaragoza, C. María de Luna, Zaragoza 50018, Spain

## ARTICLE INFO

### Keywords:

Implicit constitutive model  
Artificial neural network  
Virtual fields method  
Recurrent neural network  
Gated-recurrent unit  
Indirect training

## ABSTRACT

Accurate material description is crucial to achieve high-quality results in computational analysis software. Phenomenological constitutive laws generalize the material behaviour observed in simple mechanical tests. The resulting empirical expressions contain parameters that need to be calibrated through an inverse optimization process. Advancements in Digital Image Correlation (DIC) techniques have enabled the extraction of non-uniform multi-axial displacement fields, facilitating the development of heterogeneous mechanical specimens and unlocking access to richer material behaviour data. Despite these advancements, constitutive models are still limited by mathematical formulations and biases due to simplifying assumptions. Artificial Neural Networks (ANNs), as universal function approximators, could potentially drive a paradigm shift in the field. ANNs do not require explicit pre-formulations, hence avoiding the need for identifying unknown parameters. Moreover, ANNs are able to implicitly learn patterns purely from data, allowing to circumvent the biases induced by the simplifying assumptions of first-order principles. However, the training of ANNs for implicit constitutive modelling is not straightforward, given the impossibility of obtaining direct measurements of stress. This paper explores the integration of Recurrent Neural Networks (RNNs) with the Virtual Fields Method (VFM) for material modelling. This approach uses global force and displacement data to indirectly train the neural network, with the equilibrium being evaluated globally through the VFM loss function. The proposed method is (i) computationally efficient by not requiring Finite Element Analysis (FEA), (ii) compatible with both full-field measurements and numerically generated data, and (iii) able to handle experimental boundary conditions.

## 1. Introduction

The widespread adoption of computational analysis software in engineering, in particular Finite Element Analysis (FEA) software, has reinforced the need for precise material data, hence drawing an increased attention to the material characterization [1,2]. Traditionally, constitutive laws are derived from phenomenological observations and physical knowledge and, therefore generalize the behaviour observed in basic mechanical tests [3,4]. The developed expressions contain empirical parameters that require calibration through an inverse optimization process, aimed at minimizing the discrepancy between experimental tests and numerical simulations [4]. For decades, constitutive parameter identification was limited to one-dimensional homogeneous deformation data,

\* Correspondence to: Department of Mechanical Engineering, University of Aveiro, Campus Universitário de Santiago, Aveiro 3810-193, Portugal.  
E-mail address: [rubel@ua.pt](mailto:rubel@ua.pt) (R. Lourenço).

<https://doi.org/10.1016/j.cma.2024.116961>

Received 5 January 2024; Received in revised form 20 March 2024; Accepted 23 March 2024

Available online 1 April 2024

0045-7825/© 2024 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

due to the restrictions imposed to the specimens' shapes and loading settings and the lack of high-resolution measurement equipment. In such conditions, an extensive number of experiments were often required to calibrate more complex models, which made the process expensive and time-consuming [5]. In recent years, the introduction of Digital Image Correlation (DIC) techniques allowed to extract non-uniform multi-axial displacement fields at a large number of measurement points, unlocking the development of heterogeneous mechanical specimens capable of yielding richer material behaviour data (e.g. [6–9], among others). Additionally, the ability to directly link the full-field data to the FEA's discretized fields, enabled more sophisticated approaches for the identification of constitutive parameters, such as: the Finite Element Model Updating (FEMU) [10] and the Virtual Fields Method (VFM) [11]. Despite the previously mentioned advancements, phenomenological constitutive models are still constrained by their mathematical formulation and are generally biased, due to the simplifying assumptions of first-order principles [12]. Moreover, a given model cannot guarantee good results in different settings from those used to calibrate it [13].

The emergence of Big Data and the exponential increase in computer processing power led to an unprecedented growth in various fields of Artificial Intelligence (AI), notably Machine Learning (ML) [14]. Among a plethora of ML algorithms, Artificial Neural Networks (ANNs) have risen to prominence as universal function approximators [15], offering superior modelling performance across a broad spectrum of applications. ANNs have the potential to lead a paradigm shift in constitutive modelling [16]. A key advantage is that these do not require to postulate explicit formulations, eliminating the need to identify unknown constitutive parameters [17]. Furthermore, ANNs can effectively leverage vast amounts of data from full-field measurements, learning patterns implicitly and, hence, mitigating the biases inherent to conventional constitutive models, given sufficient data and model capacity [12]. The applicability of ANNs for constitutive modelling was first explored by Ghaboussi et al. [18], who demonstrated promising results with a Feed-Forward Neural Network (FFNN) trained to capture the cyclic behaviour of concrete. This laid the foundations for subsequent developments and the expansion to other materials, with an early application to metal behaviour attributed to Furukawa and Yagawa [19], who extended the concept to viscoplasticity using a state-space representation. Notable foundational contributions also include the implementation of an ANN material into a FEA code [20] and the introduction of an analytical tangent stiffness matrix for ANN-based material models [21]. Advancements in computational resources enabled to explore more sophisticated architectures, such as Recurrent Neural Networks (RNNs). RNNs are well suited for implicit constitutive modelling due to built-in memory cells that capture path-dependency effects [22].

Most approaches reported in the literature for ANN-based material modelling fall within the standard supervised learning paradigm. The training process relies on labelled data pairs, with strain tensors typically provided as input features to predict the corresponding stress state. Depending on the problem, internal variables of the traditional constitutive model (e.g. accumulated plastic strain, isotropic or kinematic hardening, or others) may additionally be used, either as features or outputs. However, these internal variables exist only numerically, lacking real experimental meaning. Moreover, stress is impossible to be measured from complex heterogeneous mechanical specimens. Even from simple uni-axial homogeneous tests, stress is obtained from indirect calculations and assumptions, such as theoretical boundary conditions. In a context where real experimental data is used to train the ANN model, such limitation imposes the training process to be carried out indirectly. Therefore, the training dataset must be composed of measurable variables only, such as displacements and/or strains and global force. During training, the variable of interest, usually the stress, is not directly taken into account in the computation of the loss function and it may be indirectly obtained from intermediate calculations. Within the framework of indirect training, this paper proposes an innovative approach by coupling the VFM with a RNN for material modelling. This allows the use of global force and displacement data to indirectly train the neural network, with the equilibrium being evaluated globally through the VFM loss function. The main advantages of this approach are: (i) the computational efficiency gained from the non-use of FEA, (ii) the compatibility with both full-field measurements and numerically generated data and (iii) the independence of the boundary conditions.

The paper is organized as follows. Section 2 provides an introduction to RNNs with gated recurrent units (GRUs) and the corresponding mathematical background. In Section 3, the VFM is introduced as well as its theoretical background. The coupled RNN-VFM indirect training methodology is also presented. Section 4 provides details about numerical implementation, data generation and training procedures. Finally, in Section 5, the indirect training methodology is validated against the standard supervised approach. Results from the RNNs inference on the validation data and the corresponding FEA implementations as material models are presented and discussed.

## 2. GRU-based RNNs

### 2.1. Introduction

Modelling path dependent plasticity using ANNs is a challenging task due to the history dependency effect [22]. Standard feed-forward architectures are unable to capture the phenomenon, as there is no mechanism to establish a relationship between the inputs in the current time step with the outputs from previous time steps [23,24]. The interplay between past and present states is a suitable use case for RNNs. These networks have cyclic connections that allow to maintain a memory of previous inputs (see Fig. 1(b)), making them ideal for tasks involving sequential or time-dependent data [25], in which material behaviour modelling is included. Within this scope, using strain sequences with a history of  $k$  length, up to the current time step  $t$ , a RNN can capture temporal dependencies, allowing it to predict the corresponding stress state accurately. These capabilities have resulted in a slew of works reported in the field [22–24,26–29].

## 2.2. Mathematical formulation

To take advantage of RNNs for implicit constitutive modelling, in the present work a variant known as Gated Recurrent Units (GRUs) is employed. GRUs are a streamlined version of Long Short-Term Memory (LSTM) units, designed to capture dependencies in sequential data efficiently, while mitigating some of the computational complexity [30]. As observed in Fig. 1(a), a basic GRU cell comprises the following key components [30,31]:

1. **Update gate ( $z_t$ )** - determines how much is retained from the previous hidden state and how much it should be updated with new information:

$$z_t = \text{sig} (W_{xz}x_t + b_{xz} + W_{hz}h_{t-1} + b_{hz}) \quad (1)$$

2. **Reset gate ( $r_t$ )** - specifies which information from the previous hidden state should be forgotten:

$$r_t = \text{sig} (W_{xr}x_t + b_{xr} + W_{hr}h_{t-1} + b_{hr}) \quad (2)$$

3. **New gate ( $n_t$ )** - computes a new candidate hidden state based on the input and previous hidden state:

$$n_t = \tanh (W_{xn}x_t + b_{xn} + r_t \odot [W_{hn}h_{t-1} + b_{hn}]) \quad (3)$$

4. **Final state ( $h_t$ )** - combines the previous hidden state and the new memory content, controlled by the update gate:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot n_t. \quad (4)$$

The variables  $h_t$  and  $x_t$  are, respectively, the hidden state and the input at time  $t$ , while  $h_{(t-1)}$  is the hidden state at time  $t - 1$ , or the initial hidden state at  $t = 0$ . The entities sig and tanh are, respectively, the sigmoid and the hyperbolic tangent activation functions and the operator  $\odot$  is the Hadamard product. The weight matrices  $W$  and bias vectors  $b$  are appended with double indices identifying the mapping between the different components of the GRU cell. A deeper description of these mathematical entities and their corresponding dimensions, in the context of the present work, is provided in Table 1.

The hidden state  $h$  is the key element behind the flow of information relating previous time steps and future predictions. In a way analogous to the internal variables of classical constitutive laws,  $h$  carries the information about the current plastic state of the material [29]. The GRU cells can be combined with layers of different types in order to build specific architectures.

