Engineering Analysis with Boundary Elements 192 (2026) 106997

Contents lists available at ScienceDirect

Engineering Analysis with Boundary Elements

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

Preconditioned Variational Physics-informed Neural Operator (P-VINO) for membranes and Mindlin plates

L.M. Boccia [a], R. Vescovini [a,*], N. Fantuzzi [b]

[a] Dipartimento di Scienze e Tecnologie Aerospaziali, Politecnico di Milano, Via La Masa 34, 20156 Milano, Italy
[b] Dipartimento di Ingegneria Civile, Chimica, Ambientale e dei Materiali - Università di Bologna, Viale del Risorgimento 2, 40136 Bologna, Italy

ARTICLE INFO

Keywords:
Physics-informed deep learning
Neural operators
Preconditioning
Mindlin plates
Structural analysis

ABSTRACT

In this work, the Preconditioned Variational Physics-informed Neural Operator (P-VINO) is proposed. This framework exploits the potential of Physics-informed Neural Operators, which are capable of learning the mapping from parameter-dependent inputs to the solution. In addition, the Physics-informed mechanism eliminates the need for a large training dataset. The framework is applied to linear membrane and Mindlin plate problems and relies on a variational formulation. Among the distinctive features of P-VINO, two are particularly relevant for overcoming key limitations in the data-free training of the Fourier Neural Operator. Firstly, a preconditioning technique is implemented to tackle ill-conditioned issues. Secondly, a FEM-based conforming strategy is implemented to preserve the FEM flexibility in defining loading and boundary conditions along any edge of the structure. The results are presented for both membranes and Mindlin plates, demonstrating that P-VINO improves convergence, energetic consistency with finite element solutions, and accurately captures the structural response even for ill-conditioned problems, suggesting its potential for fast parametric structural analyses.

1. Introduction

Driven by a combination of algorithmic advances and increased availability of computational resources [1,2], deep neural networks have gained tremendous popularity in the last decade. Their application spans across a wide range of disciplines, structural mechanics being one of the most extensively investigated.

Early applications of deep learning in structural mechanics rely on classical neural networks, such as multilayer perceptrons (MLPs), employed as black-box surrogates mapping a finite set of input parameters to output quantities of interest. These architectures are theoretically grounded in the universal approximation theorem for functions [3] and different applications are found in the field of structures. For instance, Bisagni and Lanzi [4] used Artificial Neural Networks (ANNs) to develop a surrogate-based post-buckling optimization of stiffened panels, with training conducted via nonlinear FE simulations. Petrolo and Carrera [5] used Artificial Neural Networks (ANNs) to define the kinematic models to establish appropriate kinematic models in 2D structural models. Other applications include zooming techniques [6], modeling of nonlinear constitutive laws [7] and stress analysis [8].

The applicability of ANNs is inherently constrained by the finite-dimensional nature of their input–output spaces: a network trained on a given discretization cannot generalize to a different one. This limitation can be overcome by techniques based on neural operators, a class of architectures designed to learn mappings between infinite-dimensional function spaces, and recognized as universal approximators of operators [9]. An example is found in DeepONet [10], which relies on the operator extension of the universal approximation theorem, and the Fourier Neural Operator (FNO) [11], which parametrizes the integral kernel in the frequency domain. Being discretization-invariant, neural operators are well-suited for the solution of PDEs and, once trained, can infer solutions that are orders of magnitude faster than traditional numerical solvers [12]. For this reason, they have been the subject of recent investigations [13–15], including applications to fluid–structure interaction [16] and brittle fracture [17].

