

---

# Physics-informed and thermodynamics-based neural networks

**Filippo Masi<sup>1</sup>, Ioannis Stefanou<sup>2</sup>**

<sup>1</sup> *The University of Sydney, Australia*

<sup>2</sup> *Nantes Université, Ecole Centrale Nantes, CNRS, Institut de Recherche en Génie Civil et Mécanique (GeM), UMR 6183, F-44000, Nantes, France*

---

*This chapter aims to offer a comprehensive introduction to the integration of prior knowledge stemming from physics and/or thermodynamics into deep learning algorithms. By means of motivating examples, we explore the capabilities and strengths of (i) Physics-Informed Neural Networks for the discovery, driven by data, of partial differential equations and of (ii) Thermodynamics-based Artificial Neural Networks for the discovery of constitutive equations of complex materials.*

## 1 Introduction

Fuelled by a continuously increasing flow of data, Machine Learning (ML) has been offering promising solutions to diverse problems in a broad spectrum of disciplines. As a result, science has experienced a shift of paradigm, where data are no longer confined to a mere supporting role, but have instead taken center stage as main protagonists in the scientific narrative. This transformative shift can be considered as the advent of a “fourth paradigm” [HTT<sup>+</sup>09], which coexists with and complements the three traditional scientific paradigms: experimental, theoretical, and computational.

Within such a context, the quest for accurate and predictive data-intensive (or data-centric) models has received and continues to receive enormous attention. Such models are often characterized by complex, highly specialized algorithms, e.g. deep neural networks, with numerous interconnected parameters that are optimized to learn those patterns and those correlations proper to a training data set. However, while data-intensive models may fit observations well, many of them face limitations in extracting interpretable information and knowledge from vast amounts of data. In addition, they can suffer from physical inconsistencies and poor generalization [KKL<sup>+</sup>21]. *Consistency* refers here to the fulfilment of laws of physics and thermodynamics. *General-*

*ization* refers to the ability of making predictions for data that do not belong to the particular data set used in the learning process (cf. Chapter 2).

To address these challenges, there has been a surging desire, in the last four-to-five years, to switch the focus from a pure data-centric vision towards a hybrid one that also accounts for first principles – namely, physics and/or thermodynamics. That is, leverage the high expressivity (i.e. approximation properties) of ML models but also embed prior knowledge (or physics-based bias) stemming from our physical understanding of the world in order to reach improved performance, reliability, and interpretability. This gave birth to numerous successful methods, sometimes referred to as *physics-informed machine learning*<sup>1</sup>. These approaches enable the enforcement of prior knowledge through the introduction of appropriate *observational*, *inductive*, and/or *learning biases* [KKL<sup>+</sup>21].

*Observational biases* use data augmentation procedures by leveraging physical principles that dictate the generation of the latter<sup>2</sup> [LJP<sup>+</sup>21]. *Inductive biases* resort to the design of specialized neural network architectures that, by construction, embed prior assumptions or physical models, e.g. [HBG<sup>+</sup>21, MSVMB21]. *Learning biases* consists of introducing physical constraints, rather than via specialized architectures, in a soft manner, by appropriately penalizing the loss functions of conventional neural networks, e.g. [RPK19]. For more, we refer to [KKL<sup>+</sup>21].

In this chapter, we focus on two emblematic examples of physics-informed machine learning. The first one consists of the seminal work carried out by Raissi, Perdikaris, and Karniadakis on “Physics-Informed Neural Networks” (PINN) [RPK19]. PINN are a class of supervised deep learning algorithms capable of encoding, through learning biases, physical laws that govern a given data set, and can be described by partial differential equations.

The second example is provided by “Thermodynamics-based Artificial Neural Networks” (TANN) [MSVMB21], that, through the hardwiring of the laws of thermodynamics within the architecture of neural networks (inductive and learning bias), enable the discovery of constitutive equations of complex materials.

After studying this chapter, we hope that the reader will be able to

- Grasp the limitations of machine learning in providing reliable and accurate descriptions of physical phenomena and, at the same time, the need for developing ML approaches accounting for physics/thermodynamics principles.
- Understand, by means of hands-on and pedagogic examples, how to introduce physical knowledge in the form of learning biases (e.g. PINN) and inductive biases (e.g. TANN) to construct high-fidelity physical representations from data.

The codes related to the hands-on example are available on ALERT Geomaterial GitHub (repository [alert-geomaterials/2023-doctoral-school](#)), while those

---

<sup>1</sup>There seems to be no strong consensus on the definition of ML models leveraging prior knowledge, mainly due to the fact that this is a very fast moving field. In this Chapter, we follow the broad definition suggested in [KKL<sup>+</sup>21]. Note that this is not the sole possibility, of course.

<sup>2</sup>For instance, augment the data by leveraging frame indifference and/or symmetries.

related to TANN can be found at [github.com/filippo-masi/TANN-multiscale](https://github.com/filippo-masi/TANN-multiscale) and [filippo-masi/Thermodynamics-Neural-Networks](https://github.com/filippo-masi/Thermodynamics-Neural-Networks).

Updated versions of this chapter are also available at [filippo-masi.github.io](https://filippo-masi.github.io).

Below, we adopt the following notation:  $\mathbf{a} \cdot \mathbf{b} = a_i b_i$ ,  $\mathbf{P} : \dot{\mathbf{F}} = P_{ij} \dot{F}_{ij}$ , and  $\operatorname{div} \mathbf{y} = \frac{\partial y_i}{\partial x_j}$ , with  $\mathbf{x}$  the spatial coordinates,  $i, j = 1, 2, 3$ . Einstein's summation is implied for repeated indices.

## 2 Physics-informed neural networks

Physics-Informed Neural Networks (PINN) are a class of supervised deep learning algorithms capable of encoding physical laws governing given data sets whose evolution can be described by partial differential equations.

The setting considers parametrized and nonlinear partial differential equations (PDE) of the form

$$\dot{\mathbf{u}} + \mathcal{N}_\gamma[\mathbf{u}] = 0, \quad (1)$$

where  $\mathbf{u} = \hat{\mathbf{u}}(\mathbf{x}, t)$  is the latent (hidden) solution and  $\dot{\mathbf{u}}$  its partial derivative with respect to time – the superposed caret in  $\hat{\mathbf{u}}$  serves to distinguish the solution from its values,  $\mathcal{N}_\gamma[\cdot]$  is a nonlinear differential operator parametrized by  $\gamma$ , with  $\mathbf{x} \in \Omega \subset \mathbb{R}^n$  the spatial coordinates and  $t$  the time coordinate. Note that expression (1) encapsulates a wide range of problems in physics including conservation laws, diffusion processes, advection-diffusion-reaction systems, and kinetic equations. For instance, the heat transfer equation<sup>3</sup> can be retrieved by selecting  $N_\gamma[\mathbf{u}] = \gamma \Delta \mathbf{u}$ , where  $\Delta \cdot$  denotes the Laplacian.

### 2.1 Methodology

By leveraging the high expressivity of neural networks and prescribed physical equations during the learning process – by means of learning biases – PINN provide a powerful and general framework, driven by data, to (i) discover solutions of PDE and/or (ii) the underlying differential equations, by leveraging expression (1).

#### 2.1.1 Solution of partial differential equations

The data-driven solution of PDE aims at finding an approximation of  $\hat{\mathbf{u}}$  such that (1) holds true for fixed model parameters  $\gamma$ , by means of a deep neural network  $\mathbf{u}_\theta$ , parametrized with respect to the network parameters  $\theta$  – that is, the set of weights and biases of each layer (see Chapter 7).

In doing so, the expression (1) is reformulated as

$$\varrho(\mathbf{x}, t) \equiv \dot{\mathbf{u}}_\theta + \mathcal{N}[\mathbf{u}_\theta], \quad (2)$$

---

<sup>3</sup> $\dot{\mathbf{u}} = \alpha \Delta \mathbf{u}$ , where  $\alpha$  is the thermal diffusivity.

where  $\varrho(\mathbf{x}, t)$  is a residual. The residual is computed by applying the chain rule for differentiating compositions of functions using automatic differentiation (cf. [BPRS18] and Chapters 7 and 8), see Figure 1. The learning process consists of the minimization of a loss function composed of a purely data-driven term and a physics-based one, namely

$$\mathcal{L} = \lambda_u \mathcal{L}_u + \lambda_{\text{PDE}} \mathcal{L}_{\text{PDE}}, \quad (3)$$

where  $\lambda_u$  and  $\lambda_{\text{PDE}}$  are weighting parameters that can be tuned to balance the interplay between the two losses.  $\mathcal{L}_u$  and  $\mathcal{L}_{\text{PDE}}$  are, respectively, a supervised loss related to the (data) measurements  $\mathbf{u}$  from the initial and boundary conditions and an unsupervised loss of the PDE, namely

$$\begin{aligned} \mathcal{L}_u &= \frac{1}{N_u} \sum_{i=1}^{N_u} \left\| \mathbf{u}_\theta \left( \mathbf{x}^{(i)}, t^{(i)} \right) - \mathbf{u}^{(i)} \right\| \\ \mathcal{L}_{\text{PDE}} &= \frac{1}{N_{\text{PDE}}} \sum_{k=1}^{N_{\text{PDE}}} \left\| r \left( \mathbf{x}^{(k)}, t^{(k)} \right) \right\|, \end{aligned} \quad (4)$$

$\|\cdot\|$  being a metric based on an arbitrary norm (cf. Chapter 2). Here  $\{(x^{(i)}, t^{(i)})\}$  denote the initial and boundary training data on  $\hat{\mathbf{u}}(\mathbf{x}, t)$ , while  $\{(x^{(k)}, t^{(k)})\}$  specify the set of collocation points sampled in the entire domain for  $\varrho(\mathbf{x}, t)$ .

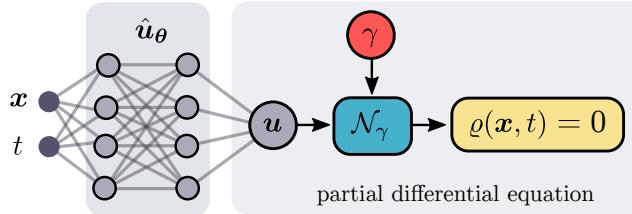


Figure 1: NN versus PINN: learning the projectile motion from

### 2.1.2 Discovery of partial differential equations

In the presence of noisy and/or incomplete measurements  $\mathbf{u}$  of the state of the system, PINN can be deployed for the data-driven discovery of PDE by means a of neural network  $\mathbf{u}_\theta$  and learning model parameters  $\gamma$  that best describe the measurement data. The expression (1) is reformulated as

$$\varrho(\mathbf{x}, t) \equiv \dot{\mathbf{u}}_\theta + \mathcal{N}_\gamma[\mathbf{u}_\theta]. \quad (5)$$

Exactly as for the discovery of the solution of PDE, the network is trained over the minimization of the loss (3), where the unknown parameters  $\gamma$  are part of the optimization problem.

While there is no theoretical guarantee of convergence to a global minimum of the loss function (3), many benchmarks and applications, e.g. [RPK19], demonstrated that PINN can achieve accurate predictions when applied to well-posed PDE with unique solutions, both in terms of data-driven discovery of solutions and of PDE. PINN are nowadays used to solve not only partial differential equations, but also fractional equations, integral-differential equations, and stochastic PDE, see [CDCG<sup>+</sup>22] for an extensive review.