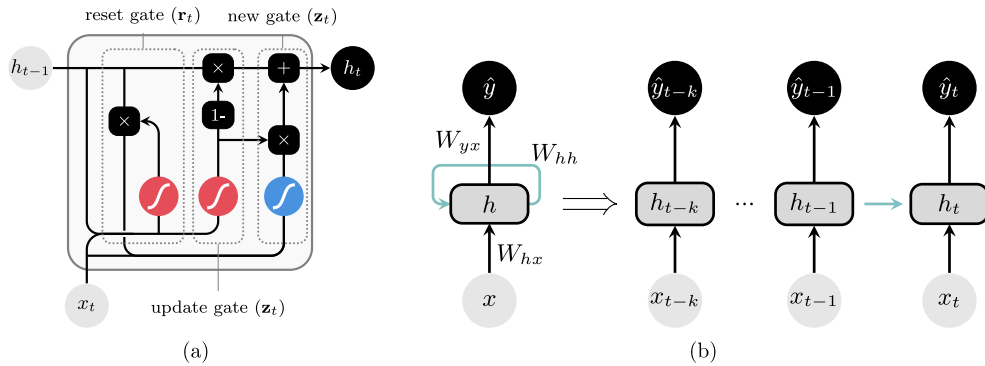


Fig. 1. Schematic representation of (a) a standard GRU cell and (b) an unrolled recurrent neural network.

Table 1

Mathematical entities, and corresponding dimensions, for a GRU-based RNN, where:  $n$  is the batch size,  $k$  the sequence length,  $n_{\text{cells}}$  the number of GRU cells,  $s_{\text{in}}$  the input size,  $s_{\text{out}}$  the output size and  $s_{\text{hid}}$  the hidden size.

Component	Entity	Dimensions	Description
Input	$x_t$	$(n \times k \times s_{\text{in}})$	Strain sequences
Previous state	$h_{t-1}$	$(n_{\text{cells}} \times n \times k \times s_{\text{hid}})$	Hidden state at time $t - 1$
Final state	$h_t$	$(n_{\text{cells}} \times n \times k \times s_{\text{hid}})$	Hidden state at time $t$
Output	$\hat{y}_t$	$(n \times s_{\text{out}})$	Stress tensor at time $t$
Update gate	$W_{x(z r n)}$	$(3 \times s_{\text{hid}} \times s_{\text{in}})$	Input to hidden weights of 1st GRU cell
Reset gate or New gate	$b_{x(z r n)}$	$(3 \times s_{\text{hid}} \times s_{\text{hid}})$	Input to hidden weights of $i$ th GRU cell
	$b_{x(z r n)}$	$(3 \times s_{\text{hid}})$	Input to hidden bias
	$W_{h(z r n)}$	$(3 \times s_{\text{hid}} \times s_{\text{hid}})$	Hidden to hidden weights
	$b_{h(z r n)}$	$(3 \times s_{\text{hid}})$	Hidden to hidden bias

### 3. The virtual fields method

#### 3.1. Introduction

The Virtual Fields Method (VFM) is a state-of-the-art technique employed in the inverse identification of phenomenological material models. First introduced by Grédiac [32], the VFM is renowned for its computational efficiency [5], with the main advantage of not requiring the Finite Element Method (FEM) for any forward calculations, avoiding the modelling of boundary conditions and the specimen's geometry [33,34]. The method is based on the Principle of Virtual Work (PVW), which states the internal virtual work, as a result of the internal forces, must be equal to the external virtual work performed by the external forces in the surface  $S$ , in the absence of body forces, such that [11,35]:

$$-\int_V \boldsymbol{\sigma} : \boldsymbol{\varepsilon}^* dV + \int_{\partial V} \mathbf{T} \cdot \mathbf{u}^* dS = 0, \quad (5)$$

where  $\boldsymbol{\sigma}$  is the stress tensor,  $\boldsymbol{\varepsilon}^*$  is the virtual strain,  $\mathbf{u}^*$  is the virtual displacement,  $V$  is the volume of the solid and  $\mathbf{T}$  is the vector of loading tractions. The virtual displacements are mathematical test functions, with no real physical meaning and independent of the measured displacements, that satisfy the equilibrium from Eq. (5) [35]. The equilibrium holds for any virtual displacement  $\mathbf{u}^*$  as long as it is continuous, piece-wise differentiable over the specimen's body and kinematically admissible, in order to satisfy the displacement boundary conditions. The virtual strains  $\boldsymbol{\varepsilon}^*$  are derived from the virtual displacements  $\mathbf{u}^*$  using the strain–displacement relationship [11]:

$$\boldsymbol{\varepsilon}^* = \frac{1}{2} (\nabla \mathbf{u}^* + \nabla \mathbf{u}^{*T}). \quad (6)$$

Besides the requirements previously enumerated, there are no specific rules for the choice of virtual fields, rendering the user with an infinity of fields to choose from [34]. Nevertheless, it is convenient to find an expression such that the virtual displacement  $\mathbf{u}^*$  would vanish or be constant along the boundary of prescribed displacements [11,33]. The reason behind the simplification is the fact that the traction distribution is often unknown and only the resultant force  $\mathbf{F}$  is available from the load cell. By doing so, the set of possible choices for the virtual fields are narrowed down and the computation of the external virtual work  $\mathbf{W}_{\text{ext}}^*$  is simplified, such that [36]:

$$\mathbf{W}_{\text{ext}}^* = \mathbf{u}^* \int_{\partial S} \mathbf{T} dS \approx \mathbf{u}^* \cdot \mathbf{F}. \quad (7)$$

Worthy of note, it is the fact that the strain measurements from full-field techniques are only available at the surface of the specimen. Thus, an assumption for the through-thickness behaviour of the material is necessary [34]. Since thin specimens are often employed in mechanical testing, it is common to reduce the problem to two dimensions by assuming a plane stress state, although different assumptions are possible. In that case, for a thin specimen of thickness  $h$ , Eq. (5) may be modified into [33]:

$$-h \int_S \boldsymbol{\sigma} : \boldsymbol{\varepsilon}^* dS + h \int_{\partial S} \mathbf{T} \cdot \mathbf{u}^* dL = 0, \quad (8)$$

where  $S$  and  $\partial S$  are, respectively, the surface and the boundary of the specimen. In full-field measurements the displacements are captured using DIC techniques in a very large number points or subsets, as such, the integral in Eq. (8) may be approximated as a discrete sum for any time stage  $t$ . Keeping that in mind and assuming a suitable set of virtual fields is used, the identification process is carried out by minimizing a loss function with respect to the constitutive parameters  $\mathbf{x}$  defined as follows:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\varepsilon}) = \sum_{i=1}^{n_{VFS}} \left[ \sum_{t=1}^{n_t} \left( \sum_{j=1}^{n_{pts}} \boldsymbol{\sigma}^j(\mathbf{x}, \boldsymbol{\varepsilon}) \cdot \boldsymbol{\varepsilon}^{*j(i)} \cdot S^j - \mathbf{W}_{\text{ext}}^* \right)^2 \right], \quad (9)$$

where  $S^j$  is the area of the  $j$ -th element and  $n_{VFS}$ ,  $n_t$ ,  $n_{pts}$  are the numbers of virtual fields, time stages and measurement points, respectively. The choice of virtual fields constitutes a key point of the VFM with a great impact in the final solution. These virtual entities are often user-defined, nonetheless, systematic procedures are available to automatically define them, namely: the stiffness-based and the sensitivity-based virtual fields [33]. It is important to refer that the User-Defined Virtual Fields (UDVFs) are tied to the user's own expertise and intuition and do not evolve in time.

#### 3.2. Sensitivity-based virtual fields

The concept behind the sensitivity-based virtual fields is to apply a perturbation to each of the model's parameters in order to obtain a stress sensitivity and its temporal evolution [33]. The stress field is the only quantity that depends directly on the constitutive parameters in the VFM, so the stress sensitivity maps highlight areas that strongly depend on a given parameter [37]. The spatial sensitivity of stress to each model parameter may be computed as:

$$\delta \boldsymbol{\sigma}^{(i)}(\mathbf{x}, \boldsymbol{\varepsilon}, t) = \boldsymbol{\sigma}(\boldsymbol{\varepsilon}, \mathbf{x}, t) - \boldsymbol{\sigma}(\boldsymbol{\varepsilon}, \mathbf{x} + \delta x_i, t), \quad (10)$$

with  $i$  being the  $i$ th term of the constitutive model parameters  $\mathbf{x}$  and  $t$  the time stage. The virtual displacements  $\mathbf{u}^*$  are found solving the following system of equations applied to a virtual mesh [37]:

$$\delta \boldsymbol{\sigma}^{(i)} = \mathbf{B} \mathbf{u}^{*(i)}, \quad (11)$$

where  $\mathbf{B}$  is the global strain–displacement matrix, used to map the virtual displacements at the nodes to every virtual strain. Considering a virtual mesh of bi-linear quadrilateral elements,  $\mathbf{B}$  results from the assembly:

$$\mathbf{B} = \bigwedge_{e=1}^{n_e} \mathbf{B}^e, \quad (12)$$

where  $n_e$  is the number of elements in the mesh and  $\mathbf{B}^e$  is the element strain–displacement matrix, defined through the spatial derivatives of the standard bi-linear nodal shape functions  $\mathbf{N}^e$  in the coordinate space  $(x, y)$ , such that [38]:

$$\mathbf{N}^e = \begin{bmatrix} N_1(x, y) \\ N_2(x, y) \\ N_3(x, y) \\ N_4(x, y) \end{bmatrix} = \frac{1}{4} \begin{bmatrix} (1-x)(1-y) \\ (1+x)(1-y) \\ (1+x)(1+y) \\ (1-x)(1+y) \end{bmatrix}, \text{ thus} \quad (13)$$

$$\mathbf{B}^e = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 & \frac{\partial N_4}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} & 0 & \frac{\partial N_4}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial y} & \frac{\partial N_4}{\partial x} \end{bmatrix}. \quad (14)$$

If the displacement is prescribed at the boundaries, the traction is unknown and, as such, the displacements must be set to zero. On the other hand, if the traction distribution is unknown but the resultant force is known, a constant virtual displacement is set at the boundaries. By applying these constraints, a modified global strain–displacement matrix  $\bar{\mathbf{B}}$  is obtained and the virtual displacements  $\mathbf{u}^*$  can finally be computed as follows [33]:

$$\mathbf{u}^{*(i)} = \text{pinv}(\bar{\mathbf{B}}) \delta \sigma^{(i)}, \quad (15)$$

with  $\text{pinv}(\bar{\mathbf{B}})$  being the pseudo-inverse of the modified strain–displacement matrix. The virtual strains  $\epsilon^*$  are then computed as:

$$\epsilon^{*(i)} = \mathbf{B} \mathbf{u}^{*(i)}. \quad (16)$$

The identification process is carried out, with the virtual fields  $\mathbf{u}^{*(i)}$  and  $\epsilon^{*(i)}$  being updated multiple times in order to minimize the loss resulting from enforcing the PVW, within an infinitesimal strain framework, as follows:

$$\mathcal{L}(\mathbf{x}, \epsilon) = \sum_{i=1}^{n_{VF3}} \left[ \frac{1}{(\alpha^{(i)})^2} \sum_{t=1}^{n_t} \left( \sum_{j=1}^{n_{ms}} \sigma^j(\mathbf{x}, \epsilon, t) \cdot \epsilon^{*(j)(i)}(t) \cdot S^j - \mathbf{W}_{ext}(t) \right)^2 \right], \quad (17)$$

with  $S^j$  being the surface area of the  $j$ -th measurement point and  $\alpha^{(i)}$  the scaling factor of the  $i$ -th virtual field, employed to guarantee similar magnitudes between different virtual fields.