One critical aspect for both classical networks and neural operators is their reliance on large collections of input–output pairs for training, often too expensive or even inaccessible in structural mechanics applications. By directly introducing physics-related information into the training process of neural networks, the above mentioned restriction can be mitigated. Several strategies have been proposed in this context [12]. A first example is given by physics-guided neural networks (PgNNs) that embed physical knowledge into the architecture or training procedure, using curated input–output datasets that comply

* Corresponding author.
E-mail address: riccardo.vescovini@polimi.it (R. Vescovini).

https://doi.org/10.1016/j.enganabound.2026.106997
Received 11 May 2026; Received in revised form 12 June 2026; Accepted 21 August 2026
Available online 1 September 2026

with physical principles [18]. Another approach refers to the physics-encoded neural networks (PeNNs). They enforce physical constraints directly into the network architecture, extending the learning capability from instance learning to continuous learning [19,20]. A third well-established approach refers to the so-called physics-informed neural networks (PINNs) [21], whose main idea consists of incorporating the governing equations as additional terms in the loss function and reducing the dependence on labeled data. Within the physics-informed paradigm, two approaches to incorporating the PDE residual into the loss can be distinguished: the strong form [22–25], in which the residual is enforced pointwise, and the weak or variational form [26–28] in which it is either integrated against test functions, as in variational PINNs (hp-VPINNs) [26], or the solution is sought as the minimizer of an energy functional, as in the Deep Ritz method [29]. Both formulations have proven particularly effective in Kirchhoff plate bending problems, where the C^1 continuity requirement poses significant difficulties in traditional mesh-based methods, while mesh-free neural network approximations satisfy this requirement naturally and avoid the cost of domain remeshing [30,31]. Domain decomposition approaches have been proposed too [32,33], and applied as a useful mean for analyzing assemblies of plate elements [23].

A recent advancement of the physics-informed approach regards the so-called neural operators, which have found application in the solution of parametric PDEs. Among these, the Physics-Informed Neural Operator (PINO) and physics-informed DeepONets [15,34,35] adopt the strong form, enforcing the PDE residual pointwise within the operator learning framework, while the Variational Neural Operator (VINO) [36] adopts the variational form, seeking the solution operator as the minimizer of an energy functional. Since this energy is integrated analytically, VINO eliminates the difficulties associated with the high-order differentiation and numerical integration of the strong-form residual. Both PINO and VINO frameworks adopt the FNO as backbone architecture, which is generally more scalable than DeepONet-based operators [15], but rely on the Fast Fourier Transform and is therefore restricted to data sampled on a Cartesian grid, which limits their applicability to non-trivial geometries. In VINO, owing to its variational formulation, this limitation is partially overcome through an immersed discretization strategy. In DeepONet-based architectures geometry-aware formulations have been developed to improve generalization across different domain geometries [37].

Inspired by the VINO framework, this work introduces the Preconditioned Variational Physics-Informed Neural Operator (P-VINO). Like VINO, P-VINO learns the solution operator as the minimizer of an energy functional, in contrast to strong-form approaches such as PINO and physics-informed DeepONet [35], which enforce the PDE residual pointwise. Going beyond VINO, P-VINO introduces two key contributions aimed at improving its applicability to the analysis of membranes and Mindlin plates. First, P-VINO employs a preconditioning strategy aimed at reshaping the energetic landscape, thereby enabling the use of simpler Fourier Neural Operator (FNO) architectures, accelerating convergence, and improving the treatment of ill-conditioned problems such as those arising for Mindlin plates under bending-dominated loading. Second, it incorporates a FEM-based conforming mesh discretization strategy, which interpolates the predicted displacement at the grid nodes onto a FEM mesh, and is shown to accelerate convergence in problems involving non-trivial geometries. This conforming strategy enables the introduction of a data-free stopping criterion based on the Clapeyron theorem for linear elastic systems. Due to the implementation challenges associated with this conforming FEM-based discretization, alternative strategies for the evaluation of the internal strain energy have also been developed and are presented within the P-VINO framework. Together, these two contributions are aimed at improving convergence speed, accuracy and the computational efficiency of the variational operator-learning approach for membrane and plate problems.

The remainder of this paper is organized as follows. The variational formulation of the linear elastic problem is briefly recalled, together with the VINO framework that is at the core of P-VINO. The newly proposed computational framework is then presented and discussed in detail, together with a critical overview of its potential applicability to parametric studies and multi-query settings and an identification of the open challenges that motivate further development. Finally, the performance of P-VINO is assessed through a series of numerical experiments.

2. Variational formulation

The P-VINO framework is developed for the analysis of membranes and plates. For this reason, the relevant equations are derived as a basis to subsequently develop the computational framework.

