

A recurrent neural network-accelerated multi-scale model for elasto-plastic heterogeneous materials subjected to random cyclic and non-proportional loading paths

Ling Wu^a, Van Dung Nguyen^{a,1}, Nanda Gopala Kilingar^a, Ludovic Noels^a

^a*University of Liege, Department of Aeronautics and Mechanical Engineering, Computational & Multiscale Mechanics of Materials, Allée de la découverte 9, B-4000 Liège, Belgium*

Abstract

An artificial Neural Network (NNW) is designed to serve as a surrogate model of micro-scale simulations in the context of multi-scale analyzes in solid mechanics.

The design and training methodologies of the NNW are developed in order to allow accounting for history-dependent material behaviors. On the one hand, a Recurrent Neural Network (RNN) using a Gated Recurrent Unit (GRU) is constructed, which allows mimicking the internal variables required to account for history-dependent behaviors since the RNN is self-equipped with hidden variables that have the ability of tracking loading history. On the other hand, in order to achieve accuracy under multi-dimensional non-proportional loading conditions, training of the RNN is achieved using sequential data. In particular the sequential training data are collected from finite element simulations on an elasto-plastic composite RVE subjected to random loading paths. The random loading paths are generated in a way similar to a random walking in stochastic process and allows generating data for a wide range of strain-stress states and state evolution.

The accuracy and efficiency of the RNN-based surrogate model is tested on the structural analysis of an open-hole sample subjected to several loading/unloading cycles. It is shown that a similar accuracy as with a FE² multi-scale simulation can be reached with the RNN-based surrogate model as long as the local strain state remains in the training range, while the computational time is reduced by four orders of magnitude.

Keywords: Artificial Neural Network, Recurrent Neural Network, Surrogate,

Email addresses: l.wu@ulg.ac.be (Ling Wu), vandung.nguyen@uliege.be (Van Dung Nguyen), ngkilingar@uliege.be (Nanda Gopala Kilingar), l.noels@ulg.ac.be (Ludovic Noels)

¹Postdoctoral Researcher at the Belgian National Fund for Scientific Research (FNRS)

1. Introduction

The application of heterogeneous materials increases rapidly in a large variety of engineering and industrial fields. For the purpose of reliable structural analyzes, accurate modeling of the mechanical response of heterogeneous materials is demanded. Although the ability of computers is developing at high speed, the coexistence of different phases and their complex geometrical arrangements make their detailed analyzes using a massive finite element discretization unreachable. Therefore, multi-scale methods were extensively developed in order to carry out a structural analysis at an affordable computational cost.

Homogenization-based multi-scale analyzes have been extensively developed, see the reviews [1–3]. In these methods, the effect of material heterogeneity is resolved at a structural material point through an homogenization process.

Among the existing different homogenization methods, computational homogenization is a purely numerical process, which solves the Boundary Value Problem (BVP) defined on a Representative Volume Element (RVE) of heterogeneous materials using a full field finite element discretization of the microstructure. This makes computational homogenization the most versatile homogenization method since it can be applied on a large variety of heterogeneous material systems including for coupled multi-physics problems [4–6]. The multi-scale analysis based on computational homogenization is usually called FE^2 analysis [7–10]. In such an approach, the macro-scale structure defines a Boundary Value Problem (BVP) which is solved by considering homogenized material properties extracted, at each (macro) material point of interest, from the coupled resolution of meso-scale BVPs, see Fig. 1(a). Although, as a multi-scale approach, FE^2 is theoretically more efficient than a direct full field finite element model, the requirement of solving the meso-scale RVE BVP in a coupled way at all the macro-scale material points involves high numerical resources in terms of time and memory, which limits the applicability of the FE^2 multi-scale simulations to reduced size problems.

Reduce Order Model (ROM) aims at providing an efficient resolution process by considering a reduced version of the full-field analyzes. This can be applied on the meso-scale BVPs involved in the multi-scale analyzes, with the general idea of trading space for time. Reduced versions of the meso-scale models are created by projection of the governing equations into suitably selected sub-spaces based on pre-off-line finite element simulations on the RVE: the number of displacement degrees of freedom is reduced by means of proper orthogonal decomposition of the displacement field [11] and a so-called hyper-reduction is further applied to reduce the computations on internal forces [12, 13]. When using these ROMs during FE^2 multi-scale simulations, the resolution of the ROM equations is coupled to the macro-scale simulation. Alternatively, the pre-off-line finite element simulations on the RVE can be used to build a surrogate model by means of constructing mapping functions, such as through kernel

methods [14], artificial neural networks etc. We refer to the review [15]. After this so-called training step, as illustrated in Fig. 1(b), the surrogate model can be used as the constitutive law of a single-scale simulation, leading to highly efficient simulations while still accounting for the micro-structure of the material.

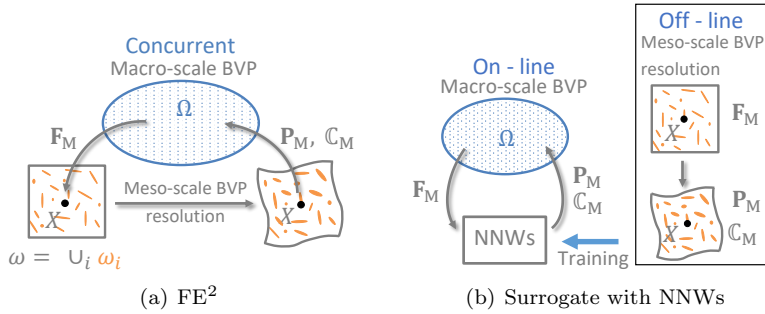


Figure 1: Homogenization-based multi-scale models: (a) A FE^2 multi-scale simulation implies the coupled resolution of BVPs at the different scales; and (b) The meso-scale BVPs resolutions during the multi-scale simulation are substituted by a surrogate model trained from pre-off-line simulations of the meso-scale BVP.

Artificial neural networks (NNWs) have increasingly attracted attention in the field of computational mechanics, especially in fields where material behavior is complex and/or intensive computation is required. A NNW, can be viewed as a function $F : \mathbf{u} \rightarrow \mathbf{v}$, in which $\mathbf{u} = [u_1 \ u_2 \ \dots \ u_{n_0}]^T$ is the input and $\mathbf{v} = [v_1 \ v_2 \ \dots \ v_{n_N}]^T$ is the output, see Fig. 2. They were used to replace parts of material constitutive laws, such as evolution laws in irreversible nonlinear analyzes for visco-plasticity [16], for cyclic plasticity [17] and in [18, 19] for constitutive law of interface. In finite element simulations, NNWs were used as surrogate for constitutive laws of non-linear materials in [20–22], and of rate-dependent materials in [23]. NNWs served as a surrogate for an elasto-plastic material model for parameters identification in [24] and for micro-mechanics model parameters identification in [25]. In this last reference, NNWs were adopted to substitute complex material homogenization constitutive laws to accelerate the massive computations involved in Bayesian inference. NNW was also adopted as a surrogate of the damaged-elastic response of meso-scale volumes of bones in [26].

When it comes to use NNWs as a surrogate model in FE^2 analyzes, this can be achieved either by approximating the strain energy density surface as suggested by [27, 28], the stress-strain responses as achieved by [29–32], or a function of both the current stress and the plastic dissipation density [22]. Although NNWs have shown to be reliable surrogate models in elasticity and in non-linear elasticity [27], when it comes to irreversible behaviors, the loading history plays an important role in RVE response involving more difficulties both in the NNW architecture definition and in its training. The NNW is actually a network of artificial neurons, which perform a weighted sum op-

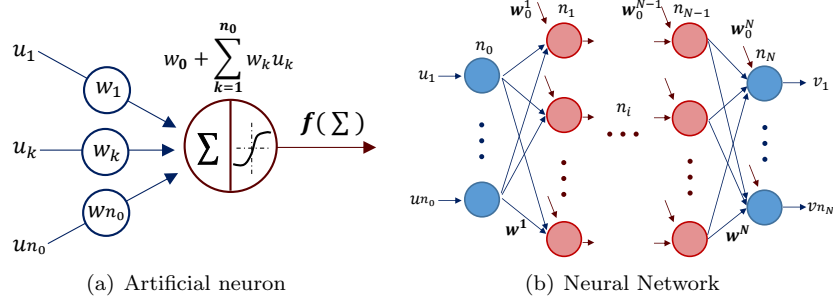


Figure 2: Description of a feed-forward Neural Network: (a) An artificial neuron; (b) An artificial Neural Network.

eration on input $(w_0 + \sum_{k=1}^{n_0} w_k x_k)$ in order to produce output through an activation function of the weighted sum $f(\Sigma)$, see Fig. 2(a). The most commonly used architecture is the feed-forward neural network in which the information moves only along the forward direction, from the input nodes and to the output nodes *via* $N - 1$ hidden layers, see Fig. 2(b). The weights w^i_{kj} , with $i = 1, \dots, N$; $k = 1, \dots, n_{i-1}$ and $j = 1, \dots, n_i$, and bias w^i_{0j} , with $i = 1, \dots, N$ and $j = 1, \dots, n_i$, see the notations in Fig. 2(b), are obtained through a so-called training of the NNW. However, since the information moves only along one direction, in case of irreversible behaviors, a feed-forward neural network can only be used as surrogate for the RVE resolution of heterogeneous material under monotonic loading [29]. For general loading conditions, state variables are needed to account for the loading history at the homogenized material level. According to the physical problem under investigation, some state variables can be defined at the meso-scale level, *i.e.* on the RVE, and serve as a part of the input, beside the strain variables, of a feed-forward neural network. In [30], using state variables with the support vector machine as a solution strategy for the decision of loading/unloading, a feed-forward network was used to extract meso-scale resolution for multi-scale failure analysis, but only 1D loading conditions were considered at the macro-scale. In [31], meso-scale plastic strains were used as state variables and were updated at each loading step through an empirical model or in combination with another feed-forward neural network. However, when the RVE has a complex micro-structure, the state variables are not always easy to be defined and their update is also troublesome because of the anisotropy induced by the historical dependent behavior, *e.g.* as in elasto-plasticity.

The difficulty inherent to the definition of history-dependent state variables can be avoided by replacing the feed-forward network with a Recurrent Neural Network (RNN). The idea behind RNNs is to make use of sequential information: RNNs are called recurrent because they perform the same task on every step input of a sequence, with the output being dependent on the previous evaluations. We can say that RNNs have a “memory”, which captures information about what has been calculated so far. A typical illustration of RNNs is pre-

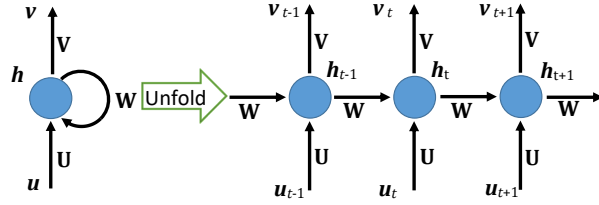


Figure 3: Recurrent Neural Network, with a hidden state passing through the input series; \mathbf{U} , \mathbf{V} and \mathbf{W} are the weights matrix related to the linear operations on the input vector \mathbf{u} , output vector \mathbf{v} and internal variables \mathbf{h} , respectively. In the recurrent neural network, the operations at step t use the internal variables evaluated at step $t - 1$.

sented in Fig. 3, in which \mathbf{u} and \mathbf{v} are respectively input and output variables and \mathbf{h} are called hidden variables. The latter are used to track the input history: instead of defining state variables manually, those historical variables can be extracted automatically during the RNNs training with sequential data. In order to account for inelastic responses, recurrent neural networks, in particular Long Short Term Memory networks (LSTMs), were used in [32] to study cyclic loading of elasto-visco-plastic materials; however only 1D loading conditions were achieved. Nevertheless, theoretically speaking, NNWs should be able to approximate any multi-dimensional non-linear function accurately under the conditions that its structure is properly designed and it has been appropriately trained. Recently, model order reduction with sub-structuring was enhanced by so-called meta-elements [33], which predict the inside and boundary history dependent responses of the sub-structures, or patches of elements, through a neural network. The neural network architecture used was a combination of convolution –to capture the spatial distribution– and recurrent –to capture the time history– neural-networks. The training was achieved by random superposition of different uni-axial and bi-axial loading modes with a random noise.