### 3.3. Coupled RNN-VFM for indirect training

Within the framework of indirect training only a handful of works were reported in the literature. For example, Xu et al. [39] proposed a symmetric positive definite neural network (SPD-NN) to predict the Cholesky factors of the tangent stiffness, with the stress tensor being computed in an incremental form from intermediate steps. Although the method provides numerically stable solutions, it applies exclusively to the non-linear segment of the material response and plasticity has to be identified through a transition function, relying on an estimate of the yield stress given by the user. Both Huang et al. [40] and Xu et al. [41] proposed a method using an ANN to replace the constitutive model in a FEA solver. The solver imposes the physical constraints during training, enabling the ANN to learn material behaviour. Similarly, Liu et al. [42] proposed integrating an ANN with FEA to form a coupled mechanical system, with inputs and outputs being determined at each incremental step. These approaches work in a way similar to the FEMU method and, while the physical correctness of the network's predictions is guaranteed, the first drawback is the need to write a FEM code based on an automatic differentiation package (e.g., Pytorch or TensorFlow), making the FEA step time-consuming. A second drawback, is the need to run multiple FEA simulations run per epoch, rendering the ANN training computationally demanding. In the domain of hyperelasticity, Huang et al. [40] proved that an ANN could be made thermodynamically consistent by predicting the hyperelastic energy instead of the stress tensor, which was computed by automatic differentiation. A similar earlier approach was shown by Man and Furukawa [43] using an energy-based loss function to indirectly train an ANN. Thakolkaran et al. [3] used an ensemble of Input Convex Neural Networks (ICNNs) to successfully learn the constitutive response of a set of well-known hyperelastic constitutive models, resorting only to full-field data. Still in the framework of hyperelasticity, in a recent work by Jeong et al. [44] an ensemble of four independent Physics-Informed Neural Networks (PINNs) was employed to extract the material behaviour from full-field data, with physical constraints being encoded within a multi-objective loss. Despite the promising results and generalization capacity shown from the last two approaches, both resort to training several independent neural network models, which adds complexity. On another recent work in the domain of hyperplasticity, Franke et al. [45] employ an advanced energy–momentum scheme for time discretization to improve the performance of a Physics-Augmented Neural Network (PANN) in dynamical simulations, ensuring energy and momentum conservation. The authors report excellent results, however it is not clear

their approach is applicable to real experimental data. Finally, notable contributions by Fläschel et al. aimed at the unsupervised discovery of an explicit closed form expression of the yield surface of a certain category of materials [46] and its extension to a general class of materials [47]. These approaches ensure physically consistent solutions, however, the authors still resort to several assumptions about the existence of internal history variables and the explicit formulations of the models.

In the current work a novel approach is proposed by coupling the VFM with a single GRU-based RNN model (Fig. 2). By doing so, one can use force and displacement data to indirectly train the neural network. The strains can be obtained from the spatial gradient of the displacements and fed as inputs to the RNN material model  $\mathcal{M}$ , which will provide the stress components. Then, during training, the equilibrium can be evaluated globally by the application of the PVW. The network's parameters are searched until the gap between the internal and the external parts of the virtual work is minimized. This methodology benefits from all the advantages of the VFM, while being compatible with experimental data coming from full-field measurements and allows to predict the full elastoplastic material response employing only one RNN model. The possibility of integrating the VFM with neural network training had previously been shown by the authors in a very foundational state on a limited-page conference paper [16]. Despite the purely academic nature, the work allowed to identify many key areas of improvement, effectively laying the foundation for a more solid approach, which is now hereby presented.

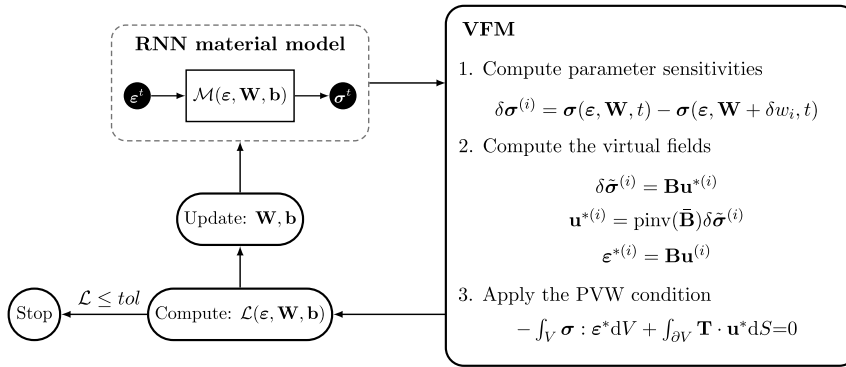


Fig. 2. Coupled RNN-VFM indirect training method for implicit constitutive modelling, using sensitivity-based virtual fields.

## 4. Materials and methods

### 4.1. Numerical DIC database

Synthetic data to train the RNN models was gathered from a set of FEA simulations of a displacement-driven heterogeneous biaxial test, using a cruciform specimen with a thickness  $h = 1$  mm and dimensions specified in Fig. 3(a) [36]. Due to geometric symmetries only one fourth of the specimen was modelled. Symmetry boundary conditions were defined at the edges  $y = 0$  and  $x = 0$ , with displacements being prescribed at each arm of the specimen, as depicted in Fig. 3(b). Several virtual experiments were generated with displacements of different magnitudes (in mm) being prescribed in paired combinations from the set:

$$(u_x, u_y) \in \{0.5, 1.0, 1.5, 2.0, 2.5, 3.0\}. \quad (18)$$

The FEA simulations were carried out with Abaqus Standard software, assuming plane stress conditions and a small strain formulation. Regarding the material behaviour, isotropic hardening was considered, with the flow stress evolving according to the Swift's law:

$$\sigma_y = K (\epsilon_0 + \bar{\epsilon}^p)^n, \quad \text{with } \epsilon_0 = \left( \frac{\sigma_0}{K} \right)^{1/n} \quad (19)$$

where  $\sigma_y$  is the flow stress,  $\bar{\epsilon}^p$  the equivalent plastic strain,  $\sigma_0$  the initial yield stress,  $K$  and  $n$  are material-specific hardening parameters. For the purpose, a soft steel material was considered, characterized by the set of constitutive parameters listed in Table 2.

The geometry was discretized using CPS4R elements and the mesh was refined according to a convergence study, from which an average element size of 0.52 mm was chosen, resulting in a total of 2121 elements. Similar refinement levels are cited in the literature for this type of specimen ([36,48], among others). The reduced integration elements were chosen in order to simulate DIC subsets from full-field measurements, thus, all the necessary field output variables were extracted at the centroid of the element. For each time step, the strain and stress tensors were extracted for all the elements and the resultant force applied at each of the specimen's arms was computed from the equilibrium of the internal forces, through the following:

$$Fl = \sum_{i=1}^n \sigma_i A_i h, \quad (20)$$

where  $F$  is the global force,  $l$  the length of the specimen,  $n$  is the number of elements,  $A$  is the area of the element and  $h$  the thickness.

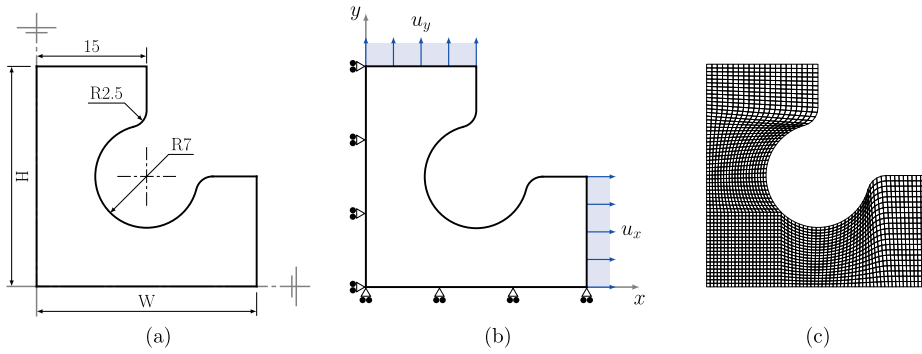
From the generated dataset, a total of 36 virtual experiments were randomly selected and split in half, such that one part was used for training and the remaining part for validation. The training set was split such that 80% of the included virtual experiments were used for training and the remaining 20% for testing. Scaling of the deformation input data was performed by first computing the mean and variance of each component of the strain tensor, in the training dataset, and then applying the following transformation (i.e. mean normalization):

$$\epsilon' = \frac{\epsilon - \boldsymbol{\mu}}{s}, \tag{21}$$

where  $\epsilon'$  is the scaled strain tensor,  $\boldsymbol{\mu} = [\mu_{\epsilon_{xx}} \ \mu_{\epsilon_{yy}} \ \mu_{\epsilon_{xy}}]$  is the vector of mean values and  $s = [s_{\epsilon_{xx}} \ s_{\epsilon_{yy}} \ s_{\epsilon_{xy}}]$  is the vector of variances of the strain tensor components, respectively. The statistical variables  $\boldsymbol{\mu}$  and  $s$  were saved in order to later apply the same transformation to the test set and during inference on the validation set, prior to organizing the data into strain sequences. For each virtual experiment, the inputs have dimensions  $(n_{\text{pts}} \times n_t \times k \times 3)$ , where  $n_{\text{pts}}$  is the number of measurement points, i.e. the number of elements,  $n_t$  is the number of time stages,  $k$  the RNN sequence length and the last dimension is the number of strain components.

