PAPER • OPEN ACCESS

Operator-aware physics-informed neural networks for forward and inverse problems in solid mechanics

MACHINE LEARNING
Science and Technology

PAPER

Operator-aware physics-informed neural networks for forward and inverse problems in solid mechanics

Amir Ghorbani Ghezeljehmeidan*, Willem Dirk van Driel and Justin Dauwels

Department of Microelectronics, TUDelft, Delft 2627, The Netherlands
* Author to whom any correspondence should be addressed.

E-mail: amir.ghorbani@tudelft.nl

Keywords: physics-informed neural networks, variational methods, eigenmodes, operator learning, inverse problems

OPEN ACCESS

RECEIVED
21 April 2026

REVISED
4 June 2026

ACCEPTED FOR PUBLICATION
22 June 2026

PUBLISHED
8 July 2026

Abstract

Physics-informed neural networks (PINNs) on complex domains are limited by input representations that encode geometry but do not reflect the physics of the governing PDE. We propose an operator-aware PINN for solid mechanics problems that embeds precomputed eigenmodes of the problem's own discrete operator as geometry and physics-aware features within a weak-form variational formulation. The displacement field is represented by a neural mapping enriched by operator-aligned features rather than being constrained to a finite element trial space. A hybrid chain-rule formulation propagates spatial derivatives through both coordinate and eigenmode branches, yielding strain fields consistent with operator structure while preserving neural expressivity. The approach maintains the stability of variational formulations while mitigating the spectral bias and geometry-encoding limitations of classical PINNs. For inverse elasticity, the same eigenmodes act as virtual test functions in a weak-form residual, mitigating the stiffness-collapse pathology. The framework provides an instance-based alternative to differentiable finite element method without labeled training data. It achieves competitive accuracy and connects spectral operator theory with physics-informed neural computation.

1. Introduction

Structures with complex internal geometry, such as ceramics, metal foams, and biological tissues, pose persistent computational challenges due to the interplay of geometric complexity, multi-physics coupling, and iterative design workflows [1–4]. The irregular void topology creates stress concentration patterns governed by long-range pore interactions [5–7], demanding fine meshes to resolve rapidly varying fields near geometric features [8, 9].

The finite element method (FEM) remains the standard tool, yet each new load case or material perturbation requires a fresh assembly-and-solve cycle, making parametric studies and optimization prohibitively expensive when many forward evaluations are needed [10, 11].

Physics-informed neural networks (PINNs) [12, 13] offer an alternative. The solution is parameterized by a neural network trained by minimizing a PDE residual (strong-form) or potential energy (weak form) [14]. The latter is known as the deep energy method (DEM). The resulting continuous and differentiable representation enables gradient-based inverse analysis and sensitivity computation via automatic differentiation. Although the mesh-free nature of PINNs makes them applicable to complex geometries [15], they tend to lose accuracy near geometric features [16]. A standard MLP operating on raw coordinates must learn both geometry and physics, leading to spectral bias [17–19] and poor accuracy where spectral bias makes the high-frequency response difficult to resolve and boundary conditions are hard to impose exactly [20, 21].

The eigenmodes of the elastic operator encode geometry, boundary conditions, and material response in a compact spectral basis. Supplying these modes and their gradients as input defines an operator-aligned feature space, allowing the network to learn corrections to a physically structured representation rather than reconstructing global interactions from coordinates alone.

1.1. Related work

Several strategies have emerged to address the limitations in PINNs. We organize the literature into four thematic groups, progressing from weak-form and FEM-integrated PINNs, input enrichment and geometric encoding, spectral and eigenfunction-based approaches, and operator learning.

(a) Weak-form and FEM-integrated PINNs. Variational and energy-based losses eliminate second-order derivatives and naturally incorporate Neumann conditions. The DEM [14, 22] and the deep Ritz method [23] pioneered this approach, with extensions to viscoelasticity [24], mixed formulations [25], elastoplasticity [26], and complementary energy [27]. In pure AD-based gradient evaluation, strain localization artifacts have been identified [28]. VPINNs [29] adopt a Petrov–Galerkin setting, FEINN [30] evaluates weak-form losses via FE quadrature with inverse extension. DFEM [31] embeds the weak form in a differentiable graph with nodal-displacement parameters. I-FENN [32] and NeuFENet [33] further integrate PINNs with FE machinery.

A common thread across these works is that the FEM mesh provides reliable quadrature and gradient machinery, while the neural network provides a flexible trial function space. Our framework inherits these weak-form advantages but enriches the network inputs with precomputed elastic eigenmodes, providing an operator-aligned representation that enables learning corrections to a physics-informed embedding rather than reconstructing global patterns from local shape functions.

(b) Input enrichment and geometric encoding. A parallel line of research provides PINNs with structured input features beyond raw coordinates to mitigate spectral bias and geometric encoding difficulties. Fourier feature mappings [19] and multi-scale features [18] (FF-PINN) combat spectral bias but encode no physical information. FBPINN [34] decompose the domain into overlapping subdomains. Δ-PINNs [35] replace Cartesian coordinates with Laplace–Beltrami (LB) eigenfunctions, dramatically improving precision in complex geometries. Finite-PINNs [36] concatenate LB eigenfunctions with coordinates for solid mechanics with chain-rule differentiation. GADEM [37] conditions on geometric latent vectors for cross-shape generalization.

Our approach differs from the input-enrichment and geometric-encoding methods discussed above [18, 19, 34, 37] in three respects, with the first two distinctions applying most directly to the Laplacian-eigenbasis methods [35, 36]. First, we replace the Laplacian eigenbasis, which encodes only domain geometry, with eigenmodes of the governing operator itself. For elasticity, this means the stiffness operator, so the resulting features are vector-valued and carry material and boundary information. This change is invisible in homogeneous isotropic problems, where LBE and elastic modes span overlapping information, but becomes essential in anisotropic, heterogeneous, or multi-physics settings, where Laplacian features are formally blind to the physics. Second, a hybrid chain-rule formulation, tailored to the vector-valued modal structure, ensures that strain fields remain consistent with the enriched displacement representation rather than ignoring the operator branch. Third, unlike coordinate-replacement or finite-element-based representations [30, 35, 36], the eigenmodes are used as input enrichment within the weak-form energy formulation, while the displacement field remains a free neural trial function rather than being restricted to a finite-element trial space or obtained by replacing the coordinates outright.

(c) Spectral and eigenfunction-based operator learning. In operator learning, Laplacian eigenfunctions serve as spectral basis. LE-NO [38] projects reaction-diffusion dynamics on Laplacian basis, NORM [39] encodes function-space mappings in Laplacian subspace, and neural spectral methods [40] learn spectral representations end-to-end. Spectral [41] and Laplace [42] neural operators extend these ideas to Chebyshev/Fourier bases and non-periodic signals.

These operator-learning methods are typically trained on parametric PDE families and often rely on substantial supervised or hybrid training data. Our approach is instance-based. A single network is trained for each PDE instance by energy minimization, with eigenmodes serving as input enrichment rather than a solution basis.

(d) Operator learning for solid mechanics. DeepONets [43, 44] and FNOs [45] with geometry extensions [46] and physics-informed variants [47, 48] achieve fast inference for parametric PDEs, with recent solid-mechanics applications [49–51] and PDE foundation models [17, 52].

Although these methods achieve remarkable inference speed, they require thousands of labeled FEM solutions for training at a cost that can exceed the benefit when only a small number of geometry variants are of interest. The operator-aware PINN (OA-PINN) targets a complementary regime providing

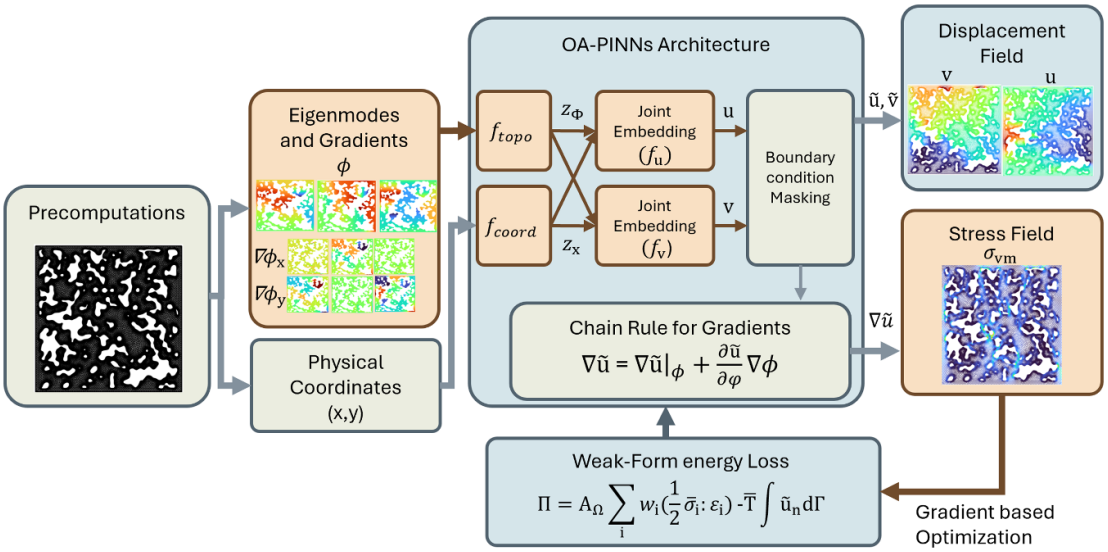

Figure 1. Operator-aware PINN framework. Elastic eigenmodes are computed once from the FE stiffness operator and supplied, together with spatial coordinates, to a dual-branch neural network trained by minimizing total potential energy.

instance-based solutions without labeled training data, at the cost of a one-time eigenmode computation per geometry. Table 1 summarizes how our approach relates to the most recent methods.

1.2. Contributions

This work proposes an operator-aware PINN that uses precomputed elastic eigenmodes as a physics-adapted feature basis. Our key insight is that elastic eigenmodes of the generalized eigenvalue problem encode critical information about domain geometry, boundary conditions, and material properties. Using these modes as input to the network alongside spatial coordinates, we provide the neural network with a compact, topology-aware representation that improves learning efficiency. The architecture is shown in figure 1.

Our main contributions are as follows.

1. **Operator-aware feature engineering.** We propose eigenmodes of the governing PDE operator as PINN input features, extending geometry-aware enrichment (Laplacian eigenfunctions, Fourier features) to physics-aware enrichment. The principle applies to any PDE with a computable self-adjoint operator (section 3).
2. **Hybrid architecture with gradient propagation.** OA-PINN processes eigenmode features and spatial coordinates through separate encoding branches (section 3.5). Crucially, we derive a chain rule that propagates AD gradients through both branches (section 3.6), unlike methods that treat enrichment features as frozen encodings.
3. **virtual field method (VFM)-based inverse material identification.** We show that the same eigenmode basis serves as virtual test functions for a weak-form equilibrium residual, enabling alternating optimization that avoids the stiffness-collapse pathology $\mathbf{C} \to 0$ inherent in joint energy minimization with learnable material parameters (section 5.6).

1.3. Organization

Section 2 reviews linear elasticity and PINN formulations. Section 3 presents the operator-aware framework, eigenmode computation, OA-PINN architecture, chain rule gradients, and weak-form training. Section 4 details the implementation, while section 5 provides numerical validation. Sections 6 and 7 discuss limitations and offer concluding remarks, respectively.

Table 1. Positioning of the proposed framework against the most closely related methods. 'Hybrid ∇' indicates AD gradient propagation through the enrichment features. 'Extends across PDEs' marks methods whose feature construction recipe applies uniformly to other self-adjoint operators. 'IMI' refers to inverse material identification. The symbols (✓), (×), and (–) denote supported, not supported, and not applicable, respectively.

Method	Operator eigenmodes	Energy loss	Hybrid ∇	Extends across PDEs	IMI
DEM [14]	×	✓	—	–	×
VPINNs [29]	×	✓[a]	—	–	×
Δ-PINNs [35]	✓[b]	×	×	×	×
FBPINNs [34]	×	×	—	–	×
Finite-PINN [36]	✓[c]	✓	✓[d]	×	×
FEINN [30]	×	✓	—	–	✓
GADEM [37]	×	✓	—	–	×
LE-NO [38]	✓[e]	×	—	×	×
This work	✓[f]	✓	✓	✓	✓

[a] Petrov–Galerkin, not energy minimisation.
[b] Scalar LBE eigenfunctions; replaces coordinates.
[c] Scalar LBE eigenfunctions concatenated with coordinates.
[d] Through scalar LBE branches.
[e] Operator-learning basis; data-driven, not physics-only.
[f] Vector-valued elastic operator eigenmodes concatenated with coordinates.

2. Background

2.1. Linear elasticity

Consider a 2D elastic body that occupies domain $\Omega \subset \mathbb{R}^2$ with boundary $\partial\Omega = \Gamma_D \cup \Gamma_N$. The displacement field $\mathbf{u} = (u, v)^\top$ satisfies equilibrium under small-strain linear elasticity.