With the aim of using RNN as a surrogate model of the meso-scale BVP finite element resolution in the context of FE^2 multi-scale simulations, this paper designs a RNN with a Gated Recurrent Unit (GRU). We note that such a GRU-based RNN has also been used recently in [34, 35] to represent the meso-scale response of 2D RVEs subjected to non-proportional loading conditions. In order to achieve accuracy under multi-dimensional non-proportional loading conditions, the sequential training data are herein collected from finite element simulations on an elasto-plastic composite RVE subjected to random loading paths. In this paper, the random loading paths are generated in a way similar to a random walking in stochastic process, in contrast to [34, 35] where respectively Gaussian process regression and cubic-spline interpolations were used to generate loading paths in the strain space, and are shown to be efficient for the training of the RNNs since it encompasses multiple possible stress-strain histories. Besides, since we formulate the problem in a finite strain setting, the inputs and outputs are formulated in order to respect the frame indifference of the meso-scale BVP. As a result, the obtained RNN can be used as an accurate

surrogate of the meso-scale BVP during FE² multi-scale simulations as shown by studying an open hole sample subjected to multiple loading cycles. Although in this paper only 2D problems are considered, we believe that the method can be extended without other difficulty than obtaining the training data, to 3D cases.

The paper is organized as follows. Section 2 summarizes the computational homogenization framework. The design of the RNN with a Gated Recurrent Unit (GRU), aiming at serving as surrogate model of the meso-scale BVP finite element resolution, and the developed training strategy are presented in Section 3. This training strategy is then applied by considering an elasto-plastic composite RVE in Section 4, where it is shown that the RNN can predict random, cyclic and non-proportional loading in an accurate way. Finally, the method is applied in the context of FE² multi-scale simulation, in which several loading cycles of an open holed sample are studied. The results predicted by the multi-scale simulation solving the meso-scale BVP using the finite-element method are compared in terms of cost and accuracy to the case in which the trained RNN is used as a surrogate. It is shown that the resolution time is dramatically decreased while accuracy is achieved as long as the loading conditions on the meso-scale BVP remain in the training range.

2. Homogenization-based multi-scale simulation and computational homogenization

In this section, we briefly summarize the theory of multi-scale simulations in the context of finite-strain mechanics. Assuming the meso-scale BVP is solved using the finite-element method, we then present the main lines of the computational homogenization framework.

2.1. Homogenization-based multi-scale simulation

The BVPs that have to be solved at the two scales are herein defined and the scale transition theory is recalled.

2.1.1. Definition of the meso-scale BVP

The meso-scale BVP is usually defined on Representative Volume Elements (RVEs) which are parallelepipedic (rectangular in 2D), with planar boundary faces $\partial\omega$, see Fig. 1. At the micro-scale, it is assumed that the classical continuum mechanics equations hold and that the time for a stress wave to propagate in the meso-scale volume element remains negligible. Considering the material points $\mathbf{x} \in \omega$, in the absence of dynamical effects the equilibrium equations read

$$\mathbf{P}_m \cdot \nabla_0 = \mathbf{0} \quad \forall \mathbf{x} \in \omega, \quad (1)$$

$$\mathbf{P}_m \cdot \mathbf{N}_m = \mathbf{T}_m \quad \forall \mathbf{x} \in \partial\omega, \quad (2)$$

where the subscript “m” refers to the local value at the micro-scale, \mathbf{P}_m is the first Piola-Kirchhoff stress tensor, ∇_0 is the gradient operator with respect to

the micro-scale reference configuration, and \mathbf{T}_m is the surface traction, per unit reference surface, on the boundary $\partial\omega$ of outward unit normal \mathbf{N}_m expressed in the reference configuration.

The micro-scale problem is completed by the local constitutive laws of the different materials at a given time t and material point \mathbf{x} , which are written as

$$\mathbf{P}_m(\mathbf{x}, t) = \mathbf{P}_m(\mathbf{F}_m(\mathbf{x}, t); \mathbf{Z}_m(\mathbf{x}, \tau), \tau \in [0, t]), \quad (3)$$

where $\mathbf{F}_m(\mathbf{x}) = \mathbf{I} + \mathbf{u}_m \otimes \nabla_0$ is the micro-scale deformation gradient tensor evaluated in terms of the micro-scale displacement \mathbf{u}_m with \mathbf{I} the second-order identity tensor, and where \mathbf{Z}_m is a set of internal variables for history-dependent processes.

2.1.2. The macro-scale BVP

Assuming no dynamical effects, the macro-scale linear momentum equation reads

$$\mathbf{P}_M(\mathbf{X}) \cdot \nabla_0 + \mathbf{b}_M = \mathbf{0} \quad \forall \mathbf{X} \in \Omega, \quad (4)$$

where the subscript ‘‘M’’ refers to the values at the macro-scale and \mathbf{b}_M is the load per unit reference volume. The boundary conditions read

$$\mathbf{u}_M(\mathbf{X}) = \bar{\mathbf{u}}_M \quad \forall \mathbf{X} \in \partial_D \Omega, \quad \text{and} \quad (5)$$

$$\mathbf{P}_M(\mathbf{X}) \cdot \mathbf{N}_M = \bar{\mathbf{T}}_M \quad \forall \mathbf{X} \in \partial_N \Omega, \quad (6)$$

where $\bar{\mathbf{T}}_M$ is the surface traction, per unit reference surface, on the Neumann boundary $\partial_N \Omega$ of outward unit normal \mathbf{N}_M expressed in the reference configuration, $\bar{\mathbf{u}}_M$ is the constrained displacement on the Dirichlet boundary $\partial_D \Omega$.

The macro-scale BVP should be completed by a relation linking the macro-scale deformation gradient $\mathbf{F}_M = \mathbf{I} + \mathbf{u}_M \otimes \nabla_0$ to the macro-scale stress tensor \mathbf{P}_M under the form

$$\mathbf{P}_M(\mathbf{X}, t) = \mathbf{P}_M(\mathbf{F}_M(\mathbf{X}, t); \mathbf{Z}_M(\mathbf{X}, \tau), \tau \in [0, t]), \quad (7)$$

where, for history-dependent processes, the relation depends on internal variables \mathbf{Z}_M . In the context of homogenization-based multi-scale simulation, Eq. (7) is the mathematical representation of the meso-scale BVP via the scale transition.

2.1.3. The scale transition

In the context of homogenization theories, the scale transition (7) is rewritten as a relation between the volume averaging processes of the respective micro-scale deformation gradient tensor $\mathbf{F}_m(\mathbf{x})$ and stress tensor $\mathbf{P}_m(\mathbf{x})$ over the meso-scale volume element ω , with

$$\mathbf{F}_M(\mathbf{X}, t) = \frac{1}{V(\omega)} \int_{\omega} \mathbf{F}_m(\mathbf{x}, t) d\mathbf{x}, \quad \text{and} \quad (8)$$

$$\mathbf{P}_M(\mathbf{X}, t) = \frac{1}{V(\omega)} \int_{\omega} \mathbf{P}_m(\mathbf{x}, t) d\mathbf{x}. \quad (9)$$

Under the definitions (8-9), the requirement of energy consistency between the different scales, which corresponds to the Hill-Mandel condition, yields

$$\mathbf{P}_M : \delta \mathbf{F}_M = \frac{1}{V(\omega)} \int_{\omega} \mathbf{P}_m : \delta \mathbf{F}_m d\mathbf{x}. \quad (10)$$

A perturbation field $\mathbf{u}'(\mathbf{x})$ is introduced in the micro-scale displacement field $\mathbf{u}_m(\mathbf{x})$, which is thus rewritten under the form

$$\mathbf{u}_m(\mathbf{x}) = (\mathbf{F}_M - \mathbf{I}) \cdot (\mathbf{x} - \mathbf{x}_0) + \mathbf{u}'(\mathbf{x}), \quad (11)$$

where \mathbf{x}_0 is a reference point of ω . According to the definition (8), this perturbation field should satisfy the condition

$$\frac{1}{V(\omega)} \int_{\omega} \mathbf{u}'(\mathbf{x}) \otimes \nabla_0 d\mathbf{x} = \frac{1}{V(\omega)} \int_{\partial\omega} \mathbf{u}' \otimes \mathbf{N}_m d\mathbf{x} = \mathbf{0}, \quad (12)$$

where \mathbf{N}_m is the outward unit normal to $\partial\omega$ expressed in the reference configuration. Using the micro-scale displacement field expressed in terms of the perturbation field, Eq. (11), the Hill-Mandel condition (10) can be rewritten as

$$\mathbf{P}_M : \delta \mathbf{F}_M = \frac{1}{V(\omega)} \int_{\omega} \mathbf{P}_m : \delta \mathbf{F}_m d\mathbf{x} = \mathbf{P}_M : \delta \mathbf{F}_M + \frac{1}{V(\omega)} \int_{\omega} \mathbf{P}_m : (\delta \mathbf{u}' \otimes \nabla_0) d\mathbf{x}, \quad (13)$$

or again after integrating by parts and using the equilibrium Eqs. (1-2), as

$$\int_{\partial\omega} (\mathbf{P}_m \cdot \mathbf{N}_m) \cdot \delta \mathbf{u}' d\mathbf{x} = \int_{\partial\omega} \mathbf{T}_m \cdot \delta \mathbf{u}' d\mathbf{x} = 0. \quad (14)$$

2.2. Computational homogenization

In the context of FE² simulations, the virtual material law (7) is implicitly defined through the concurrent finite-element resolution of the meso-scale BVP through the scale transition formalism presented in Section 2.1.3. We here give the main lines of this finite-element resolution, also called computational homogenization. More details on the implementation can be found in [36].

2.2.1. Definition of the constrained micro-scale finite element problem

The resolution of the meso-scale BVP governed by the strong form (1-2) needs to satisfy the scale transition Eqs. (12) and (14). Therefore, its weak form is formulated using trial and test functions part of the minimum kinematic vector field $\mathcal{U}^{\min}(\omega)$ defined from Eq. (12) as

$$\mathcal{U}^{\min}(\omega) = \left\{ \delta \mathbf{u}' \in \mathcal{H}(\omega) \mid \int_{\partial\omega} \delta \mathbf{u}' \otimes \mathbf{n}_m d\mathbf{x} = 0 \right\}, \quad (15)$$

where $\mathcal{H}(\omega)$ is the Hilbert space.

The weak form of the meso-scale BVP is then stated as finding $\mathbf{u}' \in \mathcal{U}(\omega) \subset \mathcal{U}^{\min}(\omega)$ such that

$$\int_{\omega} \mathbf{P}_m(\mathbf{u}') : (\delta \mathbf{u}' \otimes \nabla_0) d\mathbf{x} = 0, \quad \forall \delta \mathbf{u}' \in \mathcal{U}(\omega) \subset \mathcal{U}^{\min}(\omega), \quad (16)$$

where $\delta \mathbf{u}' \in \mathcal{U}(\omega)$ is a test function belonging to an admissible kinematic vector field $\mathcal{U}(\omega)$ subset of the minimum kinematic field \mathcal{U}^{\min} . The resolution of this weak form (16) satisfies the Hill-Mandel condition (10) as it always makes the second term of the right hand side of Eq. (13) vanish.