**Table 2**  
Constitutive parameters for the soft steel material.

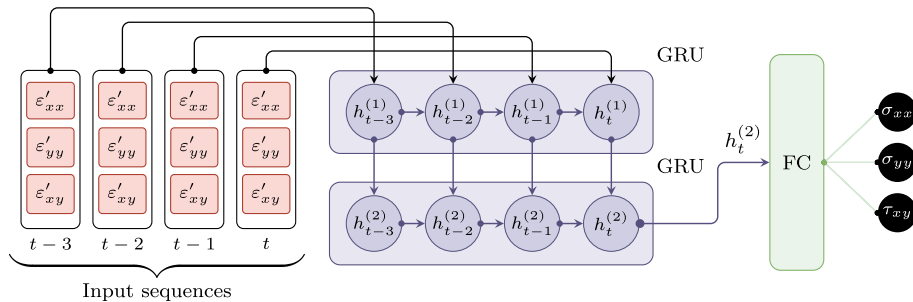
$E$ [GPa]	$\nu$	$\sigma_0$ [MPa]	$K$ [MPa]	$n$
210	0.33	160	565	0.26



**Fig. 3.** Biaxial test virtual experiment setup: (a) general dimensions of the cruciform specimen, (b) set of applied boundary conditions and (c) mesh density.

#### 4.2. ANN architecture

An implicit material model was developed based on a RNN consisting of an array of two GRU layers, followed by a fully-connected hidden layer with 32 neurons. The recurrent architecture processes strain data in the form of time sequences, unfolded using a sliding window of width  $k = 4$  time steps. Each sequence took into account the strain components of the last three time stages up to, and including, the current stage  $t$ . For each mechanical test, a padding of zeros was prepended to the initial time stage, allowing the strain sequencing to be initiated from that point in order to avoid loss of information for the first four time steps. A schematic representation of the architecture is depicted in Fig. 4.



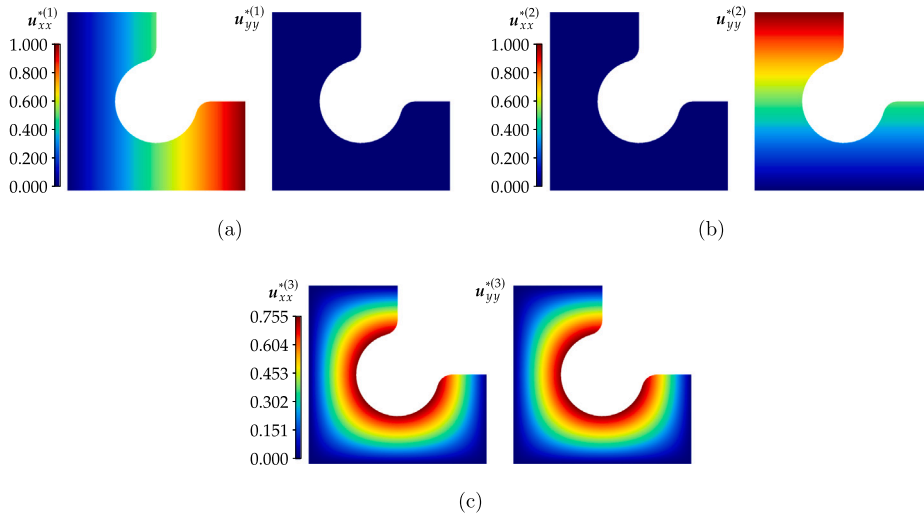
**Fig. 4.** GRU-based RNN for material behaviour modelling. Sequences of length  $k = 4$  of the scaled strain tensor  $\epsilon' = [\epsilon'_{xx} \ \epsilon'_{yy} \ \epsilon'_{xy}]$  are taken as input features and mapped to the stress tensor  $\sigma = [\sigma_{xx} \ \sigma_{yy} \ \tau_{xy}]$ , obtained as output. The green rectangular block depicts a linear fully-connected (FC) layer.

### 4.3. Training procedure

Two GRU-based RNN models with identical architectures and hyperparameters were trained in a Python environment using the PyTorch library. A first model (Dir-RNN) was trained employing a standard supervised learning approach, with labelled data. A second model (Ind-RNN) was trained employing the indirect training methodology based on the VFM. The transfer learning approach was initially used to validate the indirect training methodology and check the correct implementation of the VFM workflow. By doing so, it is expected that the Ind-RNN model to show faster convergence and lower error [49], benefiting from prior knowledge gained from direct training. More importantly, the loss should not degenerate if the integration of the VFM proves suitable for RNN training. For the Ind-RNN, the set of user-defined virtual fields from Table 3 was selected. For each virtual field, the variables  $W$  and  $H$  represent the width and the height of the specimen, respectively, while  $X$  and  $Y$  are the material coordinates in the reference configuration. According to [36], the first two virtual fields allow to isolate each component of the external virtual work, along the  $x$ - and  $y$ -direction, thus enabling the use of the corresponding components of global force acting on each arm of the specimen. The third virtual field allows to activate all the components of stress when calculating the internal virtual work, while discarding the contribution of the external virtual work. A graphical representation of these virtual fields is available in Fig. 5.

**Table 3**  
Set of user-defined virtual fields selected to train the Ind-RNN model.

Virtual field (i)	1	2	3
$u_{xx}^{*(i)}$	$\frac{X}{W}$	0	$\sin\left(\frac{X}{W}\pi\right)\sin\left(\frac{Y}{H}\pi\right)$
$u_{yy}^{*(i)}$	0	$\frac{Y}{H}$	$\sin\left(\frac{X}{W}\pi\right)\sin\left(\frac{Y}{H}\pi\right)$
$\varepsilon_{xx}^{*(i)}$	$\frac{1}{W}$	0	$\frac{\pi}{W}\cos\left(\frac{X}{W}\pi\right)\sin\left(\frac{Y}{H}\pi\right)$
$\varepsilon_{yy}^{*(i)}$	0	$\frac{1}{H}$	$\frac{\pi}{H}\sin\left(\frac{X}{W}\pi\right)\cos\left(\frac{Y}{H}\pi\right)$
$\varepsilon_{xy}^{*(i)}$	0	0	$\frac{1}{2}\left[\frac{\pi}{H}\sin\left(\frac{X}{W}\pi\right)\cos\left(\frac{Y}{H}\pi\right) + \frac{\pi}{W}\cos\left(\frac{X}{W}\pi\right)\sin\left(\frac{Y}{H}\pi\right)\right]$



**Fig. 5.** Contour maps for the components  $u_{xx}^{*(i)}$  and  $u_{yy}^{*(i)}$  of the selected set of user-defined virtual displacements.

Training was carried out using the Adam optimizer with a weight decay factor of  $10^{-3}$ . A linear warmup scheduler was employed in order to increase the learning rate from an initial value of  $10^{-3}$  up to  $5 \times 10^{-3}$  within the first five epochs. From that point onward, a gradual decay was applied according to a cosine annealing scheduler defined as [50,51]:

$$\eta_t = \eta_{\min} + \frac{1}{2} (\eta_{\max} - \eta_{\min}) \left( 1 + \cos\left(\frac{T_{\text{cur}}}{T_{\text{max}}}\pi\right) \right), \quad (22)$$

where  $\eta_t$  is the current learning rate,  $\eta_{\min}$  and  $\eta_{\max}$  are the minimum and maximum learning rates,  $T_{\text{cur}}$  and  $T_{\text{max}}$  are the current iteration and the maximum number of iterations, respectively. The maximum learning rate  $\eta_{\max}$  was set to  $5 \times 10^{-3}$  and the minimum learning rate  $\eta_{\min}$  to  $10^{-3}$ . Before each epoch, the order by which the virtual experiments were processed was shuffled. The



corresponding input data was also shuffled along the first dimension and fed into the network in the form of small mini-batches of size  $(66 \times n_t \times k \times 3)$  due to GPU memory limitations. The training process was set to run during a maximum of 10000 epochs, nonetheless, an early-stopping criterion was defined in order to prevent overfitting, such that training would be interrupted if no further improvement was registered in the test loss function  $\mathcal{L}$  after 500 epochs. For the Dir-RNN model the loss was based on the mean squared error between the RNN predicted stress,  $\hat{\sigma} = \mathcal{M}(\theta, \epsilon)$  and the target stress  $\sigma$ , defined as:

$$\mathcal{L}_{\text{MSE}}(\theta, \epsilon) = \frac{1}{2n} \sum_{i=1}^n (\hat{\sigma}(\theta, \epsilon) - \sigma)^2, \tag{23}$$

where  $n$  is the number of training samples. For the Ind-RNN model, the loss function from Eq. (9) was slightly adapted for the context of implicit constitutive modelling with RNNs, where the set of trainable parameters  $\theta$  can be seen, in a way, analogous to the classical model parameters  $\mathbf{x}$ , such that:

$$\mathcal{L}(\theta, \epsilon) = \sum_{i=1}^{n_{VFS}} \left[ \sum_{t=1}^{n_t} \left( \sum_{j=1}^{n_{ps}} \hat{\sigma}^j(\theta, \epsilon) \cdot \epsilon^{*j(i)} \cdot S^j - \mathbf{W}_{ext}^* \right)^2 \right]. \tag{24}$$

According to Martins et al. [36] the VFM-based loss should be normalized by the external virtual work. However, the virtual field 3 does not contain that contribution, hence, the loss has to be normalized by the maximum value of the internal virtual work.

#### 4.4. UMAT implementation

A second step to validate the trained RNN models, involved incorporating them as material constitutive models in the FEA programme Abaqus Standard, through a custom user material subroutine (UMAT). The language of choice for the UMAT development was C++ due to the existence of an official C++ implementation of PyTorch (LibTorch). The latter offers most of the functionality of its Python counterpart while maintaining a high-level syntax, which made it easier to deal with the ANN-related operations in the C++ environment. The architecture and data flow for the UMAT implementation are described in the flowchart of Fig. 6. The PyTorch-based RNN models had to be converted to a serializable format in order to work in the a Python-independent environment. All the RNN-related operations were kept separate from the UMAT subroutine (umat.cpp) and implemented in a function deployed as a dynamic-link library (DLL) (annlib.dll). The UMAT works as the interface between Abaqus Standard and the latter, essentially dealing with the input/output of the necessary variables and updating the strain history, which is stored as a state variable to be used on the next time step. The DLL function is invoked in the UMAT code and receives the strain history, the mean and the variance of the training data as input arguments. Within the DLL function, the strain history is converted to tensor format and scaled to have zero mean and unit variance. The RNN model is loaded persistently into memory in the code's first execution. The scaled strain history is then fed as input to the RNN in order to predict the stress tensor, which is returned as an output back to the UMAT, along with the stiffness matrix. For the latter, the user could either use the elastic stiffness matrix, provided a sufficiently small time-step during the simulation, or it could be easily computed from the RNN outputs, via an autodifferentiation method available in LibTorch. If this is the case, it should be noted that the computed jacobian may not be a symmetric positive definite matrix, which most likely would cause instabilities and convergence problems. The UMAT implementation for the ANN-based material models was validated with one-element tests, specifically uniaxial traction along  $x$ - and  $y$ -directions and a shear test.

