arXiv:2507.05291v2 [cs.LG] 13 Oct 2026

PHYSICS-INFORMED GRAPH NEURAL NETWORKS TO RECONSTRUCT LOCAL FIELDS CONSIDERING FINITE STRAIN HYPERELASTICITY

PUBLISHED IN INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
126(24) (2025) E70193
https://doi.org/10.1002/nme.70193

Manuel Ricardo Guevara Garban[1,2*] **Yves Chemisky**[1] **Michaël Clément**[2] **Étienne Prulière**[3]

[1]Univ. Bordeaux, CNRS, Bordeaux INP, I2M, UMR 5295, Talence 33400, France
[2]Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, Talence 33400, France
[3]Arts et Metiers Institute of Technology, CNRS, Bordeaux INP, I2M, UMR 5295, Talence 33400, France

ABSTRACT

We propose a physics-informed machine learning framework called *P-DivGNN* to reconstruct local stress fields at the micro-scale, in the context of multi-scale simulation given a periodic micro-structure mesh and mean, macro-scale, stress values. This method is based in representing a periodic micro-structure as a graph, combined with a message passing graph neural network. We are able to retrieve local stress field distributions, providing average stress values produced by a mean field reduced order model (ROM) or Finite Element (FE) simulation at the macro-scale. The prediction of local stress fields are of utmost importance considering fracture analysis or the definition of local fatigue criteria. Our model incorporates physical constraints during training to constraint local stress field equilibrium state and employs a periodic graph representation to enforce periodic boundary conditions. The benefits of the proposed physics-informed GNN are evaluated considering linear and non linear hyperelastic responses applied to varying geometries. In the non-linear hyperelastic case, the proposed method achieves significant computational speed-ups compared to FE simulation, making it particularly attractive for large-scale applications.

Keywords Graph neural networks · Multi-scale simulation · Machine learning · Architectured materials · Local stress field reconstruction · Physics Informed Neural Networks

1 Introduction

Technological innovation is driving material science to new heights, especially in the development of modern health devices, vehicles, and renewable energy systems. These applications demand lightweight, multi-functional components that maintain mechanical integrity. Additive manufacturing, along with other emerging processing technologies, has significantly expanded the range of achievable architectured geometries, potentially overcoming traditional design constraints. However, designing and optimizing these structures requires extensive simulations to evaluate mechanical responses under various loading conditions. Multi-scale optimization that demands multiple iterations, varying geometry at both the micro and macro-scale as well as material parameters is particularly costly. Further considering non-linear behaviors such as plasticity and non-linear kinematics render such optimization process unrealistic with full-field simulation tools.

*Corresponding author: manuel.guevara-garban@u-bordeaux.fr
Abbreviations: GNN, graph neural networks; FEM, finite element method; ROM, reduced order model.

1.1 Multi-scale simulation

A computationally efficient way to simulate mechanical problems that have two different characteristic scales (or more) lays in the use of homogenization theory. If the scales are well separated, we can split the problem into microscopic and macroscopic parts. These two boundary condition sub-problems are coupled: Considering appropriate boundary conditions, the microscopic problem provides the homogenized properties (mean stress, tangent stiffness matrix) that are required to solve the macroscopic problem. Considering a linearized two-scale finite element (FE) simulation at each time increment, the macroscopic problem requires such information at each integration point. The resolution of the macroscopic problem provides in return the predicted strain that allows to define the appropriate local boundary condition for the microscopic one. This method is referred to as the FE^2 method [1, 2]. Considering linear elastic response, the microscopic problem, which consists of six linear simulations in 3D (and three in 2D) for the determination of the homogenized stiffness matrix, need only to be solved once since the stiffness matrix is time independent. Nevertheless, if the microstructure varies inside the macroscopic domain (e.g., foam with a density gradient), a different homogenization problem needs to be solved at each integration point. Also, considering a design process such as shape optimization, a high number of microscopic simulations may be required to determine an optimal shape. In many cases, homogenized properties are not sufficient in the design process, especially considering the response to strength and fatigue. In these cases, a local criterion based on the stress components is required, so the full local stress field needs to be computed. Note that the computation of full fields comes naturally considering FE^2 methods, which suffers from computational efficiency. More efficient FE^2 have been proposed recently [3], however the computational time is still not compatible with an iterative optimization approach for multi-scale problems.

Prior to the development of FE^2 methods, micro-mechanical approaches have been developed, considering several approaches [4] to link the average local fields of different phases with the global average. Such relation is restricted to specific geometry (i.e., ellipsoidal inhomogeneities), while approximations were developed for other geometries, the field being described as a piecewise field per phase [5].

1.2 Reduced order models

To mitigate the high costs associated with multi-scale simulations and following micro-mechanical approaches, reduced order models (ROM) have been proposed to simplify simulations at the micro-scale level while retaining essential microstructure features and behaviors, such as mean stress values. Such ROM, that have added plastic modes to better represent the local fields of non-linear heterogeneous microstructures, have been developed in [6] and extended in [7] considering the use of Proper Orthogonal Decomposition (POD) to identify those plastic modes. The plastic modes are therefore utilized to better represent strain localization that appear in phases subjected to the development of plastic strains.

AI-based approaches were also developed to drastically cut down computational times [8], at the cost of retaining only the average stress field and tangent modulus.

Another local field estimation has been introduced in [9], referred to has the self-consistent clustering analysis. In this approach, local fields are approximated as a group of clusters with similar mechanical features. This method is very relevant to determine the homogenized response of complex heterogeneous structures without the determination of specific local non-linear deformation modes that are microstructure-dependent.

Nonetheless, all these methods are not dedicated to the precise definition of local fields, but rather focus on satisfying, computational efficient, estimation of the global response of non-linear, multi-scales structures.

1.3 Deep learning methods

Recently, Machine Learning (ML) approaches have been extensively employed for surrogate modeling [10, 11, 12] and predict full fields, not restricted to the specific case of multi-scale modeling. Deep Learning methods, such as Convolutional Neural Networks (CNNs) [13], have been employed to predict full stress fields on 2D geometries, as demonstrated by the framework "StressNet," which utilizes CNNs to estimate stress fields in cantilevered structures [14]. Furthermore [15] proposed to use a U-Net [16] architecture as an end-to-end ML-driven framework to accelerate multi-scale mechanics simulations of heterogeneous macro-structures. The U-Net architecture has also been employed for predicting stress fields in additively manufactured metals with intricate defect networks [17].

Other CNN-based methods have explored the use of generative models, such as Generative Adversarial Networks (GANs) [18] and Diffusion Models [19]. For example, [20] proposed a GAN model for predicting strain and stress tensors in hierarchical composite microstructures. Similarly, [21] used Diffusion Models to estimate stress fields in 2D geometries.

While CNNs have demonstrated success in predicting stress fields in mechanics, their applicability is often limited when extending to unstructured geometries. This limitation stems from the need to use a regular grid (2D or 3D) as input for CNNs, which poses significant challenges when working with meshes that have varying refinement sizes. In such cases, local features at different resolutions are essential for accurate predictions, and grid-based representations struggle to capture these multi-scale dependencies. Point-cloud-based methods such as the deep operator network proposed in [22], have been introduced to address this limitation by leveraging 1D convolutional layers for field predictions on parameterized geometries. Although, point-cloud-based approaches are inherently limited for field predictions on mesh data due to their inability to explicitly capture topological connectivity and local geometric relationships, both of which are essential for accurately representing multi-scale dependencies and intricate features in stress field predictions.

To overcome the limitations of CNNs in simulating unstructured geometries, Graph Neural Networks (GNNs) [23] have gained increasing attention. By operating directly on graph-structured data, such as mesh data, GNNs can effectively capture spatial relationships and multi-scale dependencies inherent in unstructured domains, offering a more flexible approach to stress field prediction. One notable contribution is *MeshGraphNet*, a GNN architecture proposed by [24]. This framework adopts an Encode-Message Passing-Decode architecture, enabling accurate representations of mesh-based simulations. *MeshGraphNet* has become a state-of-the-art approach for predicting scalar fields on mesh data [25, 26, 27].

Despite these advances, integrating physical constraints such as equilibrium conditions and periodic boundary conditions into GNN frameworks remains a challenge. Some exploratory works aim to incorporate physics-informed priors, also known as Physics-Informed Neural Networks (PINNs) [28], into GNNs. For instance, [29] proposed a GNN-based solver for partial differential equations (PDEs) considering boundary conditions, while [30] introduced a thermodynamics-informed GNN to predict the temporal evolution of dissipative dynamic systems. However, enforcing equilibrium constraints on predicted stress fields using GNNs remains an unresolved issue. [31] Explored the concept of divergence-free neural networks for fluid mechanics, using automatic differentiation, even so, this approach remains experimental and difficult to scale on larger meshes due to the expensive use of automatic differentiation.

1.4 Contributions

In this work, we propose a machine learning method as a deep learning based localization operator. As most of the Reduced Order Models for Computational Homogenization degrades the information about local fields for the sake of computational efficiency, durability design requires such information to efficiency be able to predict damage and or local failure. A localization problem is formulated such that local mechanical field (e.g., stress) can be determined from the average mechanical stress.

Our contributions are listed as follows: (i) we introduce a novel physics-informed loss function that constrains the neural network to predict local stress fields that follows the local equilibrium state condition (i.e. $\operatorname{div}\sigma = 0$) during the training phase; (ii) we use a graph periodic representation implemented to enhance the representation of periodic boundary conditions for the GNN model during training and inference phases; (iii) validations are performed on a 2D test case of a hole plate, varying both the radius and position of the hole to further evaluate the prediction capacity of the proposed model; (iv) the proposed model is tested in reconstructing a local stress field from a non linear hyperelastic simulation in finite strain showing an important computation speed-up compared to FEM.

2 Background

2.1 Homogenization

In legacy multiscale approaches, three key components are required: (i) a constitutive model for each phase (ii) the identification of localization tensors which relate the global and local responses and (iii) techniques to transition and bridge the different scales.

Constitutive models will be described in this article considering elastic and non-linear hyperelastic models. Note that if not fully described to focus on the development of the GNN-based reduced order model, all the models have been developed such that the *tangent modulus* is defined so that it can be fully integrated in finite element codes to build the ground truth simulation database.

Micromechanical methods utilized in the literature of heterogeneous materials can be classified in two main categories – *mean-field approaches* and *full-field approaches*.

Mean-field approaches is often referred to methods that consider the phase-averaged response of the various constituents. Thoses approaches typically leverage the seminal result of [32] and his Equivalent Inclusion Method [32,

33, 34] to account for the presence of multiple inhomogeneities. In a few cases, analytic approaches can be utilized to describe the behavior of RVEs, for example in the case of composites reinforced with spheres or cylinders [35, 36, 37].

Full-field approaches are those in which the position dependent field values are determined and then averaged for the effective, macroscopic response. The latter category utilize numerical techniques, e.g. the finite element method (FEM), to solve representative computational domains – repeating unit cells (UC) for periodic materials and representative volume elements (RVEs) for statistically homogeneous materials [38]. Also here, analytical methods exists in the case of periodic laminates, even if the constituents exhibit a non-linear behavior [39].

Note that the identification of the overall response of a composite with periodic microstructure requires most of the times efficient computational tools like finite element or finite volume methods. To reduce the computational cost of FE calculations, a method that utilizes Fast Fourier Transforms has been developed [40]. Analytical or semi-analytical homogenization solutions exist for elastic multilayered composites [41] and for elastic fiber composites with square or hexagonal symmetry [42, 43, 44, 45].

2.1.1 Linearized, incremental formulation

The relation between the stress, the strain and the temperature is often very complicated in nonlinear materials. For hyperelastic and dissipative mechanisms where internal variables are involved, considering an incremental formulation adapted to numerical implementation in simulation codes, such relation is expressed in an implicit way. The problem has been already linearized in time, using for instance an approach based on the well known return mapping algorithm scheme [46]. Details about this approach are provided in [47].

In multiscale analysis, the two problems (macroscale and unit cell) are solved simultaneously, using an iterative scheme: from the macroscale analysis the macroscopic strains are used in the unit cell problem to compute the microscale variables to determine the unknown macroscopic stress. In the following the description of the non-linear, periodic homogenization framework closely follows

For the macroscale analysis there are two types of increments for a variable b:

- The time increment between the steps n and $n+1$,
$$\Delta\bar{b}^{(n+1)} = \bar{b}^{(n+1)} - \bar{b}^{(n)}. \tag{1}$$
- The FE iterative increment between the steps m and $m+1$,
$$\eth\bar{b}^{(n+1)(m)} = \bar{b}^{(n+1)(m+1)} - \bar{b}^{(n+1)(m)} = \Delta\bar{b}^{(n+1)(m+1)} - \Delta\bar{b}^{(n+1)(m)}. \tag{2}$$

For the microscale analysis there are three types of increments for a variable b:

- The time increment between the steps n and $n+1$,
$$\Delta b^{n+1} = b^{n+1} - b^{n}. \tag{3}$$
- The macroscopic tangent moduli increment between the steps m and $m+1$,
$$\eth b^{(n+1)(m)} = b^{(n+1)(m+1)} - b^{(n+1)(m)} = \Delta b^{(n+1)(m+1)} - \Delta b^{(n+1)(m)}. \tag{4}$$
- The unit cell iterative increment between the steps m^* and m^*+1,
$$\delta b^{(n+1)(m+1)(m^*)} = b^{(n+1)(m+1)(m^*+1)} - b^{(n+1)(m+1)(m^*)} = \Delta b^{(n+1)(m+1)(m^*+1)} - \Delta b^{(n+1)(m+1)(m^*)}. \tag{5}$$

To avoid lengthy expressions and following [47], every quantity that refers to the time step $n+1$ will be presented without index. Only quantities that refer to the previous time steps are presented with the exponent (n).

2.1.2 Unit cell problem

In this section we introduce the following tensor operators using Einstein notation:

$$\mathbf{C} = \mathbf{A} : \mathbf{B} \qquad \text{with} \qquad C_{i_1\, i_2 \ldots i_{n+m}} = \quad A_{i_1\, i_2 \ldots i_n\, pq}\, B_{pq\, i_{n+1}\, i_{n+2} \ldots i_{n+m}}\,,$$

$$\mathbf{C} = \mathbf{A} \,\tilde{:}\, \mathbf{B} \qquad \text{with} \qquad C_{i_1\, i_2 \ldots i_{n+m}} = \quad A_{i_1\, i_2 \ldots i_n\, pq}\, B_{i_{n+1}\, i_{n+2} \ldots i_{n+m}\, pq}\,,$$

$\mathbf{A} : \mathbf{B}$ is often called the double contraction product. When $\mathbf{A}$ and $\mathbf{B}$ are both second order tensors, the resulting $\mathbf{C}$ is a scalar and the operators : and $\tilde{:}$ are equivalent.

1. As input, the unit cell problem is informed with the macroscopic strain and macro-strain increment, provided exclusively by the analysis in the macroscale, which means that they do not iterate during the calculations, and thus there is no δ increment in $\bar{\varepsilon}$. The microscopic stress tensor at the time step $n+1$ can be expressed in an iterative linearized fashion as

$$\sigma^{\text{updated}} = \sigma + \mathbf{D}^{\varepsilon} : \delta\varepsilon, \quad \delta\varepsilon = \text{grad}_{\text{sym}}\delta\widetilde{\mathbf{u}}, \tag{6}$$

where $\mathbf{D}^{\varepsilon}$ denotes the mechanical tangent modulus and grad_{sym} is the symmetric gradient. To ensure the existence and the uniqueness of solution of the microscale problem, the fourth order tensor $\mathbf{D}^{\varepsilon}$ need to be positive definite at all times. Using the above linearized expressions, the micro-equilibrium is written as

$$\text{div}\left(\sigma + \mathbf{D}^{\varepsilon} : \text{grad}\delta\widetilde{\mathbf{u}}\right) = 0, \tag{7}$$

Here the gradient is equivalent to the symmetric gradient since $\mathbf{D}^{\varepsilon}$ is assumed symmetric. These conservation laws are accompanied by appropriate microscopic constitutive laws to update the current stress σ. With regard to boundary conditions, periodicity of $\delta\widetilde{\mathbf{u}}$, and antiperiodicity of $\sigma \cdot \mathbf{n}$ are imposed. The global system of equations is solved iteratively using a Newton-Raphson Method. When the convergence is achieved (i.e. when $\delta\widetilde{\mathbf{u}}$ is almost zero), the time increments and the actual values of the fluctuating term $\widetilde{\mathbf{u}}$ are evaluated, allowing all the microscopic quantities and their averaged (macroscopic) counterparts to be computed. Moreover, the tangent modulus $\mathbf{D}^{\varepsilon}$ at all microscopic points can be calculated when the convergence is achieved, using the local constitutive models. These moduli are essential to obtain their macroscopic counterparts in the second part of the unit cell problem.