This variational statement (16) of the Hill-Mandel condition was introduced in [37, 38], and is carried out by defining specific boundary conditions on the meso-scale volume element, with the detailed implementation provided in [36]. There are several commonly applied boundary conditions on the meso-scale volume elements ω , such as Kinematic Uniform Boundary Conditions (KUBCs), Periodic Boundary Conditions (PBCs), Zero Average Fluctuation Boundary Conditions (ZAFBCs), Static Uniform Boundary Conditions (SUBCs), *etc.*

In this work, the PBCs is adopted for which the admissible kinematic vector field \mathcal{U} is defined by

$$\mathcal{U}^{\text{PBC}}(\omega) = \left\{ \mathbf{u}' \in \mathcal{H}(\omega) \mid \mathbf{u}'(\mathbf{x}^+) = \mathbf{u}'(\mathbf{x}^-), \right. \\ \left. \forall \mathbf{x}^+ \in \partial\omega^+ \text{ and corresponding } \mathbf{x}^- \in \partial\omega^- \right\} \subset \mathcal{U}^{\min}(\omega), (17)$$

where the parallelepiped faces have been separated in opposite surfaces $\partial\omega^-$ and $\partial\omega^+$.

2.2.2. Homogenization

The computational homogenization refers to the resolution of the constrained micro-scale finite element problem (16). This process can be itemized as follows:

- A chosen RVE which represents the micro structure of the heterogeneous material, is discretised using the finite element method;
- Using the deformation gradient tensor \mathbf{F}_M , the micro-scale finite element problem is formulated by its weak form (16) and its boundary condition (17);
- The application of periodic boundary conditions on non periodic micro-structures follows the interpolation method [39];
- The constrained micro-scale finite element problem is iteratively solved following the multiple-constraint projection method set up in [40] and detailed in [36];
- The extraction of the meso-scale response includes
 1. The extraction of the homogenized first Piola-Kirchhoff stress tensor $\mathbf{P}_M = \frac{1}{V(\omega)} \sum_e \int_{\omega^e} \mathbf{P}_m d\mathbf{x}$; and
 2. The extraction of the fourth order macro-scale material tensor $\mathbb{C}_M = \frac{\partial \mathbf{P}_M}{\partial \mathbf{F}_M}$ required at the macro-scale in order to perform the Newton-Raphson iteration of the multi-scale analysis.

The computational homogenization method is thus seen as a finite element analysis which computes the meso-scale responses \mathbf{P}_M and \mathbb{C}_M of a certain RVE

under given \mathbf{F}_M . Since this iterative process needs to be carried out at each Gauss point of the macro-scale finite element, and has to be repeated for each macro-scale iteration, it is the most computationally expensive process of FE² analysis. In order to speed up the multi-scale analysis, a surrogate model is adopted to replace the costly homogenization computation.

3. Design and training of the Recurrent Neural Networks (RNNs) as a surrogate model

In the previous section, the virtual material law (7) was obtained through the computational homogenization of the meso-scale BVP as presented in Section 2.2. In this section we aim at substituting this finite-element resolution by a surrogate model. In that case, the computational homogenization of the meso-scale BVP is performed pre-off-line and the results are used to train the surrogate model.

In this work, in order to account for history-dependent behaviors, we consider as surrogate model a Recurrent Neural Networks (RNNs). RNNs have a “memory” under the form of hidden variables \mathbf{h} used to track the input \mathbf{u} history when predicting the output \mathbf{v} , see Fig. 3. These hidden variables can play a comparable role to the internal variables \mathbf{Z}_M in the analysis of an history-dependent physical processes. During a nonlinear finite-element analysis, the physical internal variables \mathbf{Z}_M defined in Eq. (7) are required to be stored at each Gauss point in order to record the loading history of the material points. In the context of a multi-scale simulation, the internal variables \mathbf{Z}_M at one Gauss point of the macro-scale discretisation actually consist in all the sets of internal variables \mathbf{Z}_m used at every Gauss points of the micro-structure associated to this macro-scale Gauss point. For a complex RVE, this leads to thousands of physical internal variables. However, the involved physical internal variables on a RVE might not be totally independent, which can be evidenced by the applications of order reduction methods when performing computational homogenization. In fact, NNWs are also used as a general algorithm for nonlinear order reduction in some applications. The RNN hidden variables naturally arise during the training and can be viewed as the nonlinear order reduction counterpart of the real physical internal variables of the RVE. These hidden variables thus play the role of the internal variables \mathbf{Z}_M in the analysis of an history-dependent physical process, and have to be stored at each Gauss point of the macro-scale finite element simulation. The set of internal variables \mathbf{Z}_M in Eq. (7) thus coincides to the hidden variables vector \mathbf{h} .

However, remembering information for long periods of time is not the default behavior of all kinds of RNNs: only special kinds of RNN are capable of learning long-term dependencies. Long Short Term Memory networks (LSTMs) are one of those kinds of specially designed RNNs and are now widely used in a large variety of problems. Another one is Gated Recurrent Unit (GRU), which is also considered as a variation of the LSTM because of their similar design.

In this section, we first present the GRU architecture which is used in this work before detailing the generation strategy of the data that will be used for

the training and testing of the RNN. Finally the training and testing of the RNN are particularized for batch training.

3.1. The structure of adopted recurrent neural network

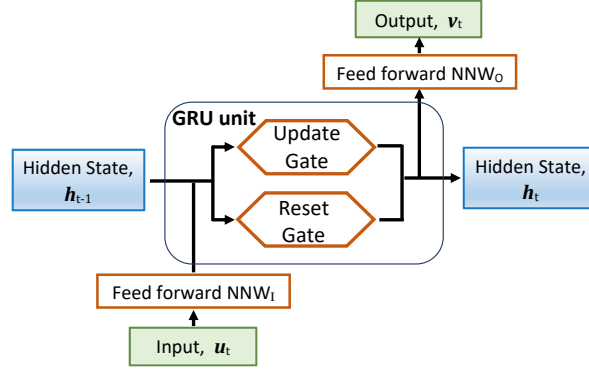


Figure 4: Recurrent Neural Network with a Gated Recurrent Unit.

In this study, GRU is chosen as surrogate for the direct finite element resolution of the RVEs. The typical functional character of GRU is illustrated by the block of a GRU unit in Fig. 4. In a model, previous hidden state information \mathbf{h}_{t-1} and current input \mathbf{u}_t need to pass through the reset and update gates, which are respectively used to decide how much of the past information can be forgotten and how much of the past information needs to be passed along to the future. More details on the GRU operations can be found in Appendix A. A basic GRU module (GRU unit) is already provided by the PyTorch library [41]. In this work, instead of using a single GRU unit, two feed-forward NNWs, see Fig. 2, are respectively added in the paths of input and output, denoted by NNW_I and NNW_O in Fig. 4. This gives more flexibility to the model than a single GRU unit, so that it can adapt to complex problems.

For the micro-structure adopted in this study, a 2D RVE under plane strain condition as it will be detailed in Section 4, the structures of the two feed-forward NNWs are respectively one hidden layer with “60” neurons for NNW_I , and two hidden layers with “100” neurons for NNW_O . The “Leaky ReLU” is chosen as the activation function in the feed-forward NNWs. The dimension of the hidden state \mathbf{h} is also “100”, meaning that the size of the internal variables \mathbf{Z}_M to be stored at each Gauss point of the macro-scale finite element discretisation is also 100.

The details of the input and output variables will be given in next section. The basic training operations of NNWs, such as computation and optimization of loss function and update of weights, follow the recommendations provided by PyTorch library [41] and are thus omitted in this paper.

3.2. Generation and preparation of data for the training and testing of the RNN

In Section 2.2, it appeared that the computational homogenization framework provides both the homogenized stress tensor \mathbf{P}_M and its derivative \mathbb{C}_M for a given homogenized deformation gradient \mathbf{F}_M . The surrogate model should naturally thus use as input \mathbf{F}_M and provides as output \mathbf{P}_M and \mathbb{C}_M .

We here detail how to reduce the dimensions of the input and output of the RNN, and how to generate the data required for its training and testing.

3.2.1. Input and output variables of the RNN

Since the resolution of the meso-scale BVP respects the frame indifference, the rigid rotation mode can be eliminated by using the Green-Lagrange strain tensor \mathbf{E}_M and the 2nd Piola-Kirchhoff stress tensor \mathbf{S}_M to describe the stress-strain relationship, instead of using directly \mathbf{F}_M and \mathbf{P}_M . Particularly for a 2D RVE under plane strain condition, we have thus reduced the input to $\mathbf{u} = \{E_{M_{XX}}, E_{M_{YY}}, E_{M_{XY}}\}$, and the output to $\mathbf{v} = \{S_{M_{XX}}, S_{M_{YY}}, S_{M_{ZZ}}, S_{M_{XY}}\}$, see Fig. 4 for the notations.

The required input of the RNN, *i.e.* \mathbf{E}_M , and of the macro-scale analysis, *i.e.* \mathbf{P}_M , are then obtained from the output of respectively the macro-scale analysis, *i.e.* \mathbf{F}_M , and of the RNN, *i.e.* \mathbf{S}_M , following

$$\mathbf{E}_M = \frac{1}{2} \left(\mathbf{F}_M^T \cdot \mathbf{F}_M - \mathbf{I} \right), \text{ and} \quad (18)$$

$$\mathbf{P}_M = \mathbf{F}_M \cdot \mathbf{S}_M. \quad (19)$$

The macro-scale material tensor \mathbb{C}_M required by the macro-scale simulation, can be deduced from the derivative $\frac{\partial \mathbf{S}_M}{\partial \mathbf{E}_M}$, which is computed either by the ‘‘Automatic Differentiation function’’ of the RNNs or by a perturbation method.

3.2.2. Data collection of the meso-scale BVP simulations

The training and testing data are provided by performing several computational homogenizations on the same RVE. Because of the frame indifference of the meso-scale BVP, \mathbf{E}_M is chosen to be the input of our RNN. However, \mathbf{E}_M can not be used directly to define the boundary condition (17) on the RVE. Therefore, the right stretch tensor \mathbf{U}_M is used as a connection between \mathbf{E}_M and \mathbf{F}_M . Indeed \mathbf{F}_M has a unique decomposition $\mathbf{R}_M \cdot \mathbf{U}_M$, where \mathbf{R}_M is a rotation tensor. For the purpose of training and testing, this rotation can be defined arbitrarily because of the frame indifference of the meso-scale BVP and, therefore, $\mathbf{R}_M = \mathbf{I}$.

The preparation of data for training and testing of the RNN is thus as follows

- Generate \mathbf{U}_M histories as detailed in Section 3.2.3;
- For a given \mathbf{U}_M , the meso-scale BVP is defined according to \mathbf{F}_M computed by

$$\mathbf{F}_M = \mathbf{R}_M \cdot \mathbf{U}_M, \quad (20)$$

and the RNN input according to

$$\mathbf{E}_M = \frac{1}{2} (\mathbf{U}_M^2 - \mathbf{I}) ; \quad (21)$$

- The resolution of the meso-scale BVP provides \mathbf{P}_M , which to be used to train and test the RNN is framed under \mathbf{S}_M using

$$\mathbf{S}_M = \mathbf{F}_M^{-1} \cdot \mathbf{P}_M . \quad (22)$$

3.2.3. Random loading path

Because the studied material system is loading history-dependent and because RNNs require sequential data for training, \mathbf{U}_M data need to be defined as a sequence. Considering the involved dimensions of \mathbf{U}_M and the large variety of possible loading/unloading paths that can locally arise during a multi-scale simulation, using proportional loading paths as a sequence is not an optimal choice.

In another context in which a feed-forward NNW was trained to predict monotonic response for different material parameters, random combinations of the input variables were used to create the training data in [25]; in this work, 500 random combinations of 6 material parameters were used for the NNW training. Compared to using the input values defined from a regular grids of high dimension data space, the random combination of input variables turned out to be more efficient. In this work we adopt a similar approach to generate sequential training data by considering series of random loading paths, which are like a random walking in a stochastic process.

