Mechanics of Materials 219 (2026) 105733

Contents lists available at ScienceDirect

Mechanics of Materials

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

Research paper

Predicting plastic strain localization in porous solids using graph neural networks

Rasmus B. Jakobsen [a,1], Tobias S. Kristensen [a,1], Joep Storm [b], Iuri B.C.M. Rocha [b], Tito Andriollo [a,*]

[a] Department of Mechanical and Production Engineering, Aarhus University, Katrinebjergvej 89 G-F, 8200, Aarhus N, Denmark
[b] Department of Civil Engineering and Geosciences, Delft University of Technology, P.O. Box 5048, 2600 GA, Delft, Zuid-Holland, The Netherlands

ARTICLE INFO

Invited Editor Ghatu Subhash

Dataset link: https://github.com/tobi2889/gnn-plastic-strain-localization

Keywords:
Surrogate modeling
Graph neural network
Plasticity
Heterogeneous material
Porous solid
Strain localization

ABSTRACT

We develop graph-based surrogate models to predict strain localization and collapse load in 2D elastic-perfectly plastic porous solids. Plastic deformation is represented using a Delaunay graph, where nodes correspond to void centers, edges represent potential shear bands, and edge values encode the local integral of the plastic work rate (PWR). Two edge-regression graph neural network (GNN) models are built. The first is a purely data-driven model (DDM) that maps local geometric features – edge length and orientation – to PWR. The second is a hybrid model (HM) that augments the GNN with a mechanistic prior from a limit load analysis model (LLAM) and learns a correction to its upper-bound bias. The models are trained on a dataset of 1200 representative volume elements with 20% porosity. The DDM can reliably reconstruct shear band patterns, but its accuracy deteriorates when applied to porosity values different from that seen in training. The HM, like the LLAM, does not fully capture the spatial pattern, but it removes the LLAM systematic bias and achieves uniformly low errors in total work rate for porosities ranging from 10% to 30%. Moreover, the HM requires less training data than the DDM and is more robust across random seeds. These results show that coupling a GNN with a physics-based prior yields a fast and data-efficient surrogate that preserves accuracy in macroscopic quantities while retaining meaningful spatial information, thereby offering a practical route to predict the collapse of porous solids while accounting for the exact locations of a very large number of voids.

1. Introduction

Accurate prediction of the mechanical response of heterogeneous materials is crucial for designing components and structures that are stronger, safer, and more sustainable (Wu and Fan, 2020; Zhu and Wu, 2023). Over the past decades, a wide range of techniques have been developed, from analytical homogenization methods to advanced multiscale numerical schemes (Gurson, 1977; Suquet, 1987; Matouš et al., 2017). Most of these approaches rely on the principle of scale separation, which enables the decomposition of the problem into a macroscale and a microscale level (Matouš et al., 2017). This assumption is valid when the characteristic microstructural length is much smaller than the length scale over which macroscopic fields vary. However, there are situations where classical homogenization methods become suboptimal — either because scale separation does not hold, or because accounting for microstructural variability makes the analysis prohibitively expensive (Yvonnet et al., 2020).

An example is the prediction of strain localization in porous ductile solids with a non-uniform pore distribution. Experimental and numerical studies have shown that plastic deformation in porous ductile solids tends to concentrate along narrow paths connecting neighboring voids, commonly referred to as strain localization bands or shear bands (Fritzen et al., 2012; Buljac et al., 2018; Xu et al., 2020; Cadet et al., 2022). The resulting shear band pattern is strongly influenced by the overall geometry and loading conditions, implying that, in principle, modeling the entire microstructure is necessary to accurately capture the mechanical response. In practical applications, the pore distribution can be obtained from advanced characterization techniques such as scanning electron microscopy or X-ray computed tomography (Withers et al., 2021). However, experimental datasets acquired in this way may contain thousands of voids. As a result, models of this kind can quickly become computationally prohibitive even for relatively small parts if the number of pores per unit volume is large.

* Corresponding author.
E-mail address: titoan@mpe.au.dk (T. Andriollo).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.mechmat.2026.105733
Received 26 January 2026; Received in revised form 27 March 2026; Accepted 14 May 2026
Available online 23 May 2026

To alleviate the computational burden, a promising strategy is to leverage prior knowledge of the deformation pattern in order to simplify the problem. An early application of this concept was given by Pijnenburg and Van Der Giessen (2003), who employed a fully resolved microstructural model while restricting the region where the strain localization could develop. Specifically, they modeled polymers with cavitated particles by using the particle centers as the nodes of a triangular finite element (FE) mesh and inserting cohesive elements along the mesh edges. The bulk elements captured the elastic response of the polymer matrix, while the cohesive element captured the plastic localization associated with the shear bands. This approach significantly reduces the computational cost of modeling the microstructure, but it still requires solving a nonlinear FE problem.

To further reduce the computational complexity, the concept of microstructure-based limit load analysis was recently introduced by Hund et al. (2025). The key idea is to exploit the tendency of shear bands to connect neighboring voids to enable the application of a simplified version of the discontinuity layout optimization (DLO) method. The DLO method (Sloan and Kleeman, 1995; Smith and Gilbert, 2007; Gilbert et al., 2014; Smith et al., 2019), which has mainly found application in geotechnical engineering, assumes elastic-perfectly plastic material behavior to exploit the power of limit load analysis. It consists in partitioning the domain into rigid blocks, which may slip relative to one another, thus creating velocity discontinuities, i.e. shear bands, along the block edges. The block velocities are computed by solving a linear programming (LP) problem, where the quantity to be minimized is the total plastic work rate (PWR) along the discontinuities, and the constraints consist of kinematic relationships enforcing no penetration between adjacent blocks. In general, the block layout is part of the solution, meaning that multiple LP problems must be solved to find the layout that provides the minimum PWR. However, in the case of porous solids the optimal block layout can be assumed to be known in advance, since shear bands are expected to form by linking neighboring voids. As shown by Hund et al. (2025), this reduces the analysis to solving a single LP problem, resulting in highly efficient algorithms for predicting shear band patterns and the maximum load-carrying capacity of the material. However, the same authors demonstrated that the approach can suffer from suboptimal accuracy, as evidenced by a systematic overestimation of the macroscopic limit load, which is likely attributable to the restrictive kinematic assumptions.

To tackle the abovementioned systematic overestimation, the present work investigates an alternative strategy based on data-driven surrogate models. In fact, data-driven surrogates have emerged as powerful alternatives to fully resolved FE simulations, particularly in multiscale mechanics where direct modeling of the microscale is prohibitively expensive. Examples of these surrogates are recurrent neural networks (Ghavamian and Simone, 2019; Wu et al., 2020), temporal convolutional networks (Wang et al., 2022; Abueidda et al., 2021), convolutional neural networks (Krokos et al., 2022; Gupta et al., 2023; Abueidda et al., 2019; Aldakheel et al., 2023), and graph neural networks (GNNs) (Storm et al., 2024; Maurizi et al., 2022; Vlassis and Sun, 2023; Frankel et al., 2022; Hendriks et al., 2025; Karapiperis and Kochmann, 2023; Krokos et al., 2024). GNNs are well suited to the present problem, particularly in the simplified 2D case, since the plastic deformation pattern can be naturally represented as a graph whose nodes correspond to void centers and edges to shear bands (Andriollo et al., 2022).

GNNs have already been used successfully in several areas of mechanics, such as solid mechanics, fluid mechanics, and interdisciplinary mechanics-related domains (Zhao et al., 2024). Several studies have also demonstrated the capability of GNNs to serve as accurate and efficient surrogates for mechanical simulations at the microstructural level. Examples include a GNN that predicts full-field microscopic strains with stresses obtained from physics-based constitutive models to accelerate multiscale simulations (Storm et al., 2024); a GNN that predict deformation, stress, and strain fields while capturing nonlinear effects such as linear hardening plasticity, wrinkling, and buckling (Maurizi et al., 2022); a graph-based autoencoder that learns low-dimensional graph embeddings of the plastic patterns (Vlassis and Sun, 2023); a GNN that predicts the crack propagation in cellular materials (Karapiperis and Kochmann, 2023); a similarity-equivariant GNN to predict global quantities and pattern transformations in hyperelastic, two-dimensional, microporous materials (Hendriks et al., 2025); and extensions to predicting the stress evolution in 3D porous materials using Bayesian and convolutional GNNs (Krokos et al., 2024; Frankel et al., 2022). However, while these studies have mainly focused on node-level predictions, much less attention has been paid to edge-level regression, where GNNs are used to predict continuous variables associated with graph edges.

In this work, we extend the GNN methodology to the relatively unexplored domain of edge-level regression and use it to predict the PWR along the edges of a graph constructed by linking neighboring voids, which are assumed to correspond to potential shear bands. This allows us to not only identify which shear bands are active but also to quantify their dissipation, providing a richer characterization of the plastic deformation process in the porous solid. To mitigate demands in terms of training data and improve generalization across void fractions, we also introduce a hybrid surrogate that augments the GNN with a physics-based prior. The prior is the microstructure-based limit load analysis model of Hund et al. (2025), and the hybrid model essentially learns to correct this reduced-order model.

The remainder of this article is structured as follows: In Section 2, a precise statement of the problem to be addressed is presented along with the method for generating the required training data. In Sections 3 and 4 the methodology and architecture of the GNN-based surrogate models are presented. The first model (Section 3) is a data-driven model (DDM), which learns the relationship between geometric features and the corresponding PWRs. The second model (Section 4) is a hybrid model (HM) where additional input features from a microstructure-based limit analysis are provided. In Section 5, the results obtained from the two approaches are benchmarked in terms of training efficiency and generalizability towards other void fractions (VFs). Finally, some concluding remarks are given in Section 6.

2. Problem definition

2.1. Assumptions and objectives

Following Hund et al. (2025), we consider a 2D porous solid consisting of an elastic–perfectly plastic matrix and circular voids of radius R. Plastic deformation in the matrix is governed by an associated flow rule based on the von Mises yield criterion with yield stress σ_y. The solid is modeled via a square representative element (RE) with edge length $L = 60R$ and fixed void volume fraction (VF) of 20% (VF20). The voids are placed randomly in the RE and a minimum void-to-void distance $d_{\text{edge}} = 0.5R$ is enforced. Periodic boundary conditions (PBCs) are applied to all sides of the RE, enabling modeling of an effectively infinite porous medium. The RE is subjected to uniaxial tensile loading under generalized plane strain until limit load conditions are attained.

Under these assumptions, the PWR localizes in shear bands, which can be mapped onto the edges of the Delaunay triangulation generated by the void centers as shown by Andriollo et al. (2022). This Delaunay-based approximation is considered reasonable for VFs in the range of 10%–30%, whereas for significantly lower fractions (e.g., 1%) the triangulation no longer provides a reliable representation of the shear-band network. The problem objectives are defined as:

1. *Predicting the shear band pattern:* Identify which potential shear bands, represented by the edges of the Delaunay triangulation, become active by quantifying the PWR of each edge.
2. *Quantifying the total rate of internal work:* Estimate the magnitude of plastic dissipation along all the active shear bands to determine the total rate of internal work of the RE for assessing the macroscopic load-carrying capacity of the solid.