2. Once the first part is completed, the residual terms of the microscopic stress components are assumed exactly zero. The strain and the equilibrium are written in the linearized, incremental form as

$$\eth\varepsilon = \eth\bar{\varepsilon} + \text{grad}_{\text{sym}}\eth\widetilde{\mathbf{u}}, \quad \text{div}\eth\sigma = \mathbf{0}, \tag{8}$$

With regard to boundary conditions, periodicity of $\eth\widetilde{\mathbf{u}}$ and antiperiodicity of $\eth\sigma \cdot \mathbf{n}$ are imposed. Considering the mechanical tangent moduli, the increment of the stress tensor is expressed:

$$\eth\sigma = \mathbf{D}^{\varepsilon} : \eth\varepsilon \tag{9}$$

Combining equations (8), (9) yields

$$\text{div}\left(\mathbf{D}^{\varepsilon} : \eth\bar{\varepsilon} + \mathbf{D}^{\varepsilon} : \text{grad}\eth\widetilde{\mathbf{u}}\right) = \mathbf{0}, \tag{10}$$

The solution of the above homogeneous system (up to an arbitrary macroscopic constant) is written in the form [48]

$$\eth\widetilde{\mathbf{u}} = \eth\bar{\varepsilon} : \chi^{\varepsilon} \tag{11}$$

where the third order tensor χ^{ε} is periodic and is referred to as corrector term. Substituting (11) in (10) yields

$$\eth\bar{\varepsilon} : \text{div}\left(\left[\mathbf{D}^{\varepsilon} + \mathbf{D}^{\varepsilon}\tilde{:}\text{grad}\chi^{\varepsilon}\right]^{T}\right) = \mathbf{0}, \tag{12}$$

The last expressions are valid for arbitrary values of the macroscale variables $\eth\bar{\varepsilon}$, only if the corrector terms satisfy the linear equations:

$$\text{div}\left(\left[\mathbf{D}^{\varepsilon} + \mathbf{D}^{\varepsilon}\tilde{:}\text{grad}\chi^{\varepsilon}\right]^{T}\right) = \mathbf{0} \tag{13}$$

Using the solution (11), the increments of the other microscopic fields are written as

$$\begin{aligned} \eth\varepsilon &= \mathbf{A}^{\varepsilon} : \eth\bar{\varepsilon} \\ \eth\sigma &= \mathbf{D}^{\varepsilon} : \mathbf{A}^{\varepsilon} : \eth\bar{\varepsilon} \end{aligned} \tag{14}$$

where

$$\mathbf{A}^{\varepsilon} = \mathcal{I} + \mathcal{I}\tilde{:}\text{grad}\chi^{\varepsilon}, \tag{15}$$

is the strain localization tensor . It is recalled that $\mathcal{I}$ denotes the fourth order symmetric identity tensor and $\mathbf{I}$ is the second order identity tensor. Averaging of these fields over the unit cell yields the macroscopic tensors

$$\bar{\mathbf{D}}^{\varepsilon} = \langle \mathbf{D}^{\varepsilon} : \mathbf{A}^{\varepsilon} \rangle \tag{16}$$

This yields the definition of a localization operator $\mathbf{A}^{\varepsilon}$ that allows to retrieve the local microscopic deformation increment from the knowledge of the average deformation tensor increment. While in practice, during finite element-based simulation of the non-linear response of a unit cell, providing periodic boundary conditions, such localization tensor is never evaluated, all methods that actually provide the tangent modulus $\bar{\mathbf{D}}^{\varepsilon}$ are able to provide those localization

tensors at every gauss point of the considered periodic mesh. A localization tensor field can therefore be constructed at node points, relying on the same interpolation functions and averaging methods that those utilized for the evaluation of the stress field.

The stress localization operator $\mathbf{B}^\varepsilon$ is deduced from $\mathbf{A}^\varepsilon$ as:

$$\begin{aligned}
&\eth\sigma = \mathbf{D}^\varepsilon \mathbf{A}^\varepsilon : \eth\bar{\varepsilon} \\
&\eth\sigma = \mathbf{D}^\varepsilon \mathbf{A}^\varepsilon \left(\bar{\mathbf{D}}^\varepsilon\right)^{-1} : \eth\bar{\sigma} \\
&\eth\sigma = \mathbf{B}^\varepsilon : \eth\bar{\sigma}\,, \quad \mathbf{B}^\varepsilon = \mathbf{D}^\varepsilon \mathbf{A}^\varepsilon \left(\bar{\mathbf{D}}^\varepsilon\right)^{-1}
\end{aligned} \tag{17}$$

The proposed method in this paper can be viewed, from this perspective, as an AI-based model of this localization operator, considering that the local stress field are directly evaluated at the graph (or mesh) node points from the information of the average stress tensor.

On the other hand, the aforementioned method is implemented in the finite element code *Fedoo*, such that non-linear constitutive laws can be utilized in a non-linear incremental periodic boundary condition framework.

2.2 Mechanical problem

The mechanical problem addressed here is the numerical evaluation of the response of multi-scale structures, considering a periodic arrangement of a representative pattern. Assuming a bridging of scale, i.e. that the local cell characteristic size is several decades smaller than the structure, the periodic homogenization theory is adapted to determine an effective response of the representative unit cell (RUC). Within such a unit cell, the local static stress equilibrium equation in the absence of volume forces writes:

$$\operatorname{div}\sigma = 0, \tag{18}$$

which reduced in 2D:

$$\operatorname{div}\sigma = \begin{pmatrix} \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} \\ \frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} \end{pmatrix} = 0, \tag{19}$$

where σ is the stress tensor and div is the divergence operator.

To solve this problem using a displacement-based approach we need: (i) a constitutive equation to define the material's response (ii) a kinematical relation that defines the strain field from the displacement field $\mathbf{u}$; (iii) appropriate prescribed boundary conditions.

In the following, the constitutive law is considered either hyperelastic or linear elastic. In this last case, small strain approximation is utilized and the relationship between σ and the small strain tensor ε may be written as:

$$\sigma = \mathbf{L} : \varepsilon, \tag{20}$$

where $\mathbf{L}$ is fourth order the linear elastic stiffness tensor.

Further considering such linearized kinematics, the relation between ε and $\mathbf{u}$ is:

$$\varepsilon = \frac{1}{2}\left(\nabla\mathbf{u} + \nabla^T\mathbf{u}\right), \tag{21}$$

where $\nabla\mathbf{u}$ denotes the displacement gradient. As for the boundary conditions, considering that we have a periodic microscopic domain, we naturally use the periodic boundary conditions [49], that can be written by:

$$\mathbf{u}(\mathbf{x}) = \bar{\varepsilon}\cdot\mathbf{x} + \tilde{\mathbf{u}}(\mathbf{x}), \tag{22}$$

where $\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}$ is the coordinates vector in the unit cell, assumed in 2D for sake of readability without loss of generality, that is described by the geometric domain Ω. $\bar{\varepsilon} = \frac{1}{V}\int_\Omega \varepsilon\, dV$ is the mean strain tensor over the representative unit cell and $\tilde{\mathbf{u}}$ is a periodic fluctuation, i.e. for two points that belong to opposite faces whose coordinates are noted respectively $\mathbf{x}^+$ and $\mathbf{x}^-$ we have $\tilde{\mathbf{u}}(\mathbf{x}^+) = \tilde{\mathbf{u}}(\mathbf{x}^-)$ and therefore it comes the condition:

$$\mathbf{u}(\mathbf{x}^+) - \mathbf{u}(\mathbf{x}^-) = \bar{\varepsilon}(\mathbf{x}^+ - \mathbf{x}^-), \tag{23}$$

that we have to enforce over the boundary of the domain for each pair of opposite faces. Note that for the microscopic strain localization problem, the mean field $\bar{\varepsilon}$ is supposed to be known considering that the macroscopic structure mechanical problem has been solved. Note that practical details about the implementation of this kind of conditions inside a FE code can be found in [50].

To solve Equations (18), (20) and (21) using the FE method we need to use the corresponding weak formulation that is:

$$\int_{\Omega} \sigma(\mathbf{u}) : \varepsilon(\mathbf{u}^{\star})\, d\Omega = 0, \tag{24}$$

$\mathbf{u}^{\star}$ being a kinematically admissible displacement field. We assume a FE discretization with n nodes that allow to build an arbitrary scalar field f from its nodal values such as:

$$f(\mathbf{x}) = \begin{pmatrix} N_1(\mathbf{x}) & N_2(\mathbf{x}) & \cdots & N_n(\mathbf{x}) \end{pmatrix} . \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix} . \tag{25}$$

$N_i(\mathbf{x})$ are the classical FE shape functions associated to node i and f_i are the nodal values of f. This discretization is used to approximate the displacement field in the context of Equation (24).

In the more general framework of finite strains, considering a nearly incompressible hyperelastic material, a strain energy density is defined as the sum of a volume-changing part and a volume-preserving part:

$$\Psi = U(J) + W(\bar{I}_1, \bar{I}_2)\,. \tag{26}$$

In the above, $J = \mathrm{Det}\mathbf{F}$ is the determinant of the deformation gradient $\mathbf{F}$, while $\bar{I}_1, \bar{I}_2$ are the first and second invariants of the volume-preserving right Cauchy-Green strain tensor $\bar{\mathbf{C}}$, that derives from the right Cauchy-Green strain tensor $\mathbf{C}$:

$$\begin{aligned} \bar{I}_1 &= J^{-2/3} I_1 \quad & I_1 &= \mathrm{tr}(C) \\ \bar{I}_2 &= J^{-4/3} I_2 \quad & I_2 &= \frac{1}{2}\left(\mathrm{tr}(C)\mathrm{tr}(C) - \mathrm{tr}(C^2)\right). \end{aligned} \tag{27}$$

Following [51], the Cauchy stress is written as:

$$\begin{aligned} \sigma &= p\mathbf{1} + \mathbf{s} \\ p &= \frac{\partial U}{\partial J} \\ J\mathbf{s} &= 2\frac{\partial W}{\partial \bar{I}_1}\mathrm{dev}\,\bar{\mathbf{b}} + 2\frac{\partial W}{\partial \bar{I}_2}\left(\mathrm{tr}\,\bar{\mathbf{b}}\,\mathrm{dev}\,\bar{\mathbf{b}} - \mathrm{dev}\,\bar{\mathbf{b}}^2\right). \end{aligned} \tag{28}$$

In Equation (28), $\bar{\mathbf{b}} = J^{-2/3}\mathbf{b}$ is the volume preserving part of the left Cauchy-Green deformation tensor $\mathbf{b} = \mathbf{F}\mathbf{F}^T$. Also, p denotes the hydrostatic pressure and $\mathbf{s}$ is the deviatoric part of the Cauchy stress σ. In the following, the nearly incompressible Neo-Hookean hyperelastic law is utilized, with:

$$\Psi = \kappa\left(J\ln J - J + 1\right) + \frac{\mu}{2}\left(\bar{I}_1 - 3\right), \tag{29}$$

In finite strain, periodic boundary conditions arise naturally from the discretization of the displacement gradient over the unit cell, i.e:

$$\mathbf{u}(\mathbf{x}^+) - \mathbf{u}(\mathbf{x}^-) = \overline{\nabla u}\,\left(\mathbf{x}^+ - \mathbf{x}^-\right). \tag{30}$$

In this case, the FE method requires to solve a non-linear system due to the non-linear relation between strain quantities and the displacement gradient. Hence, a Newton-Raphson algorithm is utilized and an incremental, linearized weak formulation is defined. The geometric nonlinearities are considered based on an updated Lagrangian approach. The

constitutive model utilized in this work is implemented in the open-source library *Simcoon* and the non linear solver is implemented in the finite element library *Fedoo*, all being part of the 3MAH set [52].

Considering the FE method, the local equilibrium equation (Equation (18)) is enforced in a weak sense with respect to the chosen approximation of the displacement field. The stress divergence will therefore not be exactly 0, and we can even use the stress divergence values as a FE approximation error indicator.

2.3 Discrete divergence operator

Computing the stress divergence values can be a complex task since the stress is often poorly approximated inside an element. For instance, for 2D linear triangle elements, the approximated stress, which is linked to the displacement gradient considering a linear elastic response, is constant over the elements, and therefore the stress divergence is always zero-evaluated. Another way to better approximate the stress divergence is to get the stress values at nodes and differentiate the FE interpolation function to compute a divergence operator from the nodal values. The construction of a stress divergence operator is briefly summarized in the following.

Once the solution is obtained, i.e. the displacement field is known (or displacement increment for incremental problems), it is easy to compute the strain and stress tensors fields from the FE discretization (Equation (25)) at any points in the domain using the strain/displacement relation and the constitutive law (i.e. Equations (21) and (20) in our case). The strain and stress fields are generally not continuous between elements, and the nodal values for stress and strain are typically determined by averaging the values from the adjacent elements.

We define the divergence operator as a matrix $\mathbf{D}$ that computes the divergence at any point in the domain from the nodal values of a 2D vector field using the FE approximation. $\mathbf{D}$ is constructed to account for the contributions of the derivatives of the shape functions at each node. That gives:

$$\mathbf{D}(\mathbf{x}) = \begin{pmatrix} \mathbf{D}_x(\mathbf{x}) & \mathbf{D}_y(\mathbf{x}) \end{pmatrix},$$

where $\mathbf{D}_x$ and $\mathbf{D}_y$ are matrices containing the derivatives of the shape functions with respect to x and y, respectively:

$$\mathbf{D}_x(\mathbf{x}) = \begin{pmatrix} \frac{\partial N_1(\mathbf{x})}{\partial x} & \frac{\partial N_2(\mathbf{x})}{\partial x} & \dots & \frac{\partial N_n(\mathbf{x})}{\partial x} \end{pmatrix} \tag{31}$$

and

$$\mathbf{D}_y(\mathbf{x}) = \begin{pmatrix} \frac{\partial N_1(\mathbf{x})}{\partial y} & \frac{\partial N_2(\mathbf{x})}{\partial y} & \dots & \frac{\partial N_n(\mathbf{x})}{\partial y} \end{pmatrix}. \tag{32}$$

In the same way, a divergence operator $\mathbf{D}_\sigma$ is built to compute the approximated stress divergence from nodal values:

$$\mathbf{div}(\sigma(\mathbf{x})) = \mathbf{D}_\sigma(\mathbf{x}) \cdot \boldsymbol{\sigma},$$

where $\boldsymbol{\sigma}$ is the vector of nodal stress components, that is built by stacking the nodal values of each stress component such as:

$$\boldsymbol{\sigma} = \begin{pmatrix} [\sigma_{xx}] \\ [\sigma_{yy}] \\ [\sigma_{xy}] \end{pmatrix} \tag{33}$$

and

$$\mathbf{D}_\sigma(\mathbf{x}) = \begin{pmatrix} \mathbf{D}_x(\mathbf{x}) & 0 & \mathbf{D}_y(\mathbf{x}) \\ 0 & \mathbf{D}_y(\mathbf{x}) & \mathbf{D}_x(\mathbf{x}) \end{pmatrix}. \tag{34}$$

Like the stress and strain fields, the divergence computed with this operator is not continuous between elements. The nodal divergence field is calculated by averaging the values at adjacent elements. In practice we build a discrete divergence matrix operator $\hat{\mathbf{D}}_\sigma$ of size $(2n, 3n)$, that directly computes the divergence nodal values ($2n$ values since there are two divergence terms per node) from the nodal stress values (3 components of stress on each node).