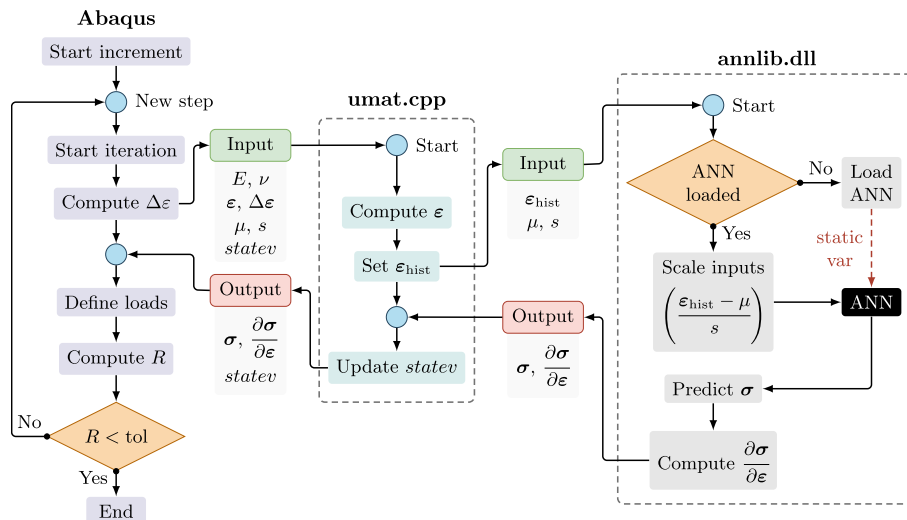


Fig. 6. Flowchart of the UMAT-RNN implementation.

## 5. Results and discussion

### 5.1. Learning curves

Regarding the Dir-RNN model (Fig. 7(a)), both train and test losses sharply decrease during the first 50 to 100 epochs, followed by a gradual evolution for the remaining epochs of training. The oscillations exhibited by both curves reveal a certain difficulty in achieving convergence. Nonetheless, the early-stopping was triggered after 1383 epochs, saving the model state at this point with a train and test loss of  $1.3182 \text{ MPa}^2$  and  $2.2190 \text{ MPa}^2$ , respectively. For the Ind-RNN model (Fig. 7(b)), both train and test losses show similar behaviour, however the initial loss is several orders of magnitude lower and the training was interrupted much earlier, with the early-stopping criteria being triggered after only 30 epochs. The model state was saved at this point, with a train and test loss of  $8.830 \times 10^{-10} \text{ J}^2$  and  $2.248 \times 10^{-9} \text{ J}^2$ , respectively. Although the loss functions for both models were measured in different units, the faster training and overall lower loss values of the Ind-RNN were to be expected due to the parameter transfer from the Dir-RNN model. In both cases, the gap between the train and test loss is not significant, indicating the RNNs did not overfit. Regarding the Ind-RNN model, the loss did not degenerate confirming the suitability of the integration of the VFM for RNN training.

### 5.2. RNN inference on validation data

To infer the quality of the RNNs predictions using unseen data, a virtual experiment with prescribed displacements  $u_x = u_y = 1.5 \text{ mm}$  was chosen as an example from the validation dataset. Here, both the global and local stress responses are presented, with the latter being evaluated at the centroid of element no. 10. For both cases, the RNN model predictions are compared against the FEA solution and the corresponding absolute difference is shown.

Regarding the global stress response, Figs. 8 and 9 show the contour maps of the stress components for the last time stage ( $n_t = 124$ ) of the virtual experiment. One can observe that the Dir-RNN model approximates the FEA solution quite well, exhibiting low absolute errors across the specimen, with the highest error magnitudes corresponding to key regions around the fillets near the central hole and on the lower-left corner. The Ind-RNN model shows similar levels of performance, however, its predictions are bound to higher absolute errors that spread to wider regions of the specimen, albeit with maximum magnitudes kept within reasonable levels. This model also has a slight tendency to underestimate the stress in specific regions around the central hole of the specimen, as denoted by the white blobs seen around those areas.

Looking at the results corresponding to the local stress response, Figs. 10 and 11 show the stress curves predicted by the Dir-RNN and Ind-RNN models, respectively, for the whole duration of the virtual experiment. The results show the predicted curves are a near-perfect match of the FEA solution, with both models showing similar levels of performance. This is supported by the corresponding absolute error curves that register very low mean and maximum values. Nonetheless, the Ind-RNN model shows, in general, slightly higher deviations for all the stress components. This was to be expected for the indirectly-trained model. For both RNN models, the error curves appear smooth, with the exception of sharp peaks, that show around the yield transition region (approx.  $n_t \in [20, 30]$ ), corresponding to the highest error magnitudes. The results successfully attest the capacity of the developed GRU-based RNN models on capturing the full elastoplastic response of the material.

### 5.3. UMAT validation

Figs. 12 and 13 show the uniaxial tensile and shear strain-stress curves obtained for the one-element tests with the UMAT implementation of the Dir-RNN and the Ind-RNN models, respectively. The curves are compared with the FEA solution and with the RNNs' inferred predictions. The corresponding absolute errors along the time stages are also shown. The results show the Dir-RNN model is capable of predicting the uniaxial behaviour very well at both rolling (RD) and transverse (TD) directions with overall low absolute errors, however some numerical instabilities were detected with the test along the  $x$ -direction (RD), resulting in slight oscillations towards the final segment of the stress curve. The lowest errors are exhibited for the uniaxial traction along the  $y$ -direction (TD), especially for the UMAT-version of the Dir-RNN model. Higher deviations were registered for the shear response. Regarding the Ind-RNN model, it can be seen that its predictions provide identical results, however, high discrepancies are observed with its UMAT implementation. Once again, the lowest errors are shown for the  $y$ -direction (TD), nonetheless the model seems unable to predict the initial state correctly. The highest instabilities seem to occur for the test along the  $x$ -direction (RD). Although these results are not perfect, it is interesting to see that the RNNs were able to capture the material behaviour from the biaxial experiments in such a way that they are still capable of performing relatively well in a completely different setting. The source of the instabilities is essentially of numerical nature induced by spurious predictions interfering with the equilibrium iterations. Taking this into account, although the feasibility of the proposed indirect training methodology is confirmed, it certainly would benefit of the application of constraints.

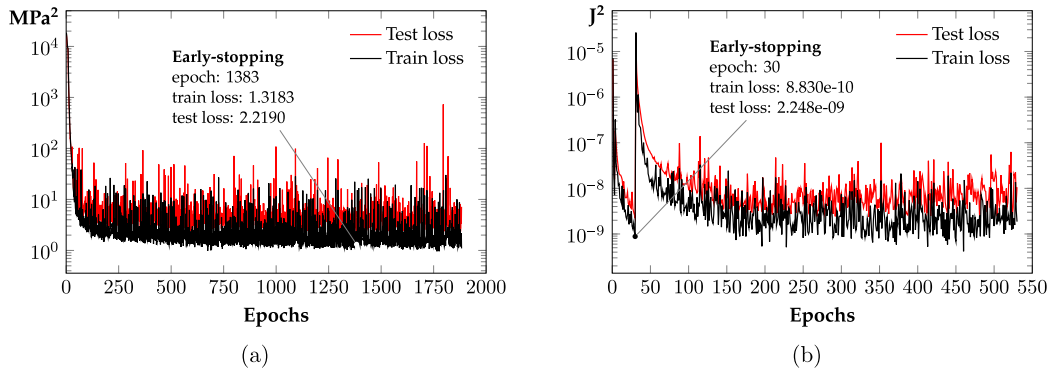


Fig. 7. Learning curves for (a) the Dir-RNN model and (b) the Ind-RNN model based on the VFM.

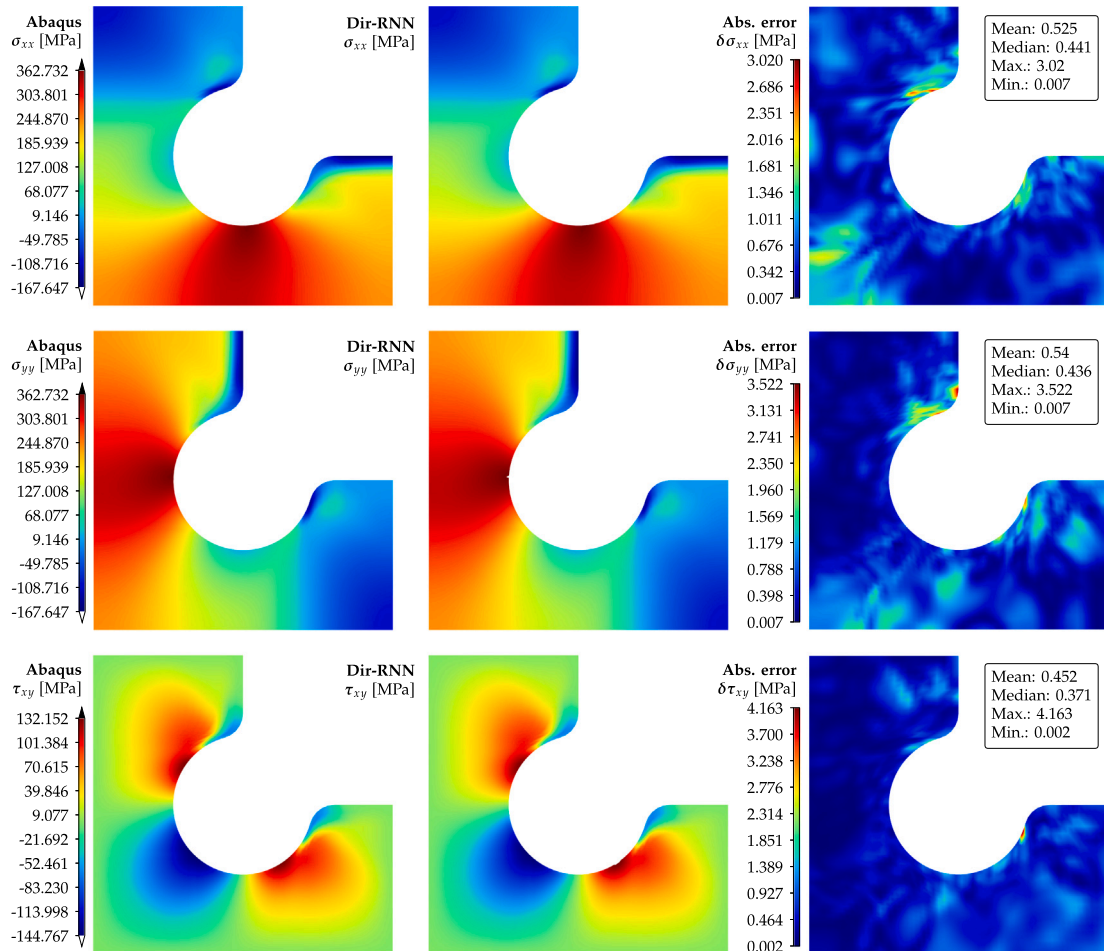


Fig. 8. Contour maps of the three components of stress for stage  $n_t = 124$  of the virtual experiment with boundary conditions  $u_x = u_y = 1.5$  mm. Comparison between the FEA solution and the Dir-RNN model predictions and corresponding absolute difference.