We consider a body occupying an open domain $\Omega \subset \mathbb{R}^d$, whose boundary $\partial\Omega$ is decomposed into two disjoint parts such that essential boundary conditions are prescribed on Γ_u through the assigned displacement field $\bar{\mathbf{u}}$, whereas natural boundary conditions are imposed on Γ_t in terms of traction field $\bar{\mathbf{t}}$. In addition, the body may be subjected to a body-force density $\mathbf{b}$ acting throughout Ω. In the present work, two cases are considered: membranes and Mindlin plate models. The first one can be easily recovered from the second by neglecting any contribution from bending, transverse shear and membrane/bending coupling.

By formulating the problem from an energy perspective, the total potential energy is written as:

$$\Pi(\mathbf{u}) = U(\mathbf{u}) - W(\mathbf{u}), \tag{1}$$

where $\mathbf{u}$ is the vector that collects the generalized displacements of the kinematic model, $U(\mathbf{u})$ denotes the internal strain energy and $W(\mathbf{u})$ the work of the external forces, whose expressions read:

$$U(\mathbf{u}) = \frac{1}{2}\int_\Omega \boldsymbol{\sigma}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{u})\,\mathrm{d}\Omega, \tag{2}$$

$$W(\mathbf{u}) = \int_\Omega \mathbf{b}\cdot\mathbf{u}\,\mathrm{d}\Omega + \int_{\Gamma_t} \bar{\mathbf{t}}\cdot\mathbf{u}\,\mathrm{d}\Gamma. \tag{3}$$

In the expressions reported above, $\boldsymbol{\sigma} \sim \{N_{\alpha\beta}\}$ in the case of membranes, while, for Mindlin plates, $\boldsymbol{\sigma} \sim \{N_{\alpha\beta}\ M_{\alpha\beta}\ Q_{3\alpha}\}^T$, having defined $\{N_{\alpha\beta}\}$, $\{M_{\alpha\beta}\}$ and $Q_{3\alpha}$ as the membrane, moment and shear resultants within the context of Budiansky-like notation [38]. Accordingly, the strains are interpreted as $\boldsymbol{\varepsilon} \sim \{\epsilon_{\alpha\beta}\}$ for membranes, and as $\boldsymbol{\varepsilon} \sim \{\epsilon_{\alpha\beta}\ k_{\alpha\beta}\ \gamma_{3\alpha}\}^T$ for plates, where $\epsilon_{\alpha\beta}$ are the membrane strains, $k_{\alpha\beta}$ the bending curvatures and $\gamma_{3\alpha}$ the transverse shear deformations.

In the present work, the domain is discretized following a finite element-like procedure based on four-node bilinear elements (Q4). Within each element, the displacement field is approximated by means of suitable shape functions as:

$$\mathbf{u}_h^{(e)}(\mathbf{x}) = \mathbf{N}^{(e)}(\mathbf{x})\,\mathbf{d}^{(e)}, \tag{4}$$

where $\mathbf{N}^{(e)}$ collects the elemental shape functions and $\mathbf{d}^{(e)}$ denotes the vector of nodal degrees of freedom; the superscript (e) refers to the generic element. Under this approximation, the total potential energy functional is rewritten in discrete form as:

$$\Pi_h(\mathbf{d}) = U_h(\mathbf{d}) - W_h(\mathbf{d}), \tag{5}$$

with Π_h, U_h and W_h denoting the discrete counterpart of the total potential energy, internal energy and external work, respectively. The subscript h indicates that the corresponding quantities are evaluated over the discretized domain. The internal energy is expressed as the sum of elemental contributions,

$$U_h(\mathbf{d}) = \sum_{e=1}^{n_{\mathrm{el}}} U^{(e)}(\mathbf{d}^{(e)}), \tag{6}$$

where n_{el} is the total number of elements, $U^{(e)}$ denotes the internal energy contribution of the eth element, and $\mathbf{d}^{(e)}$ is the vector of nodal

degrees of freedom associated with that element. The external work is written directly in global discrete form as