Kinematics. The strain tensor is

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} \\ \varepsilon_{xy} & \varepsilon_{yy} \end{bmatrix}, \quad \varepsilon_{xx} = \frac{\partial u}{\partial x}, \quad \varepsilon_{yy} = \frac{\partial v}{\partial y}, \quad \varepsilon_{xy} = \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right). \tag{1}$$

Constitutive law. For isotropic material with Young's modulus E and Poisson's ratio ν, we have

$$\boldsymbol{\sigma} = \lambda \operatorname{tr}(\boldsymbol{\varepsilon})\,\mathbf{I} + 2\mu\boldsymbol{\varepsilon}, \quad \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \quad \mu = \frac{E}{2(1+\nu)}. \tag{2}$$

Strong form. Equilibrium without body forces:

$$\nabla \cdot \boldsymbol{\sigma} = \mathbf{0} \quad \text{in } \Omega, \quad \mathbf{u} = \mathbf{u}_D \text{on } \Gamma_D, \quad \boldsymbol{\sigma}\mathbf{n} = \mathbf{t}_N \text{on } \Gamma_N. \tag{3}$$

2.2. Weak form and potential energy

The principle of minimum potential energy states that the equilibrium displacement minimizes the total potential energy:

$$\Pi(\mathbf{u}) = \underbrace{\int_\Omega \frac{1}{2}\boldsymbol{\sigma} : \boldsymbol{\varepsilon}\, \mathrm{d}\Omega}_{U_{\text{int}}} - \underbrace{\int_{\Gamma_N} \mathbf{t}_N \cdot \mathbf{u}\, \mathrm{d}\Gamma}_{W_{\text{ext}}}, \tag{4}$$

where the colon denotes double contraction: $\boldsymbol{\sigma} : \boldsymbol{\varepsilon} = \sigma_{xx}\varepsilon_{xx} + \sigma_{yy}\varepsilon_{yy} + 2\sigma_{xy}\varepsilon_{xy}$.

In a discretization of nodal finite elements with P1 triangles, the internal energy can be approximated as

$$U_{\text{int}} \approx A_\Omega \sum_{i=1}^{N} w_i \tfrac{1}{2}\boldsymbol{\sigma}(\mathbf{u}_i) : \boldsymbol{\varepsilon}(\mathbf{u}_i), \tag{5}$$

where A_Ω is the total domain area and w_i are normalized FEM-consistent lumped nodal area weights so that $\sum_i w_i = 1$. Similarly, the external work of uniform traction on a straight boundary is approximated by a one-dimensional trapezoidal rule over the boundary nodes. This is the discrete energy formulation that we adopt for training the OA-PINN.

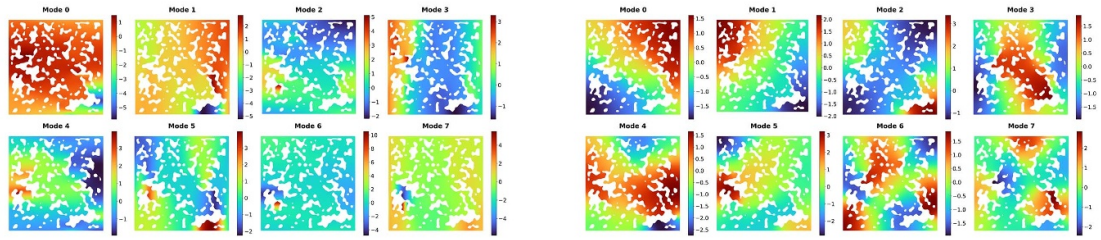

Figure 2. Example elastic eigenmodes used as operator-aware features. Shown on the left are the non-rigid eigenmodes for the elastic operator and on the right for the LBE. Colors indicate the modal amplitude values.

2.3. Physics-informed neural networks

Standard PINN formulation. Parameterize displacement as a neural network $\mathbf{u}_\theta(\mathbf{x})$, and define the loss:

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{PDE}} + \lambda_D \mathcal{L}_{\text{BC}} + \lambda_N \mathcal{L}_{\text{traction}}, \tag{6}$$

where $\mathcal{L}_{\text{PDE}}$ penalizes the equilibrium residual at interior collocation points, $\mathcal{L}_{\text{BC}}$ enforces Dirichlet conditions, and $\mathcal{L}_{\text{traction}}$ enforces Neumann conditions. This strong-form approach requires second derivatives via autodiff (introducing numerical noise), problem-dependent balancing of loss weights λ_D, λ_N, careful sampling of collocation points, and suffers from spectral bias that makes sharp stress gradients difficult to resolve [53].

Weak-form PINNs. Replacing strong-form residuals with energy functional minimization equation (4) improves stability and naturally incorporates traction boundary conditions. However, most implementations still use only raw coordinates as inputs, limiting the representational power on complex geometries.

3. Operator-aware method

3.1. Eigenvalue problem formulation

We compute elastic eigenmodes by solving the generalized eigenvalue problem

$$\mathbf{K}\boldsymbol{\phi}_k = \lambda_k \mathbf{M}\boldsymbol{\phi}_k, \tag{7}$$

where $\mathbf{K} \in \mathbb{R}^{2N\times 2N}$ is the FE stiffness matrix, $\mathbf{M} \in \mathbb{R}^{2N\times 2N}$ is the consistent mass matrix, $\boldsymbol{\phi}_k$ are vector-valued eigenmodes with x- and y-components at each node, and $\lambda_k \geqslant 0$ are the corresponding eigenvalues.

Dirichlet boundary conditions are optionally enforced by eliminating the corresponding DOFs from equation (7) before solving. Examples of the resulting elastic eigenmodes on the porous domain, compared to the LBE eigenfunctions, are shown in figure 2.

The cost of this one-time eigen analysis is comparable to a few linear FEM solves and is amortized over all subsequent OA-PINN training and inference on that geometry. We discard the three rigid-body modes and, for the 2D porous benchmarks, the two lowest global bending modes, by skipping the first $k_{\text{skip}} = 5$ eigen-pairs before constructing the feature tensor. All reference (ground-truth) solutions were computed with COMSOL Multiphysics 6.2, run on CPU on a single 16-core Intel workstation. The resulting symmetric system is solved within approximately 5–15 s depending on the problem and solver.

3.2. Gradient recovery

Each vector eigenmode $\boldsymbol{\phi}_k$ contains $2N$ nodal displacement values. We recover spatial gradients using FE-consistent element-wise computation with area-weighted nodal averaging, summarized in algorithm 1.

This yields four gradient fields per vector mode: $\partial\phi_{k,x}/\partial x$, $\partial\phi_{k,x}/\partial y$, $\partial\phi_{k,y}/\partial x$, and $\partial\phi_{k,y}/\partial y$, each in $\mathbb{R}^N$.

3.3. Feature tensor construction

From K vector eigenmodes, we build scalar feature tensors

$$\Phi \in \mathbb{R}^{N\times C}, \quad \frac{\partial\Phi}{\partial x}, \frac{\partial\Phi}{\partial y} \in \mathbb{R}^{N\times C}, \tag{8}$$

Algorithm 1. Eigenmode gradient recovery.

1: **for** each element e with nodes $\{i,j,k\}$ **do**
2: Compute constant element gradients $\nabla\phi_{k,x}^{(e)}, \nabla\phi_{k,y}^{(e)}$ from P1 shape function derivatives
3: Accumulate to nodes: $(\nabla\phi)_i^{\text{acc}} \leftarrow (\nabla\phi)_i^{\text{acc}} + A_e \cdot \nabla\phi^{(e)}$
4: Accumulate weights: $w_i \leftarrow w_i + A_e$
5: **end for**
6: Nodal gradients: $(\nabla\phi)_i = (\nabla\phi)_i^{\text{acc}} / w_i$

where $C = 2K$ channels stack the x- and y-displacement components of each mode:

$$\Phi = \left[\phi_{1,x},\ \phi_{1,y},\ \phi_{2,x},\ \phi_{2,y},\ \ldots,\ \phi_{K,x},\ \phi_{K,y}\right]. \tag{9}$$

3.4. Normalization

Per-channel z-score normalization improves training stability:

$$\Phi_c^{(n)} = \frac{\Phi_c - \mu_c}{\sigma_c}, \quad \mu_c = \frac{1}{N}\sum_{i=1}^{N}\Phi_{i,c}, \quad \sigma_c = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\Phi_{i,c} - \mu_c)^2}, \tag{10}$$

with consistent gradient scaling: $(\partial\Phi_c^{(n)}/\partial x) = (\partial\Phi_c/\partial x)/\sigma_c$ and likewise for $\partial/\partial y$, so that the relative contribution of each mode to the chain-rule gradients is invariant under normalization.

To avoid sign ambiguity across eigenmode computations, we enforce a deterministic sign convention by flipping each mode so that its nodal mean is nonnegative; the same sign is applied to its gradients.

3.5. OA-PINN architecture

The OA-PINN processes topology features $\boldsymbol{\phi}(\mathbf{x}) \in \mathbb{R}^C$ and spatial coordinates $\mathbf{x} = (x,y) \in [0,1]^2$ through parallel encoding branches:

$$\begin{aligned} \mathbf{z}_\phi &= f_{\text{topo}}\left(\boldsymbol{\phi}; \theta_{\text{topo}}\right) \in \mathbb{R}^H, \\ \mathbf{z}_x &= f_{\text{coord}}\left(\mathbf{x}; \theta_{\text{coord}}\right) \in \mathbb{R}^H, \\ \mathbf{z} &= [\mathbf{z}_\phi, \mathbf{z}_x] \in \mathbb{R}^{2H}, \\ u(\mathbf{x}) &= f_u(\mathbf{z}; \theta_u), \quad v(\mathbf{x}) = f_v(\mathbf{z}; \theta_v), \end{aligned} \tag{11}$$

where $f_{\text{topo}}, f_{\text{coord}}, f_u, f_v$ are multilayer perceptrons with tanh activations. Concretely, each encoder consists of two hidden layers of width $H = 256$, producing latent representations $\mathbf{z}_\phi, \mathbf{z}_x \in \mathbb{R}^{N\times 256}$. These are concatenated and fed to two displacement heads (each with two hidden layers) that output $u_\theta, v_\theta \in \mathbb{R}^N$.

3.6. Chain rule for displacement gradients

Accurate strain computation requires displacement gradients $\nabla\tilde{\mathbf{u}}$. Since $\tilde{\mathbf{u}}(\mathbf{x})$ depends on both coordinates and eigenmode features $\boldsymbol{\phi}(\mathbf{x})$, we apply the total derivative:

$$\frac{\partial\tilde{u}}{\partial x} = \underbrace{\left.\frac{\partial\tilde{u}}{\partial x}\right|_{\boldsymbol{\phi}}}_{\text{coordinate branch}} + \underbrace{\sum_{c=1}^{C}\frac{\partial\tilde{u}}{\partial\phi_c}\frac{\partial\phi_c}{\partial x}}_{\text{topology branch}}, \tag{12}$$

and analogously for $\partial\tilde{u}/\partial y$, $\partial\tilde{v}/\partial x$, and $\partial\tilde{v}/\partial y$.

The first term is obtained by automatic differentiation with respect to the coordinate inputs, treating $\boldsymbol{\phi}$ as fixed. The second term is the dot product between the network's sensitivity to each feature channel $\partial\tilde{u}/\partial\phi_c$ (also obtained by autodiff) and the precomputed eigenmode gradients $\partial\phi_c/\partial x$ at each node. Since the eigenmode gradients $\nabla\boldsymbol{\phi}$ are stored once in memory, only lightweight matrix-vector products are needed during training. Applying equation (12) to all displacement components:

$$\nabla\tilde{\mathbf{u}} = \begin{bmatrix}\partial\tilde{u}/\partial x & \partial\tilde{u}/\partial y \\ \partial\tilde{v}/\partial x & \partial\tilde{v}/\partial y\end{bmatrix} = \nabla\tilde{\mathbf{u}}\big|_{\boldsymbol{\phi}} + \frac{\partial\tilde{\mathbf{u}}}{\partial\boldsymbol{\phi}}\nabla\boldsymbol{\phi}. \tag{13}$$

Strains then follow from the standard small-strain relations.

Algorithm 2. OA-PINN training.

Input: Mesh, eigenmodes $\Phi, \nabla\Phi$, material (E, ν), traction T
Initialize: Network parameters θ
Optimizer: AdamW with lr $= 5 \times 10^{-4}$, weight decay 10^{-5}
for epoch $= 1$ to N_{epochs} **do**
  Forward: evaluate $\tilde{\mathbf{u}}_\theta$ at all nodes
  Gradients: compute $\nabla\tilde{\mathbf{u}}$ via chain rule equation (12)
  Energy: compute $\Pi(\theta)$ via equation (15)
  Backward: $\nabla_\theta \Pi$ via autodiff
  Update: $\theta \leftarrow \theta - \alpha\nabla_\theta \mathcal{L}$ (AdamW step)
  Schedule: cosine annealing
end for

3.7. Weak-form energy training

To improve numerical conditioning, we work with normalized Lamé parameters of equation (2): $\overline{\lambda} = \lambda/E = \nu/[(1+\nu)(1-2\nu)]$ and $\overline{\mu} = \mu/E = 1/[2(1+\nu)]$, so that the normalized stresses are

$$\begin{aligned} \overline{\sigma}_{xx} &= \left(\overline{\lambda} + 2\overline{\mu}\right)\varepsilon_{xx} + \overline{\lambda}\,\varepsilon_{yy}, \\ \overline{\sigma}_{yy} &= \overline{\lambda}\,\varepsilon_{xx} + \left(\overline{\lambda} + 2\overline{\mu}\right)\varepsilon_{yy}, \\ \overline{\sigma}_{xy} &= 2\overline{\mu}\,\varepsilon_{xy}, \end{aligned} \tag{14}$$

with physical stresses recovered as $\boldsymbol{\sigma} = E\overline{\boldsymbol{\sigma}}$. Training minimizes the discretized total potential energy

$$\Pi(\theta) = A_\Omega \sum_{i=1}^{N} w_i \tfrac{1}{2}\,\overline{\boldsymbol{\sigma}}_i : \boldsymbol{\varepsilon}_i \;-\; \bar{T}\int_{\Gamma_N} \tilde{u}_n \,\mathrm{d}\Gamma, \tag{15}$$

where $\bar{T}$ is the applied traction vector and $\tilde{u}_n$ is the displacement component in the traction direction. The double contraction uses the tensor convention: $\overline{\boldsymbol{\sigma}} : \boldsymbol{\varepsilon} = \overline{\sigma}_{xx}\varepsilon_{xx} + \overline{\sigma}_{yy}\varepsilon_{yy} + 2\overline{\sigma}_{xy}\varepsilon_{xy}$. The boundary integral is evaluated with a one-dimensional trapezoidal rule over boundary nodes. The training loss is $\mathcal{L}(\theta) = \Pi(\theta)$ (algorithm 2), scaled by a constant factor for numerical conditioning.