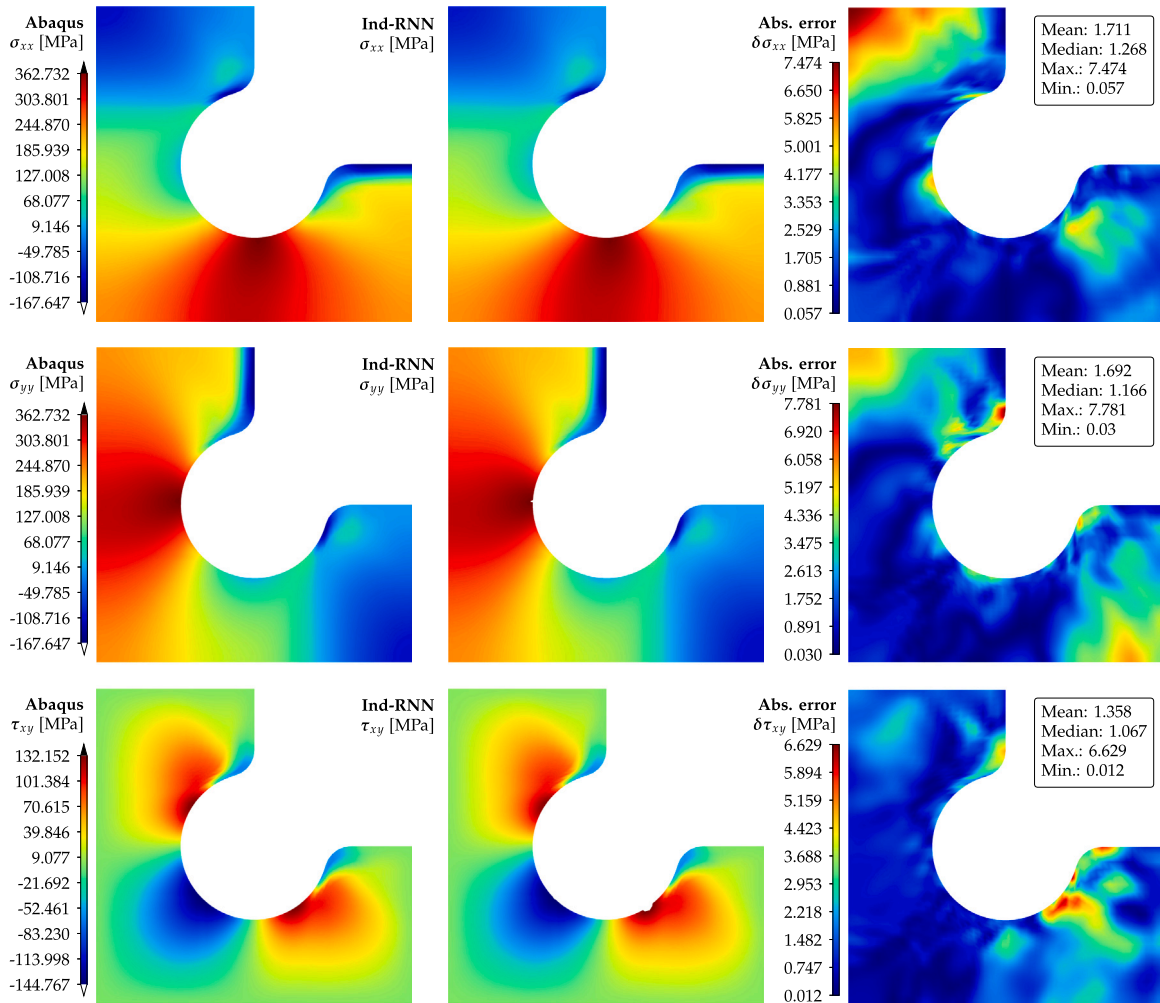


Fig. 9. Contour maps of the three components of stress for stage  $n_t = 124$  of the virtual experiment with boundary conditions  $u_x = u_y = 1.5$  mm. Comparison between the FEA solution and the Ind-RNN model predictions and corresponding absolute difference.

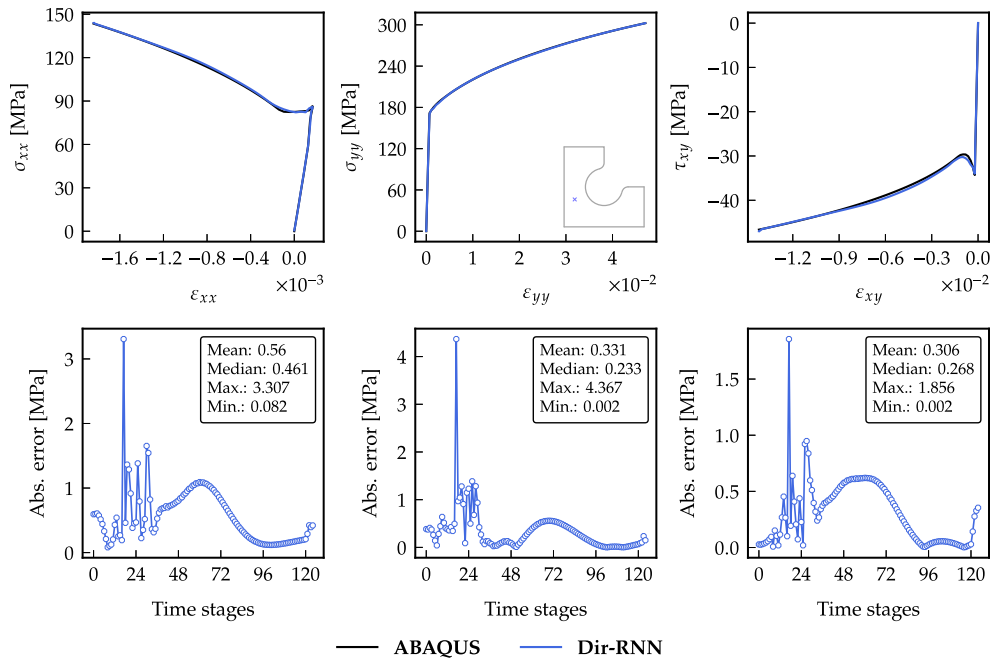


Fig. 10. Local stress evaluated at element 10, for the virtual experiment with boundary conditions  $u_x = u_y = 1.5$  mm. Comparison with FEA solution and Dir-RNN inference, with corresponding absolute error.

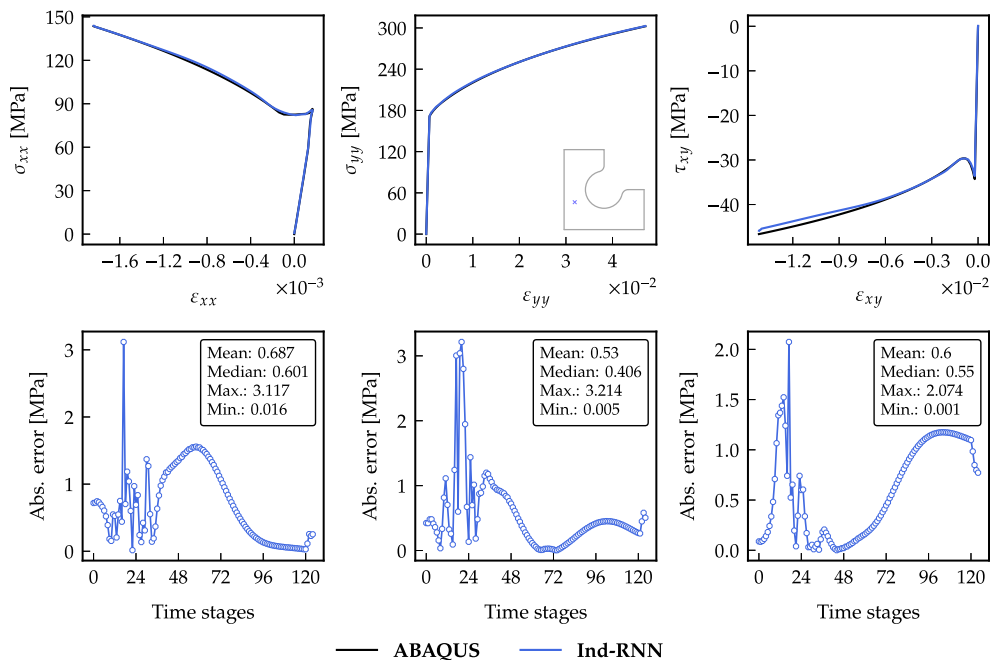


Fig. 11. Local stress evaluated at element 10, for the virtual experiment with boundary conditions  $u_x = u_y = 1.5$  mm. Comparison with FEA solution and Ind-RNN inference, with corresponding absolute error.