It is worth noting that the first objective is more stringent than the second. Once the edge-wise PWR distribution is known, the total rate of internal work of the RE follows directly by summation over all edges, so Objective 1 implies Objective 2. The converse, however, does not hold. Different shear band patterns may lead to the same global dissipation, meaning that knowledge of the total rate of internal work alone is insufficient to uniquely determine the underlying localization pattern.

As we aim to predict the limit state of the solid, we are not concerned with creating a surrogate for general FE simulations. Full-field displacements, stress–strain paths and elastic properties are thus out of the scope of this work. Similarly, the relatively narrow constraints on the problem setup (such as circular pores and uniaxial tension) allow us to focus on comparing the data-driven aspects to the existing limit load analysis model without modifications. Future work could focus on relaxing these assumptions and constraints. For example, while the limit analysis theory used here is invalid for strain-hardening materials, we suspect that the surrogate models could still learn to make accurate predictions for them.

2.2. Data generation

To train and evaluate the surrogate models, a graph dataset is generated. The procedure is concisely described here. For a more detailed overview, we refer to Hund et al. (2025). First, a total of 1200 unique REs are created. For each RE, the predicted PWR field at limit load is computed using the FE software Abaqus (Systèmes, 2021) and exported on a 1200×1200 pixel grid (see Fig. 1a). Pixel values are then assigned to the edges of the Delaunay triangulation constructed from the void location. For this purpose, each Delaunay triangle is subdivided into three sub-triangles by connecting its centroid to the vertices. For any given edge, the edge-wise PWR $\dot{w}$ is computed as the discrete area integral of the pixel-wise PWR over the two sub-triangles adjacent to that edge (see Fig. 1b).

The resulting graph dataset consists of nodes representing the void centers in the microstructure, and edges representing the potential shear bands from the Delaunay triangulation, with each edge associated with a PWR value $\dot{w}$. We remark that by assigning a continuous PWR value to each potential shear band, there is no need to introduce a threshold to distinguish between active and inactive bands. For consistency and comparability, the $\dot{w}$ is scaled by $(R\sigma_y\dot{\epsilon})$, where $\dot{\epsilon}$ is the assumed strain rate at the macroscale, while the edge length l is scaled by the void radius R. The dataset is split 75%/15%/10% for training, validation, and test purposes, respectively.

Additionally, an extended dataset is created by considering 50 REs for each of the following VFs: 10% (VF10), 15% (VF15), 25% (VF20), 30% (VF30). This is done to test the ability of the surrogate models to generalize to other VFs without re-training them.

The FE simulations used for data generation of the 1200 REs with VF20 are performed in Abaqus 2022 (Systèmes, 2022), and are executed on the DelftBlue high-performance cluster at Delft University of Technology (Delft High Performance Computing Centre (DHPC), 2024). The FE simulations of the 50 REs for each of the other four VFs (VF10, VF15, VF25, VF30) are performed in Abaqus 2021 (Systèmes, 2021), and are executed on the AU PRIME high-performance cluster at Aarhus University (Aarhus University Prime, 2025).

3. Data-driven model

3.1. Graph fundamentals

A graph, $G = (\mathcal{V}, \mathcal{E})$, is a data structure that consists of nodes, $\mathcal{V}$, and edges, $\mathcal{E}$. The nodes represent entities, while the edges represent relationships between these entities. An edge that links node $n \in \mathcal{V}$ to node $m \in \mathcal{V}$ is denoted $(n, m) \in \mathcal{E}$. Edges can be directed or undirected. A directed edge only transfers information from one node to another, while an undirected edge transfers information between both nodes. This is equivalent to having two opposite-directed edges, meaning that if $(n, m) \in \mathcal{E}$, then $(m, n) \in \mathcal{E}$. If two edges share the same node, they are called neighbors. The set of all neighboring edges that share the same node n is denoted by $\mathcal{N}(n)$. Information of a graph can be in terms of node features, $\mathbf{x}_n$, edge features, $\mathbf{x}_{(n,m)}$, but also at the level of the graph itself, depending on the application.

3.2. Edge-regression graph neural network

The DDM considered in this work is based on a GNN, which is a type of artificial neural network designed to operate on graph data (Hamilton, 2020; Bishop and Bishop, 2024). The objective of a GNN is to learn relationships between the input features, i.e., the quantities provided to the network for each node and edge, and the target variables, i.e., the quantities the network is trained to predict. In the present case, both the target and input features are defined at the edges. Specifically, the targets are the edge-wise PWR values obtained from the FE simulations, while the input features consist of the edge length and its orientation relative to the global loading direction.

Training a GNN to predict a continuous variable at the edge level is referred to as edge regression. The goal of edge regression is to learn edge embeddings, which serve as compact, information-rich representations of the edges. These embeddings are generated by encoding the edge features through a Multi-Layer Perceptron (MLP) as

$$\mathbf{h}^{(0)}_{(n,m)} = \mathrm{MLP}_{encode}(\mathbf{x}_{(n,m)}) \tag{1}$$

where $\mathbf{x}_{(n,m)}$ and $\mathbf{h}_{(n,m)}$ denote the features and encoded embedding, respectively, for edge (n, m).

Initially, the embeddings contain information only about the individual edge they represent. To incorporate information about the broader graph, message passing layers (MPLs) are used. During each message-passing iteration k, the embedding $\mathbf{h}^{(k)}_{(n,m)}$ of each edge $(n, m) \in \mathcal{E}$ is updated using a three-step procedure. First, all embeddings are passed through a MLP to compute an edge message $\mathbf{m}^{(k)}_{(n,m)}$:

$$\mathbf{m}^{(k)}_{(n,m)} = \mathrm{MLP}^{(k)}_{message}\left(\mathbf{h}^{(k)}_{(n,m)}\right) \tag{2}$$

Next, the messages associated with edges that share a common node are aggregated into one node message $\mathbf{m}^{(k)}_{\mathcal{N}(n)}$:

$$\mathbf{m}^{(k)}_{\mathcal{N}(n)} = \sum_{m \in \mathcal{N}(n)} \mathbf{m}^{(k)}_{(n,m)} \tag{3}$$

Finally, each embedding $\mathbf{h}^{(k)}_{(n,m)}$ is updated by concatenating it with the node messages $\mathbf{m}^{(k)}_{\mathcal{N}(n)}$ and $\mathbf{m}^{(k)}_{\mathcal{N}(m)}$, and passing the concatenated vector through a MLP. To reduce over-smoothing, a problem that can occur when embeddings become indistinguishable as a result of too many MPLs, the MLP output is added to the original embedding (residual connection). Accordingly, the updated embedding is computed as

$$\mathbf{h}^{(k+1)}_{(n,m)} = \mathrm{MLP}^{(k)}_{update}\left(\mathbf{h}^{(k)}_{(n,m)} \oplus \mathbf{m}^{(k)}_{\mathcal{N}(n)} \oplus \mathbf{m}^{(k)}_{\mathcal{N}(m)}\right) + \mathbf{h}^{(k)}_{(n,m)} \tag{4}$$

where the symbol $(\oplus)$ indicates the concatenation operation.

After running K iterations of the MPLs, the final embedding of an edge can be decoded. The decoding is performed by passing the final embedding $\mathbf{h}^{(K)}_{(n,m)}$ through a MLP to compute the predicted PWR as

$$\hat{\dot{w}}^{pred}_{(n,m)} = \mathrm{MLP}_{decode}\left(\mathbf{h}^{(K)}_{(n,m)}\right) \tag{5}$$

Each edge in our model is undirected. For implementation reasons, this is handled by representing each undirected edge as two opposite directed edges, which results in two predictions: one for the edge (n, m) and one for the edge (m, n). When the edge is flipped from (n, m) to (m, n), so is the source and target node in the concatenation. The $\mathrm{MLP}^{(k)}_{update}$ is therefore fed with two different vectors, leading to two slightly

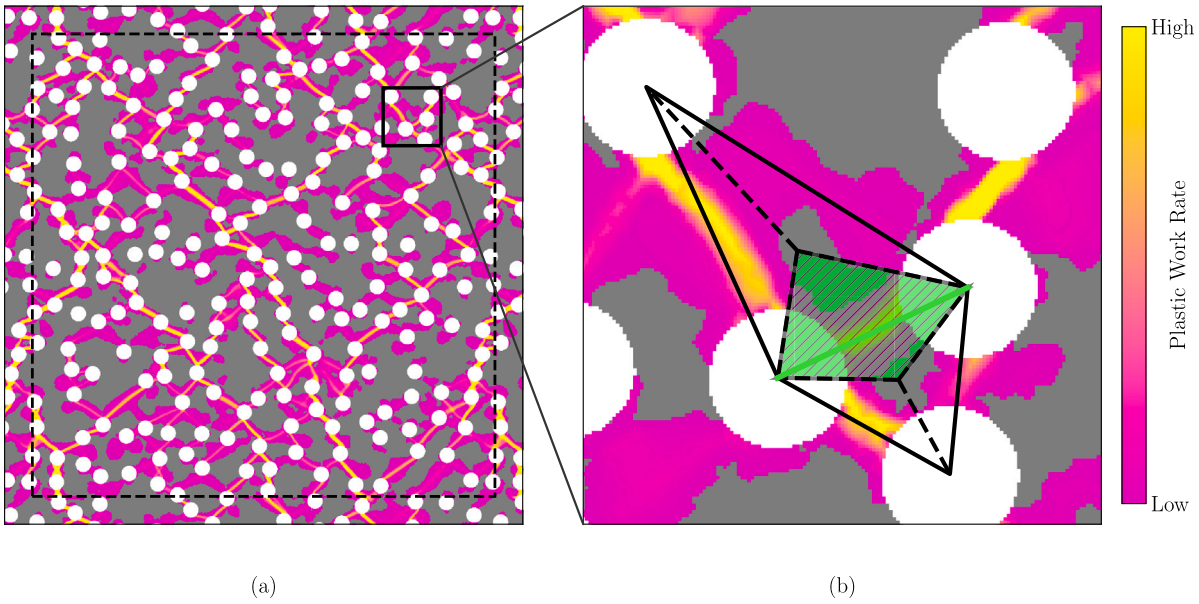

Fig. 1. (a) Example of RE microstructure with overlaid PWR field, periodically extended beyond the RE boundary (dashed bounding box). (b) Two Delaunay triangles (solid black) sharing an edge e (green), and related sub-triangles defined by centroid–vertex splits (dotted black). The two sub-triangles adjacent to e (hatched green) define the area over which the PWR is summed to obtain the PWR associated with e.

different outputs. To obtain a unique value, we compute the mean of the two predictions:

$$\dot{w}^{pred}_{(n,m)} = \frac{1}{2}\left(\hat{\dot{w}}^{pred}_{(n,m)} + \hat{\dot{w}}^{pred}_{(m,n)}\right) \tag{6}$$

The complete architecture of the DDM is illustrated in Fig. 2. The model is trained by minimizing the error between the predicted and true PWR values in the training data, quantified by the following loss function based on the Mean Squared Error (MSE):

$$\mathcal{L}_{MSE} = \frac{1}{|\mathcal{E}|}\sum_{(n,m)\in\mathcal{E}}\left(\dot{w}^{true}_{(n,m)} - \dot{w}^{pred}_{(n,m)}\right)^2 \tag{7}$$

where $\dot{w}^{\text{true}}_{(n,m)}$ denotes the target value at edge (n, m).