The divergence of the stress field at node i can be expressed as:

$$\mathbf{div}(\boldsymbol{\sigma})|_{\mathbf{x}=\mathbf{x}_i} = \begin{pmatrix} (\hat{\mathbf{D}}_\sigma \cdot \boldsymbol{\sigma})_i & (\hat{\mathbf{D}}_\sigma \cdot \boldsymbol{\sigma})_{n+i} \end{pmatrix}, \tag{35}$$

where $\mathbf{div}(\boldsymbol{\sigma})|_{\mathbf{x}=\mathbf{x}_i} \in \mathbb{R}^2$ is a 2D vector containing the two terms of the divergence, presented in Equation (19) for a given node i at coordinates $\mathbf{x}_i$. In this last expression, the index $n + i$ is related to the second term of the divergence vector at node i.

This discrete operator provides a practical method to evaluate the stress divergence across the computational domain and enables its use in error estimation or for iterative improvement of the FE solution. The construction of $\mathbf{D}$ depends on the chosen element type and the corresponding shape functions.

In this work, we propose to use this discrete operator to compute a physics-informed loss that constrains the predictions of the deep learning framework during the training phase to adhere to the equilibrium state condition.

2.4 Graph Neural Networks

The deep learning model used for this study is referred as Graph Neural Networks (GNN), this model operating on graph data as input. Graphs are an abstract representation of relational data defined as a set of vertices (or nodes) noted $\mathcal{V} = \{v_1, v_2, ..., v_n\}$ and a set of unordered pairs of vertices noted $\mathcal{E}$ called edges representing connections or relationships between vertices. Each edge $e \in \mathcal{E}$ is an element of the form (v_i, v_j). We can associate a feature node vector $\boldsymbol{v}_i \in \mathbb{R}^d$ of dimension d to each node $v_i \in \mathcal{V}$ and a feature edge vector $\boldsymbol{e}_{ij} \in \mathbb{R}^k$ to each edge $(v_i, v_j) \in \mathcal{E}$ of dimension k. The feature vectors can be used to encode various properties of nodes and edges, such as categorical information or numerical measurements.

GNNs rely on a mechanism called Message Passing Layers (MPL), where each message passing operation consists in propagating and aggregating information across nodes in a graph. The goal of this mechanism is to iteratively update the features of each node by exchanging information "messages" with its neighbors. Information that is stored in edges can participate to the message passing operation.

A message passing layer is a local operation that is performed on a given node v_i taking into account the information of its set of neighbors $j \in \mathcal{N}(i)$ and the corresponding edge information e_{ij}.

The Message Passing GNN can be seen as a generalization of Convolutional Neural Networks [13], since both methods operate locally. However, GNNs are not restricted to grid data as is the case of CNN. GNN can indeed be applied to very general structures of data, which can be represented as graphs. Finite Element meshes are particularly adapted to a graph representation, since they share structural characteristics as they both involve nodes (vertices) and connections (edges). The latter represents the geometric connectivity between nodes. GNNs appear as a very adapted framework for the task of local stress field reconstruction. Moreover, FE mesh data can present different resolutions in terms of local mesh refinement, which is difficult to represent in a grid based approach.

3 Local Equilibrium Constraint Graph Neural Networks for local stress reconstruction

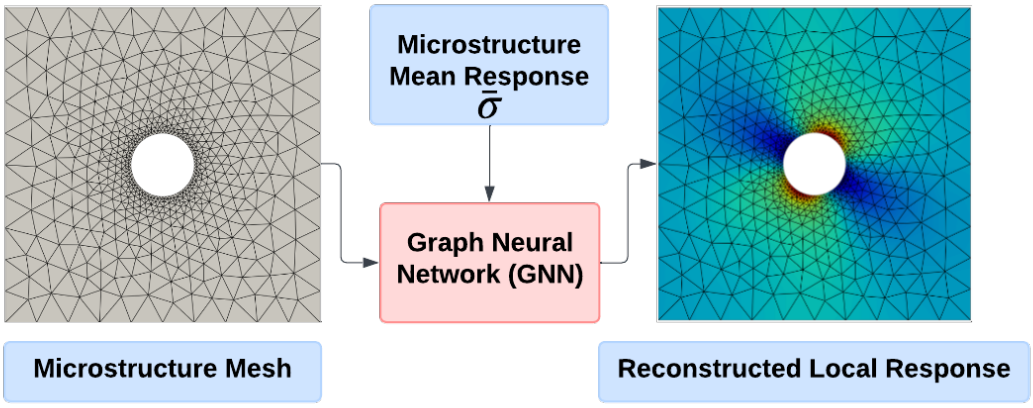


Figure 1: Overview of the proposed local stress reconstruction method. The mean microstructure stress response $\bar{\sigma}$, is uniformly assigned to all nodes of the input mesh. A Graph Neural Network (GNN) then acts as a localization operator, predicting the corresponding local stress field σ_i at each node.

The Graph Neural Network will be applied next as a localization model, with the task of reconstructing the local stress field σ on a periodic mesh $\mathcal{M}$, given only the mean mechanical response $\bar{\sigma}$ (see Figure 1). Local field reconstruction is understood here as predicting the stress field values at each node of the mesh, i.e. at each vertex of the corresponding

GNN. Note that since this graph corresponds to a FE mesh, interpolation functions can be used to reconstruct a continuous piecewise mechanical field.

The microstructure mesh is therefore represented as a graph that encodes topological features such as node positions and mean mechanical properties. The mean mechanical response $\bar{\sigma}$ is assigned as a uniform attribute to all nodes, while topological features, such as node positions, vary across nodes. Edge information is represented by the Euclidean distance between nodes. To account for periodicity, we introduce an additional preprocessing step in which edges are added between boundary nodes. This approach significantly enhances message passing within the model, giving improved results that will be presented in Section 4.3. The model incorporating periodic edges is referred to as *P-GNN*, which serves as the baseline model in place of a classical GNN.

Node and edge features of the input graph are projected to a latent representation using a Multi-Layer Perceptron (MLP) encoder. The resulting latent graph is processed by a Message Passing GNN framework (as described in Section 2.4), and finally, the latent graph is decoded by an MLP decoder, producing local stress values as output. This GNN architecture, commonly known as *MeshGraphNet*, has been successfully applied to scalar fields predictions on mesh-based data[25, 24, 27] and represents the state-of-the-art in this domain.

As most methods rely on the comparison with numerical simulation as ground truth, we recall here that the mechanical equilibrium is satisfied in a weak sense, i.e. considering the volume integral of the domain. Additionally, we propose a physics-informed loss function that constrains the network to predict local stress fields that satisfy the local static stress equilibrium condition described in Section 2.2. The model that implements this physics-informed loss is referred to as *P-DivGNN*.

Section 3.1 provides a detailed description of how the microstructure mesh is represented as a graph. Section 3.2 explains the Encode-Message Passing-Decode mechanism, and Section 3.3 presents the proposed physics-informed loss used during training.

3.1 Representing mesh as a graph

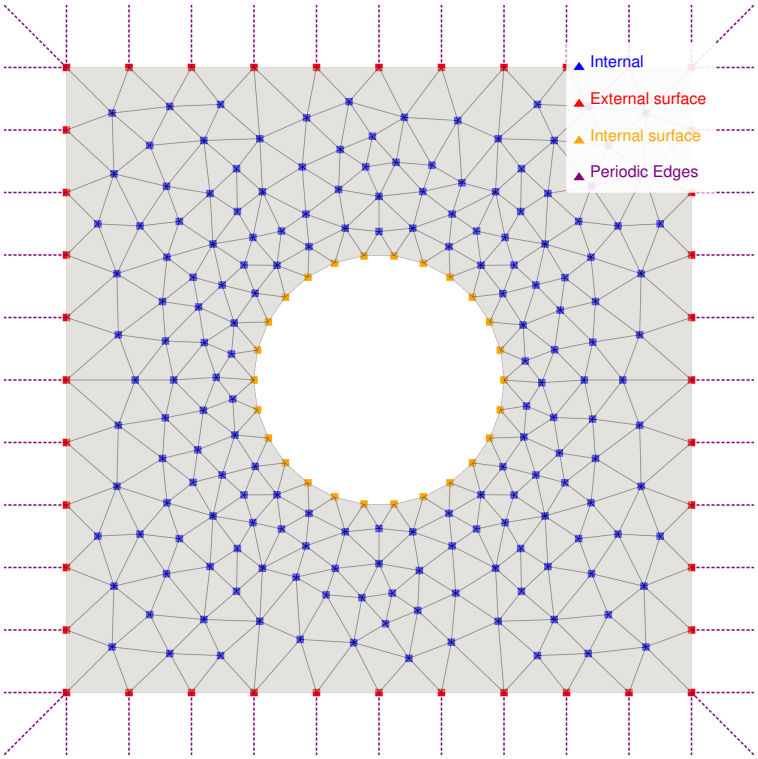


Figure 2: Illustration of the graph representation of a mesh, node labeling in function of topological features and periodic edges added to reinforce message passing.

To construct a graph representation of the periodic mesh, mesh nodes are labeled according to their position. As shown in Figure 2, nodes are classified into internal nodes, internal surface nodes, and external surface nodes using a label α.

To enhance message passing, additional edges are introduced between boundary nodes. The feature information for these additional edges is set to zero. An alternative method for representing periodicity is to merge boundary nodes, but this approach makes it challenging to assign Cartesian positions to the affected nodes.

We define $\mathcal{M} = (\mathcal{V}, \mathcal{E})$ as an undirected homogeneous graph representing a FE mesh. Each node $v_i \in \mathcal{V}$ is associated with a feature vector $\boldsymbol{v}_i = (\bar{\sigma}, \mathbf{x}_i, \alpha_i)$, where:

$\bar{\sigma} \in \mathbb{R}^3$ represents the mean stress values $(\bar{\sigma}_{xx}, \bar{\sigma}_{yy}, \bar{\sigma}_{xy})$ uniformly assigned to all nodes.

$\mathbf{x}_i \in \mathbb{R}^2$ denotes the spatial coordinates of node v_i.

$\alpha_i \in \{-1, 0, 1\}$ is an identifier indicating whether the node belongs to an internal surface, an internal node, or is subject to periodic boundary conditions (as illustrated in the legend of Figure 2).

Each edge $(v_i, v_j) \in \mathcal{E}$ is associated with a feature vector $\boldsymbol{e}_{ij} \in \mathbb{R}$ that encodes the Euclidean distance $\text{dist}(\mathbf{x}_j, \mathbf{x}_i) \in \mathbb{R}$ between the spatial coordinates of nodes v_i and v_j. To account for periodicity, additional edges are introduced between boundary nodes, with the edge feature vector for these edges set to zero. This periodic edge enhancement is visualized in Figure 2, where purple edges represent the added periodic edges. This approach facilitates more effective message passing and improves the model's ability to capture periodicity within the mesh.

3.2 Implemented Model

As illustrated in Figure 3 the model implemented for this study follows a *Encode-Message Passing-Decode* architecture, a framework that has been successfully used to predict stress, strain, and displacement fields in mesh data [24, 25, 26].

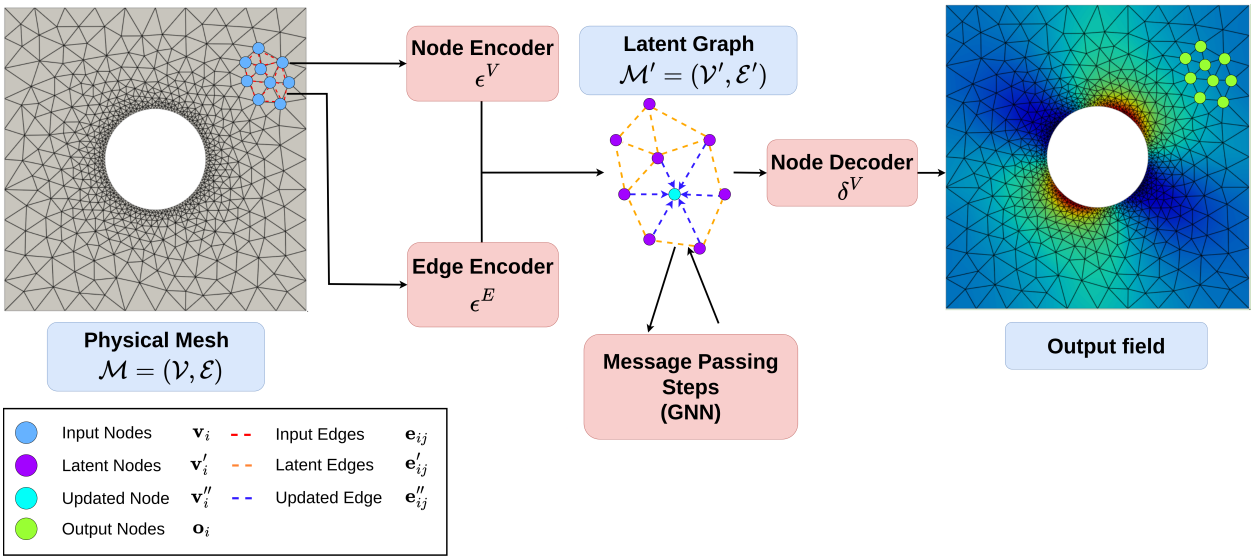


Figure 3: Schematics of the proposed model illustrating *Encode-Message Passing-Decode* architecture. Nodes $\boldsymbol{v}_i$ and edges $\boldsymbol{e}_{ij}$ of the input graph are embedded through the ϵ^V and ϵ^E encoders respectively. This embedded graph is then processed by a GNN which will update the nodes and edges embeddings through message-passing steps. Finally, the node embeddings are decoded with δ^V to obtain the output field. Here, only a small subgraph is shown for visualisation purposes, but in practice the whole graph is processed by the architecture.

The proposed GNN model is used as a mapping function:

$$\text{GNN}(\boldsymbol{v}_i, \mathcal{M}) \mapsto (\hat{\sigma}_{xx_i}, \hat{\sigma}_{yy_i}, \hat{\sigma}_{xy_i}) \tag{36}$$

allowing to estimate local stress fields values $\hat{\sigma}_i$ for each node $\boldsymbol{v}_i$ using mesh structure features represented as a graph $\mathcal{M}$.

The architecture of the GNN consists of three components:

Encoder: The encoder processes the input node feature vectors $\boldsymbol{v}_i$, and edge feature vectors $\boldsymbol{e}_{ij}$, projecting them into corresponding latent spaces, $\boldsymbol{v}'_i \in \mathbb{R}^H$ and $\boldsymbol{e}'_{ij} \in \mathbb{R}^H$ with H the dimension of the latent space. This transformation is

achieved using two independent multi-layer perceptrons (MLPs): ϵ^V for node features and ϵ^E for edge features. The transformations are defined as follows:

$$\boldsymbol{v}_i' = \epsilon^V(\boldsymbol{v}_i), \quad \boldsymbol{e}_{ij}' = \epsilon^E(\boldsymbol{e}_{ij}), \tag{37}$$

Using the transformed features, the embeddings of the input graph $\mathcal{M}$ are updated giving a latent graph noted $\mathcal{M}' = (\mathcal{V}', \mathcal{E}')$. It is important to emphasize that this encoding process operates solely on the feature vectors of the graph and does not consider its structural information and the topology of the graph is not changed neither.

Message Passing: This module operates on the latent graph $\mathcal{M}'$, incorporating the graph's structural features (e.g., the connections between nodes) through L message-passing steps. During each step, the module updates the edge and node feature vectors using two separate multi-layer perceptrons: ϕ for edge features and γ for node features. For a given message step layer $l \in [1, 2, \ldots, L]$ the transformations are defined as follows:

The edge feature vectors $\boldsymbol{e}_{ij}'$ are updated based on the features of the connected nodes and the edge itself:

$$\boldsymbol{e}_{ij}'' = \phi^{(l)}(\boldsymbol{v}_i', \boldsymbol{v}_j', \boldsymbol{e}_{ij}') \quad \text{where} \quad \boldsymbol{e}_{ij}'' \in \mathbb{R}^H. \tag{38}$$

The node feature vectors $\boldsymbol{v}_i'$ are updated using the aggregated messages from their neighbors, $\mathcal{N}(i)$, as follows:

$$\boldsymbol{v}_i'' = \gamma^{(l)}\left(\boldsymbol{v}_i', \sum_{j \in \mathcal{N}(i)} \boldsymbol{e}_{ij}''\right) \quad \text{where} \quad \boldsymbol{v}_i'' \in \mathbb{R}^H. \tag{39}$$

Unlike the Encoder module, the message-passing module explicitly uses the graph's structural information by incorporating the set of neighbors, $\mathcal{N}(i)$, for each node. The number of message-passing steps, L, is a hyperparameter that determines how far information propagates across the graph.