$$W_h(\mathbf{d}) = \mathbf{d}^T \mathbf{f}, \tag{7}$$

where $\mathbf{f}$ is the global vector of nodal forces and $\mathbf{d}$ is the global vector of nodal degrees of freedom. Irrespective of whether the structure is a membrane or a plate, the vectors collecting the generalized strains and force resultants are written in standard finite element form as

$$\boldsymbol{\varepsilon} = \mathbf{B}\mathbf{d}^{(e)}, \quad \boldsymbol{\sigma} = \mathbf{C}\,\boldsymbol{\varepsilon} \tag{8}$$

where $\mathbf{B}$ is the strain–displacement matrix associated with the element's kinematics and $\mathbf{C}$ denotes the constitutive law of the kinematic model at hand. The corresponding elemental internal energy is expressed as

$$U^{(e)} = \frac{1}{2}\,\mathbf{d}^{(e)T}\mathbf{K}^{(e)}\mathbf{d}^{(e)}, \tag{9}$$

where $\mathbf{K}^{(e)}$ is the stiffness matrix of the element, given by

$$\mathbf{K}^{(e)} = \int_{\Omega^{(e)}} \mathbf{B}^T \mathbf{C}\,\mathbf{B}\,\mathrm{d}\Omega, \tag{10}$$

with $\Omega^{(e)}$ denoting the domain of the eth element. For Mindlin plates, the stiffness matrix can be conveniently decomposed in four contributions as:

$$\mathbf{K}^{(e)} = \mathbf{K}_m^{(e)} + \mathbf{K}_{mb}^{(e)} + \mathbf{K}_b^{(e)} + \mathbf{K}_s^{(e)}, \tag{11}$$

with $\mathbf{K}_m^{(e)}$, $\mathbf{K}_{mb}^{(e)}$, $\mathbf{K}_b^{(e)}$, and $\mathbf{K}_s^{(e)}$ denoting the membrane, membrane–bending coupling, bending, and transverse shear contributions, respectively. To prevent locking, selective integration is adopted. The shear factor is 5/6 for all the test cases presented next. For laminated plates, use is made of the lamination parameters, which provide a compact and convenient encoding of the laminate stacking sequence information and are particularly advantageous as inputs to the neural operator, given the linear dependence of the constitutive law on these quantities, which simplifies the learning process [39].

3. Overview of VINO

Starting from the variational formulation introduced above, the VINO framework [36] defines a physics-informed training strategy in which the neural operator is trained by minimizing the energy functional of the problem, which in the present case takes the form of the discrete total potential energy introduced in Eq. (5). To this end, VINO adopts the Fourier Neural Operator (FNO) as its backbone architecture, a class of models designed to approximate mappings between infinite-dimensional function spaces and recognized as universal approximators of nonlinear continuous operators [9]. More specifically, given the exact solution operator

$$G : \mathcal{A} \to \mathcal{U}, \qquad a \mapsto u, \tag{12}$$

where $\mathcal{A}$ denotes the space of admissible input fields and $\mathcal{U}$ the corresponding solution space, the neural operator seeks a parametric approximation G_θ such that

$$G_\theta(a) \approx G(a), \qquad \forall a \in \mathcal{A}, \tag{13}$$

To provide a clear understanding of the specific neural-operator architecture adopted in this work, the FNO is illustrated in Fig. 1.

The input field $a(x)$, sampled on a uniform equispaced grid, is first lifted to a higher-dimensional latent representation through a pointwise operator P, thereby increasing the representational capacity of the model [40]. The resulting latent field $v_0(x)$ is then processed through a sequence of l_i Fourier layers, indexed by $i = 0, \ldots, L-1$. In each layer, a local pointwise linear transformation W_i is combined with a global spectral contribution obtained by mapping the latent field to Fourier space via $\mathcal{F}$, filtering the retained modes through learnable complex-valued weights R_i, and mapping back via $\mathcal{F}^{-1}$ [11]. The two contributions are summed and passed through a nonlinear activation function $\sigma(\cdot)$, introducing nonlinearity through the repeated composition of linear and spectral operators. After L layers, a final pointwise projection operator Q maps the latent representation to the solution space $u(x)$.