3.3. Physical considerations

When designing graph-based models for physical systems, it is important to consider which symmetries the model should respect. These symmetries determine which transformations of the input data should leave the output unchanged. For the present DDM, permutation invariance is fundamental, as reordering the nodes by changing their numbering in the Delaunay triangulation must not affect the predictions. This property is satisfied by aggregating information from neighbors independently of index order, see Eq. (3). Similarly, the model should be translation invariant, since shifting the entire microstructure in space should not alter the mechanical response. Accordingly, the voids are not labeled with coordinates, as only their relative position matters.

Rotational transformation, however, requires more careful consideration. The system studied here is subjected to a uniaxial tensile load applied in a fixed, global direction, which introduces an inherent anisotropy. Consequently, the response of the RE depends not on the absolute orientation of the geometry, but on the relative orientation between the edges and the loading direction. This distinction is crucial when discussing rotational symmetry.

If both the microstructure and the loading direction were rotated together, for example, by 90°, the underlying physical problem would remain unchanged. In such a case, a rotationally equivariant model would be expected to reproduce the same predictions in the rotated coordinate system, since the geometry–load configuration is identical in relative terms. However, in the present work, the loading direction is fixed, and only the microstructure orientation varies. Rotating the RE while keeping the load direction constant changes the relative orientation between each edge and the applied stress field, thereby producing a different physical response. For the case of a 90° rotated RE, our model would yield the same result as the unrotated RE subjected to a 90° rotated tensile loading.

The DDM is therefore not rotationally equivariant by design. The loading direction breaks rotational symmetry, meaning that the model cannot be invariant to rotations. Instead, the orientation of each edge relative to the load is explicitly encoded through the edge features. This design enables the GNN to distinguish between edges that are parallel, perpendicular, or oblique to the applied load, while still maintaining invariance to node permutations and translations.

3.4. Hyperparameter selection

For any deep learning architecture, hyperparameters have a considerable influence on model performance. For GNNs, these include the number of MPLs, the number of edge embeddings, and optimization-related choices such as learning rate, dropout, and batch size. Performing a careful hyperparameter selection is therefore crucial. In the present work, the DDM has been tested under a wide range of hyperparameter combinations, and the final choice reflects the best trade-off observed between accuracy, robustness, and computational cost. For a more thorough description of the hyperparameter optimization, we refer to Appendix.

3.5. Performance

In this section we briefly evaluate how well the DDM generalizes across VFs, in order to show a limitation that motivates the introduction of the HM. A more detailed evaluation is presented in Section 5.

Fig. 3 overlays the target and predicted PWR distributions for the VF20 test set, as well as for the test sets of VF10 and VF30. The counts are shown on a logarithmic axis to reveal the heavy-tailed nature of the data. It can be seen that while the target distributions exhibit VF-dependent scale shifts, the distributions predicted by the DDM cluster around a similar magnitude across VFs. This behavior is expected given

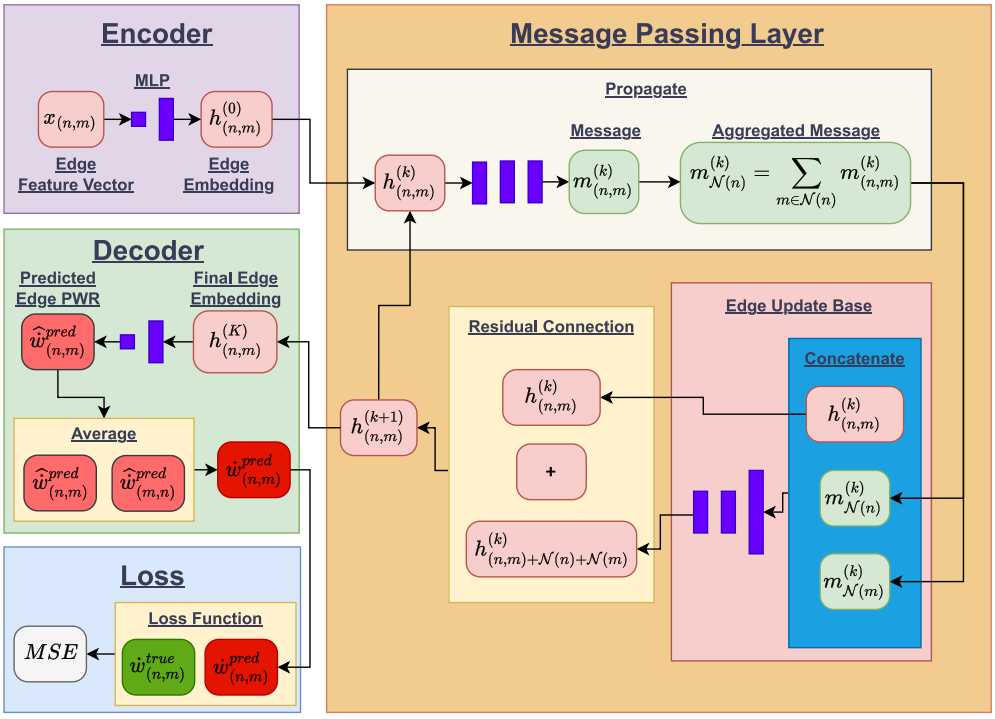

Fig. 2. Architecture of the data-driven model (DDM).

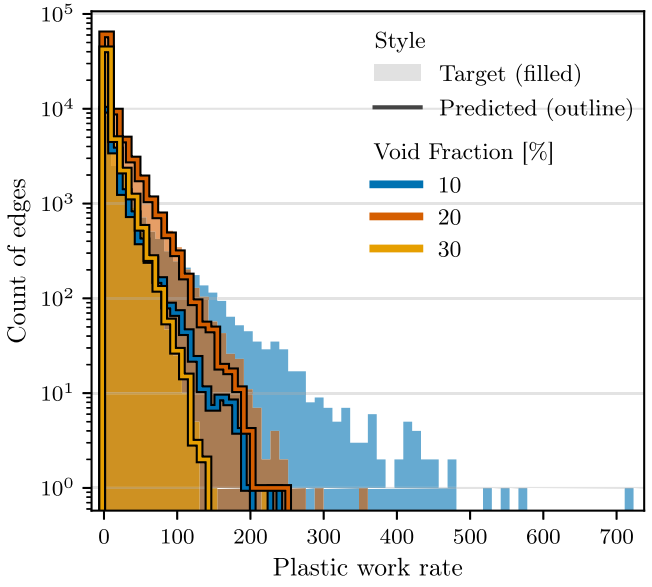

Fig. 3. Target and predicted PWR distributions. Predictions are based on the DDM trained on a void fraction of 20%.

the current inputs. The DDM only "sees" the local edge length and orientation relative to the load. These features encode anisotropy but do not tell the model how the VF rescales dissipation across an RE. Under a change in VF, the model has no mechanism to rescale edge magnitudes coherently, making the DDM largely interpolate/extrapolate around the VF20 statistics instead of adjusting to the correct scale. This means that while the DDM may be able to predict which edges are active or not, it fails to generalize the magnitude of the PWR values to other VFs. This motivates the adoption of a physics-aware prior as described in the next section.

4. Hybrid model

The HM couples the physics-based, limit-load analysis model (LLAM) proposed by Hund et al. (2025) with an edge-regression GNN to predict the total rate of internal work. We first present the key components of the LLAM – referring the reader to the original article for a detailed description – and then explain how it is used to construct the HM.

4.1. Microstructure-based limit analysis

The starting point of the LLAM proposed by Hund et al. (2025) is to partition the RE domain into rigid triangular blocks based on the Delaunay graph of the void centroids. The block velocities are taken as the primary kinematic variables, and it is assumed that the blocks may slide relative to one other along their common edges, but separation or interpenetration are not permitted due to plastic incompressibility. For each edge (n, m), this requirement is formulated as

$$\Delta v^N_{(n,m)} = 0$$

where $\Delta v^N_{(n,m)}$ is the normal velocity jump across the edge.

The edge-wise PWR is computed as

$$\dot{w}_{(n,m)} = \frac{\sigma_y}{\sqrt{3}}\, l^*_{(n,m)}\, |\Delta v^\tau_{(n,m)}|, \qquad l^*_{(n,m)} = l_{(n,m)} - 2R.$$

where $\Delta v^\tau_{(n,m)}$ is the tangential velocity jump, $l_{(n,m)}$ is the void centroid to centroid spacing, and $l^*_{(n,m)}$ is the effective shearing length. By expressing the tangential velocity jump in terms of two nonnegative discontinuity parameters:

$$\Delta v^\tau_{(n,m)} = p^+_{(n,m)} - p^-_{(n,m)}, \qquad p^+_{(n,m)}, p^-_{(n,m)} \geq 0,$$

the edge-wise PWR can be equivalently written as

$$\dot{w}_{(n,m)} = \frac{\sigma_y}{\sqrt{3}}\, l^*_{(n,m)}\, (p^+_{(n,m)} + p^-_{(n,m)}) \tag{8}$$

The total rate of internal work is obtained by summing over all the edges:

$$\dot{W}_{\text{tot}} = \sum_{(n,m)\in\mathcal{E}} \dot{w}_{(n,m)} = \frac{\sigma_y}{\sqrt{3}}\,\mathbf{g}^T\mathbf{p},$$

where the arrays $\mathbf{g}$ and $\mathbf{p}$ contain the values of the effective shearing length and discontinuity parameters, respectively, for all the edges.