A random loading path $\{\mathbf{U}_{M_t}\}$ is defined by a sequence of right stretch tensor, such as $\mathbf{U}_{M_0}, \mathbf{U}_{M_1}, \dots, \mathbf{U}_{M_N}$, where $\mathbf{U}_{M_0} = \mathbf{I}$ is the starting stage of the loading path. The loading increment, $\Delta\mathbf{U}_{M_n} = \mathbf{U}_{M_n} - \mathbf{U}_{M_{n-1}}$, is a random vector which permits the loading path to change the loading direction at each step, and thus to cover the maximum of non-proportional loading conditions within a limited number of steps.

As a symmetric second-order tensor, the increment of the right stretch tensor, $\Delta\mathbf{U}_{M_n}$, admits a spectral decomposition form expressed as,

$$\Delta\mathbf{U}_M = \Delta\lambda_1 \mathbf{n}_1 \otimes \mathbf{n}_1 + \Delta\lambda_2 \mathbf{n}_2 \otimes \mathbf{n}_2 + \Delta\lambda_3 \mathbf{n}_3 \otimes \mathbf{n}_3 , \quad (23)$$

where the eigenvectors \mathbf{n}_1 , \mathbf{n}_2 , and \mathbf{n}_3 control the direction of the loading path, and the eigenvalues $\Delta\lambda_1$, $\Delta\lambda_2$, and $\Delta\lambda_3$ control the increment size. Therefore, the generation of $\Delta\mathbf{U}_M$ requires the generation of a set of orthogonal vectors \mathbf{n}_1 , \mathbf{n}_2 , and \mathbf{n}_3 , and eigen-values $\Delta\lambda_1$, $\Delta\lambda_2$ and $\Delta\lambda_3$.

Considering the 2D case, \mathbf{n}_3 is fixed and the random orthogonal vectors \mathbf{n}_1 , \mathbf{n}_2 can be obtained by generating a uniformly distributed angular random variable with realization $\alpha \in [0, \pi)$, with

$$\mathbf{n}_1 = [\cos \alpha \ \sin \alpha]^T \quad \text{and} \quad \mathbf{n}_2 = [-\sin \alpha \ \cos \alpha]^T . \quad (24)$$

The three eigenvalues are generated in order to satisfy $\sqrt{\Delta\lambda_1^2 + \Delta\lambda_2^2 + \Delta\lambda_3^2} \leq \Delta R$, where ΔR is a defined upper bound of the step size. We note that sub-steps may be needed in order to guarantee the numerical convergence of micro-scale resolution in the case $\sqrt{\Delta\lambda_1^2 + \Delta\lambda_2^2 + \Delta\lambda_3^2}$ is too large. Still considering the 2D case, only $\Delta\lambda_1^2$ and $\Delta\lambda_2^2$ have to be generated, and it can easily be achieved by a random split of a random variable with realization $\mathcal{R} \in (0, \Delta R^2]$, or $\in (\Delta R_{\min}^2, \Delta R^2]$ to avoid too small steps to bring new information. Practically the random split is achieved by considering two independent random variables uniformly distributed with realizations $\mathcal{R} \in (\Delta R_{\min}^2, \Delta R^2]$ and $\theta \in [0, 2\pi)$, yielding

$$\Delta\lambda_1 = \sqrt{\mathcal{R}}\cos(\theta), \text{ and } \Delta\lambda_2 = \sqrt{\mathcal{R}}\sin(\theta). \quad (25)$$

Finally, the random walk or path is stopped when a criterion on the reached strain is met; *e.g.* when the eigen-values of \mathbf{U}_{MN} are such that $\sqrt{\sum_i (\lambda_i - 1)^2} > R_{\max}$, where R_{\max} is a given critical value.

In this work, $\Delta R = 5 \times 10^{-3}$ and $R_{\max} = 0.1$ are used to generate random loading paths. Besides thousands of random loading paths (each of them having tens, hundreds or thousands of computation points depending on the path), tens of proportional cyclic loading paths are also used to create training data. For the latter the loading direction and reversal time are also randomized. Indeed, some strain state histories might not be covered by random loading paths, and proportional loading paths can help the training data to reach a better coverage of the strain space.

3.3. Training with data in batches

Before training our NNW, all the input and output features must be standardized because the input/output features may not have the same scale, and the input of the activation functions might be out of their active range. For each feature, noted with χ , a simple linear operation is performed, which reads

$$\underline{\chi} = \frac{(\chi - \chi_m)}{\chi_s}, \quad (26)$$

with

$$\chi_m = \frac{(\chi_{\min} + \chi_{\max})}{2} \quad \text{and} \quad \chi_s = \frac{(\chi_{\max} - \chi_{\min})}{2}, \quad (27)$$

where χ_{\max} and χ_{\min} are the maximum and minimum values of this feature among all the available data used for training so that the features are mapped to the range $[-1, 1]$.

Since the training data sequences are obtained by a random walking process and since the random walking process will be terminated when a critical strain measure is reached, the length of the resulting data sequences are different for each path. While RNNs are typically able to take as inputs sequences of variable size, training data will usually be fed in batches to speed up the training process. In order to use batches to train from the data, it is needed to ensure that each sequence within the training data is of equal size. In most cases, padding is done by filling up sequences that are too short with 0 values and trimming sequences

that are too long. In our case, long sequences are trimmed from their end and for short sequences, both zero padding at their beginning and repeatedly adding the last element at their end are applied. Among the more than 9000 generated sequential data, 30% of them are used for training. The length of these data varies from 70 to 2000, with around half of the data having a length lower than 200 and more than 96% of them having a length lower than 1000. Therefore they were first all set to have a length of 200 for fast training, and then set to the length of 1000 for further training.

The developed RNN is characterized by hidden state variables that should be initialized for the first input of a sequence. Usually, hidden state variables are initialized to 0 for a wide variety of applications. However, after a few trial and errors of training, the value " - 1" is adopted to initialize the hidden state variables in our application.

4. Trained surrogate model of an elasto-plastic composite RVE

In this section, we apply the training strategy developed in Section 3 on an elasto-plastic composite micro-structure. The composite RVE is first described and then the accuracy of the trained RNN is assessed by comparing the results of the finite element simulations performed on the RVE with the RNN prediction for new loading paths, *i.e.* for loading conditions that were not used during the training process.

4.1. Description of the meso-scale BVP

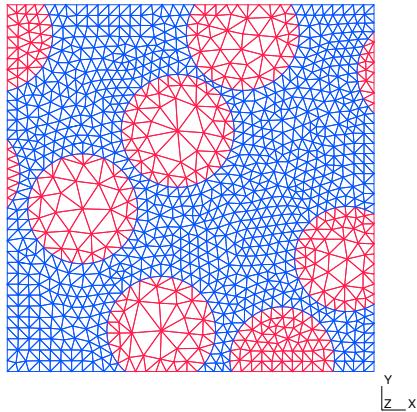


Figure 5: Finite element mesh of the microstructure volume element of dimension $0.02\text{mm}\times 0.02\text{mm}$ used to generate the data-base.

The micro-scale simulations are performed on a 2D RVE of a 39.9% continuous fiber reinforced elasto-plastic matrix material whose micro-structure is depicted in Fig. 5.

4.1.1. Fiber material model

The fibers obey a hyperelastic law based on an elastic potential

$$\psi_f(\mathbf{C}) = \frac{K_f}{2} \ln^2 J + \frac{\mu_f}{4} (\ln \mathbf{C})^{\text{dev}} : (\ln \mathbf{C})^{\text{dev}} , \quad (28)$$

where $J = \det \mathbf{F}$ is the Jacobian, $\mathbf{C} = \mathbf{F}^T \cdot \mathbf{F}$ is the right Cauchy strain tensor, $(\bullet)^{\text{dev}}$ denotes the deviatoric part of an arbitrary second-order tensor \bullet , K_f and μ_f correspond to respectively the bulk and shear moduli of the material. Then, the stress on the fiber phase is computed by

$$\mathbf{P} = \frac{\partial \psi_f(\mathbf{F})}{\partial \mathbf{F}} = K_f \mathbf{F}^{-T} \ln J + \mathbf{F}^{-T} \cdot [\mu_f \ln \mathbf{C}^{\text{dev}}] . \quad (29)$$

4.1.2. Matrix material model

The matrix obeys a finite strain J_2 elasto-plastic constitutive model [42]. The deformation gradient \mathbf{F} is decomposed into the reversible elastic part \mathbf{F}^e and the irreversible plastic part \mathbf{F}^p such that $\mathbf{F} = \mathbf{F}^e \cdot \mathbf{F}^p$. The elastic potential energy is defined as

$$\psi_m(\mathbf{C}^e) = \frac{K_m}{2} \ln^2 J + \frac{\mu_m}{4} (\ln \mathbf{C}^e)^{\text{dev}} : (\ln \mathbf{C}^e)^{\text{dev}} , \quad (30)$$

where $\mathbf{C}^e = \mathbf{F}^{eT} \cdot \mathbf{F}^e$, and K_m , and μ_m correspond to the bulk and shear moduli of the matrix material. The first Piola-Kirchhoff stress tensor \mathbf{P} derives from the elastic potential (30) following

$$\mathbf{P} = \frac{\partial \psi_m(\mathbf{F}; \mathbf{F}^p)}{\partial \mathbf{F}} = K_m \mathbf{F}^{-T} \ln J + \mu_m \mathbf{F}^e \cdot [\mathbf{C}^{e-1} \cdot (\ln \mathbf{C}^e)^{\text{dev}}] \cdot \mathbf{F}^{p-T} . \quad (31)$$

The elastic part \mathbf{F}^e and the plastic part \mathbf{F}^p of the deformation gradient are obtained through a J_2 plastic flow expressed in terms of the Kirchhoff stress. The Kirchhoff stress $\boldsymbol{\kappa} = \mathbf{P} \cdot \mathbf{F}^T$ is first computed by Eq. (31) as

$$\boldsymbol{\kappa} = K_m \ln J \mathbf{I} + \mu_m \mathbf{F}^e \cdot [\mathbf{C}^{e-1} \cdot (\ln \mathbf{C}^e)^{\text{dev}}] \cdot \mathbf{F}^{eT} . \quad (32)$$

The equivalent von Mises stress is then calculated through the deviatoric part of $\boldsymbol{\kappa}$, *i.e.* $\tau_{eq} = \sqrt{\frac{3}{2} \boldsymbol{\kappa}^{\text{dev}} : \boldsymbol{\kappa}^{\text{dev}}}$. According to the J_2 -plasticity theory, the von Mises stress criterion reads

$$f = \tau_{eq} - \tau_y^0 - R(\gamma) \leq 0 , \quad (33)$$

where f is the yield surface, τ_y^0 is the initial yield stress, γ is the equivalent plastic strain and where the isotropic hardening stress $R(\gamma)$ takes the form

$$R(\gamma) = Y [1 - \exp(-k\gamma)] , \quad (34)$$

with Y and k material constants. The evolution of \mathbf{F}^p is determined by the normal plastic flow theory following

$$\dot{\mathbf{F}}^p = \dot{\gamma} \mathbf{N} \cdot \mathbf{F}^p , \quad (35)$$

where \mathbf{N} is the normal to the yield surface, see [42] for more details.

Finally, the material properties used in this work are reported in Tab. 1.

Table 1: Material properties for fibers and matrix.