In VINO, the FNO backbone is combined with a physics-informed loss function defined through the energy formulation of the governing problem. The nodal values returned by the FNO on its uniform equispaced grid are naturally used to define a partition of the domain into rectangular elements, over which the displacement field is approximated by standard bilinear shape functions. This allows the elemental strain energy to be expressed in closed form as

$$U^{(e)} = U^{(e)}\left(a,\, b,\, \mathbf{C},\, \mathbf{d}^{(e)}\right), \tag{14}$$

where a and b denote the element half-dimensions. The expression depends solely on the element geometry, the material properties, and the nodal displacements, requiring neither numerical differentiation nor numerical quadrature during training. The significant technical challenge associated with the efficient evaluation of neural-operator derivatives [34] is thus avoided, improving both computational efficiency and numerical robustness, and enabling training without labeled data [36].

Due to its reliance on a uniform Cartesian grid [11,12], the capability of FNO to handle non-trivial geometries is limited. In data-driven settings, this limitation has motivated the development of geometry-aware FNO, such as Geo-FNO [41]. However, to the best of the authors' knowledge, no such methodology exists for the variational physics-informed setting. Within VINO [36], this issue can instead be partially addressed by adopting an immersed Cartesian discretization, in which the physical domain is embedded into the FNO uniform grid and elements are classified as fully interior or partially exterior, with the energy contribution adjusted accordingly.

The main steps of the VINO training procedure are summarized in the block diagram shown in Fig. 2.

The process starts with the definition of the parameter space on which the neural operator is trained. Then, N samples $\{a^i\}_{i=1}^N$ are selected to provide an adequate representation of this space. For each sample, a multichannel input tensor is assembled and passed through the neural operator. Each channel represents a spatial function of x and y, sampled on the uniform equispaced grid, and encodes the relevant parametric information. The operator then returns the displacement field u_θ on the background Cartesian grid. These predicted nodal values are subsequently used to evaluate the internal energy through the closed-form expression of Eq. (14) and to compute the external work, so that the loss functional can be assembled for each sample. The operator parameters are subsequently updated through a gradient-based optimization procedure. Once training is completed, obtaining the solution for a new sample in the parameter space only requires assembling the corresponding input tensor and performing inference.

4. Moving beyond VINO: the P-VINO framework

While the VINO framework described above provides an effective and computationally efficient strategy for a broad class of problems, its direct application to structural plate analysis reveals two specific limitations that motivate the new advanced features proposed in this work.

The first limitation concerns the treatment of boundary conditions in problems where essential boundary conditions are prescribed on curved boundaries or on segments not aligned with the Cartesian grid. Since the immersed Cartesian discretization relies on rectangular elements whose nodes do not, in general, coincide with the physical boundary, the prescribed displacements cannot be imposed in a geometrically exact manner at the boundary-intersected elements. As a consequence, an approximation error is introduced in the enforcement of the kinematic constraints. More generally, since the immersed

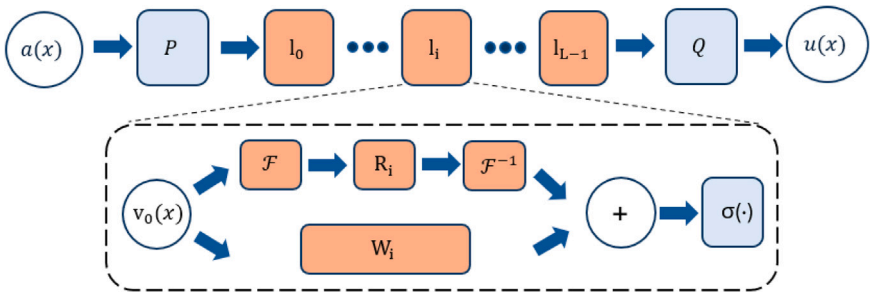


Fig. 1. FNO architecture.