The underlying success depends on accounting for physical biases, employing a sufficiently expressive neural network architecture, and an adequate number of collocation points ( $N_{\text{PDE}}$ ) – being the sampling choice a crucial aspect, cf. [LP21].

## 2.2 Hands-on example

Let us consider as an example the (one-dimensional) motion of a projectile, governed by the ordinary differential equation (ODE)  $\ddot{u}(t) + g = 0$ , that has the following general solution

$$u(t) = u(0) + \dot{u}(0)t - \frac{1}{2}gt^2,$$

where  $u$ ,  $\dot{u}$ , and  $\ddot{u}$  are the trajectory, velocity, and acceleration of the projectile, respectively;  $t$  is the time, and  $g$  is the gravitational acceleration.

This example should ring a bell to those who read Chapter 2, on regression methods. There, we considered exactly the same problem but addressed it with regularized polynomial regression. Here, we aim at learning the same governing equation but instead by using traditional artificial neural networks and physics-informed ones.

We proceed by reusing the data set generated in Chapter 2, Section 7, and add a Gaussian measurement noise on the data representing the trajectory,  $u^{(i)}$  – following the formalism introduced above. In doing so, we use the code hereinafter:

```
t_i = X[0:50:2].copy() # from Chapter 2, Sect. 7
u_i = y[0:50:2].copy() # from Chapter 2, Sect. 7
u_i += 0.1 * np.multiply(np.random.normal(0,1,25), u_i) # add noise
u_i[0] = 0.0 # noise-free initial condition
t_i = torch.tensor(t_i)[:,None] # expand dim
u_i = torch.tensor(u_i)[:,None] # expand dim
```

Figure 2 displays the exact solution of the ODE for  $t \in [0, 1]$  s and the noisy data measurements  $u^{(i)}$  used for training,  $t \in [0, 0.5]$  s.

Then, we continue by defining a neural network (in PyTorch), that we will train (i) in the traditional way and (ii) following the general setting provided by PINN. The code below serves to construct a general neural network architecture with inputs of dimension  $i\_dim$ , outputs of dimension  $o\_dim$ , hidden layers with  $h\_dim$  nodes, and Tanh activations:

```
import numpy as np
import torch
torch.manual_seed(123)
```

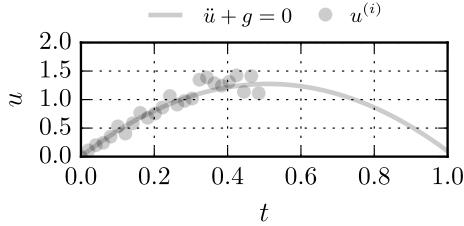


Figure 2: Projectile motion: exact solution of the ordinary differential equation  $\ddot{u} + g = 0$  and noisy data measurements  $u^{(i)}$  used for training,  $t \in [0, 0.5]$  s.

```
np.random.seed(123)
class NeuralNet(torch.nn.Module):
    '''Feed-forward neural network'''
    def __init__(self, hyper_params, dtype=torch.float32):
        super(NeuralNet, self).__init__()
        self.dtype = dtype
        self.NN = self.constructor(hyper_params) # neural net
        self.gamma = torch.nn.Parameter(torch.tensor(1.)) # gamma in PINN
    def constructor(self, hyper_params):
        '''Construct neural net'''
        i_dim, o_dim, h_dim = hyper_params
        dim = i_dim
        layers = torch.nn.Sequential()
        for hdim in h_dim:
            layers.append(torch.nn.Linear(dim, hdim, dtype=self.dtype))
            layers.append(torch.nn.Tanh())
            dim = hdim
        layers.append(torch.nn.Linear(dim, o_dim, dtype=self.dtype))
        return layers
    def forward(self, x):
        '''Forward pass'''
        return self.NN(x)
```

In order to train the neural network, without any learning bias, we proceed as it follows:

```
dtype=torch.float32
NN_params = [1,1,[12,12]] # [i_dim, o_dim, [h_dim, ...]]
model = NeuralNet(NN_params,dtype) # build neural net
optimizer = torch.optim.Adam(model.parameters(),lr=1e-2) # Adam optimizer
verbose_training = 1000 # display loss
n_epochs = 10000 # number of epochs
for i in range(n_epochs):
    optimizer.zero_grad()
    u_i_pred = model(t_i) # make predictions
    MSE_u = torch.mean((u_i_pred-u_i)**2) # MSE_u
    MSE_u.backward() # backpropagate loss
    optimizer.step()
    if i % verbose_training == 0:
        print('epoch:{:5.0f}'.format(i),
              '- train_loss:{:8.6f}'.format(MSE_u.item()))
```

Note that the architecture is composed of two hidden layers, with 12 nodes each. The evolution of the predictions of the network, at varying of the number of epochs, is shown in Figure 3(a). Without any surprise, the neural network (progressively) massively overfits the training data, and the associated noise. As a result, it cannot neither generalize nor extrapolate. Indeed, we can notice that the predicted “projectile” seems to find itself amidst a tempestuous whirlwind, in the training range ( $t \in [0, 0.5]$  s), while it *magically* levitates, defying common sense and the laws of physics (i.e., gravity), in the extrapolation range ( $t > 0.5$  s).

Let us see, at this point, what happens if we insert prior knowledge based on physics. We train the PINN analog of the same neural network and in order to discover the underlying governing equation, we assume the following physical model:

$$\varrho(t) \equiv \ddot{u}_\theta + \gamma,$$

where  $\gamma$  is a trainable parameter – in the code snippet above, `NeuralNet.gamma`. Reformulating, we are telling the network that the second order derivative of the solution is equal to an undetermined constant, that the network will eventually identify.

During the learning process, we sample uniformly distributed time (collocation) points  $t^{(k)} \in [0, 1]$  s. Automatic differentiation is used to compute  $\ddot{u}_\theta$  relying on PyTorch’s functionality `torch.autograd.grad()`:

```
t_k = torch.linspace(0,1,20).view(-1,1).requires_grad_(True) # collocation pt
model = NeuralNet(NN_params,dtype) # build PINN
optimizer = torch.optim.Adam(model.parameters(),lr=1e-2) # Adam optimizer
verbose_training = 1000 # display loss
n_epochs = 10000 # number of epochs
lambda_PDE = 0.2 # weighting parameter
for i in range(n_epochs):
    optimizer.zero_grad()
    u_i_pred = model(t_i) # make predictions for t_i
    MSE_u = torch.mean((u_i_pred-u_i)**2) # MSE_u
    u_k_pred = model(t_k) # make predictions for t_k
    dudt = torch.autograd.grad(u_k_pred, t_k, torch.ones_like(u_k_pred),
        create_graph=True)[0] # computes du/dt
    dduddt = torch.autograd.grad(dudt, t_k, torch.ones_like(dudt),
        create_graph=True)[0] # computes d^2u/dt^2
    physics = dduddt+model.gamma # computes the residual
    MSE_PDE = lambda_PDE*torch.mean(physics**2) # MSE_PDE
    loss = MSE_u + MSE_PDE # add two loss terms together
    loss.backward() # backpropagate joint loss
    optimizer.step()
    if i % verbose_training == 0:
        print('epoch:{:5.0f}'.format(i),
              '- train_loss:{:8.6f}'.format(MSE_u.item()))
```

Figure 3(b) displays the evolution of the predictions of the physics-informed network, as the training advances: much better, isn’t it?

Thanks to the (physical) learning bias we introduced, the physics-informed network not only possesses good generalization capabilities, but is also able to extrapolate<sup>4</sup>

---

<sup>4</sup>A disclaimer is necessary concerning the capability of the trained PINN model to extrapolate. In general, there is no guarantee, as for all ML models, that the network enables extrapolation. In this example,

with high accuracy. By evaluating, at the end of the learning process, the  $\gamma$  parameter, we can indeed notice that the network finds a good approximation of the gravitational acceleration, namely  $\text{gamma} = 10.58 \text{ m/s}^2$ .

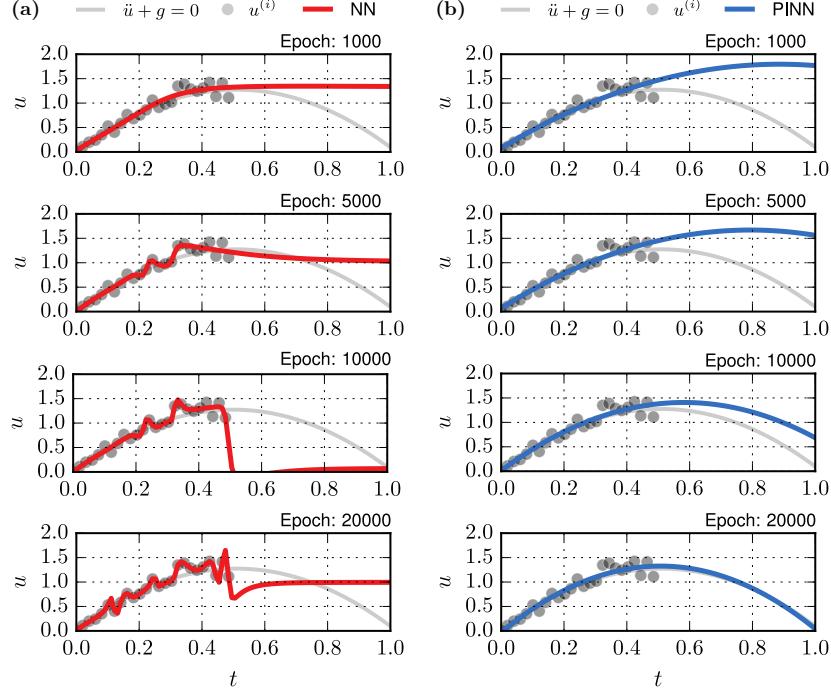


Figure 3: Learning the projectile motion from noisy measurement data using a traditional neural network (NN) (a) and a physics-informed neural network (PINN) (b).

Before concluding this example, it is worth drawing an interesting analogy between the addition of physical biases and regularization strategies in ML. In particular, we have seen in Chapter 2, paragraph 7.1, how regularized regression models based on the minimization of the  $\ell_1$  and  $\ell_2$  norm can provide similar accuracies compared to the physics-informed neural network we have deployed for predicting the projectile motion. It is important to understand that such a similarity is not simply a coincidence. To fix the ideas, let us evaluate, in Figure 4, the evolution of the  $\ell_1$  and  $\ell_2$  norms of the parameters of the traditional neural network and its physics-informed sibling. The latter is characterized by a much smaller norm compared to the traditional neural network – with a 70% decrease. This means that the physical bias acts as a sort of regularizer, shrinking the value of most of the network parameters  $\theta$  to zero. The opposite is also true: the benefits of regularization techniques lie in the fact that they enable the discovery of simple models – which, most of the time, are more physical

---

extrapolation in the domain  $t \in [0, 1] \text{ s}$  is made possible thanks to the adopted sampling strategy for the collocation points, i.e.  $t^{(k)} \in [0, 1] \text{ s}$ .

than very sophisticated ones. In an essence, both techniques (physics and regularization) allow for the implementation of a “*lex parsimoniae*” (see Chapter 2, Section 7, and [KB22]).