Fiber		Matrix				
K_f [GPa]	μ_f [GPa]	K_m [GPa]	μ_m [GPa]	τ_y^0 [MPa]	Y [MPa]	k [-]
16.67	12.50	2.50	1.15	100	20	30

4.2. Comparisons of RNN predictions to the direct finite element simulations

As previously said, around 9000 loading paths were used to generate the micro-scale finite simulations data-base, from which 30% were used to train the RNN. The accuracy of the trained RNN is first evaluated by the Mean Squared Error (MSE), which reads

$$L_{\text{MSE}} = \frac{1}{n} \sum_{i=1}^n (\underline{v}_i - \hat{\underline{v}}_i)^2, \quad (36)$$

where \underline{v}_i and $\hat{\underline{v}}_i$ are respectively the actual and the predicted normalized outputs. The number of loading paths used for the training is controlled by the decrease of the MSE. However, because of the limitation of the computer memory, and in order to remain efficient, we proceed by adding successive batches of limited number of training load paths and by performing warm-start training, *i.e.* performing a training using the previously trained parameter values and the training state variables of the optimization process, until the MSE reaches a value below the targeted threshold. A final value of $\text{MSE} \approx 4.3 \times 10^{-6}$ was obtained for the RNN with the testing data, *i.e.* the data that were not used for training, having a length of 200. The accuracy is about $\text{MSE} \approx 1.5 \times 10^{-5}$ for the testing data having a length of 1000.

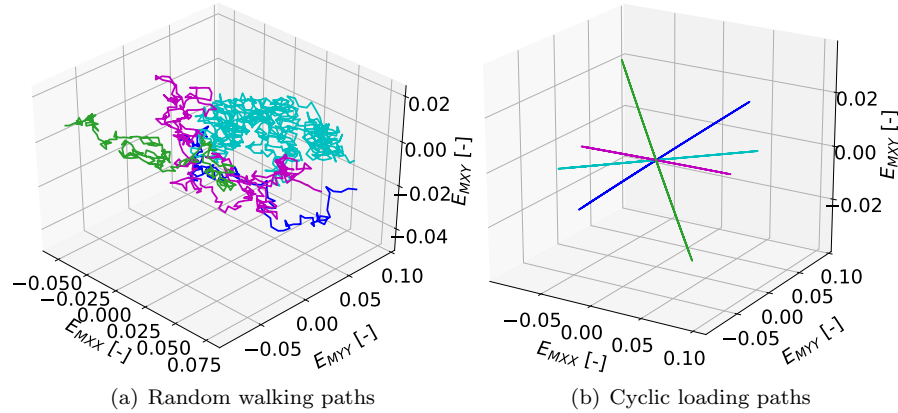


Figure 6: Illustration of the loading paths considered to test the accuracy of the RNN: (a) Four loading paths obtained using a random walking process; and (b) Four cyclic loading conditions of random orientation and reversal times.

The accuracy of the trained RNN is also assessed intuitively by comparing the RNN predictions to the direct finite element simulations, still for loading

cases which were not part of the training data. We successively consider as testing data, four random loading paths and four cyclic loading paths as illustrated in Fig. 6 in terms of the $\{E_{M_{XX}}, E_{M_{YY}}, E_{M_{XY}}\}$ -trajectories.

4.2.1. Testing on random loading paths

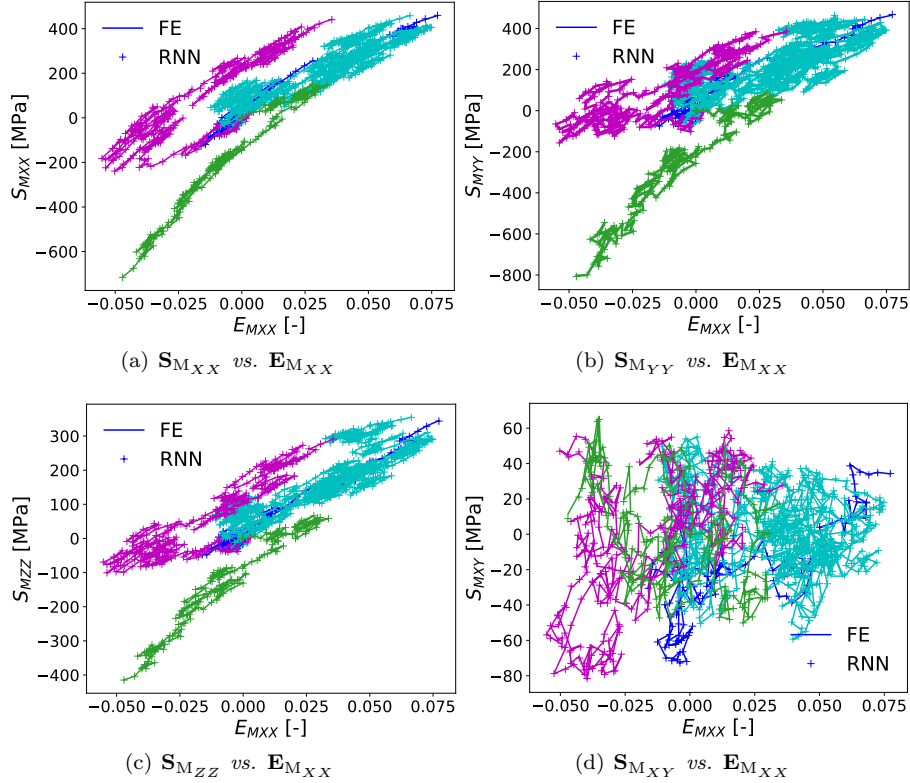


Figure 7: Comparison between the stress history predicted by the RNN and computed by the direct finite element analysis on the RVE for the random walking paths reported in Fig. 6(a).

The four random loading paths used for the testing are illustrated in Fig. 6(a). The corresponding stresses, $\{S_{M_{XX}}, S_{M_{YY}}, S_{M_{ZZ}}, S_{M_{XY}}\}$ obtained with both the finite element resolution of the meso-scale BVP and with the trained RNN are presented in Fig. 7 with the same color as their respective loading paths. A good agreement between the RNN predictions and the finite element results can be seen despite the complexity of the loading conditions.

4.2.2. Testing on cyclic loading paths

Cyclic random loading paths are also used to evaluate the accuracy of the trained RNN, by considering the four proportional loading cycles defined by $\{E_{M_{XX}}, E_{M_{YY}}, E_{M_{XY}}\}$ illustrated in Fig. 6(b). The comparison between the RNN predictions with the results obtained by direct finite element analyzes of

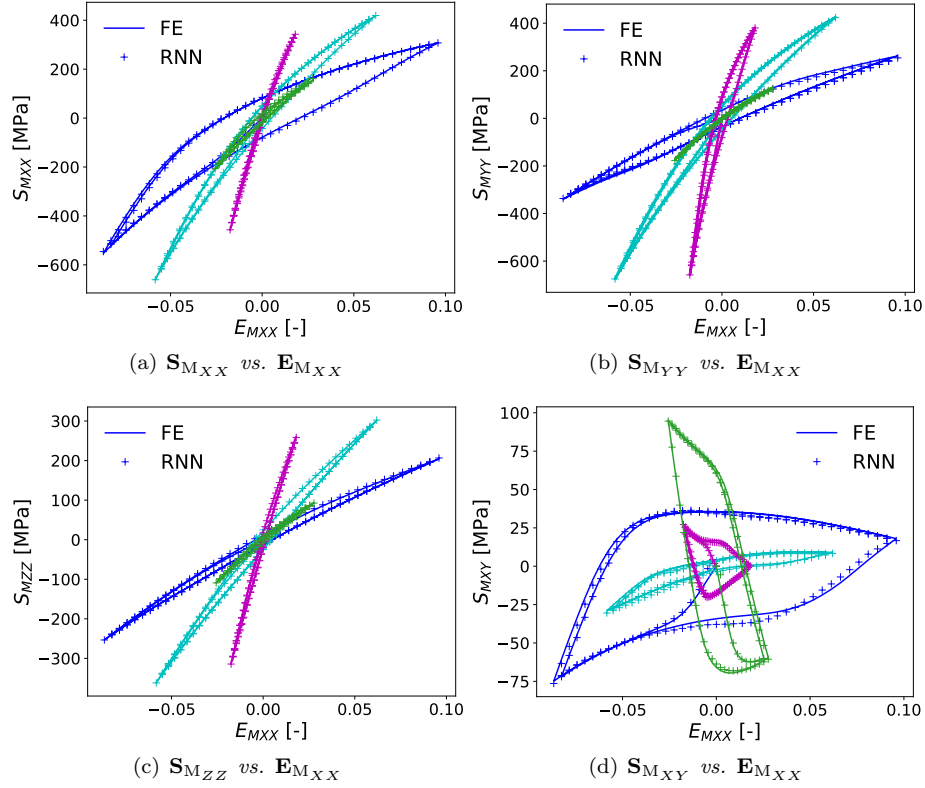


Figure 8: Comparison between the stress history predicted by the RNN and computed by the direct finite element analysis on the RVE for the random cyclic loading paths reported in Fig. 6(b).

the cyclically loaded is reported in Fig. 8 in terms of the $\{S_{M_{XX}}, S_{M_{YY}}, S_{M_{ZZ}}, S_{M_{XY}}\}$ evolution, and with the same color as their respective loading path description. These figures show that the trained RNN can predict the different cyclic responses with accuracy.

4.2.3. Discussion

Although good predictions are observed in Figs. 7 and 8, a few less accurate results are also presented in Fig. 9. For two selected loading paths, we can see that the stresses $S_{M_{XX}}$, $S_{M_{YY}}$ and $S_{M_{ZZ}}$ are well predicted by RNN, see Figs 9(a)-9(c). However only the general trend of the shear stress $S_{M_{XY}}$ is captured and the relative error of the RNN prediction is high, see Fig. 9(d).

This poor prediction only happens at low shear stress state, see the stress scale in Fig. 9(d) and is due to the loss function adopted during the training of our RNN. Indeed, the training is a process that identifies the parameters of the NNWs through a minimization of the defined loss function. The MSE, Eq. (36), is used as a loss function in this work. Since it is measured as the average

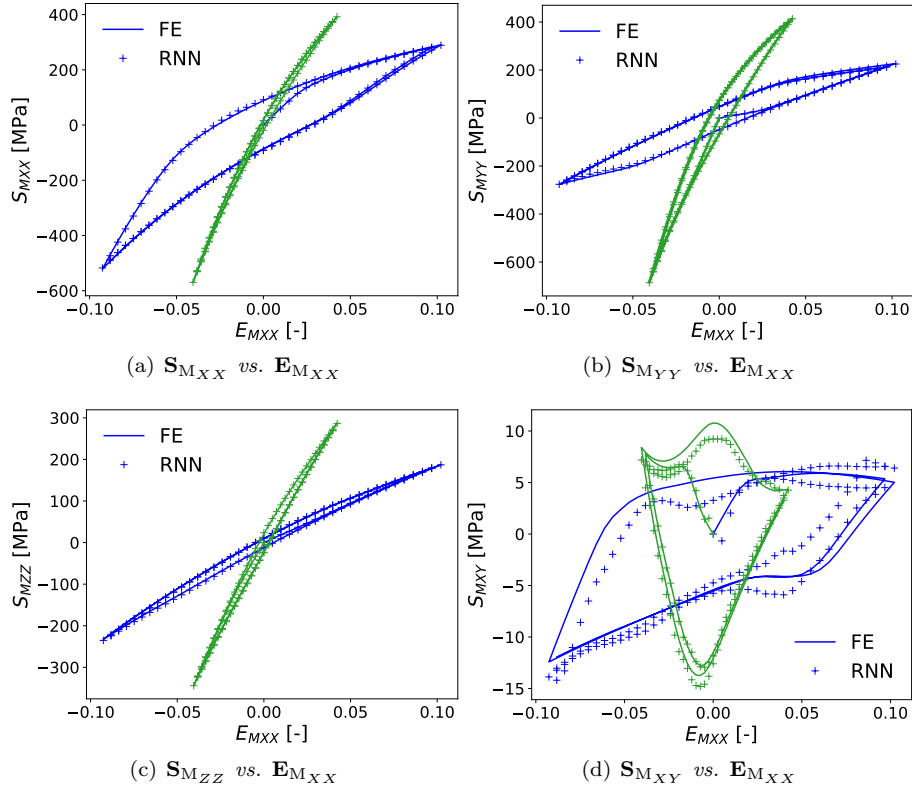


Figure 9: Comparison between the stress history predicted by the RNN and computed by the direct finite element analysis on the RVE for loading paths resulting in lower shear stress $S_{M_{XY}}$.

of the squared difference between the normalized predictions \hat{v}_i and the actual normalized values v_i , only the absolute error is minimized during the training. So when a batch of data is used in NNW training, the data with high stress state dominate the optimization process. This issue can be improved by adding weight in loss function, with a higher weight considered for low stress states or by using relative error when defining the loss function. However, because the inaccuracy only appears for stress component more than one order of magnitude lower than the governing response, we keep this trained RNN for the multi-scale simulation.