4. Implementation details

4.1. Mesh generation and preprocessing

Triangular meshes are generated with `gmsh` or the `triangle` library at typical resolutions (∼30 000 DOFs). Quadrilateral elements, when present, are split into two triangles for consistency. Element areas and FEM-consistent lumped nodal weights (each node receives one-third of the area of every element to which it belongs) are precomputed once.

4.2. Eigenmode computation

Eigenpairs are obtained via shift-invert Lanczos targeting the K smallest positive eigenvalues with shift $\tau = -10^{-6}$ and a regularization of $10^{-10}\,\mathbf{M}$ added to $\mathbf{K}$ for numerical stability. Fixed-boundary DOFs are eliminated before solving. The eigen-analysis is implemented in Python (NumPy/SciPy). For each geometry we assemble the sparse generalized eigenproblem on a linear triangular mesh and solve it with `scipy.sparse.linalg.eigsh` in shift–invert mode for the lowest modes. The wall-clock cost of this one-time step is exactly the 'Pre (s)' column of tables 3–6.

4.3. Neural network training

We use PyTorch 2.0+ with the following hyperparameters; topology encoder $\mathbb{R}^C \to 256 \to 256$; coordinate encoder $\mathbb{R}^2 \to 256 \to 256$; displacement heads $\mathbb{R}^{512} \to 256 \to 256 \to 1$ (with small-magnitude final-layer initialization); tanh activations throughout; AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, lr $= 5 \times 10^{-4}$, weight decay $= 10^{-5}$); cosine annealing schedule; gradient clipping at max norm 1.0; full-batch evaluation at all mesh nodes per epoch. The architecture and the optimizer settings were chosen by light empirical tuning guided by standard practice for deep-energy method PINNs. The same architecture, optimizer, and schedule were fixed across all benchmarks and shared identically by every neural baseline, so that the comparisons isolate the effect of the input representation rather than per-problem tuning.

Table 2. Summary of the benchmark problems. '*S*' is scalar Laplacian eigenmodes; '*V*' is elastic vector eigenmodes.

#	Problem	Type	C	Epochs	Γ_{bot}	Γ_{top}	Key parameters
1a	Isotropic tension	V	32	5000	$\mathbf{u}=\mathbf{0}$	$\bar{T}_y$=0.1 Pa	E=1 Pa, ν=0.3
1b	Isotropic shear	V	36	5000	$\mathbf{u}=\mathbf{0}$	$\bar{T}_x$=0.1 Pa	E=1 Pa, ν=0.49
1c	Orthotropic tension	V	32	5000	$\mathbf{u}=\mathbf{0}$	$\bar{T}_y$=0.01 Pa	E_1=1, E_2=0.5, G_{12}=0.1 Pa, ν=0.3
1d	High-porosity tension	V	36	8000	$\mathbf{u}=\mathbf{0}$	$\bar{T}_y$=0.01 Pa	E=1 Pa, ν=0.3; porosity≈50 %
2	Neo-Hookean	V	48	5000	$\mathbf{u}=\mathbf{0}$	$\bar{T}_y$=0.01 Pa	E=1 Pa, ν=0.3
3	Thermoelastic (stg. 1)	S	16	3000	T=0	$\bar{q}$=1	$k=1$
	Thermoelastic (stg. 2)	V	32	5000	$\mathbf{u}=\mathbf{0}$	$\bar{T}_y$=0.001 Pa	E=2 Pa, $\alpha=1.2\times10^{-5}$
4	Inverse material	V	48	5000×10	$\mathbf{u}=\mathbf{0}$	$\bar{T}_y=\bar{T}_x$=0.01 Pa	E_1, E_2, G_{12} learnable

5. Numerical experiments

For all experiments, we use the same architecture, optimizer, and learning-rate schedule described in section 4. All models are trained on a single NVIDIA A6000 GPU.

5.1. Boundary condition enforcement

Dirichlet conditions on the bottom edge ($y=0$) are imposed by multiplying the raw output of the network $\hat{u}$ by a smooth mask $m(y)$. At $y=0$, $u=\hat{u}\cdot 0=0$ (Dirichlet satisfied), while the normal derivative is

$$\frac{\partial}{\partial y}[u]\Big|_{y=0} = \hat{u}(x,0) \neq 0 \quad \text{in general,} \tag{16}$$

so the derivative $\partial u/\partial y|_{y=0}$ is unconstrained. The bottom boundary behaves as a standard Dirichlet face with free normal derivative.

5.2. Benchmark problems

We evaluate the proposed framework on problems spanning different categories of PDE physics; linear and nonlinear solid mechanics, coupled multi-field systems, and inverse material identification. Table 2 collects the key parameters. We denote by $q^{\text{FEM}}(x)$ the FEM reference field and by $q^{\text{OA}}(x)$ the corresponding prediction of OA-PINN (displacement or stress). The point-wise error is

$$e(x) = q^{\text{OA}}(x) - q^{\text{FEM}}(x). \tag{17}$$

For visualization, we introduce a range-normalized error

$$\tilde{e}(x) = \frac{e(x)}{\max_{y\in\Omega} q^{\text{FEM}}(y) - \min_{y\in\Omega} q^{\text{FEM}}(y)}. \tag{18}$$

Unless stated otherwise, all error colormaps show $\tilde{e}(x)$ and use symmetric 99th-percentile clipping of $\tilde{e}$ to define the color scale, so that the plots reflect the typical error level in the majority of the domain while preventing a small number of extreme hot-spot outliers from dominating the visualization.

The accuracy of the PINN predictions (tables 3–6) is evaluated using the relative L^2 error, defined as:

$$\varepsilon_{L^2} = \frac{\left\|q^{\text{OA}} - q^{\text{FEM}}\right\|_2}{\left\|q^{\text{FEM}}\right\|_2} \times 100\% \tag{19}$$

where $q \in \{u, v, \sigma_{\text{vm}}\}$.

To enable fair comparison, all neural baselines (Coord-Only PINN, Vanilla DEM, FF-PINN, FEINN, Finite-PINN) use the same parameter budget, the same AdamW optimizer with cosine annealing, the same epoch budget per problem, and the same full-batch evaluation on the FE mesh. For FF-PINN we additionally swept the Fourier scale $\sigma \in \{1, 5, 10, 20, 50, 100\}$ and report the best result in each table.

5.3. Problem 1: forward elasticity on a porous domain

The first benchmark evaluates the OA-PINN on a realistic porous microstructure under three sub-cases that progressively increase the role of displacement coupling and material anisotropy. The domain is a unit square that contains irregularly shaped pores arranged in a three-scale hierarchy (primary, secondary, tertiary), yielding a solid area fraction of approximately 0.70.

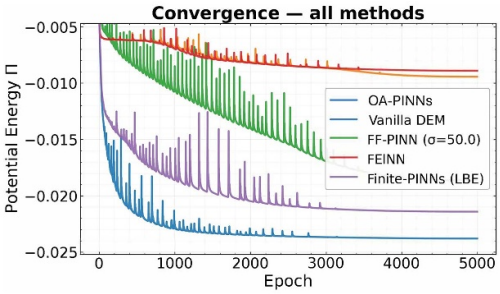

Figure 3. Problem 1(a) training convergence; potential energy Π vs epoch for all baselines.

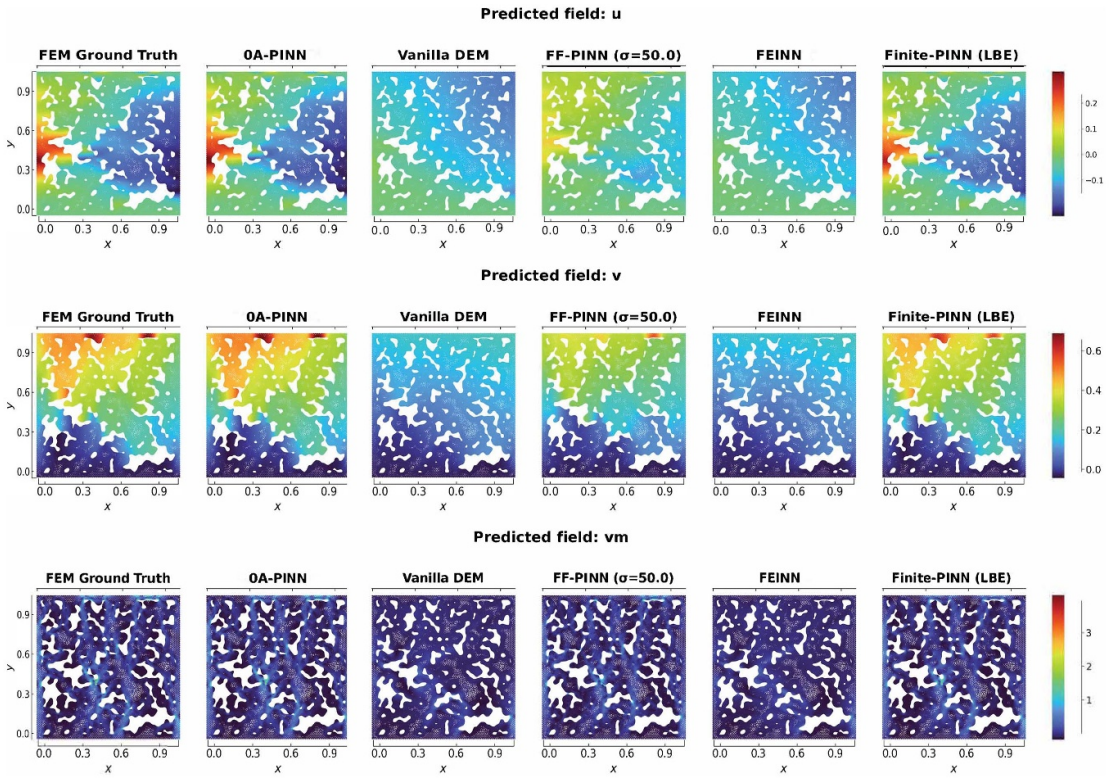

Figure 4. Problem 1(a), isotropic tension on a porous domain. The panels show displacement magnitude and stress fields.

Boundary conditions (common to all sub-cases): $u = v = 0$ on Γ_{bot} via the linear mask $m(y) = y$; lateral edges and pore surfaces are traction-free. FEM reference solutions are obtained with a direct sparse solver on the same mesh.

(a) Isotropic uniaxial tension. Uniform vertical traction $\bar{T}_y = 0.1$ Pa is applied on Γ_{top} ($\bar{T}_x = 0$). The total potential energy reads

$$\Pi[\mathbf{u}] = \tfrac{1}{2}\int_{\Omega} \bar{\boldsymbol{\sigma}} : \boldsymbol{\varepsilon}\, \mathrm{d}\Omega \; - \; \bar{T}_y \int_{\Gamma_{\text{top}}} v \, \mathrm{d}\Gamma. \tag{20}$$

The predicted fields are shown in figure 4, with training convergence in figure 3 and quantitative errors reported in table 3. The von-Mises error colormap is scaled to emphasize localized stress hot-spots; therefore, a predominantly dark panel indicates that the corresponding method fails to reproduce these localized stress concentrations.

(b) Isotropic pure shear. Uniform horizontal traction $\bar{T}_x = 0.1$ Pa is applied on Γ_{top} ($\bar{T}_y = 0$). The response is governed by the shear strain $\varepsilon_{xy} = \frac{1}{2}(\partial u/\partial y + \partial v/\partial x)$, which couples the two displacement components through the off-diagonal term. A nearly incompressible Poisson ratio ($\nu = 0.49$) is chosen

Table 3. Benchmark comparison. All NN baselines are parameter-matched. FF-PINN uses best σ from the sweep.

Method	u	v	VM	Π	Pre (s)	Train (s)	#Params
OA-PINN	0.0295	0.0144	0.1682	-2.379×10^{-02}	4.4	239.9	536 650
Coord-Only PINN †	24.1885	169.0077	0.9139	-1.246×10^{01}	4.4	146.5	536 650
Vanilla DEM	0.7807	0.5841	0.7217	-9.433×10^{-03}	0.0	173.9	535 093
FF-PINN ($\sigma = 50.0$)	0.6601	0.2394	0.4854	-1.863×10^{-02}	0.0	719.8	536 282
FEINN	0.7772	0.6093	0.7247	-8.901×10^{-03}	12.5	239.8	535 093
Finite-PINNs (LBE)	0.1520	0.0746	0.3202	-2.141×10^{-02}	4.4	241.6	536 650

Ablation of the hybrid chain rule, not an independent baseline. The derivatives are calculated only from the coordinates, not using the hybrid formulation.