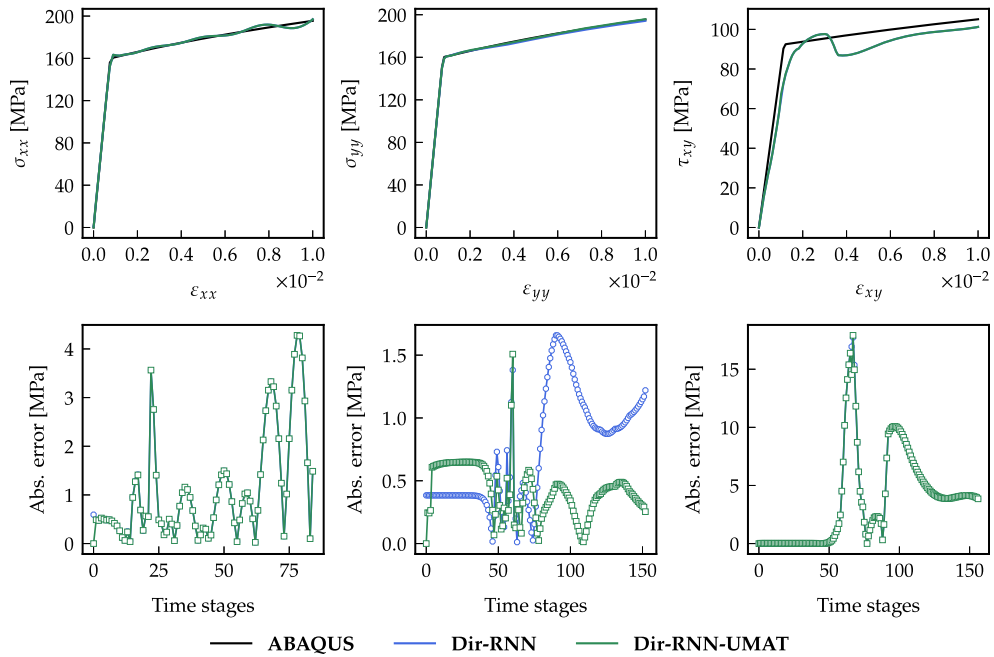


Fig. 12. Uniaxial tensile and shear strain-stress curves. Comparison between the FEA solution based on the Swift's law (Eq. (19)) and the UMAT implementation of the Dir-RNN model, with the corresponding absolute error along the time stages.

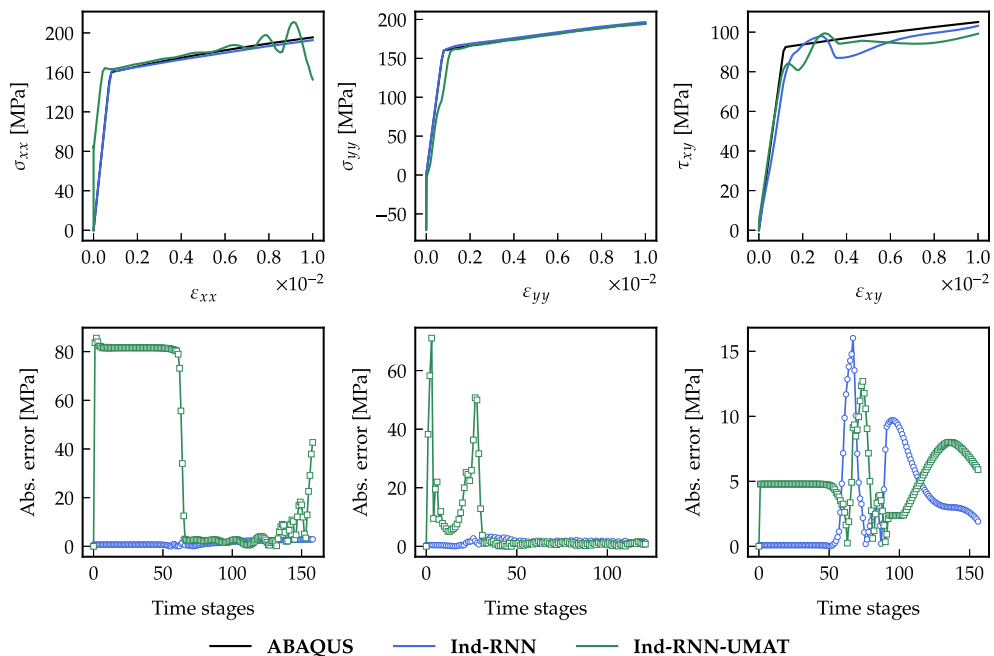


Fig. 13. Uniaxial tensile and shear strain-stress curves. Comparison between the FEA solution based on the Swift's law (Eq. (19)) and the UMAT implementation of the Ind-RNN model, with the corresponding absolute error along the time stages.

## 6. Concluding remarks

A new methodology using the VFM to indirectly train RNNs for implicit constitutive modelling was presented. The novel method was compared against the standard supervised learning approach, applied to a GRU-based RNN architecture. The purpose of the comparison was two-fold: to check the suitability of the VFM implementation when applied to neural network training and to validate the novel implementation, both as an isolated model and when incorporated in a FEA software. It was shown that the VFM-based approach using user-defined virtual fields was suitable for the context of RNN training. The predicted stress curves and contour maps from model inference confirmed that the indirectly trained model was capable of the same level of performance as the standard labelled training approach, when transfer learning was employed. Further validation studies were conducted with one-element tests, taking into consideration the incorporation of the models here presented into a FEA software. In this context, the standard model performed the best and has shown to be more stable, while the VFM-based model response was hindered by spurious predictions. These problems are expected to be solved with constrained optimization procedures in future works. However, the focus of the present paper was aimed at validating the VFM-based training methodology, which is deemed suitable for the purpose. Future analyses will also be aimed at evaluating the capabilities of the VFM-based training method, without transfer learning and further extension of the analysis with sensitivity-based virtual fields.

## Disclaimer

The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

## CRediT authorship contribution statement

**Rúben Lourenço:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Petia Georgieva:** Writing – review & editing, Supervision, Resources, Funding acquisition. **Elias Cueto:** Writing – review & editing, Supervision, Resources. **A. Andrade-Campos:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Funding acquisition, Formal analysis, Conceptualization.

## Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Ruben Lourenco reports financial support and travel were provided by the Portuguese Foundation for Science and Technology. A. Andrade-Campos reports financial support, administrative support, equipment, drugs, or supplies, and travel were provided by European Commission. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Acknowledgements

Rúben Lourenço acknowledges the Portuguese Foundation for Science and Technology (FCT) for the financial support (grant no. 2020.05279.BD), co-financed by the European Social Fund, through the Regional Operational Programme CENTRO 2020. This project has received funding from the Research Fund for Coal and Steel (grant agreement no. 888153). This project was also supported by the European Union NextGenerationEU, through the National Recovery and Resilience Plan of the Republic of Bulgaria (project no. BG-RRP-2.004-0005). This paper was also supported by the projects UIDB/00481/2020 - FCT, Portugal (doi: [10.54499/UIDB/00481/2020](https://doi.org/10.54499/UIDB/00481/2020)), UIDP/00481/2020 - FCT, Portugal (doi: [10.54499/UIDP/00481/2020](https://doi.org/10.54499/UIDP/00481/2020)), UIDB/50008/2020 - FCT (doi: [10.54499/UIDB/50008/2020](https://doi.org/10.54499/UIDB/50008/2020)) and UIDP/50008/2020 - FCT (doi: [10.54499/UIDP/50008/2020](https://doi.org/10.54499/UIDP/50008/2020)). This work was also supported by the Spanish Ministry for Digital Transformation and the Civil Service, through the ENIA 2022 Chairs for the creation of university-industry chairs in AI (grant no. TSI-100930-2023-1).