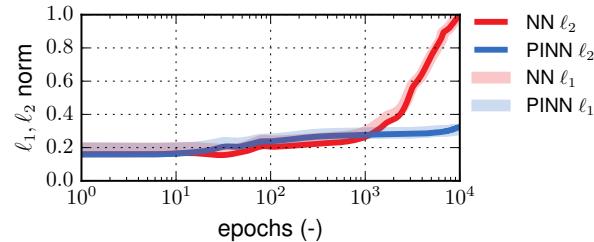


Figure 4: Evolution, at training, of the  $\ell_1$  and  $\ell_2$  norm (scaled with respect to the maximum value) of the neural network parameters with and without learning biases based on physics, respectively PINN and NN.

### 3 Thermodynamics-based neural networks

Accurate models for the behavior of materials are of fundamental importance in material science and (geo-)mechanics. However, heuristic constitutive models, traditionally derived from first principles (thermodynamics) and empirical approaches (to ensure calibration over experiments), can hardly describe the behavior of all complex materials that display path-dependency and possess multiple inherent scales. In such contexts, multiscale approaches are commonly preferred as they enable to capture the effects of the fine material scales (here denoted with *microstructure*) on the average (*macroscopic*) behavior. However, the bottleneck of these approaches lies in the cumbersome and time-consuming ways with which the auxiliary problem is being solved which eventually hinders their application in real-case scenarios.

In recent years, ML, particularly deep learning, has been offering new ways to reduce the computational burden of multiscale approaches. Indeed, taking advantage of the high expressivity of neural networks (see Chapters 7 and 8), deep learning can enable the modeling of several aspects of the behavior of complex materials and show promise in building high-fidelity replicas or *digital twins*<sup>5</sup>.

Yet, constitutive modeling based on neural networks has been facing two major issues

<sup>5</sup>Let us define the concepts of *virtual* and *digital twins*. We do so by adopting the definitions proposed in [CCAC<sup>+</sup>20]. A *virtual twin* refers to the emulation of a physical system by one or more mathematical models that describe the behavior of the former. To fix the ideas, numerical models, such the Finite Element Method (FEM) or the Discrete Element Method (DEM), are examples of virtual twins. A *digital twin* – term originally coined by General Electric to depict the digital replica of an engine produced in their manufacturing facilities – is a data-intensive representation (or model) of the same physical system. As opposite to its virtual siblings, digital twins are usually characterized by reduced computational complexity and can be deployed for real-time decision-making.

– inconsistency and poor generalization, cf. Section 1 – that hinder the applicability of classical “black-box” (i.e., physics-agnostic) approaches [GS98, LS03, LBS09].

To this, new developments were recently proposed to tackle the lack of physical consistency in neural networks and the consequent constitutive models. In particular, it has been demonstrated that it is possible to structure neural networks in a way that they deliver, though inductive and/or learning biases (cf. Section 1), thermodynamics-/physics-consistent constitutive representations, see e.g. [HBG<sup>+</sup>21, YZYK22, KFM<sup>+</sup>22]. Among these approaches, the Thermodynamics-based Artificial Neural Networks (TANN, [MSVMB21, MS22, MS23]), based on the theory of internal state variables, are able to uncover constitutive equations from the laws of thermodynamics. As a direct consequence of the thermodynamically-consistent architecture, TANN deliver accurate predictions of the material behavior and show remarkable generalization abilities in presence of unseen loading paths, both within the range of the training set (interpolation) and outside (extrapolation).

Herein, we focus on presenting the methodology and the underlying theoretical framework of TANN. Through a pedagogical, example we illustrate how thermodynamics-based neural networks enable to construct accurate, robust, and reliable digital twins/replicas of granular media. This approach not only captures the overall macroscopic material behavior but also allows to discover the (hidden) internal variables and to characterize the time evolution of the microstructure at extremely reduced computational cost. Finally, we demonstrate how to deploy the same framework for speeding-up state-of-the-art multi- and fine-scale simulations.

### 3.1 Theoretical framework

We start by briefly recalling the general continuum thermodynamic setting.

The local form of the energy balance and dissipation rate inequality, at the level of an arbitrary unit volume element<sup>6</sup>  $\mathcal{V}$  and with respect to the reference configuration, read

$$\dot{e} + \operatorname{div} \mathbf{q} = \mathbf{P} : \dot{\mathbf{F}} + \mathbf{r}, \quad (6)$$

$$\dot{d} = \dot{\eta} - (T^{-1}\mathbf{r} - \operatorname{div}(T^{-1}\mathbf{q})) \geq 0, \quad (7)$$

where  $e$  and  $\dot{e}$  are the volume density of the internal energy and its local time derivative;  $\mathbf{P}$  is the first Piola-Kirchhoff stress tensor;  $\mathbf{F}$  is the deformation gradient and  $\dot{\mathbf{F}}$  its rate of change;  $\mathbf{q}$  is the heat flux vector;  $\mathbf{r}$  is the volume density of possible external source terms;  $\dot{d}$  is the volume density of the total (mechanical plus thermal) dissipation rate;  $\eta$  the volume density of entropy; and  $T$  the absolute temperature. Note that all above quantities are functions of the time  $t$ , i.e.  $\mathbf{F} \equiv \mathbf{F}(t)$ , but for the sake of simplicity, we voluntarily omitted, and continue to do so, such explicit way of writing.

---

<sup>6</sup>The notion of unit volume element or unit cell, stems from considerations related to the mathematical theory of asymptotic homogenization. All quantities of interest, except when explicitly written, are herein expressed in the form of volume averages, e.g.  $\psi = \frac{1}{|\mathcal{V}|} \int_{\mathcal{V}} \psi(\mathbf{x}) d\mathbf{x}$ , where  $\mathbf{x} \in \mathcal{V}$  are the spatial coordinates and  $|\mathcal{V}|$  is the volume. For more, we refer to [Mie02, MS22].

From the combination of the above two expressions – that is, the first and second law of thermodynamics – we obtain the Clausius-Duhem inequality, namely

$$\begin{aligned} d &= \mathbf{P} : \dot{\mathbf{F}} - \dot{e} + T\dot{\eta} - T^{-1}\mathbf{q} \cdot \mathbf{m} \\ &= \mathbf{P} : \dot{\mathbf{F}} - (\dot{\psi} + \dot{T}\eta) - T^{-1}\mathbf{q} \cdot \mathbf{m} \geq 0 \end{aligned} \quad (8)$$

the latter involving the Helmholtz free energy density (per unit volume),  $\psi = e - Ts$ , and denoting with  $\mathbf{m}$  the temperature gradient. Whenever  $\mathbf{m} = 0$ , the Clausius-Duhem inequality becomes

$$d = \mathbf{P} : \dot{\mathbf{F}} - \dot{\psi} - \dot{T}\eta \geq 0, \quad (9)$$

which has to hold at each time  $t$ , and where  $d$  is the internal (i.e., mechanical) dissipation rate.

Relying on the aforementioned expression, we derive constitutive restrictions on material processes. In particular, we follow the seminal work of Coleman and Gurtin [CG67] and introduce the internal state variables  $z_j$ , with  $j = 1, 2, \dots, n_z$  – where  $n_z$  is the number of internal state variables. Note that the internal state variables or, internal variables for short, are introduced to account for the influence of (dissipative) micromechanisms on the constitutive behavior of materials. At this point, we continue by assuming the free energy density as a function of the internal variables and the state variables,  $\mathbf{F}$  and  $T$  – that is,

$$\psi = \hat{\psi}(T, \mathbf{F}, \mathbf{z}), \quad (10)$$

where the superposed caret in  $\hat{\psi}$  serves to distinguish the free energy function from its values and  $\mathbf{z}$  denotes the internal state vector  $\mathbf{z} = (z_1, z_2, \dots, z_{n_z})$ . Notice that, alternatively, one may consider, rather than  $\mathbf{F}$ , its elastic part in Eq. (10), see e.g. [Rub01, Ein12, Daf22]. However, here we disregard such a choice as we aim at developing a constitutive modeling approach that is independent of the requirement of the decomposition of the deformation gradient, or its rate of change, into an elastic and a plastic part.

It should also be mentioned that, for a given material, the free-energy function is not unique for a given deformation gradient (for example, the free energy values  $\psi$  and  $\psi + c$ , with  $c$  a constant, will give the same stress-strain response and dissipation rate). From Equation (10), it follows that

$$\dot{\psi} = \partial_{\mathbf{F}}\hat{\psi} : \dot{\mathbf{F}} + \partial_T\hat{\psi} \dot{T} + \partial_{\mathbf{z}}\hat{\psi} \cdot \dot{\mathbf{z}}, \quad (11)$$

where  $\partial_{\mathbf{F}}\hat{\psi} = \frac{\partial\hat{\psi}}{\partial\mathbf{F}}$ . We continue by substituting expression (11) into the Clausius-Duhem inequality (9), and obtain the following expression for the internal dissipation rate

$$(\mathbf{P} - \partial_{\mathbf{F}}\hat{\psi}) : \dot{\mathbf{F}} - (\eta + \partial_T\hat{\psi}) \dot{T} - \partial_{\mathbf{z}}\hat{\psi}(T, \mathbf{F}, \mathbf{z}) \cdot \dot{\mathbf{z}} - d = 0, \quad \forall \dot{T}, \dot{\mathbf{F}}, \dot{\mathbf{z}} \quad (12)$$

with  $-\partial_{\mathbf{z}}\hat{\psi}(T, \mathbf{F}, \mathbf{z})$  being the thermodynamic forces. It follows that, for satisfying equality (12) for any  $\dot{T}, \dot{\mathbf{F}}, \dot{\mathbf{z}}$ , every admissible thermodynamic process must satisfy

the following constitutive restrictions hereinafter, holding true at any time  $t$ ,

$$\mathbf{P} = \partial_{\mathbf{F}} \hat{\psi}(T, \mathbf{F}, \mathbf{z}), \quad (13)$$

$$\eta = -\partial_T \hat{\psi}(T, \mathbf{F}, \mathbf{z}), \quad (14)$$

$$d = -\partial_{\mathbf{z}} \hat{\psi}(T, \mathbf{F}, \mathbf{z}) \cdot \dot{\mathbf{z}} \geq 0. \quad (15)$$

A last ingredient allows the closure of the constitutive relationship: the evolution equation for the internal variables, based on (10) and (12-15) takes the form

$$\dot{\mathbf{z}} = \mathbf{f}(T, \mathbf{F}, \mathbf{z}) \quad (16)$$

such that the dissipation inequality (15) holds true. The above equation can cover a wide variety of constitutive models, including rate-dependent and rate-independent plasticity theory through the introduction of Lagrange multipliers (see [EHN07]). Note that, in some particular cases, e.g. when postulating the existence of a dissipation potential, analytical expressions for  $\mathbf{f}$  that satisfy the dissipation inequality by construction, can be retrieved [MM94, EHN07]. Similarly, evolution equations could be formulated as Onsagerian conductivity equations satisfying the dissipation inequality (see [Gur96, Ván03, EL18], among others). Yet, the above structure does not originate directly from thermodynamic considerations and may not be general enough, as it requires ad-hoc assumptions on the form of the internal variables rates and the thermodynamic forces. Thus, in the following, we opt for the more general expression (16).

### 3.1.1 The quest for internal state variables

The aforementioned thermodynamics framework builds upon the knowledge of the internal variables,  $\mathbf{z}$ . And, whilst TANN can effectively leverage any a priori identified set of internal variables (as we will see in the next paragraph), the key question is: *how can we identify internal state variables?* Traditionally, to determine the number and nature of internal variables, we must identify the internal mechanisms and phenomena that influence the behavior of the particular material at hand. This approach, referred to as the *Art of modelling* [Mau15], requires adaptation to each specific application and material and is hardly scalable. To face this difficulty, there has been a recent interest for devising approaches that automatically reveal – or discover – the internal variables independent of the type of the material, see e.g. [VBE08, HBG<sup>+</sup>21, MS22].