5. Multi-scale analysis comparison on open hole sample

In this section we introduce the RNN as a surrogate of the material law (7) so that macro-scale analyzes can be conducted. We then study an open hole sample subjected to several loading/unloading cycles and compare the response obtained with a fully coupled FE² analysis with the one obtained using the

surrogate model. In order to discuss on the accuracy of the surrogate model and to show its limitations, we consider cycles within the range of the training data, but also out of this range. Finally, the computational cost breakdown is analyzed for both approaches.

5.1. Implementation of RNN surrogate model in multi-scale analyzes

The implementation of the RNN as a surrogate model of the material law (7) follows the traditional implementation of material laws in a finite element code. The hidden variables \mathbf{h} of the RNN are saved as internal variables at each Gauss point, while the trained RNN is saved at the material law level. In particular, since our RNN is coded and trained in Python with the library PyTorch [41], it is stored as a script module which can be loaded either from Python or from a compiled C++ code. In this work this last option is selected in order to ensure the computational efficiency. The macro-scale material tensor \mathbb{C}_M , which is defined by $\frac{\partial \mathbf{P}_M}{\partial \mathbf{F}_M}$, can be computed either from “Automatic Differentiation function” of NNWs –in which case it is actually $\frac{\partial \mathbf{S}_M}{\partial \mathbf{E}_M}$ which is evaluated and which is transformed to get \mathbb{C}_M – or by a perturbation method. In our case, the two methods yield the same simulation results, however, it turns out that using perturbation is more efficient than using the “Automatic Differentiation function”.

5.2. Numerical application on an open hole sample

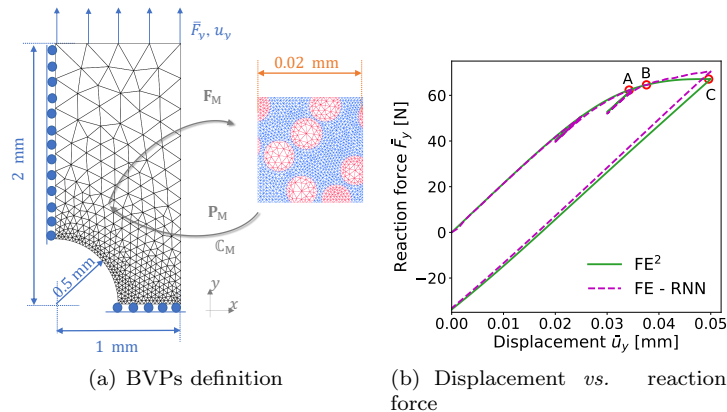


Figure 10: Multi-scale simulation: (a) Definition of the macro-scale and meso-scale BVP; and (b) Displacement *vs.* reaction force curves obtained by the FE² and the FE-RNN analyzes.

In this section we consider, at the macro-scale, the open hole sample illustrated in Fig. 10(a), while the meso-scale BVP is defined by the RVE described in Section 4.1. The open hole sample is subjected to the boundary condition illustrated in Fig. 10(a): the sample is loaded on its top edge under controlled

displacement, which increases from 0 to 0.05 mm with two partial unloading-reloading cycles in the middle and a final decrease to 0. Because of the geometrical and loading symmetry, only one quarter of an open hole sample is considered for the simulations.

As a reference, a concurrent multi-scale analysis, denoted by FE^2 , is conducted in which the meso-scale BVP is solved using the finite element method in a concurrent way with the macro-scale BVP. The resulting reaction force to displacement curve is presented in Fig. 10(b). The response predicted by the RNN as surrogate model is also reported in Fig. 10(b). The overlap of two displacement-reaction force curves shows that a good accuracy is obtained with the RNN surrogate model including up to point “B” after the second unloading-reloading cycle. We will show later that point “B” corresponds to the range limit of the RNN training. However, although the range limit of the training is reached and a slight divergence of the two curves can be seen after the loading point “B” in Fig. 10(b), the elastic unloading and reverse compression responses are still well reproduced with the RNN surrogate model.

Figure 11 compares the strain and stress distributions obtained using the RNN surrogate model and the FE^2 simulations. The Green-Lagrange strain component $E_{M_{YY}}$ and the Cauchy stress component $\sigma_{M_{YY}}$ deduced from the second Piola Kirchhoff stress tensor following

$$\boldsymbol{\sigma}_M = J_M^{-1} \mathbf{F}_M \cdot \mathbf{S}_M \cdot \mathbf{F}_M^T, \quad (37)$$

where $J_M = \det(\mathbf{F}_M)$, are used for visualization, on the one hand because the Green-Lagrange strain directly gives an indication on the the strain state with respect to the training range, and on the other hand because the Cauchy stress tensor is easier to interpret than the second Piola Kirchhoff stress tensor. Figures 11(a) and 11(b) illustrate the strain and stress distributions in the sample at loading point “A” referenced in Fig. 10(b) just before the second unloading. In these figures, it can be seen that the RNN surrogate model predicts strain and stress distributions in good agreement with the ones of the FE^2 simulation. At loading point “B”, as denoted in Fig. 10(b) after the second loading/unloading cycle, although the two displacement-reaction force curves overlap and although the stress distributions obtained by the two methods are similar, see Fig. 11(c), the maximum strain, $E_{M_{YY}}$, obtained by RNN surrogate model is slightly underestimated compared to the FE^2 simulation, see Fig. 11(c). Finally, at loading point “C” referred on Fig. 10(b) at maximum loading, the reaction force is over-predicted and the maximum strain $E_{M_{YY}}$ predicted by the RNN surrogate model is much lower than the one computed by the FE^2 simulation. The location where the stress $\sigma_{M_{YY}}$ reaches its maximum value is not really accurate either, see Figs. 11(e) and 11(f).

The observed inaccuracy of the RNN surrogate model at point “C” is explained by investigating the range of data used during training. As a reminder $R_{\max} = 0.1$ was used as the termination criterion for generating the random loading paths in Section 3.2.3, and we here below extract the maximum and minimum values of input features from all the obtained loading paths used for

training the RNN, yielding

$$\begin{aligned} E_{M_{XX}} &\in [-0.0984, 0.1088], \\ E_{M_{YY}} &\in [-0.0985, 0.1091], \quad \text{and} \\ E_{M_{XY}} &\in [-0.0743, 0.0734]. \end{aligned} \tag{38}$$

When considering the strain distribution at point ‘‘A’’, Fig. 11(a), it appears that we remain in this range, while the strain at point ‘‘B’’ is close to its boundary, see the maximum strain, $E_{M_{YY}}$, in Fig. 11(c) emphasized by green circle. This explains why the predictions are slightly less accurate at point ‘‘B’’. Furthermore, at loading point ‘‘C’’, the maximum value of the strain $E_{M_{YY}}$ reached with FE^2 simulation is out of the range of the training data, see the strain distributions in Fig. 11(e) emphasized respectively by green and red circles. This explains the poor prediction observed in Fig. 10(b) at this loading point ‘‘C’’, which results from an inaccurate stress prediction by RNN, as illustrated in Fig. 11(f) and emphasized by the green and red circles. It is known that NNWs have a weak capability of extrapolation, so it is not surprising that the NNWs prediction is inaccurate when the inputs are out of the range of their training data. We note that a strategy could be developed to shift to another (Reduced Order) model when the strain state get out of the training range as in [29].

At the end, we can state that RNNs can be used as an efficient and accurate surrogate for meso-scale BVP in computational multi-scale analyzes, if they are trained with adequate data whose range is wide enough to make sure that all the possible input will fall in it.

5.3. Cost comparison

Table 2: Computational cost breakdown.

Table 2: Computational cost breakdown.		
Pre-off-line	FE^2	FE-RNN
Data generation	-	9000 \times 2 hour-cpu
Training	-	3 day-cpu
On-line	FE^2	FE-RNN
Multi-scale simulation	18000 hour-cpu	0.5 hour-cpu

The computational cost breakdown of the two simulation methodologies is reported in Table 2.

The computation time of the coupled FE^2 multi-scale simulation took around 30 hours using 600 processors on a cluster. In comparison the analysis with the RNN surrogate model required around one half hour on a single Core i5-Intel processor, yielding a tremendous computational resource decrease.

Of course the RNN-FEM model required some time to be set up. Generating the data for the 9000 loading paths required around 2 hour per path generation. Since this step is fully scalable, this was achieved in around 2 days using 350 processors. The training of the RNN using 30% of the data was a time-consuming step of about 3 days on a single Core i5-Intel processor. Nevertheless, once the RNN is trained it can be used to conduct other simulation at no extra-cost.

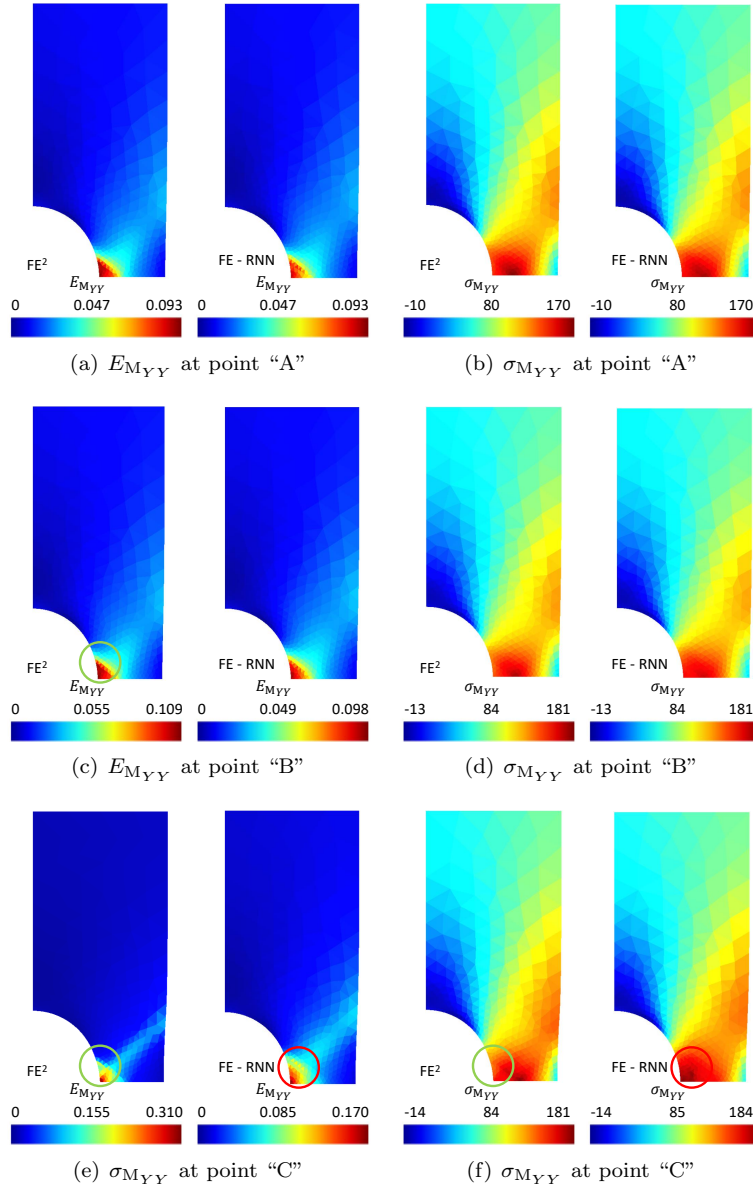


Figure 11: Comparisons of the Green-Lagrangre strain and Cauchy stress distributions of an open-hole sample obtained by the finite element simulation using the RNN surrogate model and by the FE^2 method; The loading points "A" to "C" refer to the markers on Fig. 10(b).