Residual connections are incorporated into the MLPs ϕ and γ. Additionally, the same parameters for ϕ and γ are reused across all L message-passing steps. After completing L steps, the resulting graph $\mathcal{M}'' = (\mathcal{V}'', \mathcal{E}'')$ is passed to the decoder.

Decoder: The decoder projects the latent node feature vectors into the desired output space. It uses an MLP, denoted δ^V, to map each node feature vector $\boldsymbol{v}_i''$ to an output vector $\boldsymbol{o}_i$ as follows:

$$\delta^V(\boldsymbol{v}_i'') \mapsto \boldsymbol{o}_i \quad \text{with} \quad \boldsymbol{o}_i \in \mathbb{R}^O. \tag{40}$$

Here, O represents the dimension of the output vector associated with each node. In this study, $O = 3$ since we aim to predict the local stress field components $(\hat{\sigma}_{xx_i}, \hat{\sigma}_{yy_i}, \hat{\sigma}_{xy_i})$ at each node of the input mesh $\mathcal{M}$.

In this implementation, we adopted the hyper-parameters from the original *MeshGraphNets* framework as detailed in [24], specifically setting the latent space dimension H to 128 and the number of message-passing steps L to 10.

3.3 Physics-Informed Loss

To guide the training process, we introduce a physics-informed loss function that enforces the equilibrium condition described in Section 2.3. The total loss is composed of two terms: (1) a normalized mean squared error (NMSE) between the GNN predictions and the ground truth, and (2) a physics-based regularization term that penalizes deviations from the divergence-free stress condition $\operatorname{div} \sigma = 0$.

Let σ denote the ground-truth stress field obtained from FE simulations, $\hat{\sigma}$ the predicted stress field from the GNN, and n the total number of nodes in the mesh $\mathcal{M}$. The loss function is defined as:

$$\mathcal{L}(\sigma, \hat{\sigma}) = \text{NMSE}(\sigma, \hat{\sigma}) + \lambda (\mathbf{div}(\hat{\boldsymbol{\sigma}}))^2. \tag{41}$$

The physics-informed term $(\mathbf{div}(\hat{\boldsymbol{\sigma}}))^2$ is computed using the divergence discrete operator (see Equation (35), Section 2.3) and is defined as the mean squared norm of the divergence over all indices i:

$$(\mathbf{div}(\hat{\boldsymbol{\sigma}}))^2 = \frac{1}{n} \sum_{i=1}^{n} \| \mathbf{div}(\hat{\boldsymbol{\sigma}})|_{\mathbf{x}=\mathbf{x}_i} \|^2, \tag{42}$$

It is important to note that the divergence term in Equation (42) is minimized using only internal nodes of the mesh (see blue nodes in Figure 2). This approach is necessary because the divergence operator applied to the stress field $\hat{\boldsymbol{\sigma}}$ is not well-suited for periodic meshes. In both, our model and FEM simulations, divergence values at the boundary nodes are significantly high. To address this issue, the divergence at these boundary nodes is set to zero during training.

The NMSE is used instead of the mean squared error (MSE) to ensure equal importance for each stress component, $c \in \{xx, yy, xy\}$. It is defined as:

$$\text{NMSE}\,(\sigma, \hat{\sigma}) = \frac{1}{n_c} \sum_{c=1}^{n_c} \frac{\sum_{i=1}^{n} (\sigma_{c,i} - \hat{\sigma}_{c,i})^2}{\sum_{i=1}^{n} (\sigma_{c,i} - \frac{1}{n} \sum_{j=1}^{n} \sigma_{c,j})^2}, \tag{43}$$

The NMSE ensures that all stress components contribute equally to the training loss and the physics-based regularization term further encourages the GNN to satisfy the equilibrium condition.

A detailed analysis of the model's convergence and performance is presented in Sections 4 and 5.

4 Experimental results on linear elastic case

To validate the proposed approach and evaluate the contribution of the divergence-free regularization introduced in Section 3.3, we consider two datasets corresponding to different mechanical regimes. The first dataset consists of FEM simulations under purely linear elastic conditions. This configuration, being computationally straightforward, serves as a baseline for assessing the effectiveness of the physics-informed regularization during training.

To further investigate the efficiency and generalization capabilities of the model, particularly the benefits of employing a Graph Neural Network (GNN) as a localization operator, a second, more complex dataset is introduced. This dataset is derived from a non-linear hyperelastic FEM model that includes geometric nonlinearities. The corresponding experiments and results are presented in detail in Section 5.

4.1 Dataset generation

In this work, we generate a dataset consisting of 10,000 two-dimensional periodic meshes of a square plate with a central hole. Mesh refinement is applied to every element belonging to the central hole's surface. The meshes are generated using the open-source meshing library *Gmsh* [53]. To introduce variability, the coordinates of the hole's center, its radius, and the mesh refinement levels are uniformly sampled (see Figure 4), resulting in meshes with different hole positions, radii, and triangle element sizes. Each mesh is paired with a local stress field computed using Finite Element simulation, as detailed in Section 2.2. Each solution is computed under periodic boundary conditions applied to the edges of the square plate, assuming plane stress conditions. The average in-plane strain components, $\bar{\varepsilon}_{xx}$, $\bar{\varepsilon}_{yy}$, and $\bar{\varepsilon}_{xy}$, are used to define these periodic boundary conditions. To ensure the database captures a wide range of loading conditions, mean in-plane strain values are uniformly sampled within the range $[-0.05, 0.05]$.

The dataset, comprising 10,000 meshes with varying hole plate geometries and mesh refinements along with their corresponding FEM-computed solutions, is partitioned into a training set (70%) and a test set (30%). All results presented in this work are obtained exclusively from the test set.

Each simulation is performed using the open-source FE Python library, *Fedoo* [52], with local stress values computed at the Gauss points of the elements. Considering the variability of the hole position and radius, note that the formulated problem is non-linear even if linear elastic response is considered. Indeed, the mapping function providing hole position, radius, average strain as prescribed boundary conditions and linear elastic material parameters to get the mechanical field over the domain is a non-linear function.

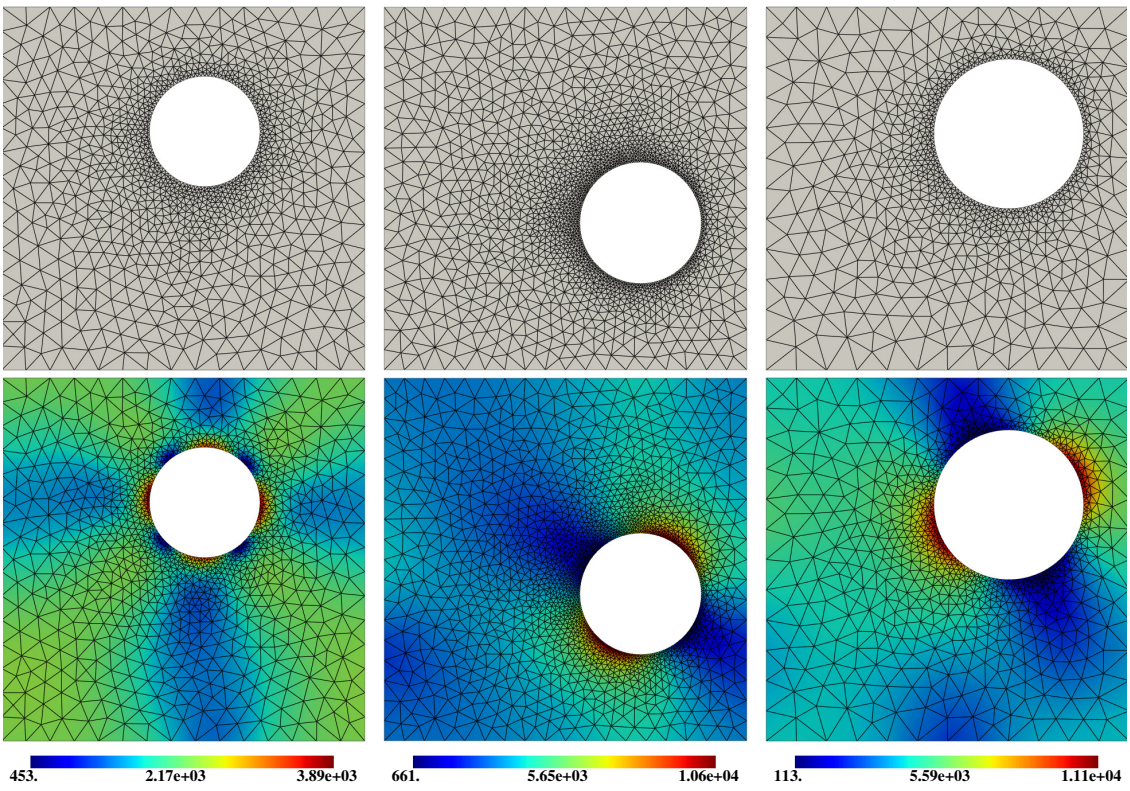


Figure 4: Top row: mesh examples of the generated database note that a mesh refinement is applied to every elements belonging to the hole surface, the mesh refinement and the global size of the triangles are different for each mesh. For this three examples the global triangles sizes corresponding to each mesh are $(7.10, 5.79, 8.35)$ and the triangle sizes belonging to the hole surface are $(1.01, 0.62, 1.07)$ Bottom row: Von Mises local stress fields (in MPa) corresponding to top row meshes.

The goal of the GNN is to reconstruct local stress fields based on their average values. During training, the input stress values corresponds to the volume average of stress fields over the representative unit cell (RUC) volume, as expressed by:

$$\bar{\sigma} = \frac{1}{V} \int_{\Omega} \sigma \, dV.$$

The GNN is designed to predict stress values at the nodes of a graph, which coincide with the structure of the FE mesh. The stress value shall be unique for each node, consequently, the finite element stress values at nodes are computed as the average of interpolated stress values from the surrounding elements.

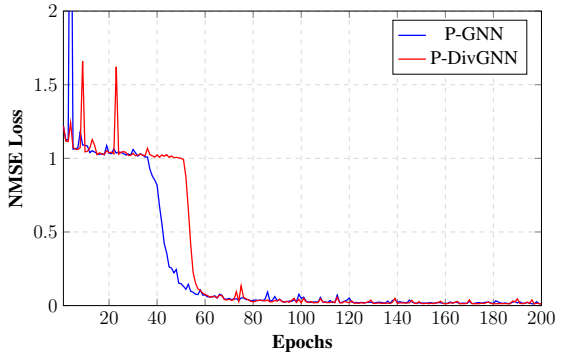
A linear elastic isotropic material model was used, characterized by a Young's modulus of 10^5 MPa and a Poisson ratio of 0.3. A linear elastic constitutive law was chosen to focus on evaluating the effects of the proposed divergence-free regularization in terms of normalized mean squared error (NMSE) convergence, the divergence of predicted fields, and reconstruction quality considering a variation of the geometric features of the problem, i.e. the graph structure. Non-linear constitutive response can be considered as an extension of such proposed methodology, considering a sequence of linearized increments of the macroscopic quantities.

To standardize the contributions of different stress components during training, feature scaling was applied by subtracting the mean and dividing by the standard deviation of the training dataset.

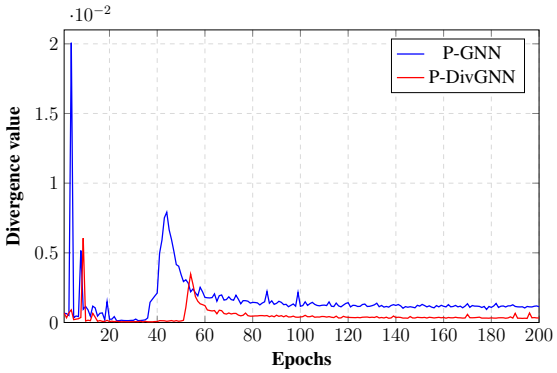
4.2 Model training

To benchmark the proposed models, *P-GNN* and *P-DivGNN*, are both trained for 200 epochs using the same training dataset and identical random seed initialization to ensure reproducibility. Model optimization is performed using the

Adam algorithm [54], with a learning rate of 10^{-2}. A mini-batching strategy is employed during training, using a batch size of 16 graphs per iteration. Early stopping is applied with a patience of 20 epochs based on the normalized mean squared error (NMSE) computed on the validation set. All experiments are conducted using the open-source library PyTorch Geometric [55] and executed on an NVIDIA RTX A4000 GPU with 16 GB of VRAM.



(a) Comparison of NMSE loss convergence in validation dataset when the proposed divergence free penalty is applied during training



(b) Divergence evolution in validation dataset during training

Figure 5: Comparison of NMSE loss and divergence evolution during training of *P-GNN* model (in blue) and *P-DivGNN* model (in red).

In Figure 5a, the evolution of the NMSE loss during training is presented. It is evident that the NMSE for *P-GNN* decreases significantly around epoch 40. In contrast, the NMSE for *P-DivGNN* shows a noticeable reduction at epoch 60. This delayed behavior can be attributed to the addition of a new term in the loss function, which increases the complexity of learning the ground truth FE solution.

Furthermore, Figure 5b examines the evolution of the divergence values of the predicted stress fields during training for both models. During the initial 40 epochs, the divergence values remain near zero. This behavior occurs because the same mean stress values are uniformly assigned to each node in the input graph. While this results in lower divergence values, the corresponding stress fields are unsatisfactory when compared to the FE solution. The regularization term hyperparameter is set to $\lambda = 10$ for balancing the contribution between data reconstruction performed by the NMSE and the divergence constraint.

It is important to note that the divergence values of the predicted stress fields using *P-DivGNN* remain consistently lower than those obtained with *P-GNN* for the majority of the training process.

4.3 Results

In our experimental setup, we assess the impact of incorporating a divergence-free penalty into the loss function during training. To this end, we compare the performance of *P-GNN* and *P-DivGNN* against finite element (FE) simulation results. The evaluation includes histogram distributions of each stress component $(\sigma_{xx}, \sigma_{yy}, \sigma_{xy})$, visual reconstructions of the stress fields, and a normalized squared error (NSE) metric to quantify deviations from the FE baseline.

The NSE is adapted from the normalized mean squared error (NMSE) introduced in Equation 3.3, but it is evaluated locally basis using nodal stress differences. It is defined as

$$\text{NSE}\left(\sigma_{c,i}, \hat{\sigma}_{c,i}\right) = \frac{(\sigma_{c,i} - \hat{\sigma}_{c,i})^2}{\sum_{k=1}^{n} (\sigma_{c,k} - \frac{1}{n}\sum_{j=1}^{n} \sigma_{c,j})^2}, \tag{44}$$

This formulation enables a consistent quantitative comparison between the GNN predicted stress fields and the corresponding FE reference fields, using the same normalization scheme employed during network optimization. Consequently, the difference plots directly reflect the error measure relevant to the model's training objective.