Herein, we adopt the approach developed in [MS22], where internal variables are identified based on the knowledge of the internal degrees of freedom of the microstructure of the material under investigation. To this end, we introduce a new quantity, referred to as *internal coordinates*  $\xi$ . These internal coordinates describe the material behavior at the microscopic scale<sup>7</sup> and encompass variables such as displacement, velocity, momentum fields, and internal force networks.

---

<sup>7</sup>Herein, we adopt the notion of microscopic scale as the finest scale of a material. Depending on the degree of fidelity one desires to achieve, *microscopic* may refer, for instance, to the scale of the grains composing a granular material or the atoms composing each grain.

According to this formalism, the identification of internal variables passes from learning low-rank representations of the internal coordinates by identifying an operator  $\mathbf{h}$ , such that

$$\mathbf{z} \equiv \mathbf{h}(\xi) \quad \text{and} \quad \psi = \hat{\psi}(T, \mathbf{F}, \mathbf{z}). \quad (17)$$

In parallel and without any loss of generality, a pseudoinverse operator  $\mathbf{g}$  can also be postulated, such that

$$\tilde{\xi} = \mathbf{g}(\mathbf{h}(\xi)) = \mathbf{g}(\mathbf{z}), \quad (18)$$

where the superposed tilde serves to distinguish a low-rank approximation of the internal coordinates from the internal coordinates themselves. To rephrase, the adopted framework implies that the determination of the internal variables boils down to identifying the operator  $\mathbf{h}$  – and, additionally, the operator  $\mathbf{g}$  – from the knowledge of the microstructural internal degrees of freedom of a material and without any a priori constitutive choice<sup>8</sup>.

### 3.2 Methodology

The thermodynamic framework presented above can be employed in the architecture of neural networks to learn physical and generalizable constitutive models. Such neural networks are referred to as Thermodynamics-based Artificial Neural Networks (TANN) [MS22]. TANN can be declined into two configurations, depending whether the internal state variables are priori known [MSVMB21] or not [MS22]. Both differential discrete-time and continuous-time formulations can be employed [MS23].

Herein, we briefly recall the main building blocks of the network (for more, we refer to [MS23]). TANN are composed of two building blocks: the free energy density network and the evolution equation network. The free energy network, shown in Figure 5, is responsible of the prediction of the material stress, from the fulfilment (inductive bias) of the thermodynamics restrictions (13-15). It consists of one network trained to predict the value of the free energy density – that is,  $\psi = \hat{\psi}_\theta(T, \mathbf{F}, \mathbf{z})$ , parametrized by the neural networks parameters  $\theta$ . Material stress, entropy, and internal dissipation rate are computed, using the constitutive restrictions (13-15), by relying on the automatic differentiation of the operator  $\hat{\psi}_\theta$  with respect to its inputs. The free energy

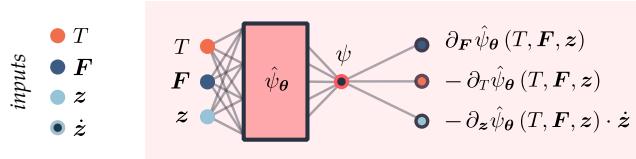


Figure 5: Thermodynamics block. All quantities refer at time  $t$ .

---

<sup>8</sup>It is worth noticing that we only consider a finite number of internal variables. Whilst this is the case for all constitutive models developed so far, one may question whether the state space of a real, non-idealized, material can be proven to be finite or not.

network is trained by minimizing a loss composed of three contributions, and namely

$$\mathcal{L}_{\text{energy}} \equiv \mathcal{L}_\psi + \mathcal{L}_{\nabla\psi} + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}}, \quad (19)$$

where  $\lambda_{\text{reg}}$  is a weighting parameter. The three terms ensure, respectively, that (i) the neural network can predict the free energy density, (ii) the gradients of the latter coincide with the material stresses, entropy density, and intrinsic dissipation rate, and (iii) the fulfilment (in the form of a learning bias) of the dissipation inequality, namely

$$\begin{aligned} \mathcal{L}_\psi &\equiv \left\| \psi - \hat{\psi}_\theta(T, \mathbf{F}, \mathbf{z}) \right\|, \\ \mathcal{L}_{\nabla\psi} &\equiv \left\| \mathbf{P} - \partial_{\mathbf{F}} \hat{\psi}_\theta(T, \mathbf{F}, \mathbf{z}) \right\| + \left\| \eta + \partial_T \hat{\psi}_\theta(T, \mathbf{F}, \mathbf{z}) \right\| + \left\| d + \partial_{\mathbf{z}} \hat{\psi}_\theta(T, \mathbf{F}, \mathbf{z}) \cdot \dot{\mathbf{z}} \right\|, \\ \mathcal{L}_{\text{reg}} &\equiv \left\| [\partial_{\mathbf{z}} \hat{\psi}_\theta(T, \mathbf{F}, \mathbf{z}) \cdot \dot{\mathbf{z}}] \right\|, \end{aligned} \quad (20)$$

with  $\|\cdot\|$  being an error metric based on an arbitrary norm (cf. Chapter 2), averaged over all data points, and  $[\cdot]$  being the Macaulay brackets. Note that the loss associated with the free energy density,  $\mathcal{L}_\psi$ , is not a necessary condition for the fulfilment of the laws of thermodynamics and the accurate prediction of the material response. The same holds true for the loss associated with the dissipation rate, which can be omitted in those cases where there is no information related to the values of the dissipation (cf. paragraph 3.3). Indeed, it can be easily proved that, in absence of the aforementioned loss terms, the free energy density network, trained only through its gradients, will still respect the restrictions (13-15), because they are hardwired in the network architecture.

The evolution equation network, shown in Figure 6, is responsible for learning the evolution equations of the internal variables by means of a neural network  $f_\theta(T, \mathbf{F}, \mathbf{z})$ . The training is performed by minimizing the error between the outputs and the rates of change of the internal variables,

$$\mathcal{L}_{\dot{\mathbf{z}}} = \left\| \dot{\mathbf{z}} - f_\theta(T, \mathbf{F}, \mathbf{z}) \right\|. \quad (21)$$

The above evolution equation can be identified either relying on the aforementioned (time-continuous) formulation or, rather, by relying on a discrete-time, incremental one.

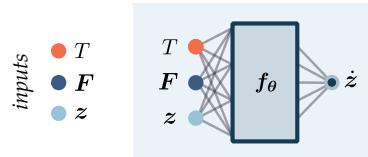


Figure 6: Evolution law block for a priori determined internal variables. All quantities refer at time  $t$ .

### 3.2.1 Data-driven identification of internal variables

To the aforementioned building blocks, a third one adds whenever no constitutive assumptions are made on the nature of the internal variables. At the heart of the identification of the internal variables and their evolution equations lies the search for an appropriate form of the functions  $\mathbf{g}$  and  $\mathbf{h}$ <sup>9</sup>. Here, we identify the latter as the neural networks of a standard autoencoder<sup>10</sup>, being the operator  $\mathbf{h}_\theta(\xi)$  the encoder and the operator  $\mathbf{g}_\theta(z)$  the decoder.

While autoencoders allow the reconstruction of the internal coordinates and their rates, a full reconstruction may not be needed to characterize the material response (see paragraph 3.3). Alternatively, one may prefer other dimensionality reduction techniques, e.g. principal component analysis, among others [Gér19, PMSJ22]. The proposed approach is general and independent on the particular choice made to identify the latent representations of the internal coordinates.

Figure 7 shows the architecture for the data-driven identification of the internal variables and evolution equations. The architecture is composed of an autoencoder, the free energy density network, and the evolution equation network. The identification of the internal variables is driven by the minimization of the reconstruction loss,

$$\mathcal{L}_{\text{recon}} = \|\xi - \mathbf{g}_\theta(\mathbf{h}_\theta(\xi))\|, \quad (22)$$

which ensures that the autoencoder can reconstruct the internal coordinates from the latent representations. Note that, at the end of the training, these latent representations,  $\mathbf{h}_\theta(\xi)$ , will coincide with the internal variables, i.e.,  $z \equiv \mathbf{h}_\theta(\xi)$ . The identification of the evolution equations follows depending on whether a discrete- or continuous-time formulation is adopted, see [MS23].

It is worth noticing that, the training of the networks composing TANN can also be performed individually. In this case, the autoencoder and the free energy density network can be first trained together, by minimization of the weighted sum of the losses  $\mathcal{L}_{\text{recon}}$ ,  $\mathcal{L}_\psi$ ,  $\mathcal{L}_{\nabla\psi}$ , and  $\mathcal{L}_{\text{reg}}$ , then, the evolution equation network is trained by minimization of the loss  $\mathcal{L}_{\dot{z}}$ .

### 3.2.2 Inference

After having trained TANN, they can be used at inference as classical constitutive models that can further be deployed in displacement-based formulations, where loading paths are expressed in terms of strain and temperature loads, i.e.

$$\mathbf{P}(t) = \text{TANN}_\theta(T(t), \mathbf{F}(t), z(t)) \quad \forall t, \quad (23)$$

---

<sup>9</sup>We implicitly assume the existence of such functions, but this might not be always the case (see differential inclusions).

<sup>10</sup>In their simplest form, autoencoders are unsupervised learning algorithms that map inputs to latent representations of minimum dimensionality and then back to themselves. Given an input  $\xi \in \mathbb{R}^n$ , we want to learn a latent representation  $z \in \mathbb{R}^l$  – where  $l \ll n$  – which is mapped back into  $\hat{\xi} \in \mathbb{R}^n$  by minimizing  $\|\xi - \hat{\xi}\|$ . This parametrization is implemented by two functions: an encoder that maps  $\xi$  into  $\mathbb{R}^l$  and a decoder that performs the opposite transformation.

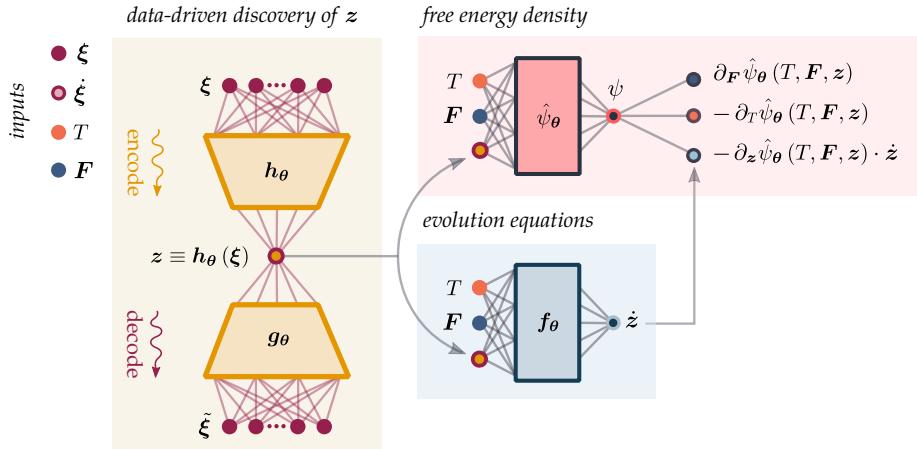


Figure 7: Evolution TANN for the data-driven identification of internal variables and governing equations. All quantities refer at time  $t$ . The internal variables are identified as the latent representations of the encoder, i.e.,  $z \equiv h_\theta(\xi)$ . The internal variables rate can be computed either in terms of finite-difference or by means of the automatic differentiation of the encoder with respect to the internal coordinates, i.e.  $\dot{z} = \partial_\xi h_\theta(\xi) \cdot \dot{\xi}$ .

where the time evolution of the internal variables is computed by solving the following initial value problem

$$\dot{z}(t) = f_\theta(T(t), F(t), z(t)), \quad z(t_0) = z_0. \quad (24)$$

The latter equation can be solved using the artillery of well-known and studied numerical integration methods. Additionally, an alternative path consists of directly learning the integral form of the evolution equations [ME23].