6. Conclusions

In this work, we designed Recurrent Neural Network (RNN) based on a Gated Recurrent Unit, to serve as a surrogate of the meso-scale BVP in the

context of computational multi-scale analyzes. With this architecture, internal variables required to account for history-dependent behaviors are implicitly represented since the RNN is self-equipped with hidden variables that have the ability of tracing loading history once the RNN is trained using sequential data. This feature of RNNs makes it very attractive in mechanical applications, in which historical dependence is commonly met for irreversible material behaviors.

In order to achieve accuracy under multi-dimensional non-proportional loading conditions, we have generated training data from a meso-scale composite RVE subjected to random loading paths. The random loading paths were generated in a way similar to a random walking in stochastic process so that a maximum of stress-strain state evolutions are used for the training of the RNNs. We have shown on the testing data that the trained RNN can predict both random non-proportional stress-strain evolutions and random cyclic loading paths.

When it comes to structural analyzes, a similar accuracy as with a FE^2 multi-scale simulation can be reached with the RNN-based surrogate model as long as the loading case remains in the training range. This has been exemplified by considering an open hole sample subjected to several loading/unloading steps, in which it was shown that the two approaches predict similar results as long as the local strain state remains in the training range, while a discrepancy in the global response is observed when the local strain get out of this range. We note that because of the training strategy, the range of training data is expressed as bounds on the different strain components. When applied on structural analyzes, the RNN-based surrogate model allows reducing the computational time by four order of magnitude as compared to FE^2 multi-scale simulations. Although the generation of data and the training stage also require important computational resources, once trained the RNN-surrogate model can be used for several simulation cases.

Although the presented methodology is exemplified with a 2D micro-structure, the 3D case is not expected to involve more difficulties other than generating larger data and requiring more training time. In this work, only stress-strain responses are considered. However, any desired physical variables, internal variables, such as plasticity, energy, and or damage *etc.* can also be extracted from the resolution of meso-scale BVP and included in the output of the RNN so that this micro-scale information remains accessible for interpretation purpose.

Acknowledgment

This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 862015 for the project “Multi-scale Optimisation for Additive Manufacturing of fatigue resistant shock-absorbing MetaMaterials (MOAMMM) of the H2020-EU.1.2.1. - FET Open Programme.

N.G. Kilingar was financed by the EnlightenIt project, grant number PDR T.0038.16 of FRS-FNRS.

The authors would like to thank Prof. Angelo Simone, Gilles Louppe, Pierre Geurts and Dr. Fariborz Ghavamian for the discussions on the RNN.

Data availability

The raw/processed data required to reproduce these findings is available on https://gitlab.uliege.be/moammm/moammmpublic/tree/master/publicationsData/2020_CMAME_RNN under the Creative Commons Attribution 4.0 International (CC BY 4.0) licence [43].

Appendix A. Details on inner-working of the GRU

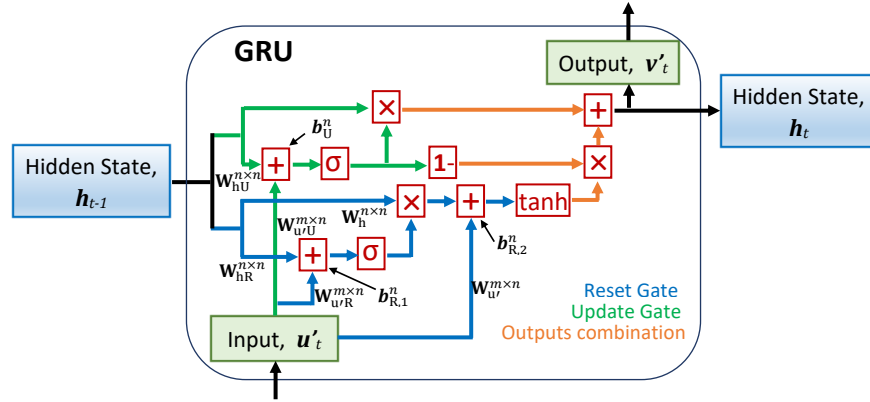


Figure A.12: Detailed GRU architecture; the trainable model parameters are the weights matrices $\mathbf{W}_{u'U}^{m \times n}$, $\mathbf{W}_{u'R}^{m \times n}$, $\mathbf{W}_u^{m \times n}$, $\mathbf{W}_{hU}^{n \times n}$, $\mathbf{W}_{hR}^{n \times n}$, $\mathbf{W}_h^{n \times n}$ and bias vectors \mathbf{b}_U^n , $\mathbf{b}_{R,1}^n$ and $\mathbf{b}_{R,2}^n$, where the superscripts m and n refer respectively to the dimensions of the input vector \mathbf{u}' and of the hidden variables vector \mathbf{h} , the output vector \mathbf{v}' having also the dimension n , and the subscripts u' , h , R , U refer respectively to the input variables, hidden variables, and constructed Reset vector and Update vector.

The structure of the GRU presented in Fig. 4 is detailed in Fig. A.12. The different operation symbols used in Fig. A.12 are

- $\boxed{+}$: the element-wise sum operator on two vectors of same dimension, which can be expressed as $\mathbf{r} = \mathbf{x} + \mathbf{y}$;
- $\boxed{1-}$: The element-wise operator on vector \mathbf{x} , which reads $\mathbf{r} = \mathbf{1} - \mathbf{x}$;
- $\boxed{\times}$: The element-wise multiplication, or Hadamard product, on two vectors, \mathbf{x} and \mathbf{y} , of same dimension, which reads $r_i = x_i y_i$.
- $\boxed{\sigma}$: The non-linear activation sigmoid function, which reads,

$$\sigma(x) = \frac{1}{1 + \exp(-x)}, \quad (\text{A.1})$$

and returns values in the range 0 to 1;

- $\boxed{\tanh}$: The non-linear activation hyperbolic tangent function.

The structure of the GRU presented in Fig. A.12 can be broken down into three main parts: the reset gate (Blue path), the update gate (Green path) and the combination of the outputs (Orange path). The three paths are now summarized:

- **Reset gate** (Blue path): uses the previous hidden state \mathbf{h}_{t-1} and current input data \mathbf{u}'_t , which in our case is the output of the input feed forward NNW_I , see Fig. 4. The weighted sum vector passes through a sigmoid function, which transforms the values to make them fall between 0 and 1 in order to obtain a Reset vector whose aim is to filter the less-important and more-important information for the subsequent steps. The previous hidden state is first multiplied by a trainable weight and then undergoes an element-wise multiplication with the Reset vector. This operation decides which information is to be kept from the previous configuration together with the new inputs. Then a non-linear activation $\boxed{\tanh}$ function is applied on the weighted sum of the current input and the product of the Reset vector and previous hidden state.
- **Update gate** (Green path): also uses the previous hidden state \mathbf{h}_{t-1} and the current input data \mathbf{u}'_t and performs similar operations as in the reset gate, but with different weight matrices. The Update vector then undergoes an element-wise multiplication with the previous hidden state (unweighted \mathbf{h}_{t-1}). The update gate helps the model to determine how much of the past information stored in the previous hidden state needs to be retained for the future.
- **Outputs combination** (Orange path): consists in an element-wise $\boxed{1-}$ operation applied on the Update vector followed by an element-wise multiplication with the output from the reset gate (unweighted). The purpose of this operation is for the update gate to determine what portion of the new information should be stored in the hidden state. Lastly, the result from the above operations is summed (unweighted) with the output from the update gate element-wise multiplied with the previous hidden state (unweighted) in order to provide the updated hidden state vector, the output vector being the same as the latter.

All the weight matrices \mathbf{W} and bias vectors \mathbf{b} used in Fig. A.12 are updated when the entire network is trained through back-propagation.

- [1] P. Kanouté, D. Boso, J. Chaboche, B. Schrefler, Multiscale Methods for Composites: A Review, Archives of Computational Methods in Engineering 16 (2009) 31–75, ISSN 1134-3060, 10.1007/s11831-008-9028-8.
- [2] L. Noels, L. Wu, L. Adam, J. Seyfarth, G. Soni, J. Segurado, G. Laschet, G. Chen, M. Lesueur, M. Lobos, T. Böhlke, T. Reiter, S. Oberpeilsteiner, D. Salaberger, D. Weichert, C. Broeckmann, Effective Properties, in: Handbook of Software Solutions for ICME, Wiley-VCH Verlag GmbH & Co.

- KGaA, ISBN 9783527693566, 433–485, doi:10.1002/9783527693566.ch6, URL <http://dx.doi.org/10.1002/9783527693566.ch6>, 2016.
- [3] J. Yvonnet, Solid Mechanics and Its Applications, in: Computational Homogenization of Heterogeneous Materials with Finite Elements, Springer International Publishing, ISBN 978-3-030-18382-0, 2019.
- [4] I. Özdemir, W. A. M. Brekelmans, M. G. D. Geers, Computational homogenization for heat conduction in heterogeneous solids, *International Journal for Numerical Methods in Engineering* 73 (2) (2008) 185–204, ISSN 1097-0207, doi:10.1002/nme.2068, URL <http://dx.doi.org/10.1002/nme.2068>.
- [5] J. Schröder, M.-A. Keip, A Framework for the Two-Scale Homogenization of Electro-Mechanically Coupled Boundary Value Problems, in: M. Kuczma, K. Wilmski (Eds.), *Computer Methods in Mechanics*, vol. 1 of *Advanced Structured Materials*, Springer Berlin Heidelberg, ISBN 978-3-642-05240-8, 311–329, 2010.
- [6] C. Miehe, D. Vallicotti, S. Teichtmeister, Homogenization and multiscale stability analysis in finite magneto-electro-elasticity, *GAMM-Mitteilungen* 38 (2) (2015) 313–343, ISSN 1522-2608, doi:10.1002/gamm.201510017, URL <http://dx.doi.org/10.1002/gamm.201510017>.
- [7] F. Feyel, Multiscale FE2 elastoviscoplastic analysis of composite structures, *Computational Materials Science* 16 (1) (1999) 344 – 354, ISSN 0927-0256, doi:[http://dx.doi.org/10.1016/S0927-0256\(99\)00077-4](http://dx.doi.org/10.1016/S0927-0256(99)00077-4), URL <http://www.sciencedirect.com/science/article/pii/S0927025699000774>.
- [8] C. Miehe, J. Schotte, J. Schröder, Computational micro-macro transitions and overall moduli in the analysis of polycrystals at large strains, *Computational Materials Science* 16 (1-4) (1999) 372–382, cited By 88.
- [9] K. Terada, M. Hori, T. Kyoya, N. Kikuchi, Simulation of the multi-scale convergence in computational homogenization approaches, *International Journal of Solids and Structures* 37 (16) (2000) 2285 – 2311, ISSN 0020-7683, doi:[http://dx.doi.org/10.1016/S0020-7683\(98\)00341-2](http://dx.doi.org/10.1016/S0020-7683(98)00341-2), URL <http://www.sciencedirect.com/science/article/pii/S0020768398003412>.
- [10] V. Kouznetsova, W. Brekelmans, F. Baaijens, An approach to micro-macro modeling of heterogeneous materials, *Computational mechanics* 27 (1) (2001) 37–48.
- [11] J. Yvonnet, Q.-C. He, The reduced model multiscale method (R3M) for the non-linear homogenization of hyperelastic media at finite strains, *Journal of Computational Physics* 223 (1) (2007) 341 – 368, ISSN 0021-9991, doi:<https://doi.org/10.1016/j.jcp.2006.09.019>, URL <http://www.sciencedirect.com/science/article/pii/S0021999106004402>.