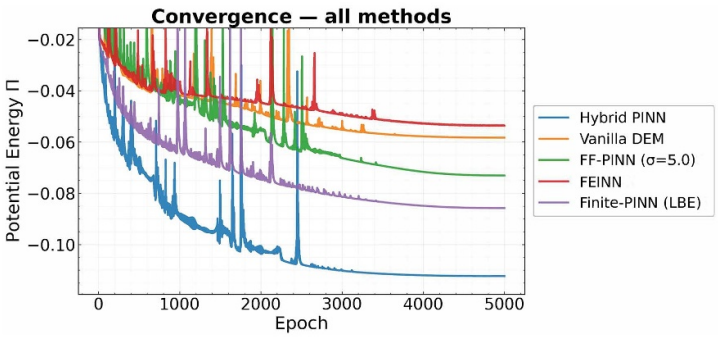

Figure 5. Problem 1(b) training convergence; potential energy Π vs epoch for all baselines.

to approach the volumetric locking regime, creating a stringent test. The nearly divergence-free constraint couples the two displacement components strongly through the volumetric term, which scalar eigenfunctions cannot represent. The energy functional becomes

$$\Pi[\mathbf{u}] = \tfrac{1}{2}\int_{\Omega} \overline{\boldsymbol{\sigma}} : \boldsymbol{\varepsilon}\, \mathrm{d}\Omega \; - \; \bar{T}_x \int_{\Gamma_{\mathrm{top}}} u\, \mathrm{d}\Gamma. \tag{21}$$

The predicted fields are shown in figure 6. Quantitative errors are reported in table 4. With momentum-based updates on a non-convex energy landscape, an individual step can overshoot and momentarily raise the energy, after which the optimizer recovers within a few iterations, as shown in figure 5. These peaks are more prominent for OA-PINN for two reasons, first, the operator-enriched model converges to a substantially lower energy, so a fixed-size excursion is visually larger and can briefly rise above the plateau of a baseline that simply never reaches that low-energy region; second, the hybrid chain-rule gradient through the eigenmode branch adds curvature that makes occasional overshoot more likely. Gradient clipping (max-norm 1.0) is already applied and bounds the gradient magnitude but does not eliminate these transients.

(c) Orthotropic uniaxial tension. The same geometry and loading direction as in sub-case (a), but with orthotropic material: $E_1 = 1$ Pa, $E_2 = 0.5$ Pa, $G_{12} = 0.1$ Pa, $\nu_{12} = 0.30$. The constitutive matrix reads

$$\mathbf{C} = \begin{bmatrix} C_{11} & C_{12} & 0 \\ C_{12} & C_{22} & 0 \\ 0 & 0 & G_{12} \end{bmatrix}, \tag{22}$$

with the plane-strain reduction $\nu_{13} = \nu_{23} = \nu_{12}$ and $\nu_{21} = \nu_{12}E_2/E_1$, $\nu_{31} = \nu_{13}$, $\nu_{32} = \nu_{23}E_2/E_1$. The stiffness coefficients are

$$C_{11} = \frac{E_1\,(1-\nu_{23}\,\nu_{32})}{\Delta},\; C_{22} = \frac{E_2\,(1-\nu_{13}\,\nu_{31})}{\Delta},\; C_{12} = \frac{E_1\,(\nu_{21}+\nu_{31}\,\nu_{23})}{\Delta}, \tag{23}$$

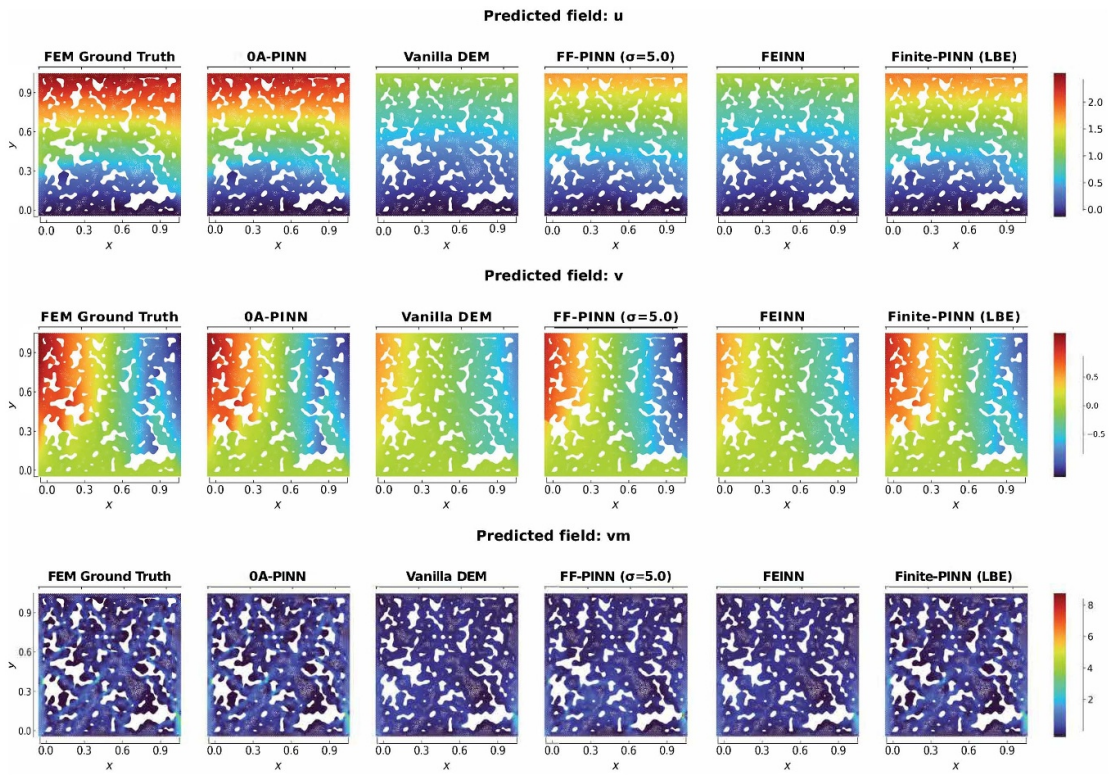

Figure 6. Problem 1(b), isotropic pure shear on a porous domain.

Table 4. Benchmark comparison. All NN baselines are parameter-matched. FF-PINN uses best σ from the sweep.

Method	u	v	VM	Π	Pre (s)	Train (s)	#Params
OA-PINN	0.0456	0.0495	0.2594	-1.144×10^{-01}	3.9	338.3	536 650
Coord-Only PINN †	45.9995	1.0034	0.9779	-2.092×10^{01}	3.9	206.0	536 650
Vanilla DEM	0.5237	0.5263	0.6837	-5.862×10^{-02}	0.0	244.5	535 093
FF-PINN ($\sigma = 5.0$)	0.2917	0.3101	0.6614	-9.483×10^{-02}	0.0	1012.2	536 282
FEINN	0.5172	0.5173	0.6822	-5.934×10^{-02}	12.5	336.9	535 093
Finite-PINNs (LBE)	0.2404	0.2000	0.4820	-9.245×10^{-02}	2.9	342.3	536 650

Ablation of the hybrid chain rule, not an independent baseline. The derivatives are calculated only from the coordinates, not using the hybrid formulation.

with

$$\Delta = 1 - \nu_{12}\nu_{21} - \nu_{23}\nu_{32} - \nu_{13}\nu_{31} - 2\nu_{21}\nu_{32}\nu_{13}\,. \tag{24}$$

This sub-case directly tests the central claim of OA-PINN. The Laplacian eigenfunctions are purely geometric. Switching from the isotropic material of the sub-case (a) to the present orthotropic material leaves all modes unchanged, despite a two-fold stiffness ratio (E_1/E_2) and a reduced shear modulus G_{12} that fundamentally alters the coupling between u and v. As illustrated, OA-PINN has a larger accuracy advantage in this sub-case (figure 9).

Training convergence is reported in figure 8 and quantitative errors are reported in table 5.

The energy functional is identical to equation (20), with $\overline{\boldsymbol{\sigma}} = \mathbf{C}\boldsymbol{\varepsilon}$ replacing the isotropic constitutive law. The orthotropic eigenmodes differ visibly from the isotropic ones (figure 7).

(d) Isotropic tension on a high-porosity domain. Same material and loading direction on domain with ∼50% porosity. The thinner ligaments produce steeper gradients near pore boundaries. This sub-case serves two purposes. First, it tests the robustness of the OA-PINN under severe geometric complexity. Second, it quantifies how the advantage of operator-aware features over scalar eigenfunctions scales with topology complexity. Predicted fields are shown in figure 11 and the convergence curve in figure 10.

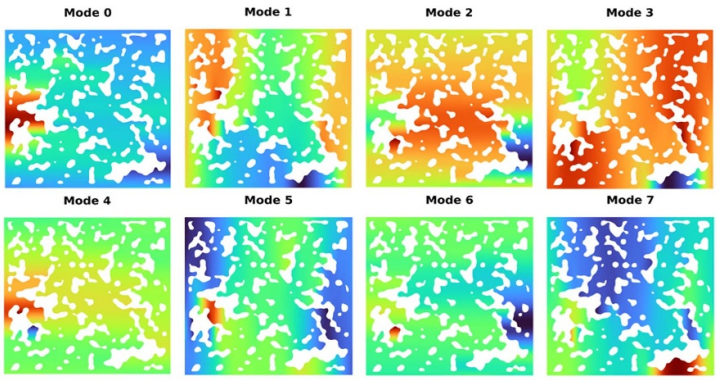

Figure 7. Example elastic eigenmodes used as operator-aware features in orthotropic material. Shown are the non-rigid eigenmodes.

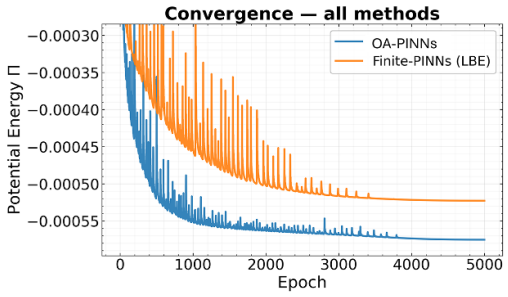

Figure 8. Problem 1(c) training convergence; potential energy Π vs epoch for all baselines.

Table 5. Benchmark comparison.

Method	u	v	VM	Π	Pre (s)	Train (s)	#Params
OA-PINN	0.0368	0.0203	0.1718	-5.754×10^{-04}	6.2	530.7	535 618
Coord-Only PINN †	10.2582	306.2466	1.0524	-1.437×10^{00}	6.2	285.6	535 618
Finite-PINNs (LBE)	0.1420	0.0600	0.2569	-5.228×10^{-04}	3	247.1	535 618

Ablation of the hybrid chain rule, not an independent baseline. The derivatives are calculated only from the coordinates, not using the hybrid formulation.

Table 6. Benchmark comparison. All NN baselines are parameter-matched. FF-PINN uses best σ from the sweep.

Method	u	v	VM	Π	Pre (s)	Train (s)	#Params
OA-PINN	0.0676	0.0392	0.2407	-4.184×10^{-05}	3.1	329.7	535 618
Coord-Only PINN †	4.1126	146.7327	1.2110	-4.053×10^{-02}	3.1	201.0	535 618
Vanilla DEM	0.9997	0.9875	0.9464	-8.738×10^{-07}	0.0	234.9	535 093
FF-PINN ($\sigma = 20.0$)	1.0136	1.0059	0.9096	-3.739×10^{-06}	0.0	954.8	535 535
FEINN	0.9997	0.9875	0.9464	-8.736×10^{-07}	9.5	279.8	535 093
Finite-PINNs (LBE)	0.4268	0.3399	0.5155	-2.892×10^{-05}	2.8	330.6	535 618

Ablation of the hybrid chain rule, not an independent baseline. The derivatives are calculated only from the coordinates, not using the hybrid formulation.

Quantitative errors are reported in table 6. As shown, Vanilla-DEM, FF-PINN, and FEINN all reach relative L^2 displacement errors of roughly 100% ($u, v \approx 0.99$–1.01) and von Mises errors above 90% and collapse to a nearly uniform field that ignores the pore-induced stress concentrations, whereas OA-PINN attains u-, v-, and von Mises errors of 0.068, 0.039, and 0.241.

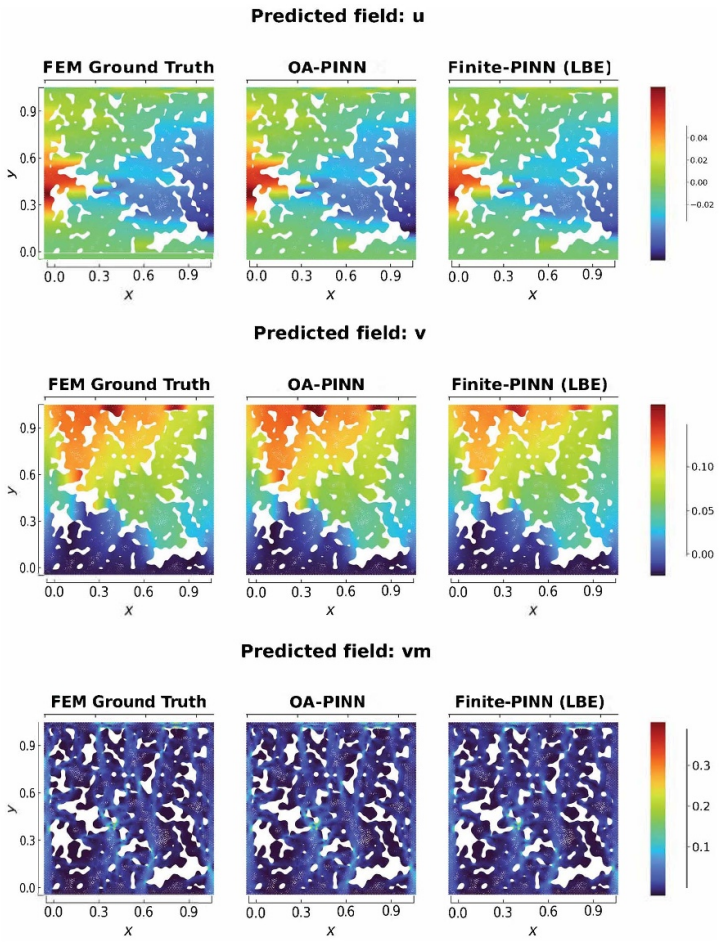

Figure 9. Problem 1(c), orthotropic tension on a porous domain.