It is also worth noticing that, in the case where one opts for a stress-based formulation, the proposed approach and formalism of TANN can still be used, but selecting the Gibbs free energy density instead of the Helmholtz free energy density.

For the case of a priori selected internal variables, only the evolution equation and free energy networks are needed. Instead, in the case where the internal variables are discovered, these are conclusively identified as the latent representations of the encoder, which is then removed completely from the architecture. However, the decoder may be kept to map  $z$  back to the internal coordinates, i.e.  $\tilde{\xi}(t) = g_\theta(z(t))$ , if the particular application at hands requires it.

In summary, TANN enable robust predictions of material responses, even for unseen data and in the presence of noise, thanks to the hardwiring of thermodynamic principles. Extensive studies on the benefits of a thermodynamics-based approach against a purely data-driven one can be found in [MSVMB21]. For the wide applicability

of the framework, we refer to [MS23] where TANN are demonstrated to accurately model a broad spectrum of complex material behaviors, from plasticity to damage and viscosity (and combination of them).

### 3.3 Digital twins of granular materials: a pedagogic example

Let us investigate, in the form of an example, how the framework of thermodynamics-based neural networks can be used to build high-fidelity thermodynamics-based digital twins of a granular system. To this end, we rely on high-fidelity replicas of the latter (i.e., a virtual twin) to generate virtual experiments that will be used in the learning process.

#### 3.3.1 Virtual twin

The reference granular material is modeled by infinitely rigid, spherical discrete particles, relying on the Discrete Element Method (DEM). The particles interact with each others through inter-particle constitutive laws of friction and their motion is governed by Newton second law of motion. The numerical analyses are conducted using the open-source platform YADE-Open DEM [KD08, SAC<sup>+</sup>21] where the equations of motion are integrated in time explicitly, using a central finite difference approximation algorithm.

Herein, we consider a purely frictional (cohesionless) granular medium, where the grains interact with each other through normal and tangential forces. The grains have a mean diameter  $d_{50} = 200 \mu\text{m}$ , drawn from a uniform distribution  $d_{50} \sim \mathcal{U}(1 - 0.0707, 1 + 0.0707)$ , and are composed of an isotropic elastic material with density  $\rho = 1800 \text{ kg/m}^3$ , Young modulus  $E = 300 \text{ MPa}$ , and poisson ratio  $\nu = 0.3$ . The grains interact through Coulomb friction interfaces, with friction angle  $\varphi = 30^\circ$ .

The granular packing is first subjected to consolidation to match prescribed stress conditions and then undergoes monotonous and cyclic drained triaxial compression. Accordingly, we first generate a periodic specimen by randomly depositing spherical particles. At the beginning, the material is prescribed with a frictional angle equal to  $0^\circ$  in order to obtain an ultimate dense packing. We proceed with an isotropic compaction of the packing to achieve a dense state and a consolidation to reach the desired homogeneous volumetric stress level,  $p = 100 \text{ kPa}$ , and zero deviatoric stress,  $q = 0 \text{ kPa}$ . The former is defined as  $p = \frac{1}{3}\sigma_{ii}$ , while the latter as the invariant of the deviatoric stress tensor,  $\sigma'_{ij} = \sigma_{ij} - p$ , namely  $q = \sqrt{2\sigma'_{ij}\sigma'_{ij}}$  – with  $i, j = 1, 2, 3$ . The final material properties are then assigned to the particles, along with their existing interactions. At this point, the packing is subjected to isotropic extension to generate states at different confining pressures, i.e.,  $p \in (20, 40, 60, 80, 100) \text{ kPa}$ .

**Virtual experiments** Using the consolidated granular packing obtained following the aforementioned protocol, we perform virtual experimental tests to generate data necessary for training, validation, and testing of TANN, and namely:

- five monotonous drained triaxial compression tests with initial confining pressures  $p \in (20, 40, 60, 80, 100)$  kPa up to a total axial deformation approximately equal to 20%, with constant strain rate equal to  $125 \text{ s}^{-1}$ ;
- 10 cyclic drained triaxial compression tests with initial confining pressure equal to 100 kPa, where the strain rates are assumed piecewise constant, in time, equal to  $\pm 125 \text{ s}^{-1}$  and of random direction.

To simulate these conditions, we implement the model by setting  $\dot{\varepsilon}_{11}$  and determining, through a servo-controller,  $\dot{\varepsilon}_v = \dot{\varepsilon}_{ii}$  at each timestep to satisfy  $\dot{p} = \dot{q}/3$ , see Figure 8. The inertia number  $I = \dot{\gamma}d_{50}\sqrt{\rho/p_c}$ , with  $\dot{\gamma}$  the shearing rate, is approximatively equal to  $10^{-5}$  in all tests. Accordingly, the material is in a quasi-static regime, with negligible strain-rate effects [MiD04, RD11].

**Stochastic representation** Granular systems are characterized by inherent spatial heterogeneity of the material properties and of the microstructure topology. As a result, the determination of the size of a representative volume element is not trivial due to the subsequent topological randomness, see [Ngu21, SAS04].

An alternative way consists of resorting to statistical ensemble approaches that leverage stochastic elementary volumes [OS06], containing a sufficient number of particles to be representative and having characteristic size exceeding that of individual grains. Following this framework, analyses based on the stochastic interpretation of multiple small, periodic stochastic elementary volumes, of characteristic size  $l$ , lead, after averaging, to the identification of the representative volume element, with characteristic size  $L$ , see Figure 8 and [PSS23]. To determine the number of stochastic elementary volumes necessary for an adequate description of the material response, Monte-Carlo analyses must be performed, cf. [PSS23].

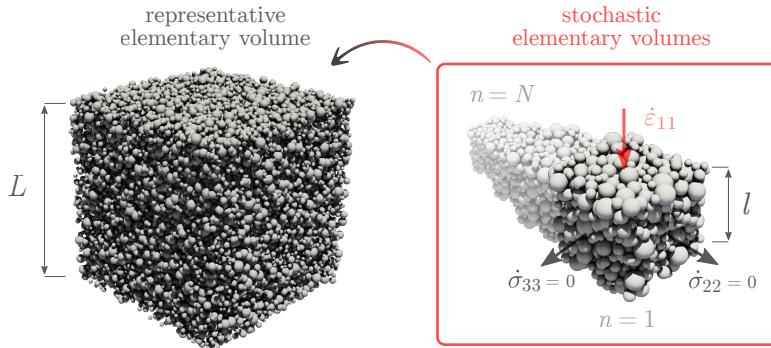


Figure 8: (re-adapted from [PSS23]) The material behavior of a granular system in terms of the representative elementary volume of size  $L$ , approximated by a stochastic ensemble of a series of  $N$  stochastic elementary volumes of size  $l$ , at the level of the microstructure.

Herein, we shall limit ourselves to only four different volumes, with size  $l \approx 8d_{50}$ , composed of  $2^{10}$  particles – the number of particles is kept at minimum in order to generate data in a short amount of time. The number of stochastic elementary volumes is obviously not enough for having realist representations of the heterogeneous nature of granular systems (as Monte-Carlo analyses would reveal). Yet, the statistical average of the four responses of the DEM models leads to a smooth material behavior, see Figure 9, not dominated by *local stick-slip* motion [PSS23] and representative enough of the response of a granular system within the limits of this pedagogic example.

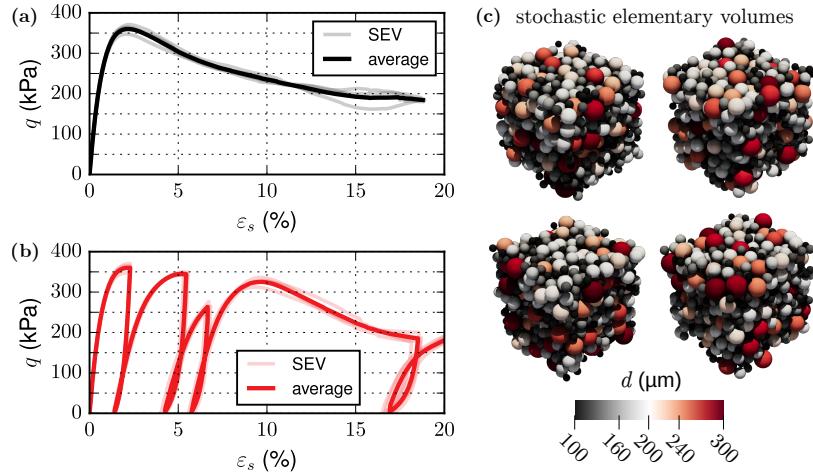


Figure 9: Monotonous and cyclic drained triaxial compression response (a), (b) of each individual stochastic elementary volume and averaged, where  $\varepsilon_s$  is the invariant of the deviatoric strain tensor,  $\varepsilon'_{ij} = \varepsilon_{ij} - \frac{\varepsilon_u}{3}$ , namely  $\varepsilon_s = \sqrt{\frac{1}{2}\varepsilon'_{ij}\varepsilon'_{ij}}$ . (c) Spatial distribution of the grains diameter in each stochastic volume ( $d_{50} = 0.2$  mm).

### 3.3.2 Digital twin

Relying on the aforementioned generated virtual experiments, we proceed with the training of a digital twin based on TANN, using a discrete-time formulation (cf. [MS22]). In particular, we select as test set one monotonous (at initial confining pressure  $p = 80$  kPa) and one cyclic path, from those previously generated. The remaining data are shuffled and split into training and validation set.

The internal coordinates are selected to be the three-dimensional displacement fields of all grains composing the system (averaged over the stochastic elementary volumes). The free energy density network consists of one hidden layer with GELU activations. The evolution equation network has three hidden layers with GELU activations. While the encoder and decoder networks are composed of two hidden layers and RELU activations.

The learning process is split in two phases. First, the energy network and the autoen-

coder are trained by minimizing the reconstruction error (22), the error on the stress predictions, i.e.,  $\|\sigma - \partial_\varepsilon \hat{\psi}_\theta(\varepsilon, \mathbf{h}(\xi))\|$ , and the dissipation inequality, rather than the total loss function (19). Such a choice is made to mimic real case scenarios where the energy and dissipation rate values might not be readily available. Second, after having identified the internal variables, according to Eq. (17), we proceed by training the evolution equation network.

It is worth mentioning that the number of internal variables necessary for an accurate reconstruction of the internal coordinates – i.e., the microscopic displacement fields – and the description of the evolution of the material stresses is not a priori known. To this end, we set the number of hidden dimensions (output of the encoder) equal to  $2^6$  and use an activity regularization based on the  $\ell_1$  norm to have an accurate low-rank reconstruction  $\xi$  with the smallest number of hidden variables. By doing so, we find that the autoencoder can accurately reconstruct the internal coordinate fields with only seven (non-zero) hidden variables – that is, compressing the information of the three-dimensional displacement fields of  $2^{10}$  particles by a factor of approximately 440 ( $32^{10}/7$ ).