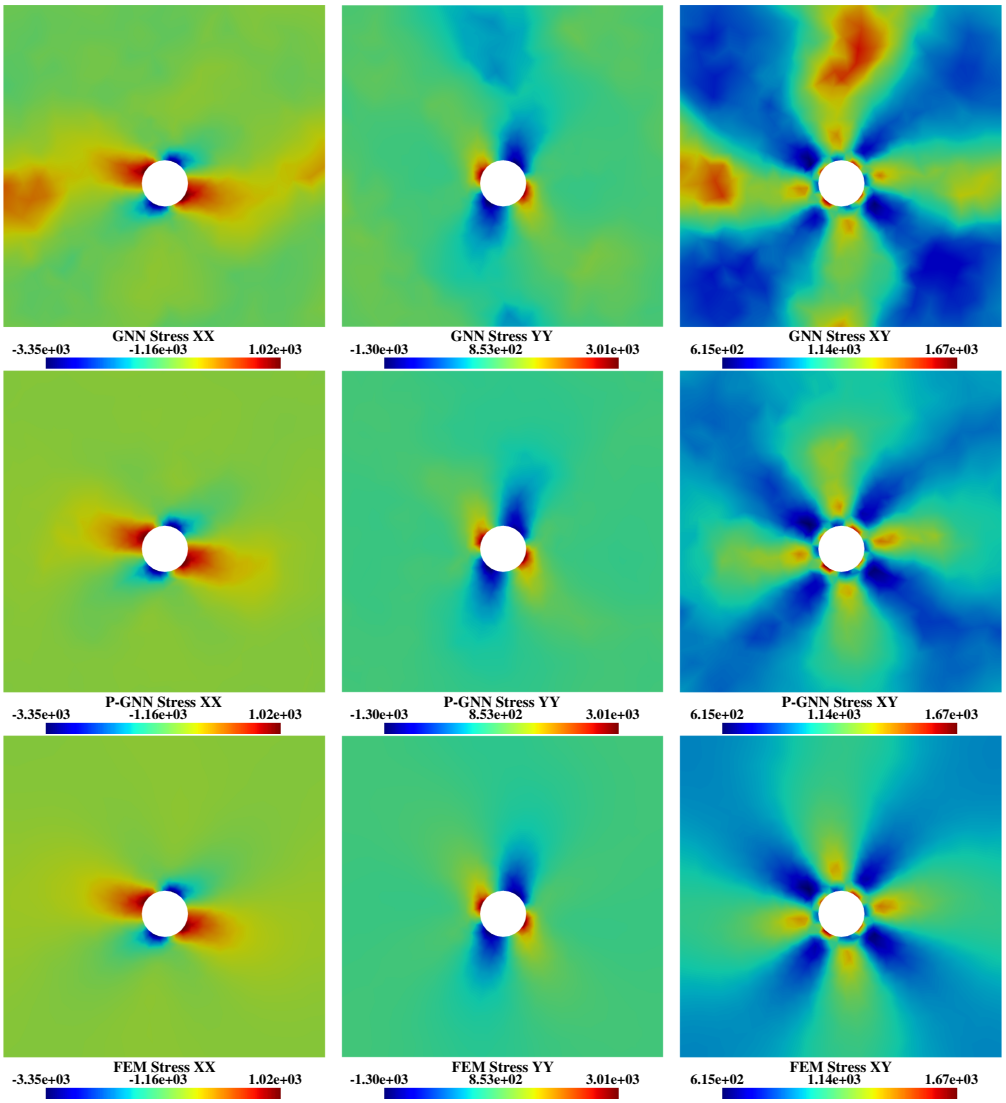


Figure 6: Local stress reconstruction comparison between classical GNN (without periodic edges) in top row, *P-GNN* in middle row, and FEM in bottom row.

As shown in Figure 6 and Figure 7, the inclusion of periodic edges between boundary nodes leads to a marked improvement in the reconstruction of local stress fields. These additional periodic edges enhance the preservation of periodicity and symmetry in the predicted stress fields, resulting in more accurate reconstructions compared to a conventional GNN. Figure 7 further emphasizes this effect through a difference plot that highlights the local reconstruction errors of the non-periodic GNN model.

This improvement is consistent across most of the test meshes. Consequently, the *P-GNN* serves as the baseline model for further comparisons instead of the classical GNN framework.

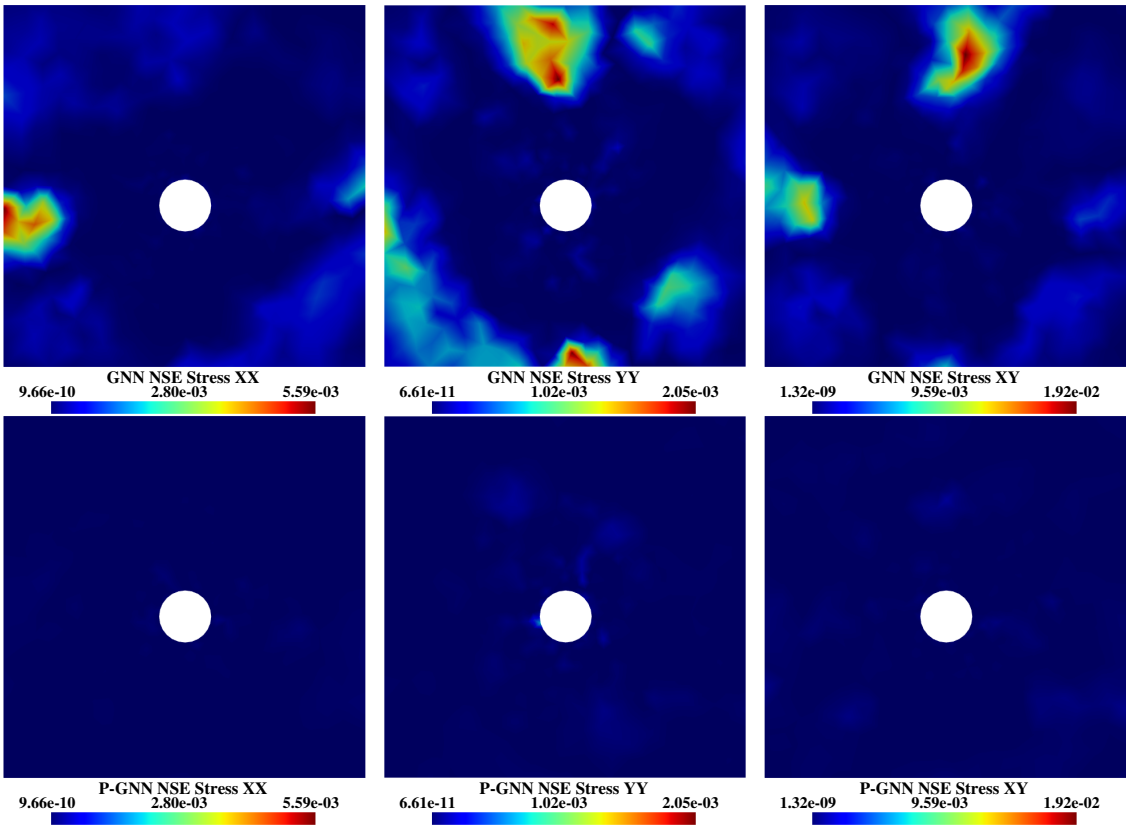
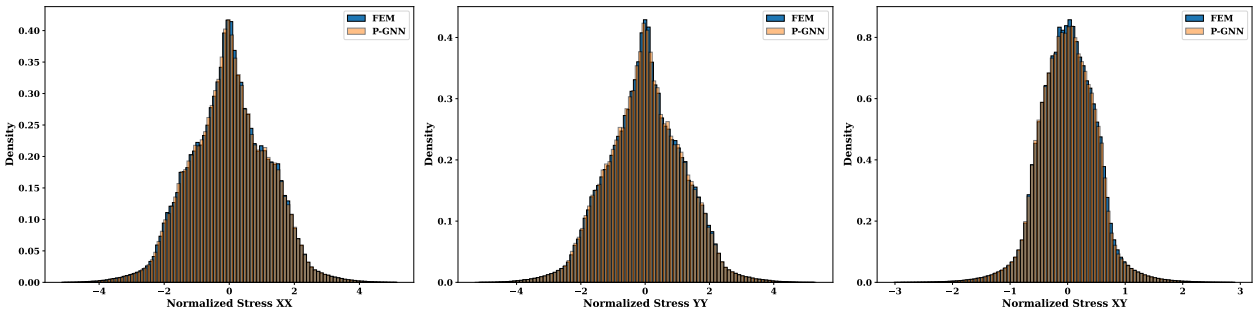
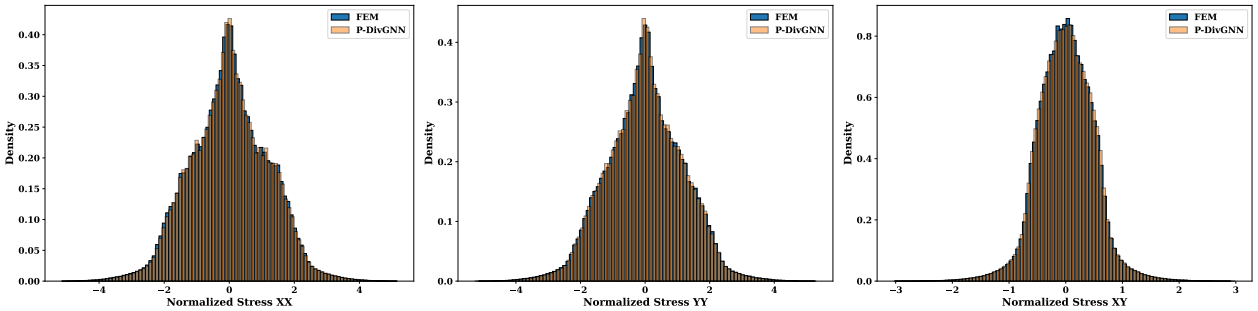


Figure 7: NSE between predicted local stress field and FEM stress field for both *P-GNN* and classical GNN (without periodic edges), error fields are plotted at node level.

It is important to note that this pre-processing step may introduce extra computational costs, particularly for very large meshes with a high number of nodes. If the representative volume element (RVE) has a squared shape (or cubic for 3D geometries), the operation is straightforward. In contrast, when the RVE does not have a regular shape, identifying boundary nodes may require the use of feature edge extraction algorithms, adding complexity to the process.



(a) Comparison distribution of all local stress fields of the test database between predicted stress fields using *P-GNN* and FE solutions



(b) Comparison distribution of all local stress fields of the test database between predicted stress fields using *P-DivGNN* and FE solutions

Figure 8: Comparison of normalized local stress distributions of the entire database between *P-GNN*, *P-DivGNN* and FEM.

In Figure 8, we compare the standardized distributions of the predictions from both proposed models, *P-GNN* and *P-DivGNN*, against the FEM results. The stress fields for the entire test set, across each direction $(\sigma_{xx}, \sigma_{yy}, \sigma_{xy})$, are projected onto a density function. The histograms effectively illustrate that, overall, the predictions from both models are close to the FE ground truth. Especially, this visualization allows to determine if specific features of the stress distribution are captured, e.g., the repartition of extreme values of the stress fields. No truncation of the extreme stress values are observed in the predicted results. Importantly, the incorporation of a divergence-free penalty does not degrade the stress values distribution but rather ensures that the predicted distributions remain consistent with the ground truth FEM solutions.

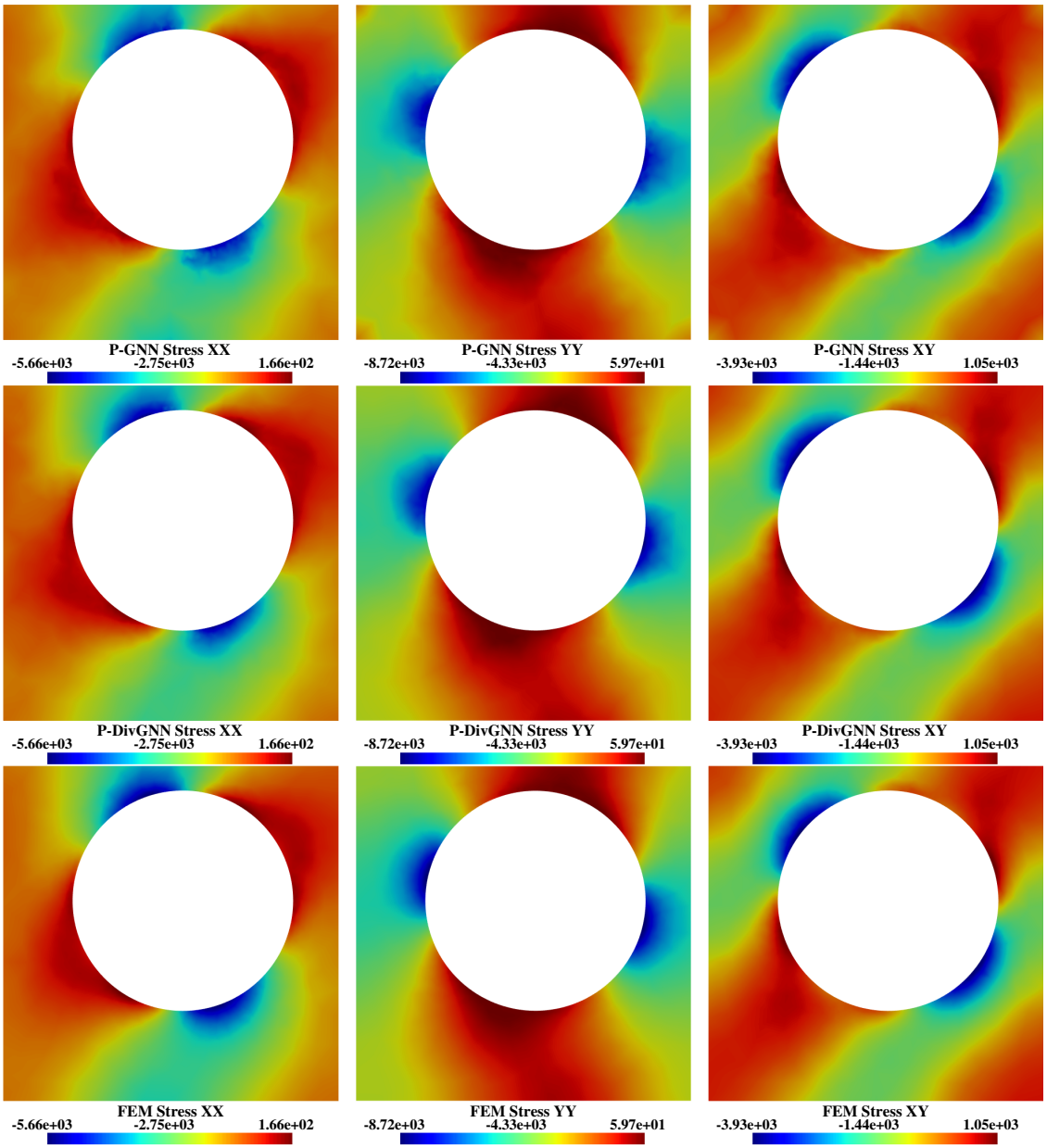


Figure 9: Comparison of local stress field reconstructions for *P-GNN* (top row), *P-DivGNN* (middle row), and FEM (bottom row) for the three stress components $(\sigma_{xx}, \sigma_{yy}, \sigma_{xy})$. Input mean stress values: $\bar{\sigma}_{xx} = -1154.90$ MPa, $\bar{\sigma}_{yy} = -2024.47$ MPa, $\bar{\sigma}_{xy} = -398.07$ MPa. The color bar scale is set to match the FE solution range.

Figure 9 compares the predictions of *P-GNN*, *P-DivGNN*, and the FE simulation for a representative example from the test set, which was not seen during training. In this example, *P-DivGNN* produces noticeably smoother stress field predictions compared to *P-GNN*, while both models generate reconstructions that are qualitatively comparable to the FE ground truth.

For a more detailed comparison, Figure 10 illustrates the NSE and error fields at the node level. Notably *P-GNN* predicts higher stress values near the mesh corners, creating artifacts that are absent in the stress fields predicted by *P-DivGNN*.