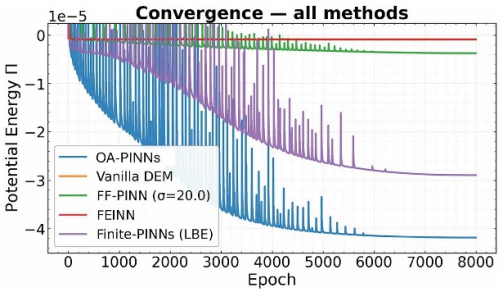

Figure 10. Problem 1(d) training convergence; potential energy Π vs epoch for all baselines.

5.4. Problem 2: Neo-Hookean hyperelasticity

This benchmark extends the framework to geometrically nonlinear elasticity in the same porous domain. The stored-energy density of a compressible Neo-Hookean solid is

$$W(\mathbf{F}) = \tfrac{1}{2}\overline{\mu}\,(I_1 - 2 - 2\ln J) + \tfrac{1}{2}\overline{\lambda}\,(\ln J)^2, \tag{25}$$

where $\mathbf{F} = \mathbf{I} + \nabla\mathbf{u}$ is the deformation gradient, $J = \det\mathbf{F}$, and $I_1 = \mathrm{tr}(\mathbf{F}^\top\mathbf{F})$.

Both displacement components are fixed on the bottom edge, $u = v = 0$ on Γ_{bot}. Vertical traction $\bar{T} = 0.01$ Pa is applied on Γ_{top}. The lateral edges and the pore surfaces are traction-free. The total potential energy reads

$$\Pi[\mathbf{u}] = \int_\Omega W(\mathbf{F})\,\mathrm{d}\Omega \;-\; \bar{T}\int_{\Gamma_{\mathrm{top}}} v\,\mathrm{d}\Gamma. \tag{26}$$

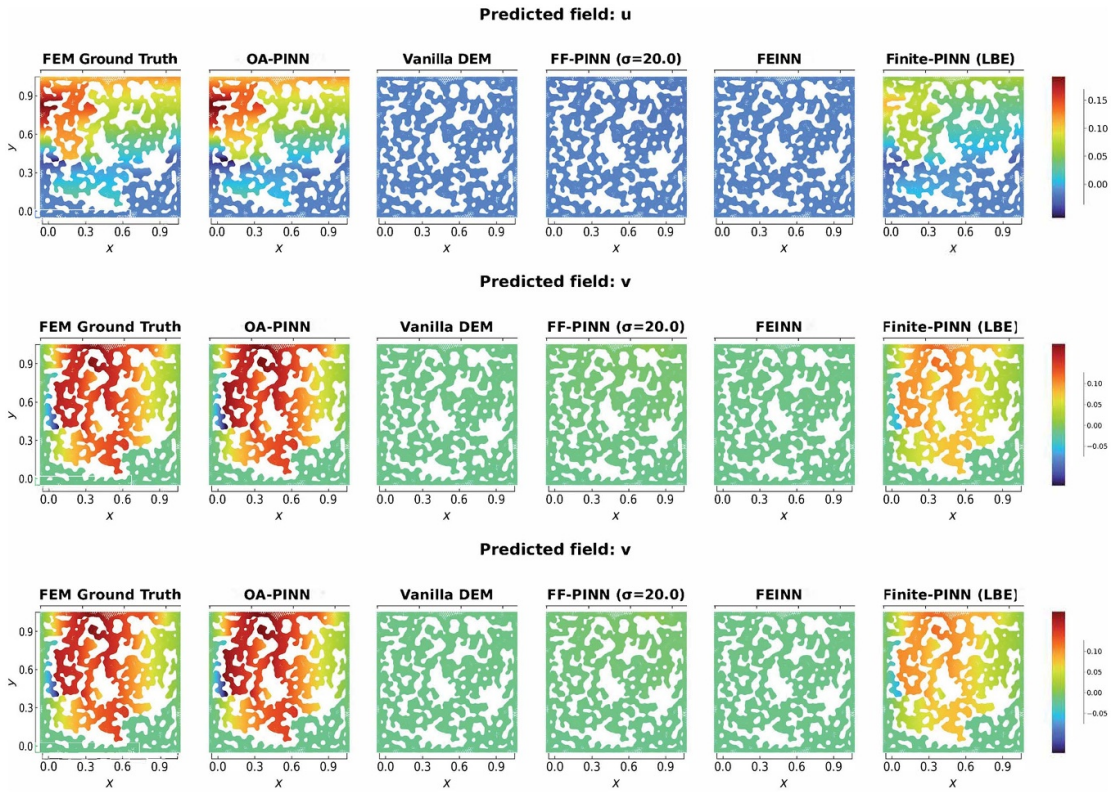

Figure 11. Problem 1(d), isotropic high-porosity tension on a porous domain.

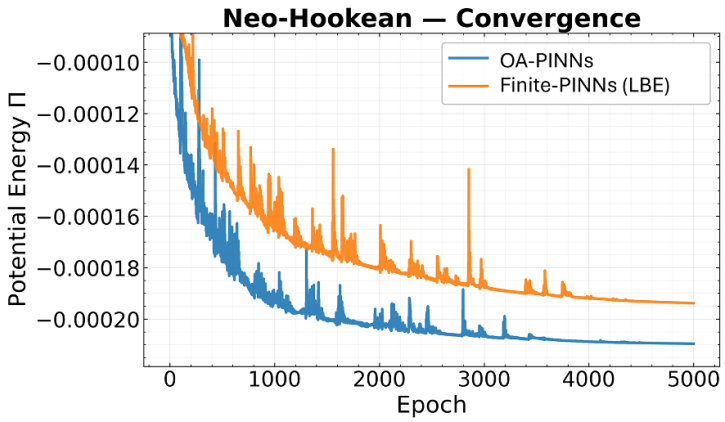

Figure 12. Problem 2 training convergence.

To prevent $\ln J \to -\infty$ during early optimization, the deformation-gradient determinant is clamped at $J_{\min} = 10^{-8}$. No Newton iteration is required since the network directly minimizes the nonlinear energy. The eigenmodes are elastic vectors, with $C = 48$. The predicted fields and errors are shown in figure 13 and the corresponding convergence curve is given in figure 12.

5.5. Problem 3: thermoelastic coupling (sequential)

A two-stage sequential solution couples thermal and elastic physics.

Stage 1 (thermal). The steady heat equation is solved using the same weak-form energy approach (thermal conductivity $k = 1$, normalized boundary flux $\bar{q} = 1$) using 16 scalar modes and 3000 epochs to obtain $T(\mathbf{x})$. The boundary conditions are identical to Problem 1, $T = 0$ on Γ_{bot} (mask y; flux unconstrained); heat flux $\bar{q} = 1$ on Γ_{top}.

Stage 2 (elastic). The frozen temperature field induces a thermal strain $\boldsymbol{\varepsilon}_{\text{th}} = \alpha T(1, 1, 0)^{\top}$ with a thermal-expansion coefficient $\alpha = 1.2 \times 10^{-5}\ \text{K}^{-1}$. The mechanical strain is $\boldsymbol{\varepsilon}_{\text{mech}} = \boldsymbol{\varepsilon}_{\text{total}} - \boldsymbol{\varepsilon}_{\text{th}}$, and

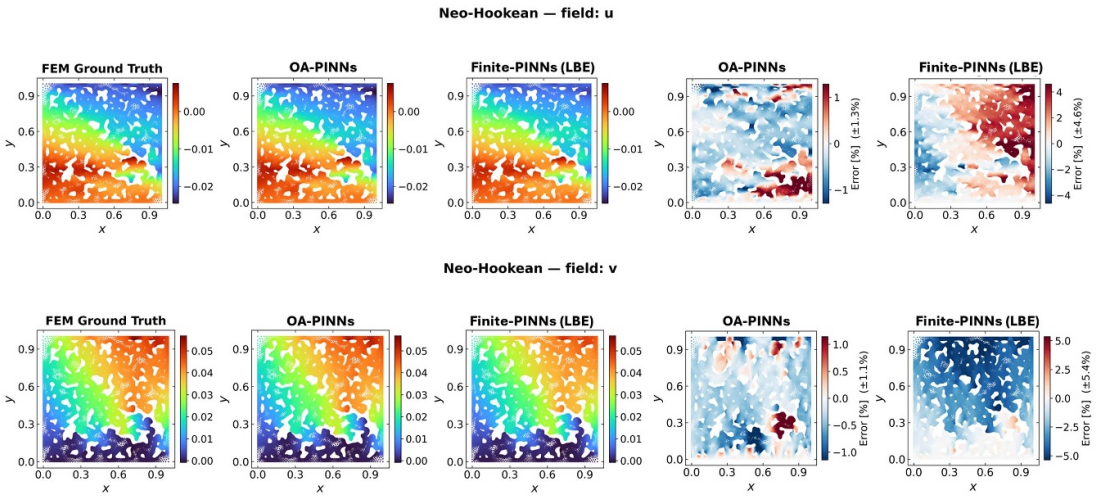

Figure 13. Problem 2, Neo-Hookean hyperelasticity on a porous domain.

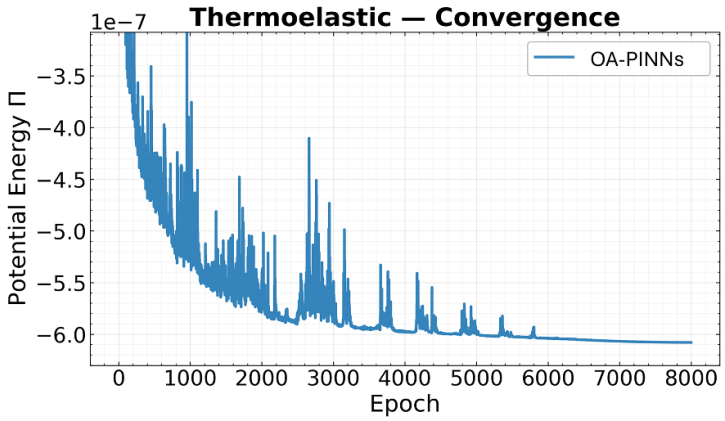

Figure 14. Problem 3 (thermoelastic, stage 2) training convergence.

the Cauchy stress follows from a plane-strain constitutive law with $E = 2$ Pa, $\nu = 0.3$. Vertical traction $\bar{T} = 0.001$ Pa on Γ_{top} (normalized by E for the network); traction-free lateral edges. The elastic energy reads

$$\Pi[\mathbf{u}] = \tfrac{1}{2}\int_{\Omega} \boldsymbol{\sigma} : \varepsilon_{\text{mech}}\, \mathrm{d}\Omega \; - \; \bar{T}\int_{\Gamma_{\text{top}}} v\, \mathrm{d}\Gamma, \tag{27}$$

where $\boldsymbol{\sigma} = \overline{\mathbf{C}} : (\varepsilon_{\text{total}} - \varepsilon_{\text{th}})$ and the integrand is $\boldsymbol{\sigma} : \varepsilon_{\text{mech}}$, rather than $\boldsymbol{\sigma} : \varepsilon_{\text{total}}$, so the thermal strain acts as a prestress rather than external body force. The convergence curve is presented in figure 14 and predicted displacement fields are shown in figure 15.

5.6. Problem 4: inverse material identification

This problem inverts the forward pipeline, given sparse, noisy displacement observations and a known applied traction, recover the unknown material stiffness tensor. The benchmark tests whether the operator-aware architecture, originally designed for forward prediction, can simultaneously serve as the backbone of a physics-constrained inverse solver.

The unknown constitutive law is an orthotropic plane-strain material described by three learnable parameters; E_1, E_2, and G_{12} (equations (22)–(24)). The inverse architecture is illustrated in figure 16.