Despite such compression, we should note that not all the so-determined hidden variables may be needed for characterizing the material response. Thus, we introduce, in parallel, a penalty based on the  $\ell_1$  norm for the weights of input layer of the free energy network (cf. Chapter 2). This allows to promote parsimony and identify only those (internal) variables that are effectively needed for predicting the material response. In doing so, we find that only four internal variables are necessary for the stress paths probed in this example.

### 3.3.3 Results

We show in Figure 10 the evolution of the volume average behavior of the reference granular material and the predictions, at inference, of TANN, for both the monotonous and the cyclic unseen path. The digital twin is found to accurately describe the complex material behavior, correctly accounting for the underlying inelastic phenomena and yielding an accurate representation of the dissipative nature of the system, subjected to multiple loading and unloading.

Particularly interesting is to observe the evolution of the discovered internal variables, shown in Figure 11, for the same monotonous and cyclic test paths. There exist two essentially different behaviors. The first two internal variables are mainly responsible for the description of the irreversible phenomena taking place before the stress peak, see Figure 11(b,e). This is particularly evident for the monotonous path (b), where the variables rapidly evolve until they reach an equilibrium state (post-peak). A similar behavior is also found for the cyclic path (e), where the multiple unloading and (re)loading repeatedly perturb the equilibrium state. The remaining two internal variables, whose evolution is depicted in Figure 11(c,f), display a behavior essentially different from that of the former as they continuously evolve, both in loading and unloading, suggesting that they may represent a macroscopic measure of dissipation mechanisms taking place at the frictional inter-particle contacts.

Note that the physical nature of the discovered internal state variables can be examined through feature extraction methods [LKS20].

Additionally, internal variables can be decoded to obtain the internal coordinates of the microstructure – the displacement fields of the grains. To demonstrate the full capabilities of TANN, we show such reconstruction in Figure 12 for the monotonous test set. The three-dimensional reconstruction of the granular packing in Figure 12(a) is made by using the predicted particles displacements (and accounting for the diameter distribution and initial grains position of one of the four stochastic elementary volumes). Figure 12(b) additionally compares the reference and predicted time evolution of the displacements of some of the grains. The framework allows to deliver accurate descriptions of the material microstructure. Note, again, that the proposed approach does not require to operate (on-the-fly) with the high-dimensional microscopic fields, but only with the identified hidden variables.

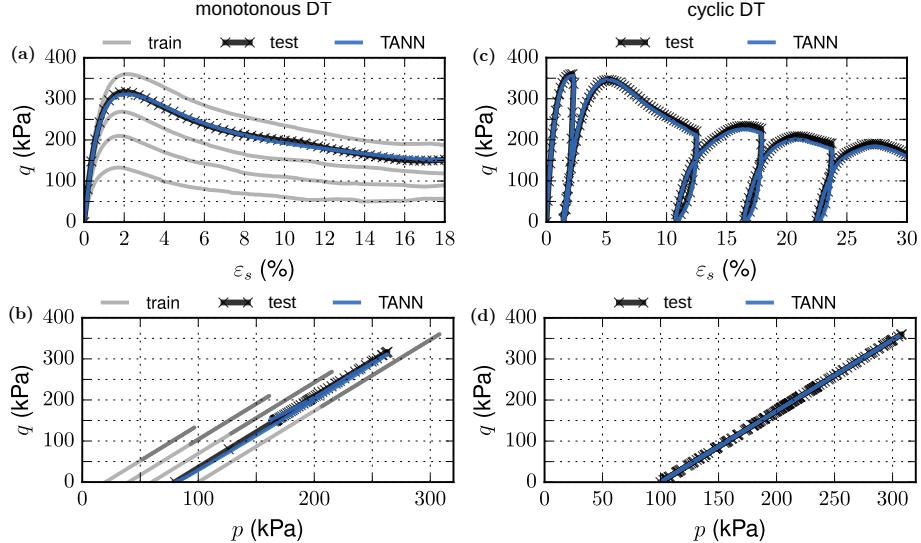


Figure 10: Comparison of the predictions of TANN with respect to unseen, test responses obtained from the statistical average of four granular stochastic elementary volumes: monotonous drained triaxial (DT, left) and cyclic (right). (a), (c)  $q - \varepsilon_s$  response and (b), (d)  $p - q$  response. For the monotonous path, we show both the training and validation sets (train), and test (test) set.

### 3.4 Speed-up multiscale simulations

We have seen so far how TANN can be deployed to construct reliable and accurate digital twins of complex and intricate materials, thanks to the universal approximation power provided by neural networks and the rigorous theoretical setting offered by

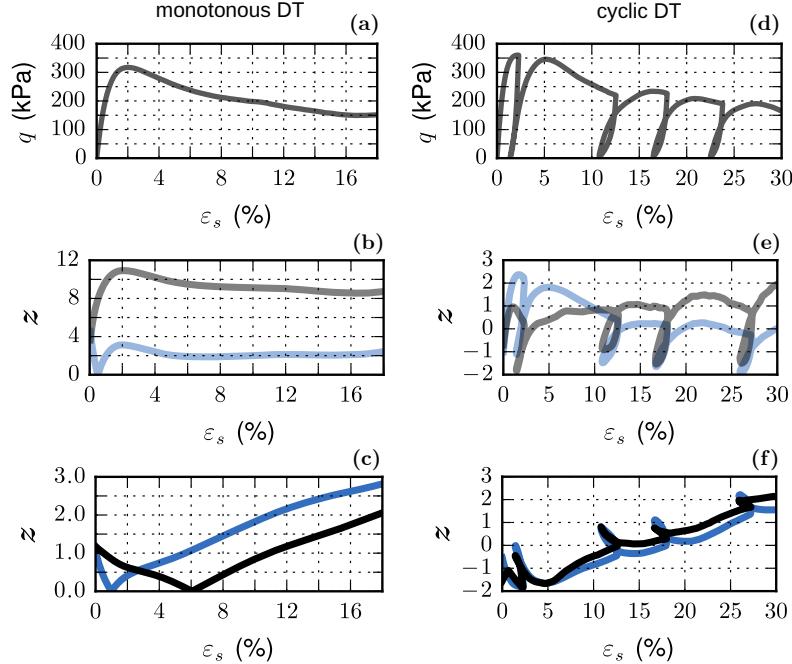


Figure 11: Evolution of the discovered internal variables for the test monotonous (left) and cyclic (right) loading paths: (a,d) reference deviatoric stress path, and (b-c,e-f) discovered internal state variables.

thermodynamics. In addition, the same framework can be used to speed-up, in a reliable manner, fine-scale simulations, relying on a multiscale approach.

Multiscale approaches integrate a number of nested computational methods at various scales, such as the Finite Element Method (FEM) and the Discrete Element Method (DEM), to obtain iterative solutions to boundary value problems at the unit cell of the microstructure (*auxiliary problem*) and, through upscaling, retrieve the effective macroscopic response [Fey03, NCDD11, NCCD14].

To fix the ideas, let us consider a large scale, macroscopic structure, whose microstructure is made of spatially (quasi-)periodic distributions of a representative unit cell  $\mathcal{V}$ . We leverage scale separation, i.e., we assume the existence of two independent scales  $\mathbf{x}$  and  $\mathbf{y} = \mathbf{x}/\epsilon$ , with  $\epsilon \ll 1$  being the dimension of the unit cell. The first scale,  $\mathbf{x}$ , is associated to the macroscale, while  $\mathbf{y}$  refers to the scale of the material microstructure. Under such considerations, we can resort to the rigorous mathematical framework of *asymptotic homogenization* [BP12] for scales bridging and, in particular, by leveraging its extension to nonlinear problems by means of an incremental formulation, see [Mie02].

Note that whenever the assumption of scale separation does not hold, for in-

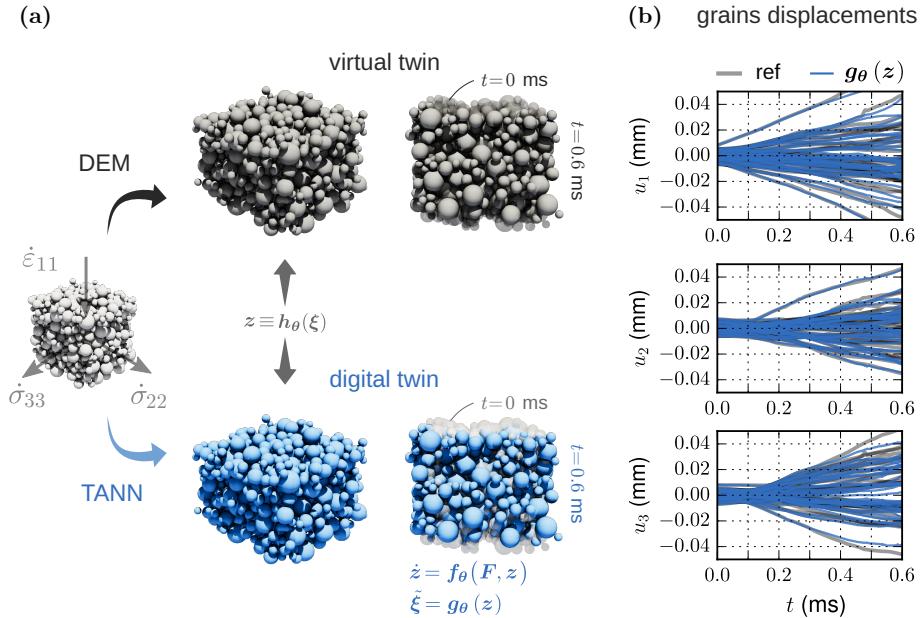


Figure 12: Internal coordinates (particles displacements) from the DEM simulations and as reconstructed from the set of discovered internal variables. (a) Reconstruction for one of the stochastic volume of the granular packing from the knowledge of the displacement fields, radii distribution, and initial position. (b) Comparison of the time evolution of (some of) the grains displacements and the predictions obtained by integrating in time the evolution equation,  $f_\theta(\cdot)$ , and decoding the internal variables,  $g_\theta(\cdot)$ .

stance, in presence of strain localization phenomena, homogenization cannot be used. However, a remedy consists of resorting to higher order continuum theories, see [SSV10, GSS<sup>+</sup>17, Var18] among others.

According to asymptotic homogenization, TANN should be trained to identify the volume average behavior of the unit cell – that is, predict the solution of the auxiliary problem. At this point, we should note that the previous example already considers training data sets related to the unit cell (of a granular material) with periodic boundary conditions. Thus the network can be directly used at inference to perform multiscale analyses. To do so, we rely on the FEM×TANN approach, as developed in [MS22]. In FEM×TANN, we perform Finite Element analyses by a straightforward replacement of classical constitutive models, at the Gauss points, with the digital twin provided by the trained network (cf. asymptotic homogenization). The tangent matrix is computed, at each Gauss integration point, by virtue of the automatic differentiation of TANN, cf. Chapter 7. For a detailed discussion on the computational accelerations available with the FEM×TANN approach, we refer to [MS22].

### 3.4.1 FEM×TANN: an example

To fix the ideas, let us consider the problem of a panel, fixed on one end, and subjected, quasi-statically, to a shearing load at the other end. Figure 13(a) depicts the initial geometry and boundary conditions. Plane strain conditions are adopted.