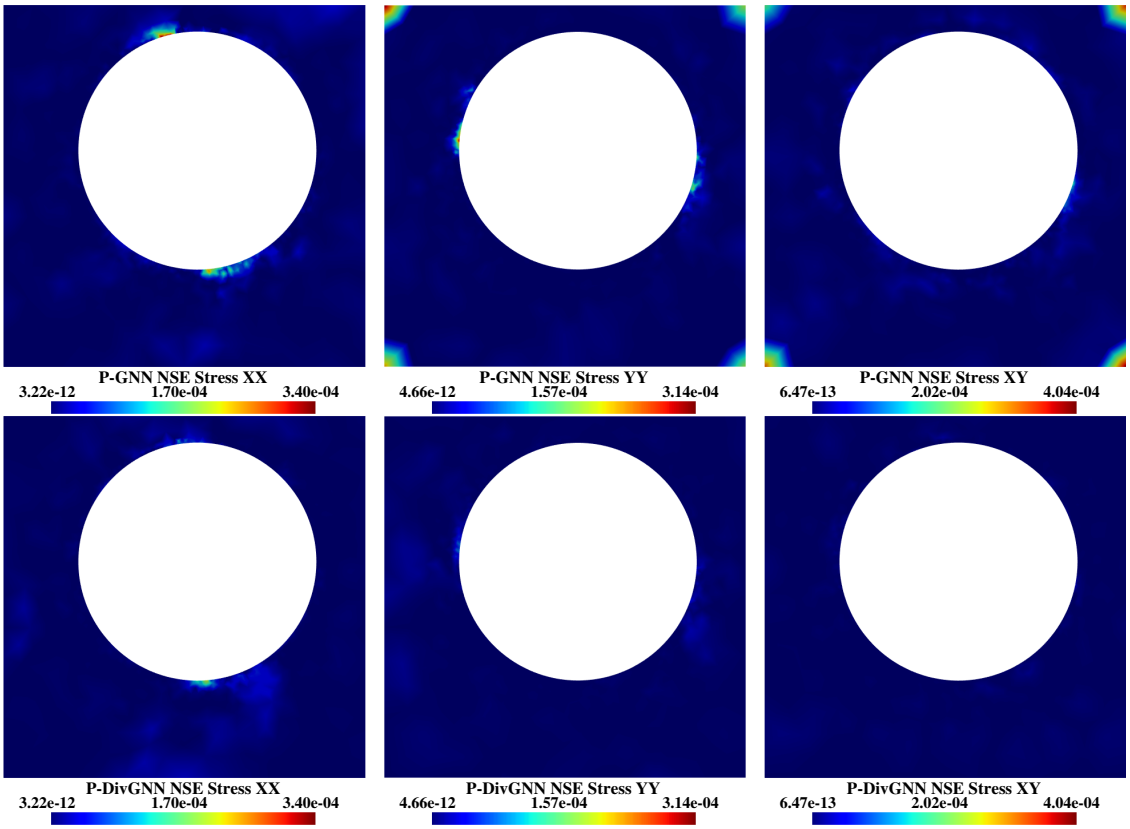


Figure 10: NSE between predicted local stress field and FEM stress field for both *P-GNN* and *P-DivGNN*, error fields are plotted at node level.

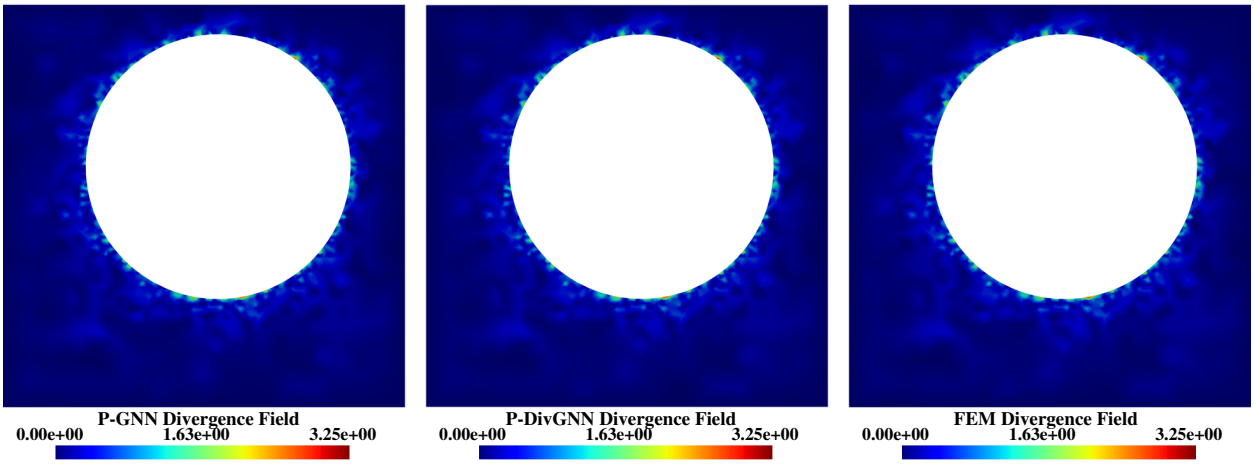


Figure 11: Comparison of norm of divergence field of local stress field between *P-GNN*, *P-DivGNN* and FEM predictions.

Furthermore, the divergence field of the predicted stress fields is analyzed, as illustrated in Figure 11. The divergence field is computed as the norm of Equation (35) at each node of the mesh for the stress fields predicted by *P-GNN*, *P-DivGNN*, and the FE simulation.

As discussed in Section 3.3, Figure 11 shows that the divergence values at the internal boundary nodes of the mesh are notably high.

It is important to note that the divergence operator used in this work does not account for periodicity when applied to external boundary nodes. Consequently, divergence values at these nodes (highlighted in red in Figure 2) are excluded from the training process and set to zero both during training and in the visualizations of Figure 11. The same exclusion applies to the nodes located along the hole boundary. This is because the stress values from the FE simulations are defined at the element level, and their divergence at the boundaries must be extrapolated using the derivatives of shape functions—an approximation deemed insufficiently accurate for training purposes.

Quantitatively, *P-DivGNN* achieves a significantly lower divergence norm (4.56×10^{-4}) than both the baseline models and the FEM reference solution, which yields a divergence of 9.43×10^{-4}. Moreover, the predicted stress field from *P-DivGNN* also exhibits the lowest normalized mean squared error, reinforcing the effectiveness of the proposed approach in producing physically consistent stress predictions.

Table 1: Comparison of divergence mean value and NMSE between GNN (without periodic edges), *P-GNN*, *P-DivGNN*, and FEM, evaluated on the whole linear elastic test dataset. The model *P-DivGNN* outperforms the baseline *P-GNN* and achieves lower divergence values than the FE simulation.

	Divergence	**NMSE**
GNN	1.13×10^{-3}	2.33×10^{-2}
P-GNN	1.11×10^{-3}	1.13×10^{-2}
P-DivGNN	$\mathbf{3.23 \times 10^{-4}}$	$\mathbf{1.08 \times 10^{-2}}$
FEM	9.18×10^{-4}	–

Results computed on the linear elastic test dataset.

Table 1 presents a quantitative comparison of the mean divergence values and NMSE for *P-GNN*, *P-DivGNN*, and FEM, evaluated over the entire test dataset. Specially, *P-DivGNN* achieves a lower mean divergence value than FEM and a reduced NMSE compared to *P-GNN*, demonstrating its improved accuracy and consistency despite adding an additional learning regularization.

This demonstrates the importance of incorporating the divergence regularization term during training, as it ensures a more physically plausible reconstruction of the stress fields.

4.3.1 Benchmark

Figure 12 presents a performance benchmark comparing the execution time of the proposed method (*P-DivGNN*) in two configurations, one with periodic edges added and the other without, vs the execution time of a FE simulation. The benchmark is conducted by progressively increasing the number of nodes in the input mesh and measuring the computation time in seconds. Adding periodic edges introduces a computational cost due to the classification of nodes based on their association with a surface.

Although the inference time of *P-DivGNN* is consistently lower than that of the FE simulation for all tested configurations, the comparison is not straightforward, as the GNN inference benefits from GPU acceleration (Nvidia RTX A4000, 16 GB VRAM), while the FE simulation runs on a CPU (Intel Xeon W-2255, 10 cores, 20 threads) using the Intel oneAPI Math Kernel Library PARDISO solver. Furthermore, the GNN requires a one-time training phase prior to inference, which is not reflected in the benchmark.

Still, these results are meaningful, even for a linear elastic problem where FE simulations are typically very fast, the proposed GNN achieves faster inference. This demonstrates the potential of the method, especially when moving toward more complex, nonlinear cases, as explored in Section 5.3.1 for a nonlinear hyperelastic case, where computational gains are expected to be even more significant.

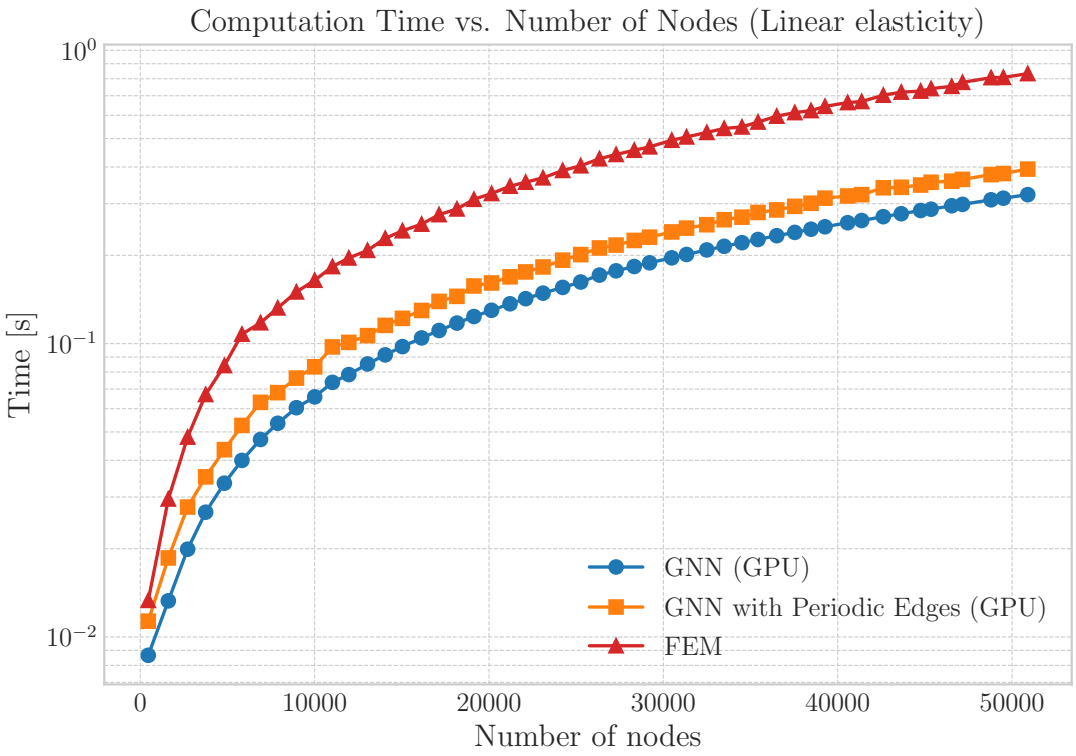


Figure 12: Performance comparison between FEM simulation *P-DivGNN* (red), *P-DivGNN* when adding periodic edges (orange) and without (blue). Tested using a CPU Intel Xeon W-2255 Processor, 10 cores, 20 threads for FEM simulation and GNN tests have been produced using a graphics card (GPU) Nvidia RTX A4000 with 16GB of VRAM where only 156MB were used.

5 Experimental results on non linear hyperelastic case

Building upon the results from the linear case (Section 4), we extend our evaluation to a dataset generated from a nonlinear hyperelastic FEM model that incorporates geometric nonlinearities. This more challenging scenario is designed to assess the robustness and performance gains achieved by using a GNN as a localization operator in complex mechanical settings.

As described in Section 2.2, the FEM simulations for this case employ an iterative Newton-Raphson scheme, leading to a significantly higher computational cost compared to the linear case. In contrast, the proposed GNN model reconstructs the local stress field directly from the mean stress field at the RVE scale, without requiring iterative solvers. This enables substantial computational savings while maintaining accuracy in stress localization.

5.1 Dataset generation

The dataset generation procedure mirrors that of Section 4.1. A total of 9795 two dimensional periodic meshes of square plate with a hole featuring variability in hole geometry and mesh refinement, were generated using the same randomized sampling procedure. The dataset is partitioned into a training set (70%) and a test set (30%). As in the linear case, all results reported in this study are obtained from the test set.

Finite Element simulations were performed under periodic boundary conditions applied to the edges of the square plate. In contrast to the linear elastic case, which used plane stress and small strain assumptions, the hyperelastic simulations were conducted under plane strain and finite strain assumptions. This choice was motivated by improved convergence of the Newton–Raphson iterative solver during FEM computations. Importantly, this modeling assumption does not affect the generalizability or performance of the GNN, which remains agnostic to the underlying kinematic framework in terms of stress field reconstruction.

To cover a broader range of mechanical responses, the average in-plane logarithmic strain components $\bar{\varepsilon}_{xx}$, $\bar{\varepsilon}_{yy}$, and $\bar{\varepsilon}_{xy}$ were uniformly sampled within the range $[-0.15, 0.15]$. In this non linear model $\bar{\varepsilon}$ is the logarithmic strain tensor whereas it is the linearized small strain tensor in the linear one. Evidently, the two strain tensor measures are equivalent for small strain values.

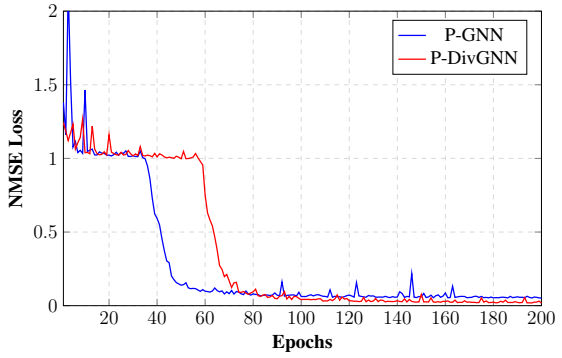
As seen in Equation (30), for finite strain problems, the periodic boundary conditions are based on the displacement gradient $\overline{\nabla u}$. The displacement gradients are computed from the sampled logarithmic strain tensors assuming a polar decomposition of $\mathbf{F}$ and enforcing an identity rotation matrix (i.e. without rigid body rotation). This gives:

$$\overline{\nabla u} = \mathbf{F} - \mathbf{1} = \exp \bar{\varepsilon} - \mathbf{1} \tag{45}$$

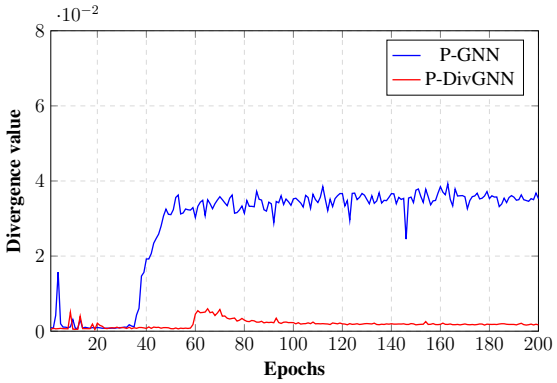
The constitutive behavior is modeled using the Neo-Hookean hyperelastic law, as defined in Section 2.2, Equation (20), with the parameters: Shear modulus $\mu = 3.0$ MPa and bulk modulus $\kappa = 10$ MPa.

5.2 Model training

To assess the convergence behavior of the models in the nonlinear hyperelastic setting, the same training protocol as described in Section 4.2 is adopted. Both *P-GNN* and *P-DivGNN* are trained for 200 epochs using the Adam optimizer, with early stopping based on validation NMSE (patience of 20 epochs), a learning rate of 10^{-2}, and mini-batches of 16 graphs. The primary difference in this setting is the choice of the physics-based regularization coefficient λ. Due to the different stress scales present in the hyperelastic dataset compared to the linear case, the regularization weight is adjusted to $\lambda = 10^{-2}$ to maintain an appropriate balance between the data fidelity and physics-informed terms in the loss function.



(a) Comparison of NMSE loss convergence in validation dataset when the proposed divergence free penalty is applied during training



(b) Divergence evolution in validation dataset during training

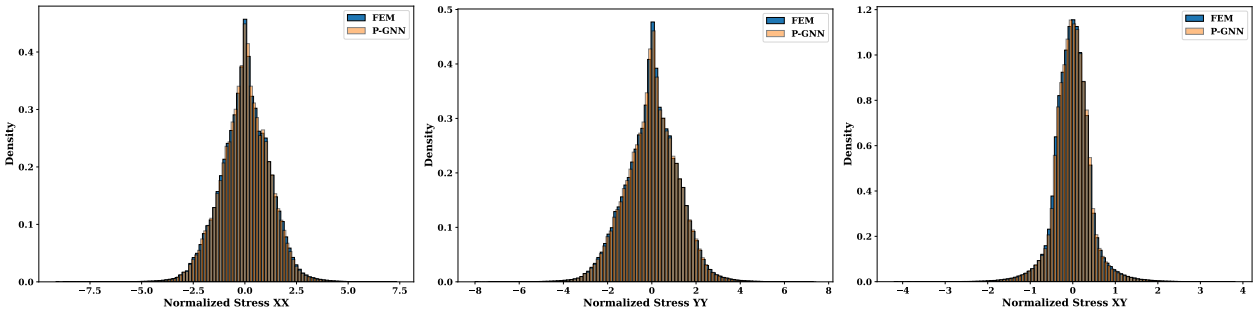
Figure 13: Comparison of NMSE loss and divergence evolution during training of *P-GNN* model (in blue) and *P-DivGNN* model (in red) in the hyperelastic dataset.