Simultaneous gradient descent on $(\theta, \mathbf{C})$ to minimize $\Pi(\mathbf{u}_\theta; \mathbf{C}) + \lambda \mathcal{L}_{\text{data}}$ leads to two pathologies. First, because the network forward pass $\hat{\mathbf{u}} = \text{net}(\phi, \mathbf{x})$ does not explicitly depend on $\mathbf{C}$, the instantaneous

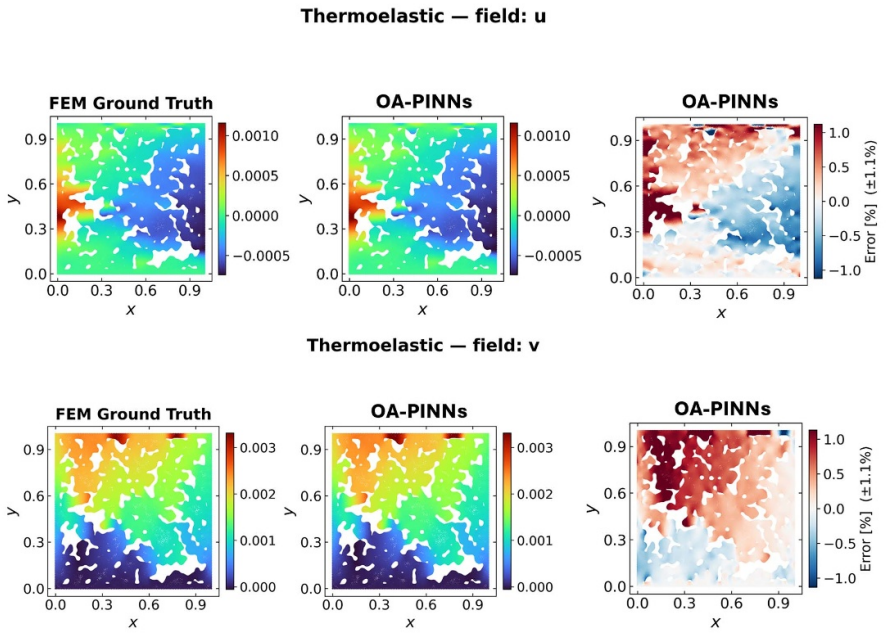

Figure 15. Problem 3; predicted displacement fields (u, v) and von Mises stress for sequential thermoelastic coupling.

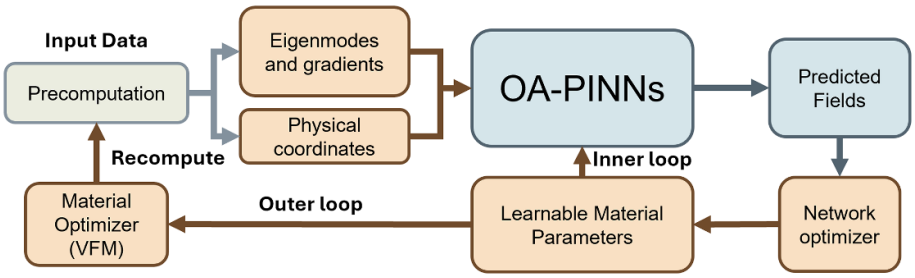

Figure 16. Inverse architecture for material properties estimation.

gradient $\partial\mathcal{L}_{\text{data}}/\partial\mathbf{C} = 0$ vanishes at every step. Second, the internal energy $U_{\text{int}} = \frac{1}{2}\int \boldsymbol{\varepsilon} : \mathbf{C} : \boldsymbol{\varepsilon}\, \mathrm{d}\Omega$ is linear in $\mathbf{C}$, so $\partial\Pi/\partial\mathbf{C} > 0$ whenever $\boldsymbol{\varepsilon} \neq 0$, biasing $\mathbf{C}$ towards the trivial null minimizer $\mathbf{C} \to 0$. The alternating VFM scheme below is a lightweight alternative that sidesteps the inner-solve cost while retaining identifiability.

We use the eigenmode basis as virtual test functions for a weak-form equilibrium residual. For each eigenmode $\boldsymbol{\phi}_k$ with precomputed virtual strain $\delta\boldsymbol{\varepsilon}_k$, equilibrium requires

$$R_k(\mathbf{C}) = \int_\Omega \boldsymbol{\sigma}\left(\mathbf{C}, \boldsymbol{\varepsilon}_{\text{pred}}\right) : \delta\boldsymbol{\varepsilon}_k \, \mathrm{d}\Omega - \int_{\Gamma_{\text{top}}} \overline{\mathbf{T}} \cdot \boldsymbol{\phi}_k \, \mathrm{d}\Gamma = 0, \tag{28}$$

where $\boldsymbol{\varepsilon}_{\text{pred}}$ is the strain field of the current network (detached from the computation graph) and $\boldsymbol{\sigma} = \overline{\mathbf{C}} : \boldsymbol{\varepsilon}_{\text{pred}}$ retains differentiability with respect to $\mathbf{C}$. The VFM loss is

$$\mathcal{L}_{\text{VFM}}(\mathbf{C}) = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{R_k}{W_{\text{scale}}}\right)^2, \tag{29}$$

normalized by $W_{\text{scale}} = \text{mean}(|W_{\text{ext},k}|)$. This loss has three properties absent from the direct approach; (i) $\partial\mathcal{L}_{\text{VFM}}/\partial\mathbf{C} \neq 0$ (through $\boldsymbol{\sigma}$); (ii) $\mathbf{C} = 0$ is not a minimizer (the traction integral is nonzero); and (iii) at $\mathbf{C} = \mathbf{C}_{\text{true}}$ and $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_{\text{true}}$ the residual vanishes exactly. Each inner loop alternates two sub-steps:

- **Network** θ: $\min_\theta \Pi(\mathbf{u}_\theta; \mathbf{C}_{\text{fixed}}) + \lambda\mathcal{L}_{\text{data}}$. The material parameters are detached; gradients flow only to the network.

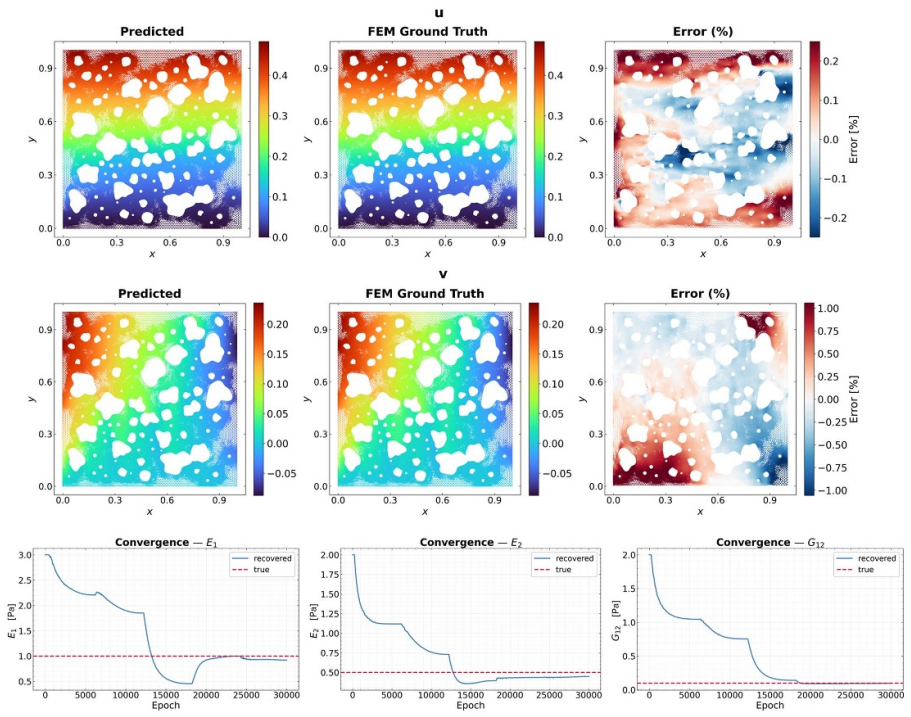

Figure 17. Problem 4; observed vs predicted displacement field (u, v) and reconstruction error at the reference noise/density setting (3% noise, 15% observations).

- **Material C**: $\min_{\mathbf{C}} \mathcal{L}_{\mathrm{VFM}}(\boldsymbol{\varepsilon}_{\mathrm{detached}}; \mathbf{C})$. Strains are detached; gradients flow only to material parameters.

A 200-epoch warmup trains the network alone before activating material updates. Ten outer loops recompute the eigenmodes each running 5000 inner epochs.

Ground-truth displacement fields are generated from an orthotropic material with $E_1 = 1$ Pa, $E_2 = 0.5$ Pa, $\nu_{12} = 0.3$, $G_{12} = 0.1$ Pa, under combined normal and shear traction $\overline{\mathbf{T}} = (\bar{T}_x, \bar{T}_y) = (T_{\mathrm{shear}}, T_{\mathrm{norm}})$ at the top edge (lateral edges are traction-free). The observation data consists of 15% of nodes sampled uniformly at random, corrupted with 3% Gaussian noise (figure 17). Initial guess: $E_1 = 3, E_2 = G_{12} = 2$ Pa. The VFM virtual fields are the 24 lowest elastic vector eigenmodes. The eigenmodes are elastic vectors, with $C = 48$. Alternating optimization recovers E_1, E_2, and G_{12} to within 1.9%, 5.2%, and 0.7% of their ground-truth values, as shown in figure 17.

6. Discussion

The numerical experiments in section 5.2 demonstrate that operator-aware features systematically improve both forward predictive accuracy and inverse material identification from sparse and noisy data. We first examine the quality of the predicted stress fields, a practically important but often underreported diagnostic for PINNs, and then organize the remaining discussion around three themes on how the network exploits the eigenmode embedding, the advantages and practical implications of operator-aware features for challenging forward problems, and the behavior of the inverse formulation, followed by an assessment of current limitations.

6.1. Stress-field accuracy

Because the OA-PINN is trained by minimizing total potential energy rather than point-wise stress residuals, stress accuracy is a derived quantity that depends on the smoothness and correctness of the predicted strain field. To assess this rigorously, figure 18 presents predicted-versus-reference scatter plots of the von Mises stress σ_{vm} at every mesh node for two representative sub-cases that isolate distinct physical challenges.

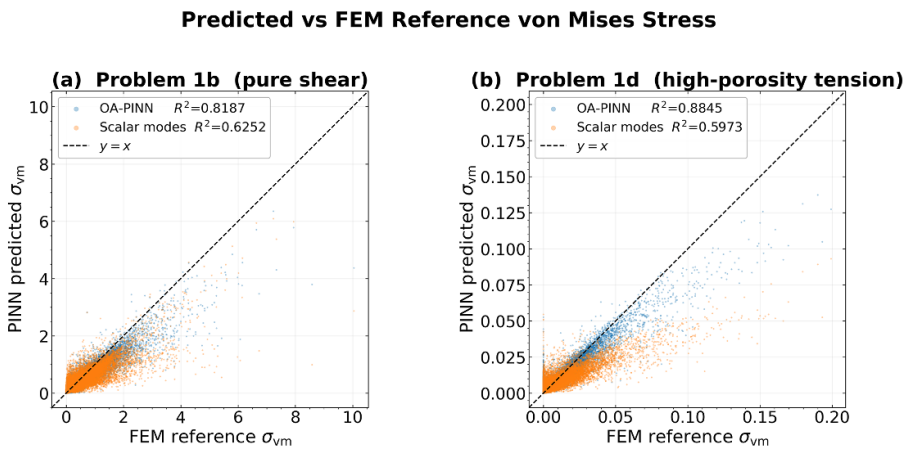

Figure 18. Predicted versus FEM reference von Mises stress at all interior nodes for (a) Problem 1b (isotropic pure shear, $\nu = 0.49$) and (b) Problem 1d (isotropic tension on a high-porosity domain).

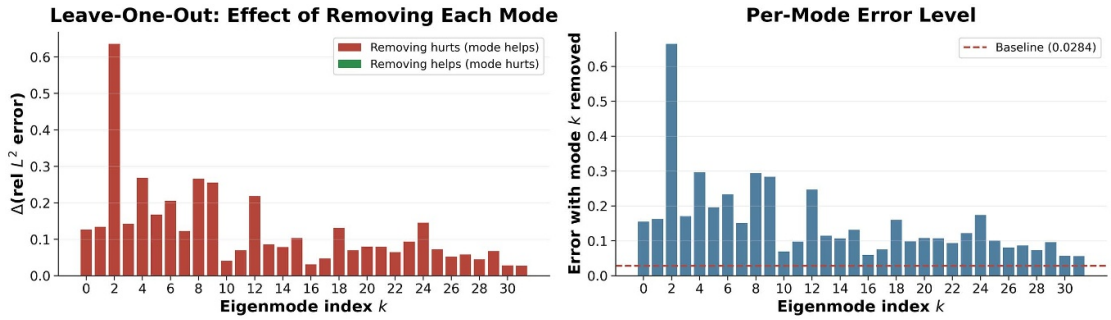

Figure 19. Leave-one-out ablation showing the change in relative L^2 displacement error when each eigenmode is removed individually (left) and the absolute error level with each mode removed (right).

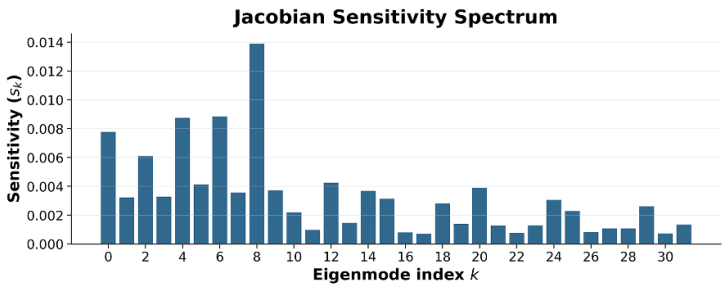

Figure 20. Jacobian sensitivity spectrum quantifying the dependence of the network output on each eigenmode. Mode index k begins after exclusion of the first five rigid-body eigenvectors.

In Problem 1b (figure 18(a)), the OA-PINN achieves $R^2 = 0.81$ versus $R^2 = 0.62$ for the scalar-mode PINNs, which systematically under-predict high stress values due to the absence of shear-coupling information in scalar modes. In the high porosity Problem 1d (figure 18(b)), the gap widens further as OA-PINN achieves $R^2 = 0.88$ while the scalar-mode PINN drops to $R^2 = 0.59$. The order-of-magnitude difference confirms that the elastic eigenmodes provide a feature basis aligned with the mechanical response, not merely the domain shape.