Collecting the ingredients above, the following linear programming problem for the block velocities $\{\mathbf{v}_i\}$ and discontinuity parameters $\{p^+_{(n,m)}, p^-_{(n,m)}\}$ is obtained:

$$\begin{aligned} \min \quad & \dot{W}_{\text{tot}} = \frac{\sigma_y}{\sqrt{3}}\,\mathbf{g}^T\mathbf{p} \\ \text{s.t.} \quad & \begin{pmatrix} \boldsymbol{\tau}^T_{(n,m)} \\ \boldsymbol{n}^T_{(n,m)} \end{pmatrix} \cdot (\mathbf{v}_a - \mathbf{v}_b) - \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot (p^+_{(n,m)} - p^-_{(n,m)}) = \hat{\mathbf{b}}_{(n,m)}, \quad \forall\,(n,m)\in\mathcal{E} \\ & p^+_{(n,m)} \geq 0, \quad p^-_{(n,m)} \geq 0, \quad \forall\,(n,m)\in\mathcal{E}. \end{aligned} \tag{9}$$

In this statement, $\boldsymbol{\tau}_{(n,m)}$ and $\boldsymbol{n}_{(n,m)}$ are unit vectors tangential and normal to the edge (n,m), and $\mathbf{v}_a$ and $\mathbf{v}_b$ are the velocities of the blocks adjacent to the same edge. The vector $\hat{\mathbf{b}}_{(n,m)}$ vanishes for interior edges, while it equals the offset determined by the prescribed PBCs for edges on the boundary of the RE, as described in Hund et al. (2025). Once the linear programming problem is solved, the PWR can be found for each edge via Eq. (8).

4.2. Model architecture

The architecture of the HM is similar to that of the DDM shown in Fig. 2, but with two important modifications. First, the predicted PWR from the LLAM is used as an edge-wise mechanistic prior supplied to the GNN. This information is introduced as an additional edge attribute, such that for each edge three attributes are considered: the PWR estimate from Eq. (8), $\dot{w}^{LLAM}$, the edge orientation, θ, and the edge length, l. Furthermore, instead of directly predicting the edge-wise PWR as in the DDM, the HM predicts the residual with respect to the LLAM prior. Let f_0 denote the GNN. The network outputs become

$$\Delta\dot{w}_{(n,m)} = f_0\left(\theta, l, \dot{w}^{LLAM}\right) \tag{10}$$

and the edge-level prediction is

$$\dot{w}^{pred}_{(n,m)} = \dot{w}^{LLAM}_{(n,m)} + \Delta\dot{w}_{(n,m)} \tag{11}$$

so that the GNN is conditioned by and corrects the LLAM solution rather than learning the full mapping from scratch. This conditioned residual-learning design leverages the broad generalizability of the LLAM while reserving data capacity for localized corrections that capture effects beyond the restricted LLAM kinematics.

The model is trained by minimizing the empirical squared 2-Wasserstein loss, which measures the discrepancy between the predicted and target edge-wise PWR distributions. Let us define the arrays $\mathbf{s} = \{\dot{w}^{\text{pred}}_{(n,m)}\}_{(n,m)\in\mathcal{E}}$ and $\mathbf{t} = \{\dot{w}^{\text{true}}_{(n,m)}\}_{(n,m)\in\mathcal{E}}$, and denote by $\mathbf{s}^{\downarrow}$ and $\mathbf{t}^{\downarrow}$ their entries sorted in the same order (descending or ascending). The Wasserstein loss is defined as

$$\mathcal{L}_{W_2^2} = \frac{1}{|\mathcal{E}|}\sum_{i=1}^{|\mathcal{E}|}\left(s^{\downarrow}_i - t^{\downarrow}_i\right)^2 \tag{12}$$

This loss is permutation-invariant and aligns quantiles (largest with largest, etc.), thus matching distributions rather than enforcing spatially correct edge-wise predictions (Frogner et al., 2015).

5. Results

The proposed DDM and HM are implemented in PyTorch Geometric (Matthias Fey and contributors, 2024). In this section, we evaluate and benchmark their predictive performance against the two objectives defined in Section 2.1. Unless stated otherwise, performance is computed edge-wise on the predicted PWR, and metrics are averaged over test REs within each VF.

5.1. Shear band pattern

Fig. 4 illustrates the DDM predictions at VF20 for five REs of the test set: the two with the lowest loss, one with median loss, and the two with the highest loss. In the low-loss REs, the model recovers both the location and topology of the dominant shear bands, and the predicted PWR values closely follow the FE targets. Most edges with negligible PWR remain inactive in the prediction, and high-PWR edges are captured with only a slight underestimation on a few. In these cases, the loss mainly reflects small local discrepancies in magnitude rather than a misplacement of the bands themselves, indicating that the DDM can accurately reconstruct the plastic localization pattern.

In contrast, the high-loss REs reveal two characteristic situations in which incorrect predictions can arise. First (Fig. 4d), when the plastic work is strongly concentrated along a few edges, the DDM systematically underpredicts the peak PWR on these dominant bands while slightly overpredicting the many low-valued edges as seen in the distribution of targets and predictions. The overall shear band pattern is still recognizable, but the contrast between active and inactive edges is compressed, leading to a large edge-wise MSE despite a visually reasonable pattern. Second (Fig. 4e), in REs where the microstructure admits two competing long shear band paths that intersect (e.g. two potential "arms" of an X-shaped pattern at roughly 45° and -45°), the model occasionally selects the wrong branch as dominant. In these ambiguous cases, the predicted main band appears rotated by about 90° relative to the FE solution, so the error is driven more by a topological misassignment of the active path than by a uniform scaling mismatch.

Fig. 5 shows how well the DDM predicts the shear band pattern for selected REs with void fractions different from the one used during training. It can be seen that, while only trained on VF20, the DDM can accurately generalize between VF15 to VF30 by predicting the dominant bands in the RE. By contrast, at VF10, the predicted bands are either misplaced or diffuse. Therefore, the DDM is able to identify the dominant shear bands for REs with similar or higher void fractions, but it underestimates the PWR values for smaller void fractions. A possible reason is that, in the latter case, much higher PWR values arise, as can be seen from the distributions in Fig. 3. These high PWR values, associated with longer edges, are not part of the VF20 training data, which can explain the inability of the DDM to extrapolate to low void fractions.

Fig. 5 illustrates also the performance of the HM. The HM, trained using the Wasserstein loss, does not target spatial correspondence and therefore, as expected, does not predict the correct shear band pattern.

A more quantitative overview of the ability of the surrogate models to predict the shear band pattern is provided in Fig. 6(a), which compares the MSE-based loss across VFs for the DDM, the HM, and the LLAM, considering the entire test set. Out of the three models, the DDM is the best at predicting the spatial pattern, as its errors are lower than both the HM and LLAM. The LLAM exhibits poor fidelity, and the HM, which is conditioned on the LLAM output, inherits this limitation. However, the HM still achieves lower spatial error than the LLAM, indicating that the HM has learned to partially correct the LLAM systematic overestimation.

Fig. 6(b) shows the Wasserstein loss for comparability. Unlike the MSE-based loss, which reflects agreement in terms of spatial pattern, the Wasserstein loss compares only the distributions of edge-wise PWR within each RE. As expected, the HM, trained directly using this loss and aided by the LLAM prior, achieves the lowest loss values across void fractions, indicating good matching of the overall PWR distribution, even when the shear band pattern does not coincide with the one predicted by the FE simulation. The DDM shows very high loss values at VF10, consistent with its scale mismatch for low-VF tails, and lower values near VF20–VF30, where its target values are closer to the training distribution. The LLAM exhibits high loss values at all void

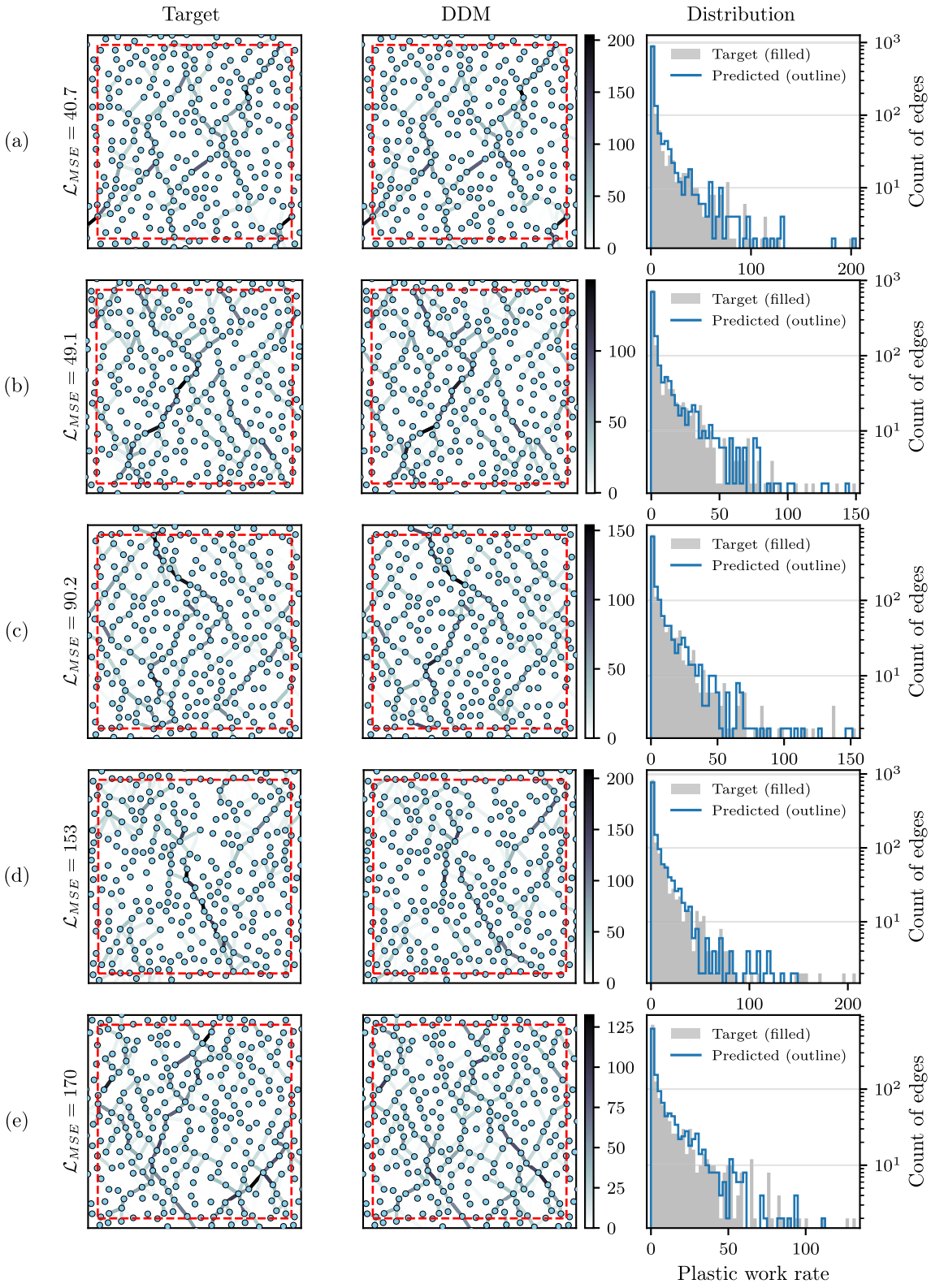

Fig. 4. Shear band pattern predicted by the DDM at VF 20%. For five REs – selected to have the lowest, second-lowest, near-median, second-highest, and highest loss in the test set – the target (left) and prediction (middle) are shown, with edge color indicating the PWR value. The right column shows the corresponding PWR distributions. The loading direction is horizontal.