The evolution of the NMSE loss during training on the nonlinear hyperelastic dataset is shown in Figure 13a. A similar trend to that observed in the linear elastic case (Section 4.2) is noted: The *P-GNN* exhibits a rapid decline in NMSE around epoch 40, while *P-DivGNN* demonstrates a delayed yet more progressive reduction, becoming significant near epoch 60. Despite the slower initial convergence, *P-DivGNN* ultimately achieves slightly lower NMSE values, suggesting that the inclusion of the physics-informed regularization term enhances the model's ability to reconstruct stress fields.

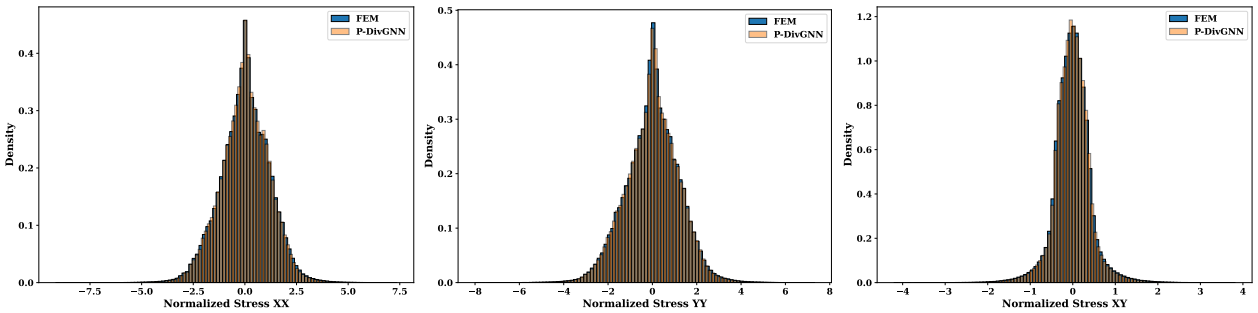
Figure 13b illustrates the divergence evolution of the predicted stress fields throughout training. As in the linear case, divergence values are initially close to zero due to the uniform application of mean stress inputs across graph nodes. However, in this nonlinear scenario, the divergence profiles of the two models diverge more noticeably as training progresses. In particular, *P-DivGNN* consistently maintains lower divergence values than *P-GNN*, indicating improved compliance with the underlying physical constraints. This distinction is more pronounced than in the linear case and highlights the benefit of incorporating the divergence penalty, especially in more complex regimes where geometric nonlinearities are present.

5.3 Results

Following the same evaluation strategy used for the linear elastic dataset, the performance of *P-GNN* and *P-DivGNN* is now assessed on the more complex nonlinear hyperelastic dataset. Figure 14 presents the distribution of predicted stress components $(\sigma_{xx}, \sigma_{yy}, \sigma_{xy})$ across the entire test set, compared to the reference FEM solutions.



(a) Comparison distribution of all local stress fields of the non-linear hyperelastic test database between predicted stress fields using *P-GNN* and FE solutions



(b) Comparison distribution of all local stress fields of the non-linear hyperelastic test database between predicted stress fields using *P-DivGNN* and FE solutions

Figure 14: Comparison of normalized local stress distributions of the entire database between *P-GNN*, *P-DivGNN* and FEM.

Figure 14 shows that both models produce stress predictions that closely align with the ground truth distributions, preserving the overall statistical structure of the stress field. These results are particularly noteworthy given the increased complexity of the hyperelastic problem, as they indicate that the proposed GNN models are capable of accurately reconstructing stress fields even in the presence of geometric nonlinearities and non linear mechanical response.

The results indicate that the inclusion of the physics-informed regularization does not impair the fidelity of the predicted stress distribution. On the contrary, it contributes to maintaining consistency with the reference FEM results, even under nonlinear loading conditions. This demonstrates the robustness of the proposed approach in capturing realistic stress responses across a broader class of mechanical behaviors.

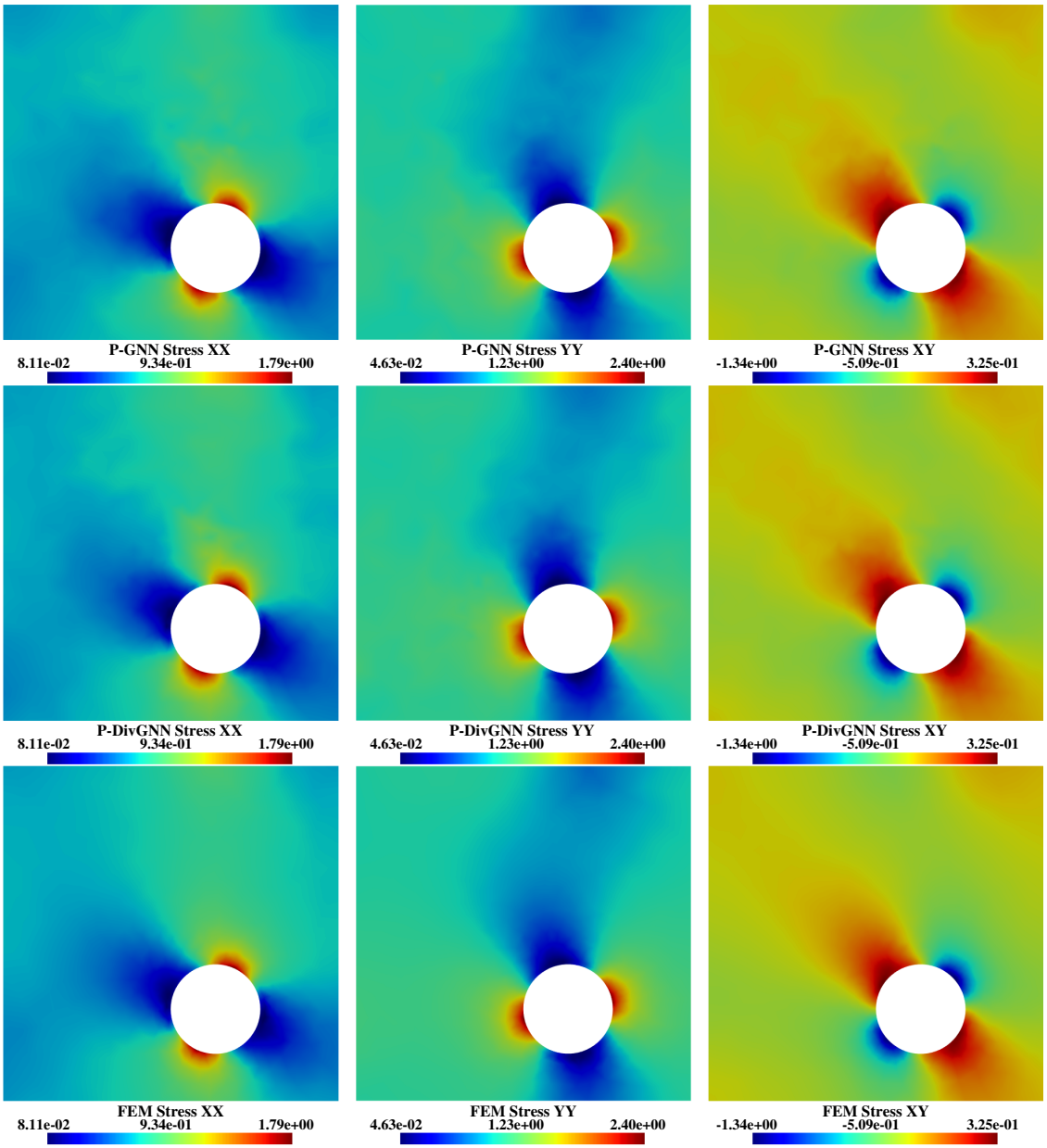


Figure 15: Comparison of local stress field reconstructions for *P-GNN* (top row), *P-DivGNN* (middle row), and FEM (bottom row) for the three stress components $(\sigma_{xx}, \sigma_{yy}, \sigma_{xy})$. Input mean stress values: $\bar{\sigma}_{xx} = 0.59$ MPa, $\bar{\sigma}_{yy} = -1.01$ MPa, $\bar{\sigma}_{xy} = 0.33$ MPa. The color bar scale is set to match the FE solution range.

Figure 15 presents a representative example from the hyperelastic test set. As observed in the linear elastic case (Section 4.3), both models provide stress field reconstructions that are qualitatively close to the FEM reference. However, the overall quality is slightly degraded in this non-linear setting. This is likely due to the increased complexity of the hyperelastic problem, which introduces geometric nonlinearities that are more challenging to learn. It is worth noting that the same GNN architecture and training settings were used for both datasets. Further performance improvements could be achieved through targeted hyperparameter tuning, such as optimizing the divergence penalty weight λ, but this was not explored in the current study.

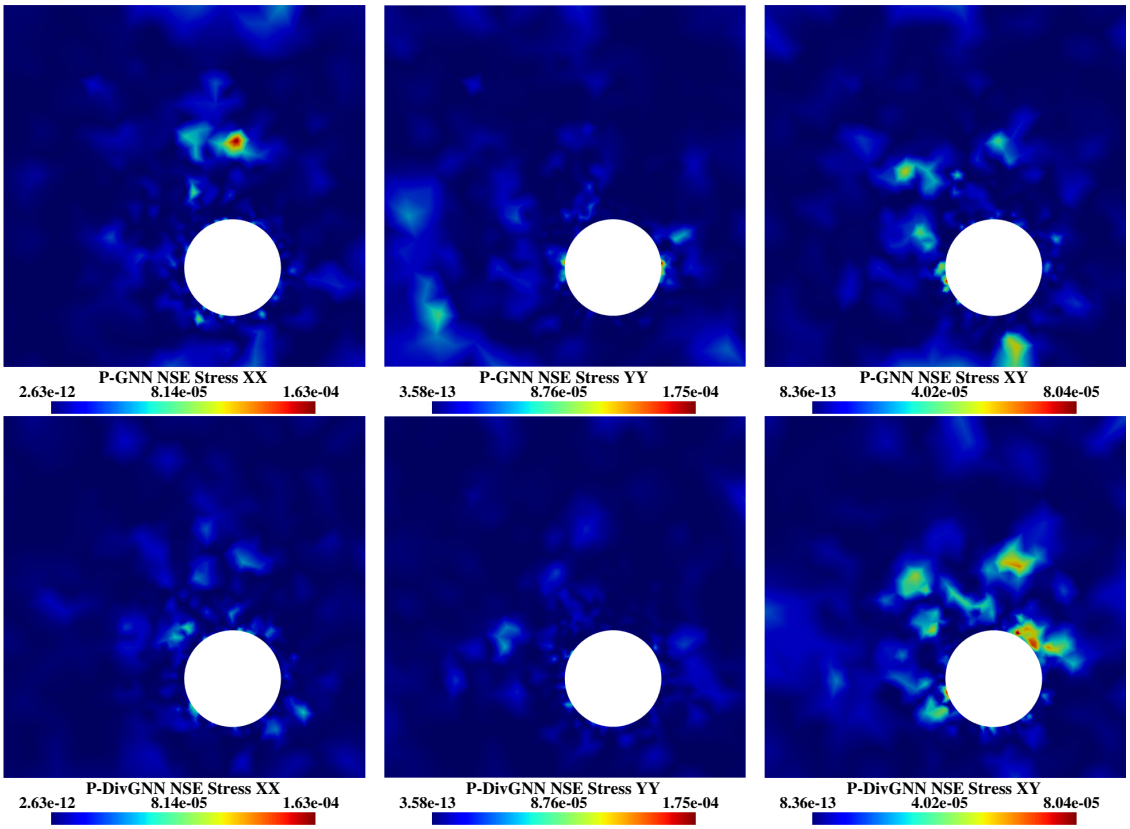


Figure 16: NSE between predicted local stress field and FEM stress field for both *P-GNN* and *P-DivGNN*, error fields are plotted at node level.

To complement the qualitative assessment, Figure 16 displays the corresponding node-level NSE error fields. The results indicate that *P-DivGNN* yields lower local errors compared to *P-GNN*, particularly in regions where the latter produces more irregular or noisy stress predictions. These observations confirm the benefit of incorporating a divergence constraint, especially in more complex, non-linear mechanical regimes.

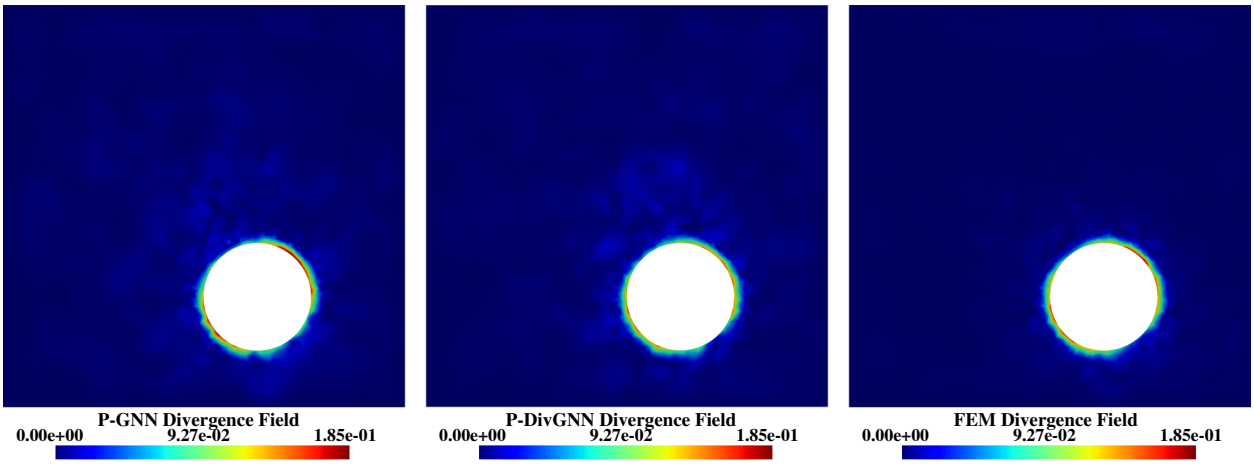


Figure 17: Comparison of norm of divergence field of local stress field between *P-GNN*, *P-DivGNN* and FEM predictions for a non-linear hyperelastic case.

Figure 17 shows the divergence norm fields corresponding to the nonlinear hyperelastic stress predictions displayed in Figure 15. The divergence is computed for the stress fields predicted by *P-GNN* and *P-DivGNN*, as well as for the

FEM reference solution. As in the linear case, the *P-DivGNN* exhibits noticeably lower divergence values than the baseline model, particularly around the hole boundary where stress gradients are most pronounced. While FEM still achieves slightly lower divergence values in the nodes near the hole surface, the *P-DivGNN* yields divergence values on the hole boundary itself that are even lower than those produced by FEM.

Table 2: Comparison of divergence mean value and NMSE between *P-GNN*, *P-DivGNN*, and FEM, evaluated on the whole non linear hyperelastic test dataset, the model *P-DivGNN* achieves lower divergence values than *P-GNN* and FE simulation.

	Divergence	**NMSE**
P-GNN	3.55×10^{-2}	5.36×10^{-2}
P-DivGNN	$\mathbf{1.56 \times 10^{-3}}$	$\mathbf{2.18 \times 10^{-2}}$
FEM	1.69×10^{-3}	–

Results computed on the non linear hyperelastic test dataset.

Table 2 presents a quantitative comparison of the mean divergence and normalized mean squared error (NMSE) over the entire test set for the nonlinear hyperelastic dataset. The results confirm the advantage of incorporating a divergence-free penalty during training. Specifically, the proposed model *P-DivGNN* achieves the lowest divergence value, outperforming both the baseline *P-GNN* and even the FEM solution. This is a notable result, as it indicates that the model not only approximates the ground truth but also enforces physical consistency more effectively than the numerical solver under the same conditions. Additionally, the *P-DivGNN* model yields significantly lower NMSE compared to *P-GNN*, demonstrating improved fidelity in reconstructing the local stress fields. These metrics, taken together with the visual and error-field comparisons discussed earlier, provide robust evidence of the benefits introduced by the divergence-aware loss formulation.