Initially, a force equal to 12 N/mm (per unit cell in width) is applied linearly in time until  $t_{\text{load}} = 0.9$  ms. The loading time is selected in order to have strain rates of the order of  $1 \text{ s}^{-1}$ , corresponding to those developed due to impact loading. The force is then maintained over a prescribed time interval, up to  $t_{\text{steady}}$ , and, finally, the structure is unloaded gradually within a time interval equal to that of the loading phase, see Figure 13(e). To capture relaxation effects due to the viscous behavior of the microstructure, the unloaded phase is maintained for an additional time interval, up to  $t_{\text{rel}}$ .

Here, the microstructure is not that of a granular material, but rather the one of an elasto-plastic lattice material whose microstructure is made of bars with an elasto-viscoplastic behavior and isotropic hardening (with Perzyna-type viscosity), see Figure 13(b). The material has the following properties  $K = 167$  GPa,  $G = 77$  GPa,  $c = 100$  MPa,  $H = 10$  MPa and  $\mu = 25$  s, where  $K$  and  $G$  are the bulk and the shear modulus,  $c$  is the material strength in simple shear,  $H$  the hardening modulus, and  $\mu$  the viscosity parameter. The microstructural bars have a constant circular cross-section equal to  $1 \text{ mm}^2$ .

Following the procedure detailed in [MS23], the digital twin of the lattice cell is built using TANN, trained over randomly generated strain-driven paths applied in the form of periodic boundary conditions. In total, 33 internal variables are identified from the microscopic total and inelastic deformation fields.

As it follows, we investigate and compare the solutions obtained from the (exact) micromechanical model with those from the homogenized one, relying on TANN. For the homogenized case, we consider a FE model consisting of 80 linear tetrahedral elements in length and 8 in height (with crossed diagonals) – the number of elements was determined by mesh convergence analyses.

First, we restrict the analysis exclusively to the loading phase and verify that the micromechanical solution converges to the homogenized one, for  $\epsilon$  tending to zero, as expected from asymptotic homogenization [Mie02]. To this end, we compare the results of the micromechanical model, with different sizes of the unit-cell  $\epsilon$ , with those obtained with the FEM×TANN approach. Figure 14(a,b) depicts the total energy and dissipation rate in function of  $\epsilon$ . For  $1/\epsilon \geq 6$ , the maximum relative error is as low as 1.7% (in energy) and 2.3% (in dissipation). The convergence of the response of the homogenized model to the response of the micromechanical simulations is achieved at  $\epsilon = 0.1$  (with an error approximately equal to 0.5%).

Then, we continue by considering the full time scale of the analysis and select the microscopic model with  $\epsilon = 0.1$  as the reference solution. Figure 13(c,d) depicts the deformed shapes of the micromechanical and homogenized models (magnified by a factor of 10), at the end of the steady phase. Contours of the free energy density in the

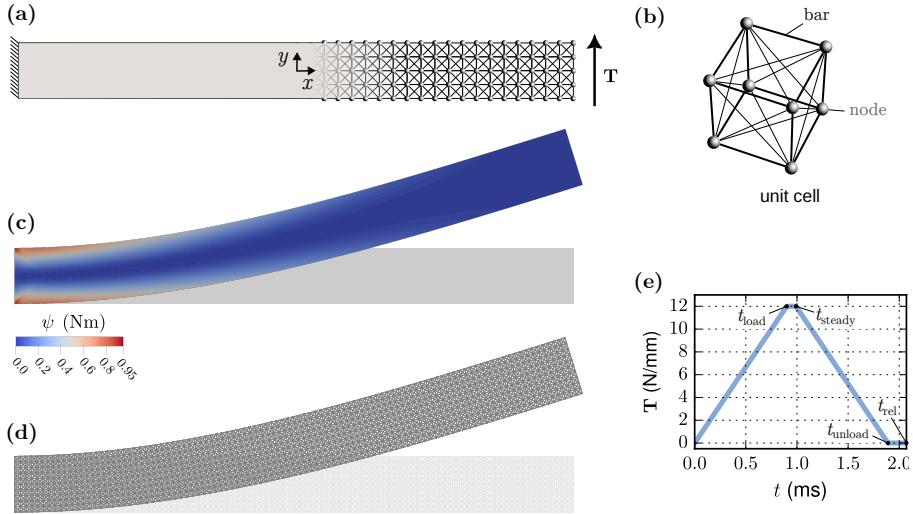


Figure 13: Multiscale problem: (a) homogenized and micromechanical model of a panel, with length 10 mm and height 2 mm, under plan strain conditions, subjected to a shearing load and (b) force-displacement response of the micromechanical ( $\epsilon = 0.1$ ) and the FEM×TANN model.

homogenized model identify the areas where high stresses develop.

As far as it concerns the kinematics of the problem, we compare, in Figure 14(c), the force-displacement response of the homogenized model with the micromechanical reference solution. For the former, we consider the (zeroth-order) approximation of the displacements, while for the latter, the microscopic displacements at the nodes are considered as reference. A good agreement between the two models is observed. During the steady phase, the material displays viscous effects that alter the displacements and deformation fields. This is captured by the homogenized model with a relative error in the displacement of 0.25%. In addition, the model correctly predicts the residual (non-zero) vertical displacement, cf. Figure 14(c).

In parallel, TANN enable the characterization of microscopic fields, specifically the internal coordinates  $\xi$ , within the homogenized model, known as *localization* in asymptotic homogenization (not to be confused with strain localization). Unlike the classical procedure [PdCOTD09], the proposed approach allows straightforward reconstruction of microscopic fields by decoding the internal variables.

Figure 15 compares the microscopic deformations at the intrados of the panel in the micromechanical model ( $\epsilon = 0.1$ ) with those obtained from the decoding of the internal variables. An excellent agreement is observed. The FEM×TANN approach accurately captures deformation redistribution during the steady phase caused by viscosity. Additionally, it provides not only overall agreement with the micromechanical

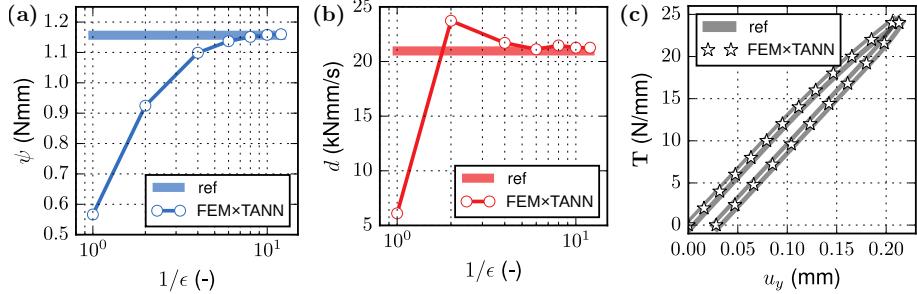


Figure 14: Total energy (a) and dissipation rate (b) of the micromechanical (ref) and homogenized (FEMxTANN) models ( $t = t_{\text{load}}$ ), at varying of the unit cell size,  $\epsilon$ .

solution for microscopic inelastic deformations but also precise prediction of the plastic region location.

Furthermore, the proposed method successfully captures complex micromechanical mechanisms, such as the phenomenon of *trapped*, locked elastic energy. At the end of the unloading phase, the inelastic deformation exceeds the total microscopic deformation (cf. Figure 15). Elastic deformations remain trapped within the plastified region of the structure, from the clamped end to the end of the plastic region ( $x \in [-5, 1.625]$  mm), persisting even without external force. Conversely, for  $x > 1.625$  mm, total and inelastic deformations are zero, and elastic deformations vanish during unloading.

It is worth noticing that predictions near the fixed end ( $x = -5$  mm) of the panel exhibit minor differences compared to the micromechanical solution due to high strain gradients with wavelengths comparable to the unit cell size. Remedies for these boundary layer effects exist (see e.g. [BP12]). Nonetheless, despite the limitations of first-order asymptotic homogenization theory, the FEMxTANN approach yields excellent results compared to the micromechanical problem and holds potential for higher-order homogenization schemes.

## 4 Conclusions

Deep learning and, in general, data-intensive (or data-centric) approaches have the potential of revolutionizing the way we perceive models in physics and in mechanics. However, a purely data-centric vision is intrinsically hindered by poor generalization and interpretability of the consequent discovered models.

Particularly inspiring, in recent years, has been the possibility of hardwiring (in soft and hard ways) first principles into deep learning algorithms and learn interpretable, robust, and high-fidelity models that respect the inherent physics [KKL<sup>+</sup>21, HBG<sup>+</sup>21, MSVMB21, KFM<sup>+</sup>22, CC22].

In this chapter, we presented two emblematic methodologies, namely Physics-Informed Neural Networks (PINN) and Thermodynamics-based Artificial Neural Net-

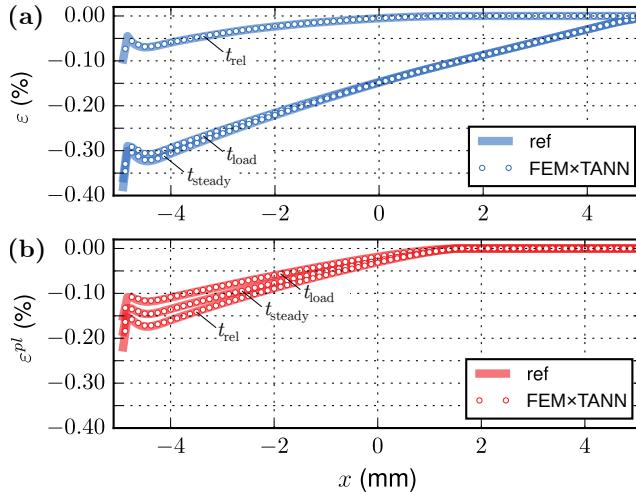


Figure 15: Microscopic deformations (total and inelastic) at the intrados of the panel at the end of the loading, steady, and unloading plus relaxation phase. In the micromechanical model ( $\epsilon = 0.1$ ), the microscopic deformations are computed from the deformations of the bars with axis parallel to the intrados. In the homogenized model, the microscopic deformations are obtained from the decoding of the identified internal variables at the Gauss points (no interpolation).

works (TANN). The former enable accounting for physical laws in the solution and discovery of partial different equations, from data. Physical knowledge is introduced through ad-hoc learning biases that consists in the minimization of a loss function related to the residual of the underlying governing equations. The latter consist, instead, of hardwiring the first and second laws of thermodynamics (i.e., energy and entropy balance) within the architecture of neural networks, via inductive and learning biases, to discover constitutive equations of complex materials, from data.

Relying on an hands-on example, we draw an interesting analogy between the integration of physics into deep learning algorithms and regularization techniques (based on  $\ell_1$  and  $\ell_2$  norms) promoting parsimony and model simplicity (see paragraph 2.2). Then, we demonstrated the possibility of identifying high-fidelity and interpretable digital replicas of complex granular materials relying on the framework of thermodynamics-based neural networks (paragraph 3.3). The latter not only capture the average material behavior but also allow to uncover the (hidden) internal state variables and efficiently track the microstructure evolution at reduced computational cost. Finally, the same approach can be deployed to accelerate computationally intensive multiscale simulations, herein exemplified by means of an application to lattice structures (paragraph 3.4).