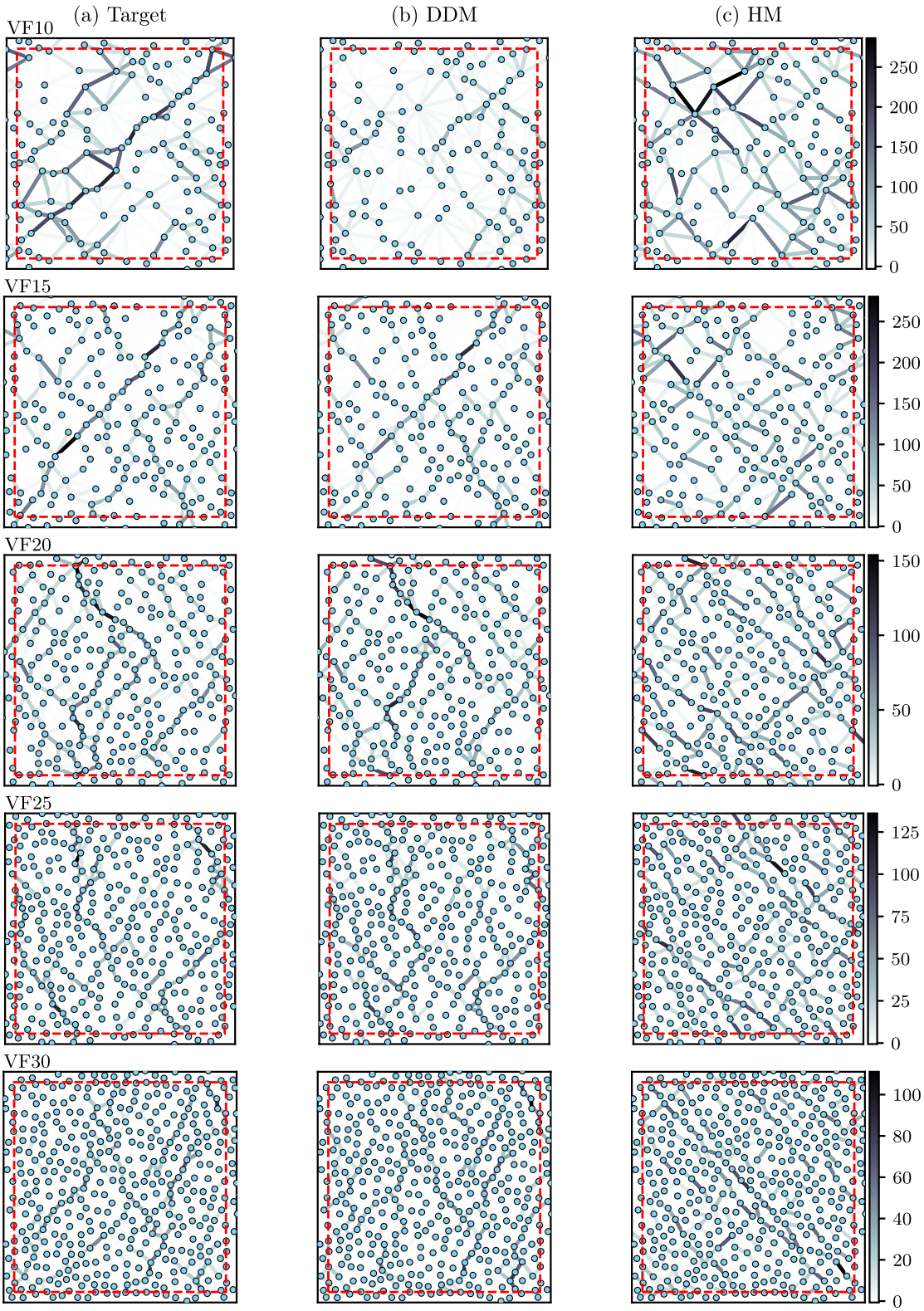

Fig. 5. Impact of void fraction on the shear band pattern prediction quality. For each test set (VF10, VF15, VF20, VF25, VF30), one RE with median loss is shown. The left column displays the target, the middle column the DDM prediction, and the right column the HM prediction. Edge color indicates the PWR value. The loading direction is horizontal.

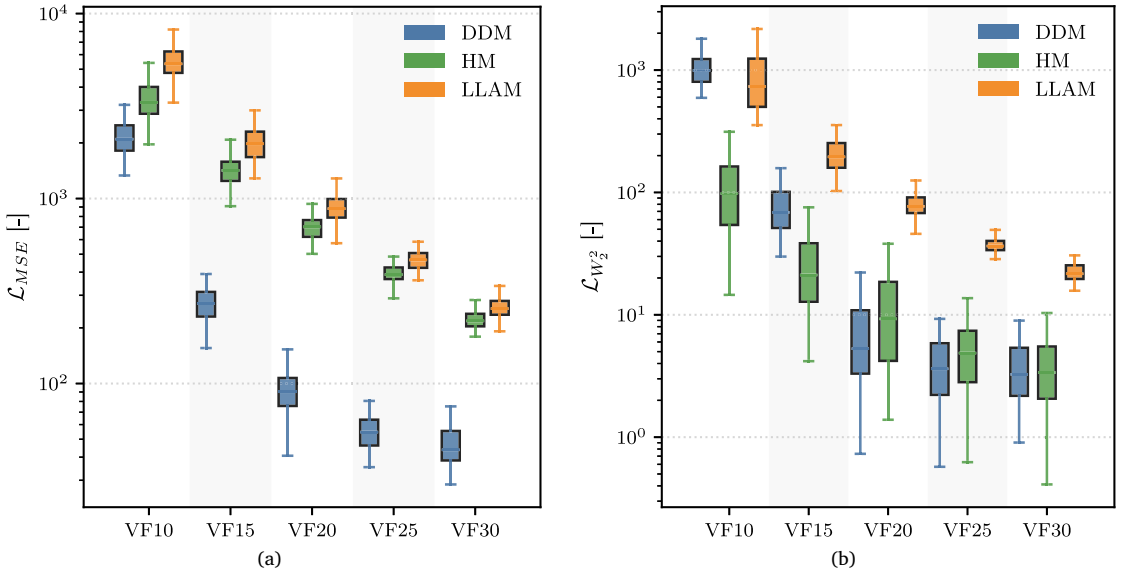

Fig. 6. Prediction error on the test set for the DDM, HM, and LLAM across void fractions. (a) Error measured using the MSE-based loss. (b) Error measured using the Wasserstein loss. Boxes show the interquartile range (IQR), and whiskers denote 1.5×IQR. The DDM is trained using the MSE-based loss, whereas the HM is trained using the Wasserstein loss.

fractions, which can be explained by the upper-bound character of its predictions.

5.2. Total rate of internal work

As discussed in relation to Fig. 3, the DDM is unable to adjust the PWR distribution when predicting at a void fraction lower than the one used for training. Particularly, at low VFs, the DDM largely preserves the VF20 amplitude scale instead of modifying the shape of the PWR distribution. In contrast, Fig. 7 shows that the HM does recalibrate its predicted PWR values across VFs, aligning the predicted distribution with the VF-dependent target scale. Therefore, the HM is expected to deliver more accurate predictions for quantities that are directly linked to the PWR distribution, such as the total rate of internal work. As already mentioned, this quantity is directly related to the macroscopic load-carrying capacity of the solid: the collapse load can be obtained by equating the total rate of internal work to the power of the external loads (Christiansen, 1996).

In this respect, Fig. 8 shows the relative error on the predicted total rate of internal work for the DDM, HM, and LLAM, across all VFs. Consistently with its upper-bound basis, the LLAM systematically overpredicts the total rate of internal work, and the overprediction increases almost linearly with the VF (Hund et al., 2025). The DDM performs best at the training void fraction (VF20), where the relative error is minimal. As the void fraction moves away from VF20, the model develops a systematic signed bias, consistently under- or over-predicting depending on the direction of the shift. It underestimates the total rate of internal work for REs with lower VF (heavier-tailed/higher-peak PWR) and overestimates it for REs with higher VF (compressed PWR range). By contrast, the HM yields uniformly low relative errors across all VFs, indicating that the HM retains the physics guidance of the LLAM while correcting its positive bias and its VF-dependent drift even when only trained on a single VF. This shows that incorporating the mechanistic prior into the HM architecture, together with the use of the Wasserstein loss during training, constitutes an effective approach for achieving accurate and robust predictions of the total rate of internal work.

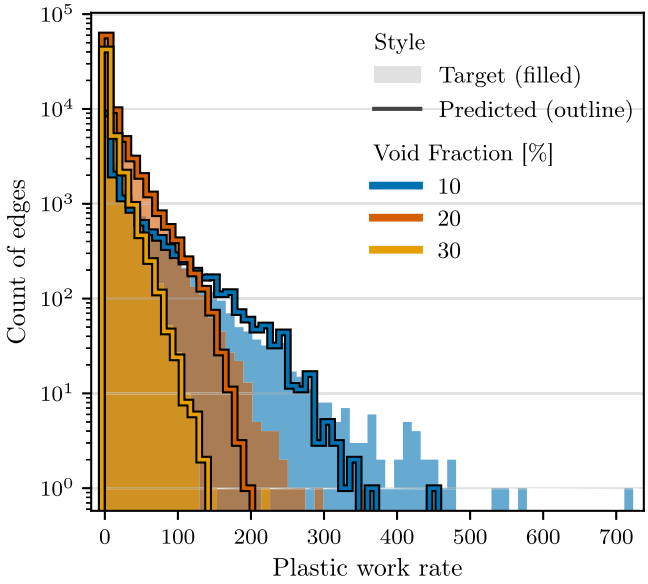

Fig. 7. Target and predicted PWR distributions. Predictions are based on the HM trained on a void fraction of 20%.

5.3. Data efficiency

This section investigates how predictive performance scales with training set size by training both surrogate models on progressively larger numbers of REs. For a given training set size, each model is trained using identical optimization settings, and the attained validation loss is recorded. A decreasing validation loss with increasing training data indicates that the model can still benefit from additional REs. Conversely, if the validation loss remains approximately constant, it can be concluded that the model has reached its optimal performance with fewer REs.