5.3.1 Benchmark

Figure 18 presents the same benchmark test as in Figure 12, but applied to the reconstruction of a local stress field in a nonlinear hyperelastic problem. In this case, the FEM solution requires an iterative Newton–Raphson scheme which introduces variability in computation time regarding the mesh geometry and input strain tensors, the fluctuations on the FEM computation reported times are explained arise from adaptive time-stepping: When the convergence criteria are not met, the solver automatically reduces the time step, increasing the number of iterations required for getting the FEM solution.

The proposed model achieves a significant computational speed-up of approximately $\approx$ 500x primarily due to the absence of using an iterative scheme and the use of highly parallelized GPU matrix operations. The computation time of GNN scales similarly to the computation time of FEM in function of the number of nodes. It is important to note that the batch prediction capability of neural networks, allowing multiple problems to be solved simultaneously, was not utilized in this study. Therefore, the reported speed-up represents a conservative estimate. In a full FE² setting, where multiple RVEs are evaluated concurrently, even greater computational gains can be expected.

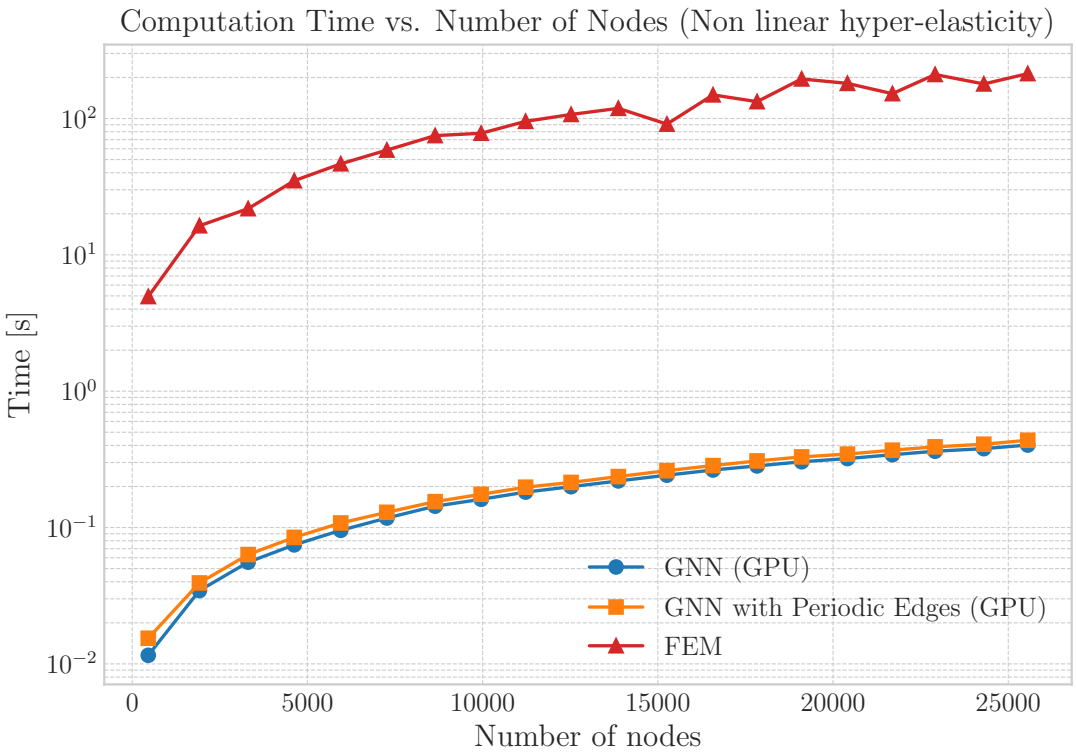


Figure 18: Performance comparison between hyperelastic FEM simulation *P-DivGNN* (red), *P-DivGNN* when adding periodic edges (orange) and without (blue). Tested using a CPU Intel Xeon W-2255 Processor, 10 cores, 20 threads for FEM simulation and GNN tests have been produced using a graphics card (GPU) Nvidia RTX A4000 with 16GB of VRAM where only 156MB were used.

6 Conclusion and perspectives

In this study, we developed a novel framework for reconstructing local stress fields from homogenized mean stress values obtained through ROM simulations. The proposed approach, named *P-DivGNN*, leverages GNNs trained using a physics-informed loss function. This loss enforces equilibrium constraints during the learning process, ensuring that the predicted local stress fields minimizes the equilibrium condition $\operatorname{div}\sigma = 0$. The framework enables the localization of stress fields within a representative unit cell (RUC) of a periodic microstructure. To achieve this, a graph is constructed from the input periodic mesh, incorporating periodicity by adding edges that connect opposing nodes of the mesh. Our results demonstrate that the inclusion of these periodic edges significantly enhances the accuracy of the predicted stress fields compared to a standard mesh to graph representation.

To evaluate the accuracy and robustness of *P-DivGNN*, we conducted tests on two distinct mechanical scenarios, both involving two-dimensional plate geometries with a central hole. In each case, the hole's position, radius, and mesh refinement were independently sampled from uniform distributions to introduce geometric variability. The first test case focused on linear elastic material behavior and was used to validate the model's predictive performance. The second test case involved a nonlinear hyperelastic material model and served to assess the model's ability to reconstruct complex stress fields under large deformations. Additionally, this nonlinear scenario highlights the significant computational speed-up achieved by the proposed method compared to conventional finite element simulations.

The results indicate that *P-DivGNN* generalizes well to unseen hole plate mesh configurations, successfully handling variations in node positions and mesh refinement. Comparative analysis with FE simulations revealed that *P-DivGNN* is competitive in terms of both stress field reconstruction accuracy and computational efficiency. The impact of the physics-informed loss was further analyzed by comparing the divergence of the predicted stress fields with that of the FE-generated fields. On average, *P-DivGNN* produces stress fields with lower divergence, indicating that, in some cases from the test database, the equilibrium state is better maintained using *P-DivGNN* than with conventional FE simulations.

In future work, it would be valuable to explore the benefits of imposing a divergence-free constraint when localizing stress fields in the context of plasticity. Additionally, extending the proposed framework to handle more complex geometries and 3D cases could further expand its applicability. For temporal sequences of linearized increments, this approach could be adapted with minimal modifications, offering the potential for significant gains in computational efficiency.

7 Code availability

The source code allowing to generate the datasets used and to reproduce the experiments presented in this article can be found at:
https://github.com/ricardo0115/p-div-gnn

Author contributions

Manuel Ricardo Guevara Garban: Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Conceptualization. **Étienne Prulière**: Writing – review and editing, Supervision, Software, Resources, Methodology, Conceptualization. **Michaël Clément**: Writing – review and editing, Supervision, Software, Resources, Methodology, Conceptualization. **Yves Chemisky**: Writing – review and editing, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Financial disclosure

None reported.

Conflict of interest

The authors declare no potential conflict of interests.

References

[1] Frédéric Feyel. Multiscale fe2 elastoviscoplastic analysis of composite structures. *Computational Materials Science*, 16(1):344–354, 1999.

[2] Frédéric Feyel. A multilevel finite element method (fe2) to describe the response of highly non-linear structures using generalized continua. 192(28):3233–3244, 2003.

[3] Nils Lange, Geralf Hütter, and Björn Kiefer. An efficient monolithic solution scheme for fe2 problems. 382:113886, 2021.

[4] Jianmin Qu and Mohammed Cherkaoui. *Fundamentals of Micromechanics of Solids*. Wiley, 1 edition, August 2006.

[5] George Dvorak. Transformation field analysis of inelastic composite materials. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 437(1900):311–327, May 1992.

[6] J.C. Michel and P. Suquet. Nonuniform transformation field analysis. *International Journal of Solids and Structures*, 40(25):6937–6955, December 2003.

[7] S. Roussette, J.C. Michel, and P. Suquet. Nonuniform transformation field analysis of elastic–viscoplastic composites. *Composites Science and Technology*, 69(1):22–27, January 2009.

[8] Aymen Danoun, Etienne Prulière, and Yves Chemisky. Fe-lstm: A hybrid approach to accelerate multiscale simulations of architectured materials using recurrent neural networks and finite element analysis. 429:117192, 2024.

[9] Zeliang Liu, M.A. Bessa, and Wing Kam Liu. Self-consistent clustering analysis: An efficient multi-scale scheme for inelastic heterogeneous materials. 306:319–341, 2016.

[10] Anindya Bhaduri, Yanyan He, Michael D. Shields, Lori Graham-Brady, and Robert M. Kirby. Stochastic collocation approach with adaptive mesh refinement for parametric uncertainty analysis. *Journal of Computational Physics*, 371:732–750, 2018.

[11] Anindya Bhaduri, Christopher S. Meyer, John W. Gillespie, Bazle Z. Gama Haque, Michael D. Shields, and Lori Graham-Brady. Probabilistic modeling of discrete structural response with application to composite plate penetration models. *Journal of Engineering Mechanics*, 147(11):04021087, 2021.

[12] Ehsan Haghighat, Maziar Raissi, Adrian Moure, Hector Gomez, and Ruben Juanes. A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. 379:113741, 2021.

[13] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[14] Zhenguo Nie, Haoliang Jiang, and Levent Burak Kara. Stress field prediction in cantilevered structures using convolutional neural networks. *Journal of Computing and Information Science in Engineering*, 20(1):011002, 2019.

[15] Ashwini Gupta, Anindya Bhaduri, and Lori Graham-Brady. Accelerated multiscale mechanics modeling in a deep learning framework. *Mechanics of Materials*, 184:104709, 2023.

[16] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI*, 2015.

[17] Brendan P. Croom, Michael Berkson, Robert K. Mueller, Michael Presley, and Steven Storck. Deep learning prediction of stress fields in additively manufactured metals with intricate defect networks. *Mechanics of Materials*, 165:104191, 2022.

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014.

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[20] Zhenze Yang, Chi-Hua Yu, Kai Guo, and Markus J. Buehler. End-to-end deep learning method to predict complete strain and stress tensors for complex hierarchical composite microstructures. *Journal of the Mechanics and Physics of Solids*, 154:104506, 2021.

[21] Yayati Jadhav, Joseph Berthel, Chunshan Hu, Rahul Panat, Jack Beuth, and Amir Barati Farimani. Stressd: 2d stress estimation using denoising diffusion model. 416:116343, 2023.

[22] Junyan He, Seid Koric, Diab Abueidda, Ali Najafi, and Iwona Jasiuk. Geom-deeponet: A point-cloud-based deep operator network for field predictions on 3d parameterized geometries. 429:117130, 2024.

[23] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Trans. Neural Networks*, 20(1):61–80, 2009.

[24] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations, (ICLR)*, 2021.

[25] Marco Maurizi, Chao Gao, and Filippo Berto. Predicting stress, strain and deformation fields in materials and structures with graph neural networks. *Scientific Reports*, 12, 2022.

[26] J. Storm, I.B.C.M. Rocha, and F.P. van der Meer. A microstructure-based graph neural network for accelerating multiscale simulations. 427:117001, 2024.

[27] Meire Fortunato, Tobias Pfaff, Peter Wirnsberger, Alexander Pritzel, and Peter W. Battaglia. Multiscale meshgraphnets. In *2nd AI4Science Workshop at the 39th International Conference on Machine Learning (ICML)*, 2022.

[28] M. Raissi, P. Perdikaris, and G.E. 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.

[29] Masanobu Horie and Naoto Mitsume. Physics-embedded neural networks: Graph neural PDE solvers with mixed boundary conditions. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[30] Quercus Hernández, Alberto Badías, Francisco Chinesta, and Elías Cueto. Thermodynamics-informed graph neural networks. *IEEE Transactions on Artificial Intelligence*, 5(3):967–976, 2024.

[31] Jack Richter-Powell, Yaron Lipman, and Ricky T. Q. Chen. Neural conservation laws: A divergence-free perspective. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[32] J. D. Eshelby. The Determination of the Elastic Field of an Ellipsoidal Inclusion, and Related Problems. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 241(1226):376–396, 1957.

[33] T. Mura. Micromechanics of Defects in Solids. In *Mechanics of elastic and inelastic solids*. Kluwer Academic Publishers, Dordrecht, Second, Revised edition, 1987.

[34] J. Qu and M. Cherkaoui. *Fundamentals of Micromechanics of Solids*. Wiley, New Jersey, 2006.

[35] Z. Hashin. The elastic moduli of heterogeneous materials. *Journal of Applied Physics*, 29(1):143–150, 1962.

[36] Z. Hashin and B. W. Rosen. The elastic moduli of fiber-reinforced materials. *Journal of Applied Mechanics*, 31:223–232, 1964.

[37] R. M. Christensen and K. H. Lo. Solutions for effective shear properties in three phase sphere and cylinder models. *Journal of the Mechanics and Physics of Solids*, 27:315–330, 1979.

[38] M. J. Pindera, H. Khatam, A. S. Drago, and Y. Bansal. Micromechanics of spatially uniform heterogeneous media: A critical review and emerging approaches. *Composites Part B: Engineering*, 40(5):349–378, 2009.

[39] G. Chatzigeorgiou, N. Charalambakis, Y. Chemisky, and F. Meraghni. Periodic homogenization for fully coupled thermomechanical modeling of dissipative generalized standard materials. *International Journal of Plasticity*, 81:18–39, 2016.

[40] H Moulinec and P. Suquet. A numerical method for computing the overall response of nonlinear composites with complex microstructure. *Computer methods in applied mechanics and englneering*, 157:69–94, 1998.

[41] A. L. Kalamkarov and A. G. Kolpakov. *Analysis, design and optimization of composite structures*. Wiley, West Sussex, 1997.

[42] R. Rodríguez-Ramos, F. J. Sabina, R. Guinovart-Díaz, and Bravo-Castillero. Closed-form expressions for the effective coefficients of a fiber-reinforced composite with transversely isotropic constituents - I. Elastic and square symmetry. *Mechanics and Materials*, 33(4):223–235, 2001.

[43] R. Guinovart-Díaz, J. Bravo-Castillero, R. Rodríguez-Ramos, and F. J. Sabina. Closed-form expressions for the effective coefficients of fibre-reinforced composite with transversely isotropic constituents. I: Elastic and hexagonal symmetry. *Journal of the Mechanics and Physics of Solids*, 49(7):1445–1462, 2001.

[44] G. Wang and M. J. Pindera. Locally-exact homogenization of unidirectional composites with coated or hollow reinforcement. *Materials and Design*, 93:514–528, 2016.

[45] G. Wang and M. J. Pindera. Locally exact homogenization of unidirectional composites with cylindrically orthotropic fibers. *Journal of Applied Mechanics*, 83:071010, 2016.

[46] J. C. Simo and T. J. R. Hughes. *Computational Inelasticity*. Springer-Verlag, New York, 1998.

[47] Georges Chatzigeorgiou, Nicolas Charalambakis, Yves Chemisky, and Fodil Meraghni. *Thermomechanical Behavior of Dissipative Composite Materials*. ISTE Press, 2018.

[48] H. I. Ene. On linear thermoelasticity of composite materials. *International Journal of Engineering Science*, 21(5):443–448, 1983.

[49] Pierre M. Suquet. Elements of homogenization for inelastic solid mechanics. In Enrique Sanchez-Palencia and André Zaoui, editors, *Homogenization Techniques for Composite Media*, pages 193–198, 1987.

[50] E. Tikarrouchine, G. Chatzigeorgiou, Y. Chemisky, and F. Meraghni. Fully coupled thermo-viscoplastic analysis of composite structures by means of multi-scale three-dimensional finite element computations. *International Journal of Solids and Structures*, 164:120–140, 2019.

[51] C.H. Liu, G. Hofstetter, and H.A. Mang. 3d finite element analysis of rubber-like materials at finite strains. *Engineering Computations*, 11(2):111–128, 1994.

[52] Etienne Prulière and Yves Chemisky. 3MAH : Un ensemble de librairies pour analyser le comportement complexe de matériaux hétérogènes. In *Colloque National en Calcul des Structures (CSMA)*, 2023.

[53] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.

[54] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[55] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv:1903.02428*, abs/1903.02428, 2019.