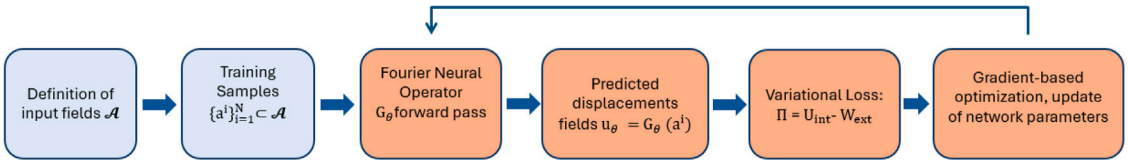


Fig. 2. VINO workflow.

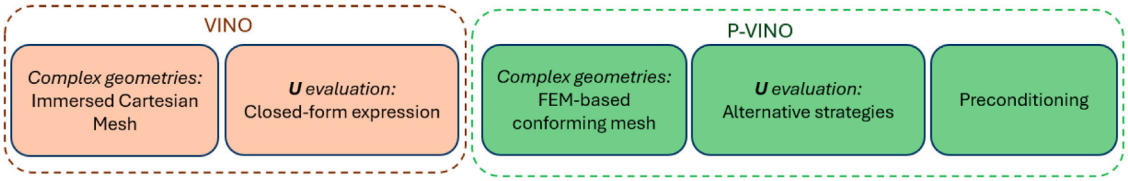


Fig. 3. VINO and P-VINO components.

Cartesian approach is a non-conforming discretization technique, the Clapeyron theorem for linear elastic systems,

$$U_h = \frac{1}{2} W_h, \tag{15}$$

which is exactly satisfied by the conforming FEM solution of $\mathbf{Ku} = \mathbf{f}$, is not guaranteed in general, and the recovered solution may exhibit a residual imbalance between internal strain energy and external work even upon convergence.

The second limitation concerns the behavior of the framework when applied to ill-conditioned variational problems. The Mindlin plate model, particularly in bending-dominated regimes, gives rise to an energy landscape whose poor conditioning makes gradient-based optimization unreliable. In practice, as will be shown next, the loss function stagnates at a plateau well above the theoretical minimum during training, preventing the neural operator from recovering an accurate solution.

All of the issues above are addressed by the P-VINO framework, whose key components are schematically illustrated in Fig. 3.

P-VINO first adopts a FEM-based conforming mesh discretization strategy, presented in Section 4.1, which replaces the immersed Cartesian background grid with a mesh that conforms to the physical domain. In doing so, the geometrically exact enforcement of boundary conditions is guaranteed and the discrete equilibrium balance restored. The conforming discretization introduces new challenges in the evaluation of the internal strain energy, so alternative evaluation strategies developed to address this issue are discussed in Section 4.2. Finally, a preconditioning strategy is introduced in Section 4.3 to reshape the energetic landscape of the variational problem, enabling convergence of the neural operator in ill-conditioned settings where the unmodified VINO framework fails to progress.

4.1. FEM-based conforming discretization

In the proposed discretization strategy, the physical domain is meshed following the same procedures commonly adopted in the finite element method, thus allowing the framework to leverage geometrical treatments that are well established and supported by decades of development. The Q4 elements are still employed, but the resulting mesh now consists of non-uniform, non-rectangular elements that conform to the physical boundary rather than rectangular elements of identical shape as in the original VINO framework [36].

Once the conforming mesh has been generated, it is superposed on the equispaced Cartesian grid required by the FNO backbone. Since the mesh nodes do not, in general, coincide with the grid nodes, the displacement field predicted by the neural operator on the regular grid must be transferred to the mesh nodes before the energy evaluation. This is achieved through a bilinear interpolation procedure: for each mesh node, the four surrounding Cartesian grid nodes are identified, and the displacement at the mesh node is recovered from the nodal displacements of the enclosing grid cell using the standard Q4 shape functions evaluated at the local coordinates of the mesh node within that cell.

A schematic comparison between the two discretization strategies is provided in Fig. 4 for a domain with a curved boundary.

In the left of the figure, the immersed Cartesian strategy is illustrated: white dots denote the background grid nodes, cells with black boundaries correspond to elements whose four nodes lie entirely within the physical domain, and those with red boundaries denote elements only partially contained within it. In the right part of the figure, it is shown the conforming strategy proposed in this work for the same domain: gray elements represent the conforming mesh, while white dots identify the underlying Cartesian grid nodes used to interpolate the displacements at the mesh nodes.