- [12] J. Hernández, J. Oliver, A. Huespe, M. Caicedo, J. Cante, High-performance model reduction techniques in computational multiscale homogenization, *Computer Methods in Applied Mechanics and Engineering* 276 (2014) 149 – 189, ISSN 0045-7825, doi:<https://doi.org/10.1016/j.cma.2014.03.011>, URL <http://www.sciencedirect.com/science/article/pii/S0045782514000978>.
- [13] D. Soldner, B. Brands, R. Zabihyan, P. Steinmann, J. Mergheim, A numerical study of different projection-based model reduction techniques applied to computational homogenisation, *Computational mechanics* 60 (4) (2017) 613–625.
- [14] D. Wirtz, N. Karajan, B. Haasdonk, Surrogate modeling of multiscale models using kernel methods, *International Journal for Numerical Methods in Engineering* 101 (1) (2015) 1–28, ISSN 1097-0207, doi:10.1002/nme.4767, URL <http://dx.doi.org/10.1002/nme.4767>.
- [15] I. Rocha, P. Kerfriden, F. P. van der Meer, Micromechanics-based surrogate models for the response of composites: A critical comparison between a classical mesoscale constitutive model, hyper-reduction and neural networks, *European Journal of Mechanics - A/Solids* 82 (2020) 103995, ISSN 0997-7538, doi:10.1016/j.euromechsol.2020.103995.
- [16] T. Furukawa, G. Yagawa, Implicit constitutive modelling for viscoplasticity using neural networks, *International Journal for Numerical Methods in Engineering* 43 (2) (1998) 195–219.
- [17] T. Furukawa, M. Hoffman, Accurate cyclic plastic analysis using a neural network material model, *Engineering Analysis with Boundary Elements* 28 (3) (2004) 195 – 204, ISSN 0955-7997, doi:[https://doi.org/10.1016/S0955-7997\(03\)00050-X](https://doi.org/10.1016/S0955-7997(03)00050-X), URL <http://www.sciencedirect.com/science/article/pii/S095579970300050X>, inverse Problems.
- [18] K. Wang, W. Sun, Meta-modeling game for deriving theory-consistent, microstructure-based tractionseparation laws via deep reinforcement learning, *Computer Methods in Applied Mechanics and Engineering* 346 (2019) 216 – 241, ISSN 0045-7825, doi:<https://doi.org/10.1016/j.cma.2018.11.026>, URL <http://www.sciencedirect.com/science/article/pii/S0045782518305851>.
- [19] M. Fernández, S. Rezaei, J. R. Mianroodi, F. Fritzen, S. Reese, Application of artificial neural networks for the prediction of interface mechanics: a study on grain boundary constitutive behavior, *Advanced Modeling and Simulation in Engineering Sciences* 7 (1) (2020) 1–27.
- [20] M. Lefik, B. Schrefler, Artificial neural network as an incremental non-linear constitutive model for a finite element code, *Computer Methods in Applied Mechanics and Engineering* 192 (28) (2003) 3265 – 3283,

ISSN 0045-7825, doi:[https://doi.org/10.1016/S0045-7825\(03\)00350-5](https://doi.org/10.1016/S0045-7825(03)00350-5), URL <http://www.sciencedirect.com/science/article/pii/S0045782503003505>, multiscale Computational Mechanics for Materials and Structures.

- [21] Y. M. A. Hashash, S. Jung, J. Ghaboussi, Numerical implementation of a neural network based material model in finite element analysis, *International Journal for Numerical Methods in Engineering* 59 (7) (2004) 989–1005, doi:10.1002/nme.905, URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.905>.
- [22] A. Zhang, D. Mohr, Using neural networks to represent von Mises plasticity with isotropic hardening, *International Journal of Plasticity* (2020) 102732, ISSN 0749-6419, doi:<https://doi.org/10.1016/j.ijplas.2020.102732>.
- [23] S. Jung, J. Ghaboussi, Characterizing rate-dependent material behaviors in self-learning simulation, *Computer Methods in Applied Mechanics and Engineering* 196 (1) (2006) 608 – 619, ISSN 0045-7825, doi:<https://doi.org/10.1016/j.cma.2006.06.006>, URL <http://www.sciencedirect.com/science/article/pii/S0045782506001940>.
- [24] M. Lefik, B. Schrefler, Artificial neural network for parameter identifications for an elasto-plastic model of superconducting cable under cyclic loading, *Computers & Structures* 80 (22) (2002) 1699 – 1713, ISSN 0045-7949, doi:[https://doi.org/10.1016/S0045-7949\(02\)00162-1](https://doi.org/10.1016/S0045-7949(02)00162-1), URL <http://www.sciencedirect.com/science/article/pii/S0045794902001621>.
- [25] L. Wu, K. Zulueta, Z. Major, A. Arriaga, L. Noels, Bayesian inference of non-linear multiscale model parameters accelerated by a Deep Neural Network, *Computer Methods in Applied Mechanics and Engineering* 360 (2020) 112693, ISSN 0045-7825, doi:<https://doi.org/10.1016/j.cma.2019.112693>, URL <http://www.sciencedirect.com/science/article/pii/S004578251930578X>.
- [26] R. Hambli, H. Katerchi, C.-L. Benhamou, Multiscale methodology for bone remodelling simulation using coupled finite element and neural network computation, *Biomechanics and modeling in mechanobiology* 10 (1) (2011) 133–145.
- [27] B. A. Le, J. Yvonnet, Q. C. He, Computational homogenization of nonlinear elastic materials using neural networks, *International Journal for Numerical Methods in Engineering* 104 (12) (2015) 1061–1084, doi:10.1002/nme.4953.
- [28] M. Bessa, R. Bostanabad, Z. Liu, A. Hu, D. W. Apley, C. Brinson, W. Chen, W. Liu, A framework for data-driven analysis of materials under uncertainty: Countering the curse of dimensionality, *Computer Methods in Applied Mechanics and Engineering* 320 (2017) 633 – 667, ISSN 0045-7825, doi:<https://doi.org/10.1016/j.cma.2017.03.037>, URL <http://www.sciencedirect.com/science/article/pii/S0045782516314803>.

- [29] F. Fritzen, M. Fernandez, F. Larsson, On-the-Fly Adaptivity for Nonlinear Twoscale Simulations Using Artificial Neural Networks and Reduced Order Modeling, *Frontiers in Materials* 6 (2019) 75, ISSN 2296-8016, doi:10.3389/fmats.2019.00075, URL <https://www.frontiersin.org/article/10.3389/fmats.2019.00075>.
- [30] J. F. Unger, C. Könke, Coupling of scales in a multiscale simulation using neural networks, *Computers & Structures* 86 (21) (2008) 1994 – 2003, ISSN 0045-7949, doi:<https://doi.org/10.1016/j.compstruc.2008.05.004>, URL <http://www.sciencedirect.com/science/article/pii/S0045794908001430>.
- [31] C. Settgest, G. Htter, M. Kuna, M. Abendroth, A hybrid approach to simulate the homogenized irreversible elastic-plastic deformations and damage of foams by neural networks, *International Journal of Plasticity* 126 (2020) 102624, ISSN 0749-6419, doi:<https://doi.org/10.1016/j.ijplas.2019.11.003>, URL <http://www.sciencedirect.com/science/article/pii/S074964191930381X>.
- [32] F. Ghavamian, A. Simone, Accelerating multiscale finite element simulations of history-dependent materials using a recurrent neural network, *Computer Methods in Applied Mechanics and Engineering* 357 (2019) 112594, ISSN 0045-7825, doi:<https://doi.org/10.1016/j.cma.2019.112594>, URL <http://www.sciencedirect.com/science/article/pii/S0045782519304700>.
- [33] A. Koepppe, F. Bamer, B. Markert, An intelligent nonlinear meta element for elastoplastic continua: deep learning using a new Time-distributed Residual U-Net architecture, *Computer Methods in Applied Mechanics and Engineering* 366 (2020) 113088, ISSN 0045-7825, doi: <https://doi.org/10.1016/j.cma.2020.113088>.
- [34] M. Mozaffar, R. Bostanabad, W. Chen, K. Ehmann, J. Cao, M. A. Bessa, Deep learning predicts path-dependent plasticity, *Proceedings of the National Academy of Sciences* 116 (52) (2019) 26414–26420, ISSN 0027-8424, doi:10.1073/pnas.1911815116.
- [35] M. B. Gorji, M. Mozaffar, J. N. Heidenreich, J. Cao, D. Mohr, On the potential of recurrent neural networks for modeling path dependent plasticity, *Journal of the Mechanics and Physics of Solids* 143 (2020) 103972, ISSN 0022-5096, doi:<https://doi.org/10.1016/j.jmps.2020.103972>.
- [36] V.-D. Nguyen, L. Wu, L. Noels, Unified treatment of microscopic boundary conditions and efficient algorithms for estimating tangent operators of the homogenized behavior in the computational homogenization method, *Computational Mechanics* 59 (3) (2017) 483–505, ISSN 1432-0924, doi:10.1007/s00466-016-1358-z, URL <https://doi.org/10.1007/s00466-016-1358-z>.

- [37] D. Peric, E. A. de Souza Neto, R. A. Feijóo, M. Partovi, A. J. C. Molina, On micro-to-macro transitions for multi-scale analysis of non-linear heterogeneous materials: unified variational basis and finite element implementation, *International Journal for Numerical Methods in Engineering* 87 (2010) 149 – 170, ISSN 1097-0207, URL <http://dx.doi.org/10.1002/nme.3014>.
- [38] J. Schröder, M. Labusch, M.-A. Keip, Algorithmic two-scale transition for magneto-electro-mechanically coupled problems: FE2-scheme: Localization and homogenization, *Computer Methods in Applied Mechanics and Engineering* 32 (2016) 253–280, ISSN 0045-7825, doi:<http://dx.doi.org/10.1016/j.cma.2015.10.005>, URL <http://www.sciencedirect.com/science/article/pii/S0045782515003242>.
- [39] V.-D. Nguyen, E. Béchet, C. Geuzaine, L. Noels, Imposing periodic boundary condition on arbitrary meshes by polynomial interpolation, *Computational Materials Science* 55 (0) (2012) 390 – 406, ISSN 0927-0256, doi:[10.1016/j.commatsci.2011.10.017](https://doi.org/10.1016/j.commatsci.2011.10.017), URL <http://www.sciencedirect.com/science/article/pii/S0927025611005866>.
- [40] M. Ainsworth, Essential boundary conditions and multi-point constraints in finite element analysis, *Computer Methods in Applied Mechanics and Engineering* 190 (48) (2001) 6323 – 6339, ISSN 0045-7825, doi:[10.1016/S0045-7825\(01\)00236-5](https://doi.org/10.1016/S0045-7825(01)00236-5).
- [41] URL <https://pytorch.org/>, accessed: 2020-04-30, 2020.
- [42] A. Cuitino, M. Ortiz, A material-independent method for extending stress update algorithms from small-strain plasticity to finite plasticity with multiplicative kinematics, *Engineering computations* 9 (1992) 437–437.
- [43] L. Wu, V. D. Nguyen, N. G. Kilinger, L. Noels, Data of A recurrent neural network-accelerated multi-scale model for elastoplastic heterogeneous materials subjected to random cyclic and non-proportional loading paths, doi:[10.5281/zenodo.3902663](https://doi.org/10.5281/zenodo.3902663), URL <https://doi.org/10.5281/zenodo.3902663>, 2020.