## References

- [1] A. Andrade-Campos, S. Thuillier, J. Martins, P. Carlone, F. Tucci, R. Valente, R.M. Paulo, R.J.A. de Sousa, Integrated design in welding and incremental forming: Material model calibration for friction stir welded blanks, *Procedia Manuf.* 47 (2020) 429–434, <http://dx.doi.org/10.1016/j.promfg.2020.04.327>, 23rd International Conference on Material Forming.
- [2] P. Weber, W. Wagner, S. Freitag, Physically enhanced training for modeling rate-independent plasticity with feedforward neural networks, *Comput. Mech.* 72 (4) (2023) 827–857, <http://dx.doi.org/10.1007/s00466-023-02316-9>.
- [3] 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>.
- [4] J. Dornheim, L. Morand, H.J. Nallani, D. Helm, Neural networks for constitutive modeling: From universal function approximators to advanced models and the integration of physics, *Arch. Comput. Methods Eng.* (2023) <http://dx.doi.org/10.1007/s11831-023-10009-y>.
- [5] J. Martins, A. Andrade-Campos, S. Thuillier, Comparison of inverse identification strategies for constitutive mechanical models using full-field measurements, *Int. J. Mech. Sci.* 145 (2018) 330–345, <http://dx.doi.org/10.1016/j.ijmecsci.2018.07.013>.
- [6] M. Conde, Y. Zhang, J. Henriques, S. Coppiters, A. Andrade-Campos, Design and validation of a heterogeneous interior notched specimen for inverse material parameter identification, *Finite Elem. Anal. Des.* 214 (2023) 103866, <http://dx.doi.org/10.1016/j.finel.2022.103866>.
- [7] B. Barroqueiro, A. Andrade-Campos, J.D. de Oliveira, R. Valente, Design of mechanical heterogeneous specimens using topology optimization, *Int. J. Mech. Sci.* 181 (2020) 105764, <http://dx.doi.org/10.1016/j.ijmecsci.2020.105764>.
- [8] M. Gonçalves, A. Andrade-Campos, B. Barroqueiro, On the design of mechanical heterogeneous specimens using multilevel topology optimization, *Adv. Eng. Softw.* 175 (2023) 103314, <http://dx.doi.org/10.1016/j.advengsoft.2022.103314>.
- [9] J. Henriques, A. Andrade-Campos, J. Xavier, On the inverse identification of sheet metal mechanical behaviour using a heterogeneous Arcan virtual experiment, in: *Material Forming*, Vol. 28, ESAFORM 2023, 2023, pp. 1131–1142, <http://dx.doi.org/10.21741/9781644902479-124>.
- [10] T. Marwala, *Finite Element Model Updating Using Computational Intelligence Techniques: Applications to Structural Dynamics*, Springer London, 2010.
- [11] F. Pierron, M. Grédiac, *The Virtual Fields Method: Extracting Constitutive Mechanical Parameters from Full-Field Deformation Measurements*, Springer Science & Business Media, 2012.
- [12] S. Avril, M. Bonnet, A.-S. Bretelle, M. Grédiac, F. Hild, P. Jenny, F. Latourte, D. Lemosse, S. Pagano, E. Pagnacco, F. Pierron, Overview of identification methods of mechanical parameters based on full-field measurements, *Exp. Mech.* 48 (2008) 381, <http://dx.doi.org/10.1007/s11340-008-9148-y>.
- [13] R. Ibáñez, E. Abisset-Chavanne, D. González, J.-L. Duval, E. Cueto, F. Chinesta, Hybrid constitutive modeling: data-driven learning of corrections to plasticity models, *Int. J. Mater. Form.* 12 (2019) 717–725, <http://dx.doi.org/10.1007/s12289-018-1448-x>.
- [14] J. Schmidt, M.R.G. Marques, S. Botti, M.A.L. Marques, Recent advances and applications of machine learning in solid-state materials science, *npj Comput. Mater.* 5 (2019) 2057–3960, <http://dx.doi.org/10.1038/s41524-019-0221-0>.
- [15] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, *Neural Netw.* 2 (5) (1989) 359–366, [http://dx.doi.org/10.1016/0893-6080\(89\)90020-8](http://dx.doi.org/10.1016/0893-6080(89)90020-8).
- [16] R. Lourenço, A. Andrade-Campos, P. Georgieva, The use of machine-learning techniques in material constitutive modelling for metal forming processes, *Metals* 12 (3) (2022) <http://dx.doi.org/10.3390/met12030427>.
- [17] Y.C. Lin, J. Zhang, J. Zhong, Application of neural networks to predict the elevated temperature flow behavior of a low alloy steel, *Comput. Mater. Sci.* 43 (4) (2008) 752–758, <http://dx.doi.org/10.1016/j.commatsci.2008.01.039>.
- [18] J. Ghaboussi, J.H. Garrett, X. Wu, Knowledge-based modeling of material behavior with neural networks, *J. Eng. Mech.* 117 (1) (1991) 132–153, [http://dx.doi.org/10.1061/\(asce\)0733-9399\(1991\)117:1\(132\)](http://dx.doi.org/10.1061/(asce)0733-9399(1991)117:1(132)).
- [19] T. Furukawa, G. Yagawa, Implicit constitutive modelling for viscoplasticity using neural networks, *Internat. J. Numer. Methods Engrg.* 43 (2) (1998) 195–219, [http://dx.doi.org/10.1002/\(SICI\)1097-0207\(19980930\)43:2<195::AID-NME418>3.0.CO;2-6](http://dx.doi.org/10.1002/(SICI)1097-0207(19980930)43:2<195::AID-NME418>3.0.CO;2-6).
- [20] M. Lefik, B.A. Schrefler, Artificial neural network as an incremental non-linear constitutive model for a finite element code, *Comput. Methods Appl. Mech. Engrg.* 192 (28–30) (2003) 3265–3283, [http://dx.doi.org/10.1016/S0045-7825\(03\)00350-5](http://dx.doi.org/10.1016/S0045-7825(03)00350-5).
- [21] Y.M.A. Hashash, S. Jung, J. Ghaboussi, Numerical implementation of a neural network based material model in finite element analysis, *Internat. J. Numer. Methods Engrg.* 59 (7) (2004) 989–1005, <http://dx.doi.org/10.1002/nme.905>.
- [22] 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, <http://dx.doi.org/10.1002/nme.7319>.
- [23] M.B. Gorji, M. Mozaffar, J.N. Heidenreich, J. Cao, D. Mohr, On the potential of recurrent neural networks for modeling path dependent plasticity, *J. Mech. Phys. Solids* 143 (2020) 103972, <http://dx.doi.org/10.1016/j.jmps.2020.103972>.
- [24] A. Danoun, E. Prulière, Y. Chemisky, Thermodynamically consistent Recurrent Neural Networks to predict non linear behaviors of dissipative materials subjected to non-proportional loading paths, *Mech. Mater.* 173 (2022) 104436, <http://dx.doi.org/10.1016/j.mechmat.2022.104436>.
- [25] L. Medsker, L.C. Jain, *Recurrent Neural Networks: Design and Applications*, CRC Press, 1999.
- [26] F. Ghavami, A. Simone, Accelerating multiscale finite element simulations of history-dependent materials using a recurrent neural network, *Comput. Methods Appl. Mech. Engrg.* 357 (2019) 112594, <http://dx.doi.org/10.1016/j.cma.2019.112594>.
- [27] M. Mozaffar, R. Bostanabad, W. Chen, K. Ehmann, J. Cao, M.A. Bessa, Deep learning predicts path-dependent plasticity, *Proc. Natl. Acad. Sci. USA* 116 (52) (2019) 26414–26420, <http://dx.doi.org/10.1073/pnas.1911815116>.
- [28] Y. Heider, K. Wang, W.C. Sun, So(3)-invariance of informed-graph-based deep neural network for anisotropic elastoplastic materials, *Comput. Methods Appl. Mech. Engrg.* 363 (2020) 112875, <http://dx.doi.org/10.1016/j.cma.2020.112875>.
- [29] C. Bonatti, D. Mohr, One for all: Universal material model based on minimal state-space neural networks, *Sci. Adv.* 7 (26) (2021) eabf3658, <http://dx.doi.org/10.1126/sciadv.abf3658>.
- [30] J. Chung, Ç. Gülçehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014, <http://dx.doi.org/10.48550/arXiv.1412.3555>, CoRR abs/1412.3555.
- [31] K. Cho, B. van Merriënboer, Ç. Gülçehre, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, 2014, <http://dx.doi.org/10.48550/arXiv.1406.1078>, CoRR abs/1406.1078.
- [32] M. Grédiac, *Principe des travaux virtuels et identification*, *CR Acad. Sci.* 309 (309) (1989) 1–5.
- [33] A. Marek, F.M. Davis, F. Pierron, Sensitivity-based virtual fields for the non-linear virtual fields method, *Comput. Mech.* 60 (2017) 409–431, <http://dx.doi.org/10.1007/s00466-017-1411-6>.
- [34] M. Grédiac, F. Pierron, Applying the virtual fields method to the identification of elasto-plastic constitutive parameters, *Int. J. Plast.* 22 (4) (2006) 602–627, <http://dx.doi.org/10.1016/j.ijplas.2005.04.007>.
- [35] 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>.
- [36] J. Martins, A. Andrade-Campos, S. Thuillier, Calibration of anisotropic plasticity models using a biaxial test and the virtual fields method, *Int. J. Solids Struct.* 172–173 (2019) 21–37, <http://dx.doi.org/10.1016/j.ijsolstr.2019.05.019>.
- [37] A. Marek, F.M. Davis, M. Rossi, F. Pierron, Extension of the sensitivity-based virtual fields to large deformation anisotropic plasticity, *Int. J. Mater. Form.* 12 (2019) 457–476, <http://dx.doi.org/10.1007/s12289-018-1428-1>.



- [38] F. Teixeira-Dias, J. Pinho-Da-Cruz, R.A.F. Valente, R.A. de Sousa, *Método dos Elementos Finitos - Técnicas de Simulação Numérica em Engenharia*, Editora ETEP, 2010.
- [39] K. Xu, D.Z. Huang, E. Darve, Learning constitutive relations using symmetric positive definite neural networks, *J. Comput. Phys.* 428 (2021) 110072, <http://dx.doi.org/10.1016/j.jcp.2020.110072>.
- [40] D.Z. Huang, K. Xu, C. Farhat, E. Darve, Learning constitutive relations from indirect observations using deep neural networks, *J. Comput. Phys.* 416 (2020) 109491, <http://dx.doi.org/10.1016/j.jcp.2020.109491>.
- [41] K. Xu, A.M. Tartakovsky, J. Burghardt, E. Darve, Learning viscoelasticity models from indirect data using deep neural networks, *Comput. Methods Appl. Mech. Engrg.* 387 (2021) 114124, <http://dx.doi.org/10.1016/J.CMA.2021.114124>.
- [42] X. Liu, F. Tao, H. Du, W. Yu, K. Xu, Learning nonlinear constitutive laws using neural network models based on indirectly measurable data, *J. Appl. Mech. Trans. ASME* 87 (8) (2020) 1–8, <http://dx.doi.org/10.1115/1.4047036>.
- [43] H. Man, T. Furukawa, Neural network constitutive modelling for non-linear characterization of anisotropic materials, *Internat. J. Numer. Methods Engrg.* 85 (8) (2011) 939–957, <http://dx.doi.org/10.1002/nme.2999>.
- [44] I. Jeong, M. Cho, H. Chung, D.-N. Kim, Data-driven nonparametric identification of material behavior based on physics-informed neural network with full-field data, *Comput. Methods Appl. Mech. Engrg.* 418 (2024) 116569, <http://dx.doi.org/10.1016/j.cma.2023.116569>.
- [45] M. Franke, D.K. Klein, O. Weeger, P. Betsch, Advanced discretization techniques for hyperelastic physics-augmented neural networks, *Comput. Methods Appl. Mech. Engrg.* 416 (2023) 116333, <http://dx.doi.org/10.1016/j.cma.2023.116333>.
- [46] M. Flaschel, S. Kumar, L. De Lorenzis, Discovering plasticity models without stress data, *npj Comput. Mater.* 8 (1) (2022) 91, <http://dx.doi.org/10.1038/s41524-022-00752-4>.
- [47] 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>.
- [48] P. Prates, M. Oliveira, J. Fernandes, A new strategy for the simultaneous identification of constitutive laws parameters of metal sheets using a single test, *Comput. Mater. Sci.* 85 (2014) 102–120, <http://dx.doi.org/10.1016/j.commatsci.2013.12.043>.
- [49] J.N. Heidenreich, C. Bonatti, D. Mohr, Transfer learning of recurrent neural network-based plasticity models, *Internat. J. Numer. Methods Engrg.* 125 (1) (2024) e7357, <http://dx.doi.org/10.1002/nme.7357>.
- [50] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An imperative style, high-performance deep learning library, in: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Curran Associates Inc., Red Hook, NY, USA, 2019, pp. 8026–8037.
- [51] I. Loshchilov, F. Hutter, SGDR: Stochastic gradient descent with warm restarts, 2017, <http://dx.doi.org/10.48550/arXiv.1608.03983>, arXiv:1608.03983.