6.2. Modal sensitivity and energy analysis

We characterized how the trained network exploits the embedding of the eigenmode using five diagnostics in Problem 1a. Leave-one-out ablation (see figure 19) confirms that every mode contributes constructively; the most influential is $k = 2$, whose removal raises the relative displacement error L^2 from 0.0284 to approximately 0.62.

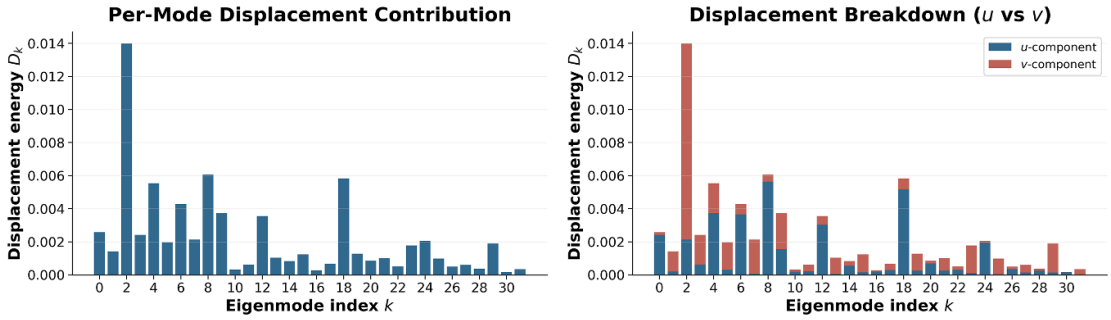

Figure 21. Per-mode displacement energy D_k (left) and u/v component breakdown (right). Mode $k = 2$ dominates through its v-component, consistent with the vertical traction direction.

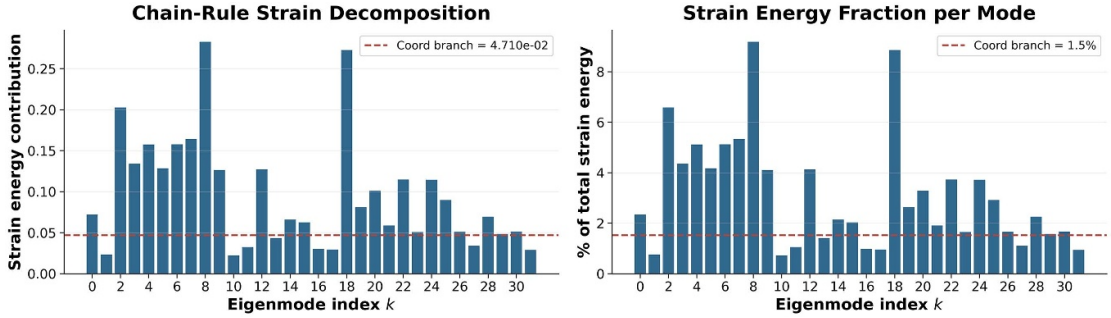

Figure 22. Absolute contribution per mode (left) and percentage of total strain energy (right). The coordinate branch contributes approximately 1.5% (dashed line).

The Jacobian sensitivity spectrum $S_k = \frac{1}{N}\sum_i \|\partial\hat{u}/\partial\phi_k\|^2$ (figure 20) confirms higher sensitivity in initial modes. The decomposition of displacement via the linearized marginal contribution $\Delta\hat{u}_k \approx (\partial\hat{u}/\partial\phi_k)\,\phi_k$ reveals that mode $k = 2$ dominates through the v-component, consistent with the applied vertical traction, while the surrounding modes accommodate the lateral Poisson contraction through the u-component (figure 21).

This directional selectivity demonstrates that the network routes vertical and horizontal displacements through distinct modal channels. The chain-rule strain decomposition (figure 22) shows that the modal branch accounts for approximately 98.5% of the total strain energy, with the coordinate branch contributing approximately 1.5% (dashed line in figure 22, right panel). A normalized side-by-side comparison of displacement and strain participation is shown in figure 23.

The participation spectrum is non-monotonic with respect to the eigenvalue index, reflecting the adaptive and nonlinear spectral weighting of the network, shaped by the eigenbasis of the operator, the applied load, and the topology, rather than a classical linear modal truncation.

Collectively, the participation spectrum is not a monotonically decaying function of the eigenvalue index. The network learns nonlinear combinations of the modal inputs, and the resulting importance distribution reflects which modes most effectively reduce the variational energy functional. This distinguishes the OA-PINN from a classical linear modal truncation, where participation would necessarily follow the eigenvalue ordering, and instead reveals an adaptive, nonlinear spectral weighting shaped jointly by the operator eigenbasis, the applied loading, and the topology.

6.3. Inverse identification robustness

To assess the practical viability of the operator-aware framework for inverse material identification, we conducted systematic studies that varied measurement noise, observation density, and modal embedding dimension. All experiments aim to recover orthotropic elastic parameters (E_1, E_2, G_{12}) from displacement observations using the VFM-based alternating optimization described in section 5.6.

(a) Influence of measurement noise. The displacement reconstruction error increases gradually with noise (figure 24). The error is relatively low because of the data-driven part matching the predictions with actual values.

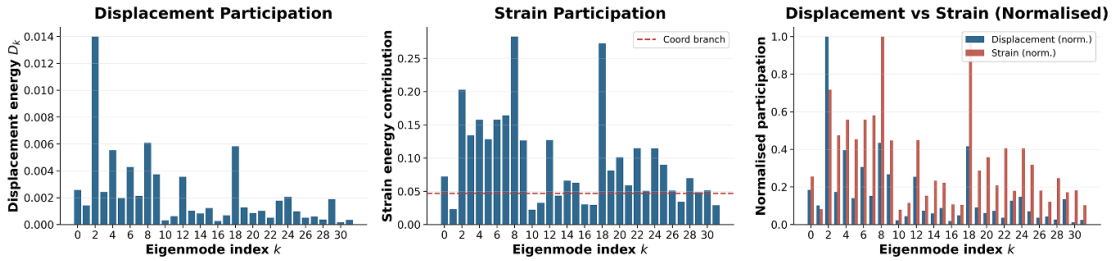

Figure 23. Displacement participation (left), strain participation (centre), and normalised comparison (right), showing correlated but non-identical mode importance rankings.

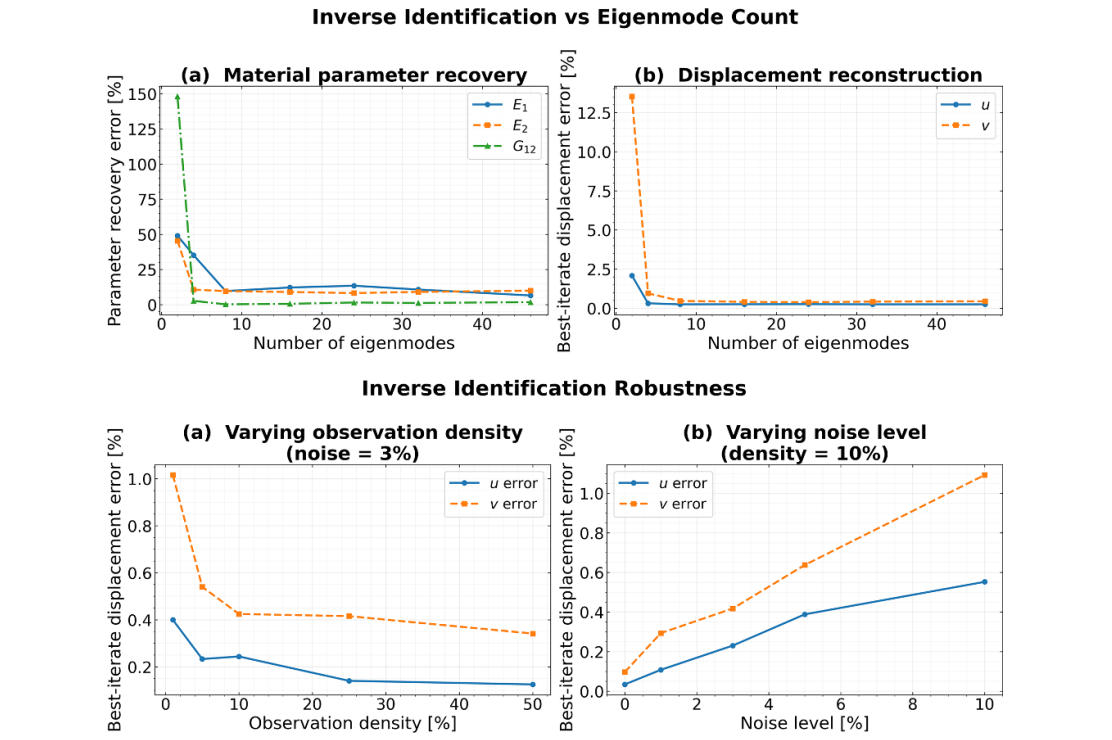

Figure 24. Robustness of inverse identification under varying number of used eigenmodes, observation density and measurement noise.

The v-component is empirically more sensitive to noise; we attribute this to the combined normal and shear loading, which couples the vertical response to G_{12} through pore-induced heterogeneity rather than through a direct G_{12}–ν coupling.

(b) Influence of observation density. Even at 1% observation density, the method produces meaningful fields; at 5% the u-error drops below 1%, with marginal improvement beyond 25% (figure 24, top). The eigenmode embedding constrains the solution space sufficiently for physically consistent reconstruction from sparse data (figure 25).

(c) Influence of modal embedding dimension. Parameter recovery is non-monotone in mode count (figure 24). A moderate number of modes balances representational capacity against identifiability, while displacement error improves monotonically.

(d) Influence of material-parameter initialization. Since joint identification of (E_1, E_2, G_{12}) is non-convex, we repeated Problem 4 from six initial guesses spanning the feasible range (near-truth, the default seed, the low- and high-bound corners, a cross-over seed reversing the physical ordering $E_1 < E_2$ with G_{12} near its lower bound, and a mixed point), holding observation density, noise, and eigenmode count fixed. As shown in figure 26, five of the six converge within three to four outer iterations to within 2%–17% for E_1, 10%–15% (E_2), and 2%–8% (G_{12}). Only the cross-over seed fails, with G_{12} collapsing to its lower bound. This failure mode can be avoided in practice by using physically plausible initial guesses that respect the expected stiffness ordering and avoid placing G_{12} at its lower bound.

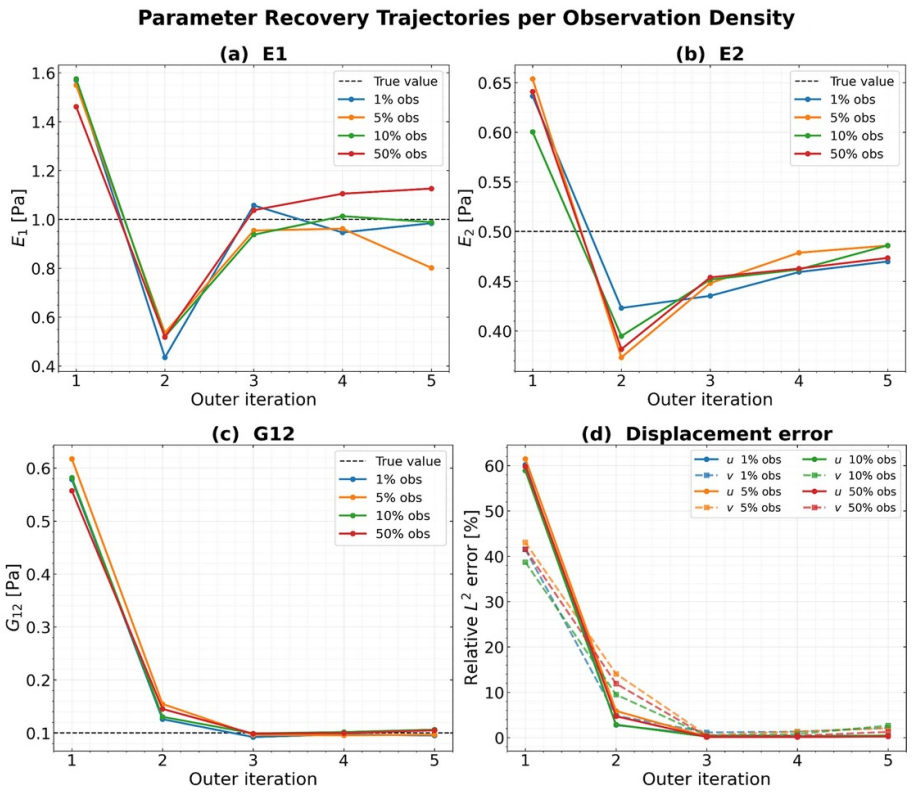

Figure 25. Problem 4; evolution of recovered moduli (E_1, E_2, G_{12}) and displacement error across outer optimization iterations, for five observation densities.

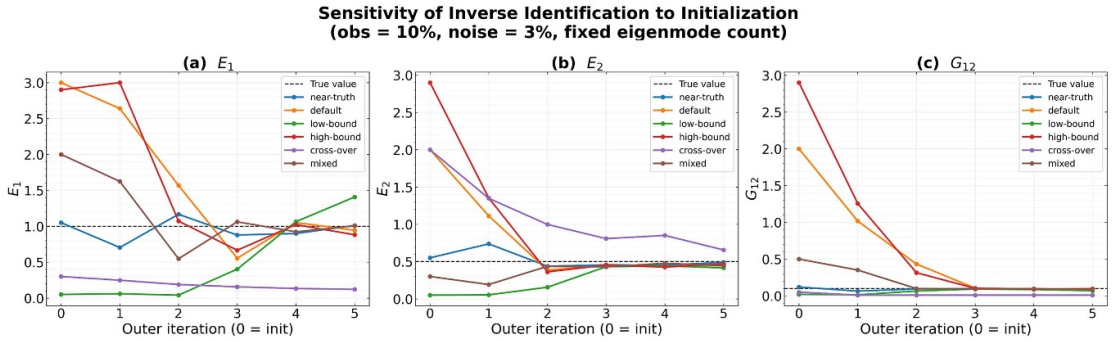

Figure 26. Sensitivity of inverse identification to the initial guess of (E_1, E_2, G_{12}). Panels (a)–(c): trajectory of each modulus across outer iterations for six initializations, with the dashed line marking ground truth.