The aforementioned examples and applications demonstrate the underlying idea of TANN in learning the constitutive behavior of intricate materials and achieving rapid

predictions at inference, thus allowing for massive accelerations of multiscale simulations. For instance, with reference to the example involving granular materials, it was observed that the computational time required by TANN to forecast the material response under a cyclic drained triaxial compression test was roughly three orders of magnitude lower compared to the extensive fine-scale simulations carried out using the discrete element method, specifically 1200/0.8 s/s.

In conclusion, the combination of physical principles with data-centric models offers a pathway to tame and describe, with high-accuracy, the complex response of intricate systems. Yet, some challenges remain to be addressed. Indeed, in mechanics, we are most of the times faced, rather than with vast amounts of data (*big data*), with very limited volumes of measurements (*small data*), characterized by an inherent heterogeneity and noise. Particular interesting is the development of physics-based deep learning algorithms that can address such a problem and deliver robust predictive models able to operate with reduced/partial information.

In addition, machine learning approaches can be employed to automatically determine meaningful variables of the system at hands and discover novel formulations, as we have seen in the example involving granular media (cf. paragraph 3.3). The same approach, driving the discovery of hidden latent (internal) variables, could be extended to address open challenges related to the choice of the descriptors of the state space of complex materials or any other physical system.

## Acknowledgements

The author F.M. would like to acknowledge Prof. Itai Einav for the fruitful discussions which have greatly enriched this chapter.

The authors would like to acknowledge the support of the European Research Council (ERC) under the European Union Horizon 2020 research and innovation program (Grant agreement ID 757848 CoQuake).

## References

- [BP12] NS Bakhvalov and G Panasenko. *Homogenisation: averaging processes in periodic media: mathematical problems in the mechanics of composite materials*, volume 36. Springer Science & Business Media, 2012.
- [BPRS18] AG Baydin, BA Pearlmutter, AA Radul, and JM Siskind. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18:1–43, 2018.
- [CC22] E Cueto and F Chinesta. Thermodynamics of learning physical phenomena. *arXiv preprint arXiv:2207.12749*, 2022.

- [CCAC<sup>+</sup>20] F Chinesta, E Cueto, E Abisset-Chavanne, JL Duval, and FE Khaldi. Virtual, digital and hybrid twins: a new paradigm in data-based engineering and engineered data. *Archives of computational methods in engineering*, 27:105–134, 2020.
- [CDCG<sup>+</sup>22] S Cuomo, VS Di Cola, F Giampaolo, G Rozza, M Raissi, and F Piccialli. Scientific machine learning through physics-informed neural networks: where we are and what’s next. *Journal of Scientific Computing*, 92(3):88, 2022.
- [CG67] BD Coleman and ME Gurtin. Thermodynamics with internal state variables. *Journal of Chemical Physics*, 47(2):597–613, 1967.
- [Daf22] YF Dafalias. Split stress rate plasticity formulation. *International Journal of Solids and Structures*, page 111494, 2022.
- [EHN07] I Einav, GT Housby, and GD Nguyen. Coupled damage and plasticity models derived from energy and dissipation potentials. *International Journal of Solids and Structures*, 44(7-8):2487–2508, 2007.
- [Ein12] I Einav. The unification of hypo-plastic and elasto-plastic theories. *International Journal of Solids and Structures*, 49(11-12):1305–1315, 2012.
- [EL18] I Einav and M Liu. Hydrodynamic derivation of the work input to fully and partially saturated soils. *Journal of the Mechanics and Physics of Solids*, 110:205–217, 2018.
- [Fey03] F Feyel. A multilevel finite element method ( $FE^2$ ) to describe the response of highly non-linear structures using generalized continua. *Computer Methods in Applied Mechanics and Engineering*, 192(28-30):3233–3244, 2003.
- [Gér19] A Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O’Reilly Media, 2019.
- [GS98] J Ghaboussi and DE Sidarta. New nested adaptive neural networks (NANN) for constitutive modeling. *Computers and Geotechnics*, 22(1):29–52, 1998.
- [GSS<sup>+</sup>17] M Godio, I Stefanou, K Sab, J Sulem, and S Sakji. A limit analysis approach based on Cosserat continuum for the evaluation of the in-plane strength of discrete media: Application to masonry. *European Journal of Mechanics-A/Solids*, 66:168–192, 2017.
- [Gur96] ME Gurtin. Generalized ginzburg-landau and cahn-hilliard equations based on a microforce balance. *Physica D: Nonlinear Phenomena*, 92(3-4):178–192, 1996.

- [HBG<sup>+</sup>21] Q Hernandez, A Badias, D Gonzalez, F Chinesta, and E Cueto. Deep learning of thermodynamics-aware reduced-order models from data. *Computer Methods in Applied Mechanics and Engineering*, 379:113763, 2021.
- [HTT<sup>+</sup>09] AJG Hey, S Tansley, KM Tolle, et al. *The fourth paradigm: data-intensive scientific discovery*, volume 1. Microsoft research Redmond, WA, 2009.
- [KB22] JN Kutz and SL Brunton. Parsimony as the ultimate regularizer for physics-informed machine learning. *Nonlinear Dynamics*, 107(3):1801–1817, 2022.
- [KD08] J Kozicki and FV Donze. A new open-source software developed for numerical simulations using discrete modeling methods. *Computer Methods in Applied Mechanics and Engineering*, 197(49-50):4429–4443, 2008.
- [KFM<sup>+</sup>22] DK Klein, M Fernández, RJ Martin, P Neff, and O Weeger. Polyconvex anisotropic hyperelasticity with neural networks. *Journal of the Mechanics and Physics of Solids*, 159:104703, 2022.
- [KKL<sup>+</sup>21] GE Karniadakis, IG Kevrekidis, L Lu, P Perdikaris, S Wang, and L Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.
- [LBS09] M Lefik, DP Boso, and BA Schrefler. Artificial neural networks in numerical modelling of composites. *Computer Methods in Applied Mechanics and Engineering*, 198(21-26):1785–1804, 2009.
- [LJP<sup>+</sup>21] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deepnet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021.
- [LKS20] PY Lu, S Kim, and M Soljačić. Extracting interpretable physical parameters from spatiotemporal systems using unsupervised learning. *Physical Review X*, 10(3):031056, 2020.
- [LP21] R Leiteritz and D Pflüger. How to avoid trivial solutions in physics-informed neural networks. *arXiv preprint arXiv:2112.05620*, 2021.
- [LS03] M Lefik and BA Schrefler. Artificial neural network as an incremental non-linear constitutive model for a finite element code. *Computer methods in applied mechanics and engineering*, 192(28-30):3265–3283, 2003.
- [Mau15] GA Maugin. The saga of internal variables of state in continuum thermo-mechanics (1893–2013). *Mechanics Research Communications*, 69:79–86, 2015.

- [ME23] F Masi and I Einav. Physics-based neural constitutive equations. In *Hybrid Computational Methods in Geotechnics*. The University of Melbourne, 2023.
- [MiD04] GDR MiDi. On dense granular flows. *The European Physical Journal E*, 14:341–365, 2004.
- [Mie02] C Miehe. Strain-driven homogenization of inelastic microstructures and composites based on an incremental variational formulation. *International Journal for Numerical Methods in Engineering*, 55(11):1285–1322, 2002.
- [MM94] GA Maugin and W Muschik. Thermodynamics with internal variables. Part I. General concepts. *Journal of Non-Equilibrium Thermodynamics*, 19:217–249, 1994.
- [MS22] F Masi and I Stefanou. Multiscale modeling of inelastic materials with Thermodynamics-based Artificial Neural Networks (TANN). *Computer Methods in Applied Mechanics and Engineering*, 398:115190, 2022.
- [MS23] F Masi and I Stefanou. Evolution TANN and the identification of internal variables and evolution equations in solid mechanics. *Journal of the Mechanics and Physics of Solids*, 174:105245, 2023.
- [MSVMB21] F Masi, I Stefanou, P Vannucci, and V Maffi-Berthier. Thermodynamics-based Artificial Neural Networks for constitutive modeling. *Journal of the Mechanics and Physics of Solids*, 147:104277, 2021.
- [NCCD14] TK Nguyen, G Combe, D Caillerie, and J Desrues. FEM × DEM modelling of cohesive granular materials: Numerical homogenisation and multi-scale simulations. *Acta Geophysica*, 62(5):1109–1126, aug 2014.
- [NCDD11] M Nitka, G Combe, C Dascalu, and J Desrues. Two-scale modeling of granular materials: a DEM-FEM approach. *Granular Matter*, 13(3):277–281, 2011.
- [Ngu21] T-K Nguyen. On the Representative Volume Element of dense granular assemblies made of 2D circular particles. In *Structural Health Monitoring and Engineering Structures: Select Proceedings of SHM&ES 2020*, pages 499–508. Springer, 2021.
- [OS06] M Ostoja-Starzewski. Material spatial randomness: From statistical to representative volume element. *Probabilistic engineering mechanics*, 21(2):112–132, 2006.
- [PdCOTD09] J Pinho-da Cruz, JA Oliveira, and F Teixeira-Dias. Asymptotic homogenisation in linear elasticity. Part I: Mathematical formulation

- and finite element modelling. *Computational Materials Science*, 45(4):1073–1080, 2009.
- [PMSJ22] G Piunno, F Masi, I Stefanou, and C Jommi. Multi-scale modelling of natural composites via Thermodynamics-based Artificial Neural Networks. In *Congrès Français de Mécanique (CFM)*. AFM, 2022.
- [PSS23] E Papachristos, I Stefanou, and J Sulem. A discrete elements study of the frictional behavior of fault gouges. *Journal of Geophysical Research: Solid Earth*, 128(1):e2022JB025209, 2023.
- [RD11] F Radjai and F Dubois. *Discrete-element modeling of granular materials*. Wiley-Iste, 2011.
- [RPK19] M Raissi, P Perdikaris, and GE Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.
- [Rub01] MB Rubin. Physical reasons for abandoning plastic deformation measures in plasticity and viscoplasticity theory. *Archives of Mechanics*, 53(4-5):519–539, 2001.
- [SAC<sup>+</sup>21] V Smilauer, V Angelidakis, E Catalano, R Caulk, B Chareyre, W Chèvremont, S Dorofeenko, J Duriez, N Dyck, J Elias, et al. YADE Documentation 3rd ed. *The Yade Project*, 2021.
- [SAS04] M Stroeven, H Askes, and LJ Sluys. Numerical determination of representative volumes for granular materials. *Computer Methods in Applied Mechanics and Engineering*, 193(30-32):3221–3238, 2004.
- [SSV10] I Stefanou, J Sulem, and I Vardoulakis. Homogenization of interlocking masonry structures using a generalized differential expansion technique. *International Journal of Solids and Structures*, 47(11-12):1522–1536, 2010.
- [Ván03] P Ván. Weakly nonlocal irreversible thermodynamics. *Annalen der Physik*, 515(3):146–173, 2003.
- [Var18] I Vardoulakis. *Cosserat continuum mechanics: with applications to granular media*, volume 87. Springer, 2018.
- [VBE08] P Van, A Berezovski, and J Engelbrecht. Internal variables and dynamic degrees of freedom. *Journal of Non-Equilibrium Thermodynamics*, 33(3):235–254, 2008.
- [YZYK22] M Yin, E Zhang, Y Yu, and GE Karniadakis. Interfacing finite elements with deep neural operators for fast multiscale modeling of mechanics problems. *Computer Methods in Applied Mechanics and Engineering*, page 115027, 2022.