As seen, the nodes of the boundary-intersected elements do not generally lie on the physical boundary of the domain, clearly highlighting the problem related to the imposition of boundary conditions on curved edges for the immersed Cartesian mesh. The conforming discretization directly addresses this limitation: since the mesh nodes now lie on the physical boundary by construction, prescribed displacements are

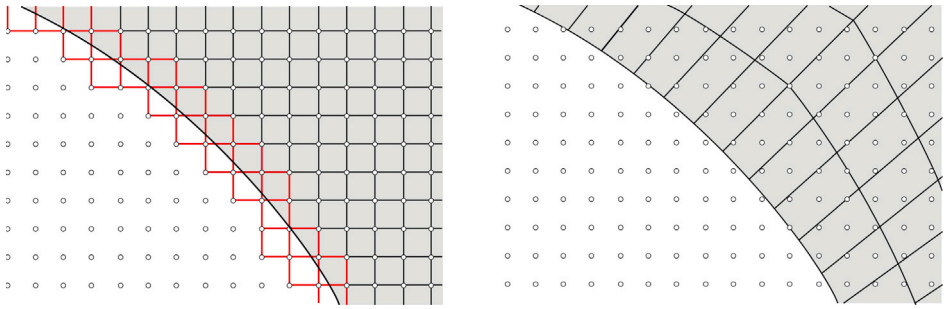


Fig. 4. Discretization strategies: (a) immersed Cartesian mesh; (b) FEM-based conforming mesh.

imposed directly at the mesh nodes, following the same procedure adopted in standard finite element analysis, without any geometrical approximation.

The use of the same conforming mesh, combined with the use of the same kinematic model and integration rules as a reference FEM solver, establishes a strong and formally precise connection between the neural operator solution and the classical finite element solution. When these conditions are met, the solution recovered by the neural operator upon convergence coincides with the FEM solution of the discrete equilibrium system. This equivalence follows directly from the uniqueness of the minimizer of the discrete total potential energy. Indeed, both methods seek the same stationary point of the discrete total potential energy Π_h over the same discrete space, and are therefore guaranteed to share the same unique solution upon convergence. This equivalence also provides a practical and data-free convergence criterion. At the FEM equilibrium solution, the Clapeyron theorem for linear elastic systems, Eq. (15), states that the internal strain energy equals half the external work. During training, the residual $|U_h - \frac{1}{2}W_h|$ can therefore be monitored as a convergence indicator: when this quantity approaches zero, the predicted displacement field is consistent with the discrete equilibrium equations and the neural operator solution has effectively converged to the FEM solution, without requiring any labeled reference solution for comparison.

4.2. Internal energy evaluation

The introduction of a FEM-based conforming mesh removes the geometric uniformity that allowed the original VINO framework to express the elemental energy contribution as a compact closed-form function of the nodal displacements, thereby avoiding any numerical differentiation or quadrature during training. To preserve this capability, the framework is extended to generic Q4 elements of arbitrary shape by adopting an isoparametric mapping following standard finite element practice, so that the internal energy can be evaluated consistently for all element geometries. Two strategies have been developed to this end, differing in how this goal is pursued and in the computational trade-offs they entail.

The first strategy extends the symbolic approach adopted in VINO to generic Q4 elements, deriving a closed-form expression for the internal energy through analytical manipulation of the isoparametric formulation. The elemental energy contribution takes the form

$$U^{(e)} = U^{(e)}\left(\mathbf{x}^{(e)}, \mathbf{C}, \mathbf{d}^{(e)}\right), \tag{16}$$

where $\mathbf{x}^{(e)}$ collects the nodal coordinates used to compute the jacobian. In practice, the resulting expression is extremely long and must be decomposed into smaller algebraic blocks to remain within the capacity of standard compilers. Still a significant overhead is incurred at the start of training due to the Just-In-Time (JIT) compilation phase required to build the computational graph [42], whose cost scales with the length of the expression. To mitigate this effect, the terms depending on the element geometry only are precomputed once offline before training begins, reducing both compilation time and per-epoch cost.