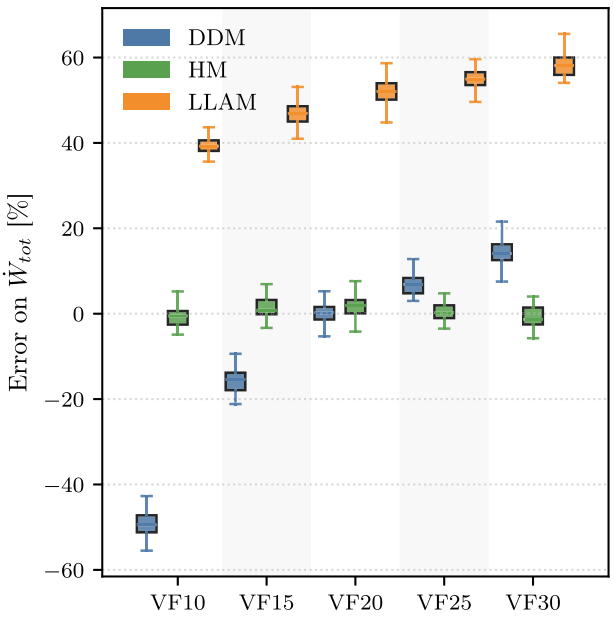

Fig. 8. Relative error in the total rate of internal work predicted on the test set by the DDM, HM, and LLAM across void fractions. Boxes show the interquartile range (IQR), and whiskers denote 1.5×IQR.

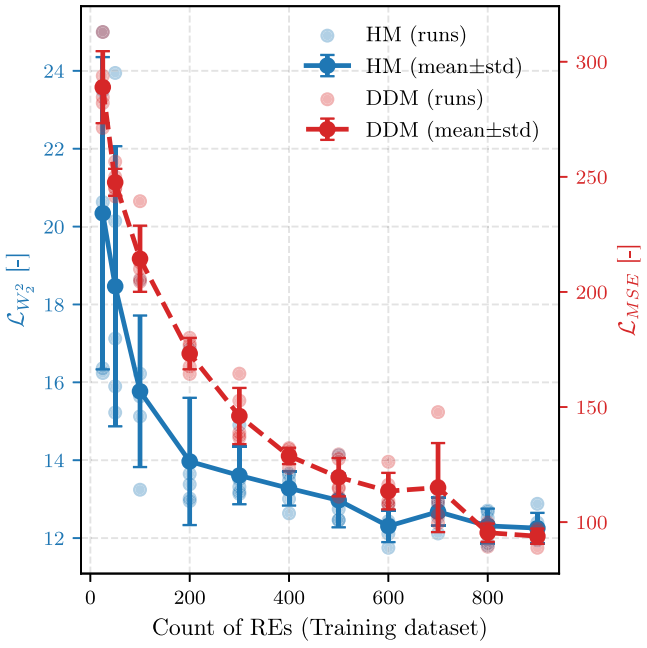

Fig. 9. Learning curves for HM and DDM. The horizontal axis reports the number of REs used for training, whereas the vertical axis reports the validation loss.

The learning curves in Fig. 9 shows a clear difference in data scaling between the two models. The HM stagnates around 600 REs, indicating data saturation under the current setup. In contrast, the DDM continues to improve up to our maximum of 900 REs, suggesting it remains data-limited. This pattern points to the HM having a higher data efficiency. The physics built into the HM supplies a strong bias, which reduces the amount of data needed to reach its performance ceiling. By contrast, the DDM must learn from data alone, requiring more REs to capture the underlying physical relationships. The benefits of the HM data efficiency are amplified by the underlying simulation costs, as the LLAM is more than 600 times quicker at solving one RE than the corresponding FE model (Hund et al., 2025).

Table 1
Correlation between the MSE and the Top-5 edge share within a RE, expressed in terms of the Spearman rank coefficient.

	VF10	VF15	VF20	VF25	VF30
ρ(MSE, S_5)	+0.775***	+0.463**	+0.307**	+0.020	+0.465***

Significance: * $q < 0.05$, ** $q < 0.01$, *** $q < 0.001$.

It is worth emphasizing that, beyond data scaling, the HM offers operational efficiency through cross-VF generalization. As shown previously, the HM maintains high performance on VFs not seen during training. The DDM, while capable of capturing the shear band pattern at VF20 and, to a large extent, at VF15, VF25, and VF30, still requires additional data generation and retraining to reliably predict the total rate of internal work at different VFs, making its usage across VFs substantially more costly.

5.4. Loss function of the data-driven model

Fig. 3 shows that high-PWR targets are underpredicted at VF10–VF20. To probe whether large errors arise when the target PWR is concentrated on a few edges, we quantify tail concentration for each RE by the Top–5 edges share

$$S_5 = \frac{\sum_{e\in\text{top-5}(\omega^{\text{true}})} \dot{\omega}_e^{\text{true}}}{\sum_{e\in\mathcal{E}} \dot{\omega}_e^{\text{true}}}, \tag{13}$$

i.e., the fraction of total PWR carried by the five edges with the largest PWR. Table 1 reports the correlation strength in terms of the Spearman rank coefficient ρ, which is robust to non-Gaussian tails and monotone rescalings. Effects are large at VF10, moderate at VF15 and VF30, smaller but significant at VF20, and non-significant at VF25. Overall, these results suggest that the model largest errors do occur when the PWR is highly concentrated on a few edges. In such cases, the shear bands connect voids into few dominant deformation paths across the RE. This localization causes most remaining edges to effectively be inactive, leading to a highly skewed PWR distribution.

The MSE-based loss penalizes large residuals quadratically, so an overestimate on a zero-valued edge or an underestimate on a rare high-valued edge leads to a large penalty. Because high-PWR edges are scarce, the model can reduce its loss by fitting the numerous low-PWR edges well while systematically underpredicting the rare, high-PWR edges. The outcome is a bias towards underestimation precisely on the critical edges where plastic deformation is concentrated, even when the model correctly identifies the main shear bands.

To reduce tail-driven errors, a solution could be to adopt a loss function less sensitive to extremes. For example, the Huber loss, which behaves like the MSE for small errors but transitions to mean absolute error for larger deviations. This hybrid nature can reduce the influence of rare, high-PWR edges, while allowing the model to learn from them, potentially resulting in more balanced predictions. Introducing the Huber loss would, however, also require specifying a new hyperparameter for the threshold at which the loss switches from MSE to mean absolute error, which could be challenging to optimize.

5.5. Seed sensitivity and robustness

The learning curves in Fig. 9 are based on the DDM and HM hyperparameter configurations that minimize the mean prediction error across all VFs. For these chosen configurations, we evaluated the surrogate models trained with the largest training set, differing only by the random seed (initialization, shuffling, and dropout). Fig. 10 shows the resulting per-VF performance considering the different seeds. Across VFs, the HM exhibits only minor variations, whereas the DDM

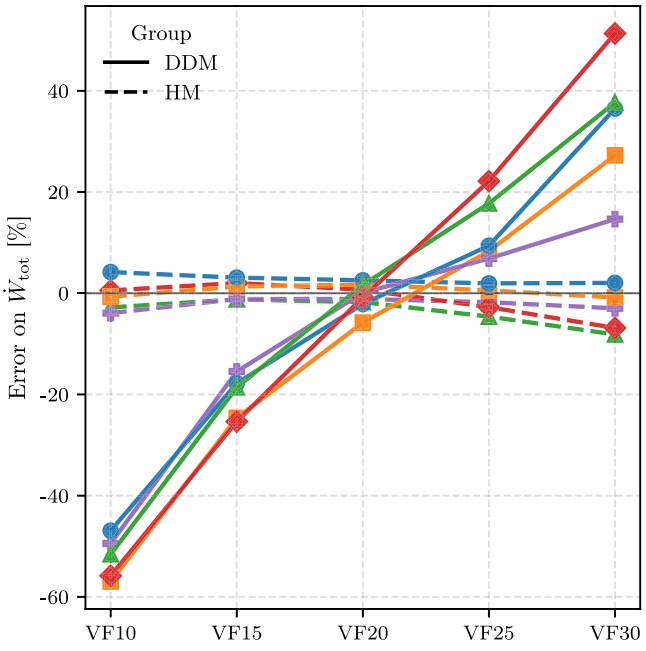

Fig. 10. Impact of the random seed on the HM and DDM prediction. Each curve corresponds to a different random seed.

shows seed-to-seed fluctuations, particularly away from the training VF. This suggests that the physics prior of the HM acts as a variance reducer, stabilizing performance.

6. Conclusions

We developed and evaluated two graph-based surrogate models for predicting strain localization pattern and macroscopic load carrying capacity in 2D elastic–perfectly plastic porous solids under uniaxial tension. Both models leverage a graph representation of the deformation, where nodes correspond to void centers, edges represent potential shear bands, and edge values encode the local PWR. The first model is purely data-driven and employs an edge-regression GNN to predict edge-wise PWR from local geometrical features. The second is a hybrid model that augments the edge-regression GNN with a mechanistic prior from limit load analysis and learns residual corrections using a permutation-invariant, distribution-matching loss.

Across extensive tests, the data-driven model accurately predicts the shear band pattern, but its accuracy deteriorates at void fractions lower than the one used for training. The hybrid model cannot reconstruct the shear band pattern, however, it delivers uniformly low errors in the total rate of internal work across all tested void fractions, effectively removing the upper-bound bias of the underlying limit-analysis prior. In addition to improved generalizability, the hybrid model requires fewer data for training, making it highly data-efficient.

These results demonstrate that coupling a GNN with a physics-based prior yields a fast and data-efficient surrogate model that preserves accuracy in macroscopic quantities while retaining meaningful spatial information. This provides a practical approach for predicting the collapse of porous solids while accounting for the exact locations of a large number of voids. As such, the present work represents a step towards developing computational models that are compatible with the large datasets generated by advanced experimental techniques, which are currently not fully exploited. Furthermore, we foresee potential applications in mesh instantiation: our method can provide insight into where shear bands are likely to form, which could be used to guide adaptive mesh refinement in high-fidelity finite element simulations, thereby improving computational efficiency.

CRediT authorship contribution statement

Rasmus B. Jakobsen: Writing – original draft, Software, Investigation, Formal analysis, Conceptualization. **Tobias S. Kristensen:** Writing – original draft, Software, Investigation, Formal analysis, Conceptualization. **Joep Storm:** Writing – review & editing, Supervision, Formal analysis, Conceptualization. **Iuri B.C.M. Rocha:** Writing – review & editing, Supervision, Formal analysis, Conceptualization. **Tito Andriollo:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Aarhus University Research Foundation, Denmark through grant no. AUFF-E-2023-9-44. The authors acknowledge the Danish e-Infrastructure Consortium (DeiC) for awarding access to the LUMI supercomputer through allocation no. DeiC-AU-S5-202400079 (LUMI project 465001634), as well as the AU GPU Cluster (Aarhus University ECE GPU Cluster, 2025), DelftBlue (Delft High Performance Computing Centre (DHPC), 2024) and AU PRIME (Aarhus University Prime, 2025) for computational resources. I. Rocha and J. Storm gratefully acknowledge the support from the TU Delft AI Labs programme.