Taken together, these experiments establish the operator-aware PINN as a promising framework for inverse material identification. The eigenmode embedding acts as a physics-informed prior that stabilizes the inverse problem under both sparse observations and moderate measurement noise, enabling accurate parameter estimates in practically relevant data-limited settings.

6.4. Limitations

The framework requires assembling and solving an eigenproblem of the stiffness operator. Although linear elastic eigenmodes remain effective for nonlinearity, path-dependent problems may require updates from the tangent stiffness operator.

Eigenmodes must be recomputed when the domain changes, making the approach most natural for repeated analyses on fixed geometries. OA-PINN is not intended to outperform a single optimized FEM solve for one forward load case; its computational appeal is instead in regimes where preprocessing and training costs can be amortized, such as repeated analysis on a fixed geometry, inverse identification, parameter studies, or uncertainty quantification. The current implementation evaluates the

energy functional at all mesh nodes per epoch (full-batch training). Extension to mini-batch or domain-decomposition strategies, combined with local eigenmode bases, is a natural avenue for improving scalability. Finally, all experiments reported here are two-dimensional; while the formulation extends directly to three dimensions (section 6.5), the computational and memory costs of the eigenmode tensor in 3D remain to be quantified.

6.5. Future work

We identify five directions for extending the proposed framework.

1. Three-dimensional extension: the OA-PINN formulation extends directly to 3D. The eigenmode tensor gains a third displacement component, and the stiffness matrix is assembled from tetrahedral elements. Separable architectures for 3D DEMs [54] suggest that memory and training costs can be significantly reduced.
2. Validation across operator classes: the operator-aware principle applies directly to self-adjoint operators with a computable and informative eigenbasis. This includes acoustic and Helmholtz problems, as well as coupled multiphysics systems, where operator-specific eigenmodes can be used within staggered OA-PINN solvers.
3. Nonlinear and history-dependent problems: linear elastic eigenmodes remain informative for moderate geometric nonlinearity. For strongly nonlinear or path-dependent problems, future work will consider incremental updates using tangent stiffness operators or hybrid approaches that combine linear eigenmodes with learned nonlinear modal subspaces.
4. Multi-fidelity and transfer learning: eigenmodes provide a natural bridge across mesh resolutions and material variations. Coarse-mesh eigenmodes can initialize fine-scale training, and previously learned modal representations can warm-start new configurations, enabling efficient multi-fidelity and transfer-learning strategies.
5. Extension to time-dependent and non-self-adjoint problems: time-dependent problems can be addressed either by a space–time formulation, with time as an additional input, or by retaining the fixed modal enrichment and advancing with an implicit time-stepping scheme. Non-self-adjoint operators such as the incompressible Navier–Stokes equations are more complex, as the advective term removes the real symmetric eigenbasis the method relies on. One approach is to enrich with the eigenmodes of the self-adjoint part of the operator for incompressible flow, the divergence-free Stokes eigenfunctions, which have served as a global trial basis for the Navier–Stokes equations [55, 56].

7. Conclusion

We proposed a general principle for PINNs on complex domains that augments spatial coordinates with eigenmodes of the problem's own discrete operator and their gradients. Accuracy improves as geometric complexity increases, with the modal branch contributing most of the strain energy. This converts the enrichment from geometry-aware to physics-aware, automatically incorporating boundary conditions, material response, and vector-valued coupling that geometry-only features cannot express. A hybrid chain rule propagates gradients through both the coordinate and eigenmode branches, keeping strain computation consistent with the operator-aware displacement representation; an ablation confirms both components are necessary. For elasticity, we demonstrated the principle across isotropic, near-incompressible, orthotropic, high-porosity, hyperelastic, and thermoelastic benchmarks.

The same eigenmode basis provides VFM virtual test functions for inverse material identification, avoiding $\mathbf{C} \to \mathbf{0}$ pathology, and recovering orthotropic moduli from 15% of nodes under 3% noise. The framework is most attractive for repeated analysis on fixed geometries, parameter studies, inverse identification, and uncertainty quantification, where the one-time pre-processing cost is amortized.

Funding

This work has been supported by European public funding through Xecs by the NL ministry EZK under the reference XECS221003 in the context of the e2LEAD project. Xecs is part of the EUREKA Clusters.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

Author contributions

Amir Ghorbani Ghezeljehmeidan 0009-0003-4597-2989
Data curation (lead), Methodology (lead), Validation (lead), Visualization (lead), Writing – original draft (equal), Writing – review & editing (equal)

Willem Dirk van Driel 0000-0001-8882-2508
Funding acquisition (lead), Project administration (equal), Resources (equal), Supervision (equal), Validation (equal), Writing – original draft (equal), Writing – review & editing (equal)

Justin Dauwels 0000-0002-4390-1568
Conceptualization (equal), Formal analysis (equal), Resources (equal), Supervision (equal), Writing – original draft (equal), Writing – review & editing (equal)

References

[1] Lindqwister W, Peloquin J, Dalton L E, Gall K and Veveakis M 2025 *Commun. Eng.* **4** 73
[2] Bardini R and Di Carlo S 2024 Computational methods for biofabrication in tissue engineering and regenerative medicine–a literature review
[3] Trebotich D, Settgast R R, Ligocki T, Tobin W, Miller G H, Molins S and Steefel C I 2024 *Front. High Perform. Comput.* **2** 1416727
[4] de Jong S D, Ghezeljehmeidan A G and van Driel W D 2025 *Microelectron. Reliab.* **172** 115797
[5] Wolcott M P 1990 *Mater. Sci. Eng.: A* **123** 282–3
[6] Haslach H 2002 *Appl. Mech. Rev.* **55** B62–B63
[7] Patnaik S, Jokar M, Ding W and Semperlotti F 2022 *npj Comput. Mater.* **8** 152
[8] Babuška I and Rheinboldt W C 1978 *Int. J. Numer. Methods Eng.* **12** 1597–615
[9] Amini D, Haghighat E and Juanes R 2023 *J. Comput. Phys.* **490** 112323
[10] Amini D, Haghighat E and Juanes R 2022 *J. Eng. Mech.* **148** 04022070
[11] Goswami S, Anitescu C, Chakraborty S and Rabczuk T 2020 *Theor. Appl. Fract. Mech.* **106** 102447
[12] Raissi M, Perdikaris P and Karniadakis G E 2019 *J. Comput. Phys.* **378** 686–707
[13] Karniadakis G E, Kevrekidis I G, Lu L, Perdikaris P, Wang S and Yang L 2022 Nat. Rev. Phys.
[14] Samaniego E, Anitescu C, Goswami S, Nguyen-Thanh V M, Guo H, Hamdia K, Zhuang X and Rabczuk T 2020 *Comput. Methods Appl. Mech. Eng.* **362** 112790
[15] Roy N 2025 *Phys. Scr.* **100** 086008
[16] Hu H, Qi L and Chao X 2024 Physics-informed neural networks (PINN) for computational solid mechanics *Numer. Framew. Appl.* **205** 112495
[17] Rahman M A *et al* 2024 Pretraining codomain attention neural operators for solving multiphysics PDEs *Advances in Neural Information Processing Systems* vol 37
[18] Wang S, Wang H and Perdikaris P 2021 *Comput. Methods Appl. Mech. Eng.* **384** 113938
[19] Tancik M, Srinivasan P P, Mildenhall B, Fridovich-Keil S, Raghavan N, Singhal U, Ramamoorthi R, Barron J T and Ng R 2020 Fourier features let networks learn high frequency functions in low dimensional domains *Advances in Neural Information Processing Systems*
[20] Cuomo S, Di Cola V S, Giampaolo F, Rozza G, Raissi M and Piccialli F 2022 *J. Sci. Comput.* **92** 88
[21] De Ryck T and Mishra S 2024 *Acta Numer.* **33** 633–713
[22] Nguyen-Thanh V M, Zhuang X and Rabczuk T 2020 *Eur. J. Mech.* A **80** 103874
[23] Weinan E and Yu B 2018 *Commun. Math. Stat.* **6** 1–12
[24] Abueidda D W, Koric S, Al-Rub R A, Parrott C M, James K A and Sobh N A 2022 *Eur. J. Mech.* A **95** 104639
[25] Fuhg J N and Bouklas N 2022 *J. Comput. Phys.* **451** 110839
[26] He J, Abueidda D, Abu Al-Rub R, Koric S and Jasiuk I 2023 *Int. J. Plast.* **162** 103531
[27] Wang Y, Sun J, Rabczuk T and Liu Y 2024 *Int. J. Numer. Methods Eng.* **125** e7585
[28] He J, Abueidda D, Koric S and Jasiuk I 2023 *Int. J. Numer. Methods Eng.* **124** 864–79
[29] Kharazmi E, Zhang Z and Karniadakis G E 2021 *Comput. Methods Appl. Mech. Eng.* **374** 113547
[30] Zhang N, Xu K, Yu Yin Z, Li K Q and Jin Y F 2025 *Comput. Methods Appl. Mech. Eng.* **433** 117474
[31] Xiong W, Long X, Bordas S P and Jiang C 2025 *Comput. Methods Appl. Mech. Eng.* **436** 117681
[32] Pantidis P, Eldababy H, Abueidda D and Mobasher M E 2024 *Comput. Methods Appl. Mech. Eng.* **425** 116940
[33] Khara B, Balu A, Joshi A, Sarkar S, Hegde C, Krishnamurthy A and Ganapathysubramanian B 2024 *Eng. Comput.* **40** 2761–83
[34] Moseley B, Markham A and Nissen-Meyer T 2023 *Adv. Comput. Math.* **49** 62
[35] Sahli Costabal F, Pezzuto S and Perdikaris P 2024 *Eng. Appl. Artif. Intell.* **127** 107324
[36] Li H, Miao Y, Khodaei Z S and Aliabadi M H 2025 *J. Mech. Phys. Solids* **203** 106222
[37] Nguyen T N K, Dairay T, Meunier R, Di Stasio J, Millet C and Mougeot M 2025 *Comput. Phys. Commun.* **316** 109757
[38] Wang J and Hao W 2025 *J. Comput. Phys.* **543** 114400
[39] Chen G, Liu X, Meng Q, Chen L, Liu C and Li Y 2024 *Natl. Sci. Open* **3** 20240001

[40] Du Y, Chalapathi N and Krishnapriyan A S 2024 Neural spectral methods: self-supervised learning in the spectral domain *12th Int. Conf. on Learning Representations, ICLR 2024*
[41] Fanaskov V S and Oseledets I V 2023 *Dokl. Math.* **108** S226–32
[42] Cao Q, Goswami S and Karniadakis G E 2024 *Nat. Mach. Intell.* **6** 631–40
[43] Lu L, Jin P, Pang G, Zhang Z and Karniadakis G E 2021 *Nat. Mach. Intell.* **3** 218–229
[44] Kovachki N, Li Z, Liu B, Azizzadenesheli K, Bhattacharya K, Stuart A and Anandkumar A 2023 *J. Mach. Learn. Res.* **24** 1–97
[45] Li Z, Kovachki N, Azizzadenesheli K, Liu B, Bhattacharya K, Stuart A and Anandkumar A 2021 Fourier neural operator for parametric partial differential equations *ICLR 2021–9th Int. Conf. on Learning Representations*
[46] Li Z, Huang D Z, Liu B and Anandkumar A 2023 *J. Mach. Learn. Res.* **24** 1–26
[47] Wang S, Wang H and Perdikaris P 2021 *Sci. Adv.* **7** eabi8605
[48] Li Z, Zheng H, Kovachki N, Jin D, Chen H, Liu B, Azizzadenesheli K and Anandkumar A 2024 *ACM/IMS J. Data Sci.* **1** 1–27
[49] Kaewnuratchadasorn C, Wang J and Kim C W 2024 Comput.-Aided Civ. Infrastruct. Eng. **39** 3435–51
[50] Goswami S, Yin M, Yu Y and Karniadakis G E 2022 *Comput. Methods Appl. Mech. Eng.* **391** 114587
[51] Hildebrand S and Klinge S 2024 Comparison of neural FEM and neural operator methods for applications in solid mechanics
[52] Herde M, Raonić B, Rohner T, Käppeli R, Molinaro R, de Bézenac E and Mishra S 2024 POSEIDON: efficient foundation models for PDEs *Advances in Neural Information Processing Systems* vol 37
[53] Rahaman N, Baratin A, Arpit D, Draxlcr F, Lin M, Hamprecht F A, Bengio Y and Courville A 2019 On the spectral bias of neural networks *36th Int. Conf. on Machine Learning, ICML 2019*
[54] Es'kin V A, Davydov D V, Gur'eva J V, Malkhanov A O and Smorkalov M E 2025 *Comput. Math. Math. Phys.* **65** 2260–75
[55] Batcho P F and Karniadakis G E 1994 *J. Comput. Phys.* **115** 121–46
[56] Baron A 2024 *Phys. Fluids* **36** 067124