Jonas Hund is gratefully acknowledged for providing the LLAM code and test data. R.B. Jakobsen and T.S. Kristensen thank Rasmus Lundgaard Madsen for insightful discussions and acknowledge the financial support of Civilingeniør Bernhard Eisenreich Sandersen og hustru Ruth Sandersens Fond through grant no. 24100059, Rudolph Als Fondet through grant no. 24110148 and 25030024, William Demant Fonden through grant no. 24-4627 and 24-4624, and FAMILIEN HEDE NIELSENS FOND through grant no. 2024-1550 and 2024-1533, which enabled their two-month research stay at TU Delft.

Appendix. Hyperparameter selection

This section presents the primary hyperparameters of the two surrogate models, discussing what their influence on the model is and how their values were either optimized or estimated. The hyperparameters are categorized into two groups: (1) those estimated based on prior experience and small-scale grid or random searches (see Appendix A.1), and (2) those selected through a more systematic grid search (see Appendix A.2).

A.1. Hyperparameters from experience

Residual Connections: Skip connections, specifically residuals added to the edge during their updates, were employed to prevent over-smoothing in deeper architectures by keeping information from the original embeddings in the update embedding as in Eq. (4). Experiments were conducted with and without residuals to evaluate their impact on information retention, and models that had residual connections performed better than those without. Experiments were also conducted by adding the initial embeddings $\mathbf{h}_{(n,m)}^{(0)}$ to the final embeddings $\mathbf{h}_{(n,m)}^{(K)}$ just before decoding. However, this did not seem to improve the model and was not implemented.

Optimizer: The learning rate controls the step size of the optimization algorithm. A too-small value can result in slow convergence, while a too-large value may cause instability. The AdamW optimizer

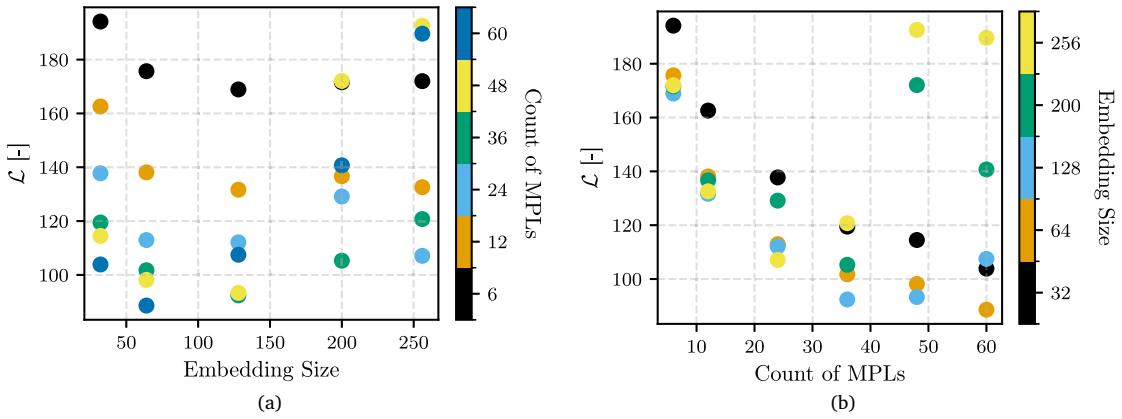

Fig. 11. Mean MSE vs Complexity of the DDM.

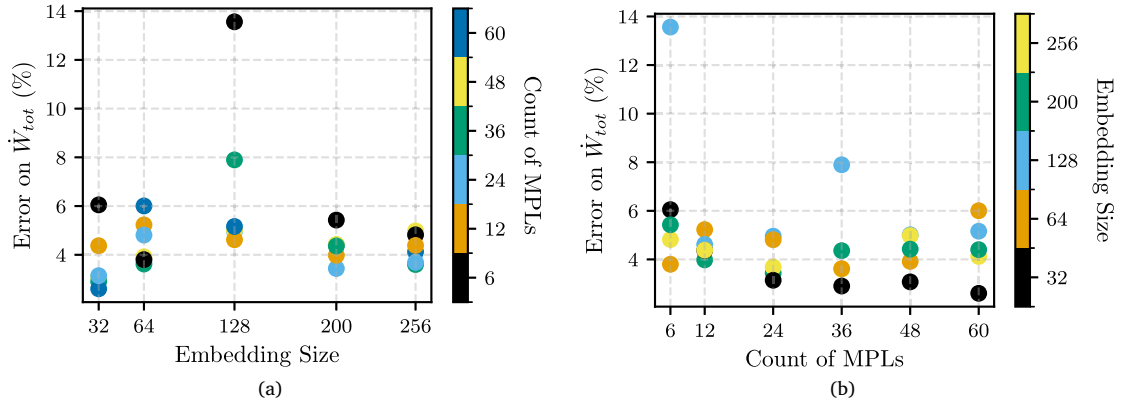

Fig. 12. Mean absolute relative error on the total rate of internal work vs Complexity of the HM.

was used, and learning rates between 0.01 and 0.0001 were evaluated. A learning rate of 0.001 gave the best results. A weight decay (L2-regularization) of 0.0001 was chosen and not investigated further.

Batch Size and Early Stopping: Training was carried out using mini-batches with a size of 30 REs per batch for quicker convergence. Early stopping was implemented based on validation loss to avoid overfitting. The threshold for early stopping was 25 epochs without improvement of the validation loss.

Random Seed: All experiments were carried out with fixed random seeds for reproducibility, where seed 20 was chosen.

Normalization: Neither the input data nor the target values were normalized, as models trained on the raw, unnormalized data gave the best results.

Activation Function: The activation function used in the MLPs is ReLU.

A.2. Hyperparameters for grid search

Number of Message-Passing Layers: The number of MPLs controls how much information is passed around the network. Increasing the number of MPLs will allow the graph to have better global awareness, but can also lead to over-smoothing and unwanted computational costs if set too high.

Embedding Size: Embedding size of the feature vectors in the MLPs plays an important role in the performance of the models. Larger embedding sizes allow the model to learn more complex patterns. However, the risk that the model overfits the training data by being too complex also increases, as well as the computational cost for training the model.

Dropout: Dropout is used to prevent overfitting by randomly deactivating a fraction of the neurons in each layer of the MLPs during training. This is done to ensure that the model does not rely too heavily on the weights of one or a few neurons.

A.3. Grid search

A systematic grid search was conducted to investigate the effect of the hyperparameters discussed in Appendix A.2. The ranges for each hyperparameter were determined on the basis of prior experience with the models. Each model configuration is trained 5 times using different seeds. The hyperparameters and the ranges of each are shown below:

- **Number of MPLs:** $\{6, 12, 24, 36, 48, 60\}$
- **Embedding size:** $\{32, 64, 128, 200, 256\}$
- **Dropout:** $\{0.5\}$

Performance for the DDM in the grid search was measured against the MSE on the validation set for VF20. Both the training set and the validation set were kept constant for all models by giving them the same seed for randomness. The outcome is shown in Fig. 11. For the HM, all model configurations were evaluated on all the VFs instead, and the performance was measured in terms of the relative error on the total rate of internal work. The outcome is shown in Fig. 12. Eventually, the configuration with 60 MPLs and 6 embeddings was selected and used to generate the results presented in this paper.

Data availability

The Python implementation of the two surrogate models is available at https://github.com/tobi2889/gnn-plastic-strain-localization. The full graph dataset will be made available on request.

References

Aarhus University ECE GPU Cluster, 2025. The ECE gpu cluster. High-performance computing facility at Aarhus University, Department of Electrical and Computer Engineering. http://gpucluster.st.lab.au.dk/.

Aarhus University Prime, 2025. PRIME laboratory documentation. LabBook page for PRIME at Aarhus University. https://labbook.au.dk/pages/viewpage.action?spaceKey=COM&title=1.+Introduction.

Abueidda, Diab W., Almasri, Mohammad, Ammourah, Rami, Ravaioli, Umberto, Jasiuk, Iwona M., Sobh, Nahil A., 2019. Prediction and optimization of mechanical properties of composites using convolutional neural networks. Compos. Struct. (ISSN: 0263-8223) 227, 111264. http://dx.doi.org/10.1016/j.compstruct.2019.111264, URL https://www.sciencedirect.com/science/article/pii/S0263822319312383.

Abueidda, Diab W., Koric, Seid, Sobh, Nahil A., Sehitoglu, Hüseyin, 2021. Deep learning for plasticity and thermo-viscoplasticity. Int. J. Plast. (ISSN: 0749-6419) 136, 102852. http://dx.doi.org/10.1016/j.ijplas.2020.102852.

Aldakheel, Fadi, Elsayed, Elsayed S., Zohdi, Tarek I., Wriggers, Peter, et al., 2023. Efficient multiscale modeling of heterogeneous materials using deep neural networks. Comput. Mech. (ISSN: 0178-7675) 72 (1), 155–171. http://dx.doi.org/10.1007/s00466-023-02324-9, URL https://link.springer.com/article/10.1007/s00466-023-02324-9.

Andriollo, Tito, Kouznetsova, Varvara, Alessandretti, Laura, 2022. Network approach for the analysis of the irreversible deformation of solids with soft heterogeneities. Phys. Rev. Mater. 6 (11), L110601. http://dx.doi.org/10.1103/PhysRevMaterials.6.L110601.

Bishop, Christopher M., Bishop, Hugh, 2024. Advances in Neural Information Processing Systems (NIPS). Springer International Publishing, London, UK, http://dx.doi.org/10.1007/978-3-031-45468-4, URL https://link.springer.com/book/10.1007/978-3-031-45468-4.eBook ISBN. (Accessed: 06 December 2024).

Buljac, Ante, Helfen, Lukas, Hild, François, Morgeneyer, Thilo F., 2018. Effect of void arrangement on ductile damage mechanisms in nodular graphite cast iron: In situ 3D measurements. Eng. Fract. Mech. (ISSN: 0013-7944) 192, 242–261. http://dx.doi.org/10.1016/j.engfracmech.2018.01.008.

Cadet, Clément, Besson, Jacques, Flouriot, Sylvain, Forest, Samuel, Kerfriden, Pierre, Lacourt, Laurent, De Rancourt, Victor, 2022. Strain localization analysis in materials containing randomly distributed voids: Competition between extension and shear failure modes. J. Mech. Phys. Solids (ISSN: 00225096) 166, 104933. http://dx.doi.org/10.1016/j.jmps.2022.104933, URL https://linkinghub.elsevier.com/retrieve/pii/S0022509622001302.

Christiansen, Edmund, 1996. Limit analysis of collapse states. In: Handbook of Numerical Analysis. In: Finite Element Methods (Part 2), Numerical Methods for Solids (Part 2), vol. 4, Elsevier, pp. 193–312. http://dx.doi.org/10.1016/S1570-8659(96)80004-4.

Delft High Performance Computing Centre (DHPC), 2024. DelftBlue supercomputer (phase 2). https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2.

Frankel, Ari L., Safta, Cosmin, Alleman, Coleman, Jones, Reese E., 2022. Mesh-based graph convolutional neural networks for modeling materials with microstructure. J. Mach. Learn. Model. Comput. (ISSN: 2689-3967) 3 (1), 1–30.

Fritzen, Felix, Forest, Samuel, Böhlke, Thomas, Kondo, Djimedo, Kanit, Toufik, 2012. Computational homogenization of elasto-plastic porous metals. Int. J. Plast. (ISSN: 0749-6419) 29, 102–119. http://dx.doi.org/10.1016/j.ijplas.2011.08.005, URL https://www.sciencedirect.com/science/article/pii/S0749641911001458.

Frogner, Charlie, Zhang, Chiyuan, Mobahi, Hossein, Araya, Mauricio, Poggio, Tomaso A, 2015. Learning with a wasserstein loss. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (Eds.), In: Advances in Neural Information Processing Systems, vol. 28, Curran Associates, Inc., URL https://proceedings.neurips.cc/paper_files/paper/2015/file/a9eb812238f753132652ae09963a05e9-Paper.pdf.

Ghavamian, F., Simone, A., 2019. Accelerating multiscale finite element simulations of history-dependent materials using a recurrent neural network. Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 357, 112594. http://dx.doi.org/10.1016/j.cma.2019.112594.

Gilbert, Matthew, Smith, Colin C., Hawksbee, Samuel J., Tyas, Andrew, 2014. Use of Layout Optimization to Solve Large-Scale Limit Analysis and Design Problems. In: Spiliopoulos, Konstantinos, Weichert, Dieter (Eds.), Direct Methods for Limit States in Structures and Materials. Springer Netherlands, Dordrecht, pp. 157–180. http://dx.doi.org/10.1007/978-94-007-6827-7_8.

Gupta, Ashwini, Bhaduri, Anindya, Graham-Brady, Lori, 2023. Accelerated multiscale mechanics modeling in a deep learning framework. Mech. Mater. (ISSN: 0167-6636) 184, 104709. http://dx.doi.org/10.1016/j.mechmat.2023.104709, URL https://www.sciencedirect.com/science/article/pii/S0167663623001552.

Gurson, A.L., 1977. Continuum theory of ductile rupture by void nucleation and growth: Part 1 - Yield criteria and flow rules for porous ductile media. J Eng. Mater. Technol.-Trans. ASME (ISSN: 15288889) 99 (1), 2–15.

Hamilton, William L., 2020. Graph representation learning. Available online at https://www.cs.mcgill.ca/~wlh/grl_book/files/GRL_Book.pdf. (Accessed: 06 December 2024).

Hendriks, Fleur, Menkovski, Vlado, Doškář, Martin, Geers, Marc G.D., Rokoš, Ondřej, 2025. Similarity equivariant graph neural networks for homogenization of metamaterials. Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 439, 117867. http://dx.doi.org/10.1016/j.cma.2025.117867, URL https://www.sciencedirect.com/science/article/pii/S0045782525001392.

Hund, Jonas, Kouznetsova, Varvara, Andriollo, Tito, 2025. Efficient prediction of strength and strain localisation in porous solids via microstructure-based limit analysis. Mech. Mater. (ISSN: 0167-6636) 201, 105208. http://dx.doi.org/10.1016/j.mechmat.2024.105208, URL https://www.sciencedirect.com/science/article/pii/S0167663624003004.

Karapiperis, Konstantinos, Kochmann, Dennis M., 2023. Prediction and control of fracture paths in disordered architected materials using graph neural networks. Commun. Eng. (ISSN: 2731-3395) 2 (1), 32. http://dx.doi.org/10.1038/s44172-023-00085-0.

Krokos, Vasilis, Bordas, Stéphane P.A., Kerfriden, Pierre, 2024. A graph-based probabilistic geometric deep learning framework with online enforcement of physical constraints to predict the criticality of defects in porous materials. Int. J. Solids Struct. (ISSN: 0020-7683) 286–287, 112545. http://dx.doi.org/10.1016/j.ijsolstr.2023.112545, URL https://www.sciencedirect.com/science/article/pii/S0020768323004420.

Krokos, Vasilis, Bui Xuan, Viet, Bordas, Stéphane P.A., Young, Philippe, Kerfriden, Pierre, 2022. A Bayesian multiscale CNN framework to predict local stress fields in structures with microscale features. Comput. Mech. 69 (3), 733–766. http://dx.doi.org/10.1007/s00466-021-02112-3, URL https://link.springer.com/article/10.1007/s00466-021-02112-3.

Matouš, Karel, Geers, Marc G.D., Kouznetsova, Varvara G., Gillman, Andrew, 2017. A review of predictive nonlinear theories for multiscale modeling of heterogeneous materials. J. Comput. Phys. (ISSN: 10902716) 330, 192–220. http://dx.doi.org/10.1016/j.jcp.2016.10.070, Publisher: Elsevier Inc..

Matthias Fey and contributors, 2024. PyTorch Geometric 2.6.1 Documentation. PyG Team, Available at: https://www.pyg.org.

Maurizi, Marco, Gao, Chao, Berto, Filippo, 2022. Predicting stress, strain and deformation fields in materials and structures with graph neural networks. Sci. Rep. (ISSN: 2045-2322) 12 (1), 21834. http://dx.doi.org/10.1038/s41598-022-26424-3.

Pijnenburg, K.G.W., Van Der Giessen, E., 2003. A novel approach to the analysis of distributed shear banding in polymer blends. Internat. J. Numer. Methods Engrg. (ISSN: 0029-5981) 58 (5), 703–721. http://dx.doi.org/10.1002/nme.792, ISSN: 1097-0207. URL https://onlinelibrary.wiley.com/doi/10.1002/nme.792.

Sloan, S.W., Kleeman, P.W., 1995. Upper bound limit analysis using discontinuous velocity fields. Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 127 (1), 293–314. http://dx.doi.org/10.1016/0045-7825(95)00868-1.

Smith, Colin, Gilbert, Matthew, 2007. Application of discontinuity layout optimization to plane plasticity problems. Proc. R. Soc. A: Math. Phys. Eng. Sci. 463 (2086), 2461–2484. http://dx.doi.org/10.1098/rspa.2006.1788.

Smith, C.C., Gilbert, M., He, L., González-Castejón, J., Ouakka, S., 2019. Recent advances in the application of discontinuity layout optimization to geotechnical analysis and design problems. In: Proceedings of the XVII European Conference on Soil Mechanics and Geotechnical Engineering. pp. 2556–2563. http://dx.doi.org/10.32075/17ECSMGE-2019-0480.

Storm, J., Rocha, I.B.C.M., van der Meer, F.P., 2024. A microstructure-based graph neural network for accelerating multiscale simulations. Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 427, 117001. http://dx.doi.org/10.1016/j.cma.2024.117001, URL https://www.sciencedirect.com/science/article/pii/S0045782524002573.

Suquet, Pierre M., 1987. Elements of homogenization for inelastic solid mechanics. In: Sanchez-Palencia, Enrique, Zaoui, André (Eds.), Homogenization Techniques for Composite Media. Springer-Verlag, pp. 193–278.

Systèmes, Dassault, 2021. Abaqus Analysis User's Guide (2021)..

Systèmes, Dassault, 2022. Abaqus 2022. Version 2022, SIMULIA, Providence, RI, USA. URL https://www.3ds.com/products-services/simulia/products/abaqus/.

Vlassis, Nikolaos N., Sun, WaiChing, 2023. Geometric learning for computational mechanics part II: Graph embedding for interpretable multiscale plasticity. Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 404, 115768. http://dx.doi.org/10.1016/j.cma.2022.115768, URL https://www.sciencedirect.com/science/article/pii/S0045782522007241.

Wang, J.J., et al., 2022. A deep learning framework for constitutive modeling based on temporal convolutional network. Int. J. Fatigue (ISSN: 0142-1123) 155, 106651. http://dx.doi.org/10.1016/j.ijfatigue.2021.106651.

Withers, Philip J., Bouman, Charles, Carmignato, Simone, Cnudde, Veerle, Grimaldi, David, Hagen, Charlotte K., Maire, Eric, Manley, Marena, Du Plessis, Anton, Stock, Stuart R., 2021. X-ray computed tomography. Nat. Rev. Methods Prim. (ISSN: 2662-8449) 1 (1), 18. http://dx.doi.org/10.1038/s43586-021-00015-4.

Wu, Hao, Fan, Guohua, 2020. An overview of tailoring strain delocalization for strength-ductility synergy. Prog. Mater. Sci. (ISSN: 00796425) 113, 100675. http://dx.doi.org/10.1016/j.pmatsci.2020.100675, URL https://linkinghub.elsevier.com/retrieve/pii/S0079642520300396.

Wu, Ling, Nguyen, Van Dung, Kilingar, Nanda, Noels, Ludovic, 2020. A recurrent neural network-accelerated multi-scale model for elasto-plastic heterogeneous materials subjected to random cyclic and non-proportional loading paths. Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 369, 113234. http://dx.doi.org/10.1016/j.cma.2020.113234.

Xu, C.L., Andriollo, T., Zhang, Y.B., Hernando, J.C., Hattel, J., Tiedje, N., 2020. Micromechanical impact of solidification regions in ductile iron revealed via a 3D strain partitioning analysis method. Scr. Mater. (ISSN: 1359-6462) 178, 463–467. http://dx.doi.org/10.1016/j.scriptamat.2019.12.018.

Yvonnet, J., Auffray, N., Monchiet, V., 2020. Computational second-order homogenization of materials with effective anisotropic strain-gradient behavior. Int. J. Solids Struct. (ISSN: 00207683) 191–192, 434–448. http://dx.doi.org/10.1016/j.ijsolstr.2020.01.006, URL https://linkinghub.elsevier.com/retrieve/pii/S0020768320300123.

Zhao, Yingxue, Li, Haoran, Zhou, Haosu, Attar, Hamid Reza, Pfaff, Tobias, Li, Nan, 2024. A review of graph neural network applications in mechanics-related domains. Artif. Intell. Rev. (ISSN: 0269-2821) 57 (11), 315. http://dx.doi.org/10.1007/s10462-024-10931-y, URL https://link.springer.com/article/10.1007/s10462-024-10931-y.

Zhu, Yuntian, Wu, Xiaolei, 2023. Heterostructured materials. Prog. Mater. Sci. (ISSN: 00796425) 131, 101019. http://dx.doi.org/10.1016/j.pmatsci.2022.101019, URL https://linkinghub.elsevier.com/retrieve/pii/S0079642522001001.