The second strategy avoids symbolic derivation entirely. The elemental stiffness matrix is computed once per element by standard numerical quadrature and stored prior to training. The internal energy contribution is then evaluated during training through the quadratic form

$$U^{(e)} = \frac{1}{2}\,\mathbf{d}^{(e)^T}\mathbf{K}^{(e)}\mathbf{d}^{(e)}, \tag{17}$$

which involves only a matrix–vector product and requires no further integration or differentiation during the optimization process.

Neither strategy is universally superior, and the choice depends on the specific problem size, mesh resolution, and training set cardinality. The symbolic approach avoids storing the stiffness matrices but incurs higher per-epoch computational cost. Furthermore, it requires re-derivation and optimization upon any modification to the constitutive model. On the contrary, the stiffness-based approach is simpler and more flexible, but its memory requirements grow with the problem size and training set cardinality.

It is worth noting that, unlike the original VINO framework where uniform rectangular elements allow the elemental energy to be integrated analytically [36], the isoparametric mapping introduced here makes numerical quadrature unavoidable in both strategies. This has the following implication: on meshes composed entirely of rectangular elements fully contained within the physical domain, analytical integration yields faster convergence than numerical quadrature at equal resolution. However, when non-trivial geometries require the immersed Cartesian strategy, the non-conforming representation of the boundary introduces an error that dominates over the choice of integration scheme. In such cases, the FEM-based conforming discretization, despite relying on numerical quadrature, achieves faster convergence and higher accuracy. Indeed, the benefit of the geometrically exact domain representation outweighs that of analytical integration. Numerical evidence supporting these claims is provided in the section of results, i.e. Section 6.2 and Section 6.1.

4.3. Preconditioning strategy

Numerical experiments, presented next, show that the VINO framework struggles to accurately capture the bending and shear energy contributions, with the loss function stagnating during training at a plateau well above the theoretical minimum. To interpret this behavior, it is useful to examine the structure of the loss function more closely.

The discrete total potential energy introduced in Eq. (5) is a quadratic function of the nodal displacement vector $\mathbf{d}$, whose Hessian, with respect to $\mathbf{d}$, is

$$\frac{\partial^2 \Pi_h}{\partial \mathbf{d}^2} = \mathbf{K}, \tag{18}$$

where $\mathbf{K}$ is the global stiffness matrix assembled from the elemental contributions introduced in Eq. (11). Since the Hessian of the loss

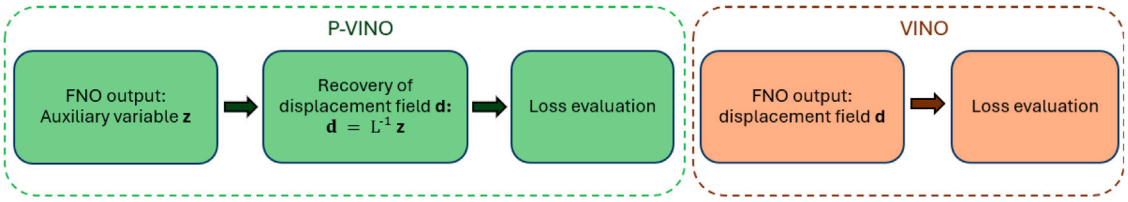


Fig. 5. Comparison between the VINO and P-VINO inference pipelines.

function coincides with $\mathbf{K}$, the conditioning of the stiffness matrix directly governs the curvature of the energy landscape and strongly influences the convergence of first-order optimization methods such as Adam [43,44]. This observation is consistent with recent analyses in the physics-informed neural network literature, where the conditioning number of the underlying differential operator has been identified as a key factor in training difficulties [45].

To identify the dominant source of ill-conditioning, the conditioning numbers of the individual stiffness contributions in Eq. (11), namely $\mathbf{K}_m$, $\mathbf{K}_{mb}$, $\mathbf{K}_b$, and $\mathbf{K}_s$, are examined separately. This analysis identifies the transverse shear contribution $\mathbf{K}_s$ as by far the most severely ill-conditioned among the four, an effect that is found to be associated with Selective Reduced Integration.

A preconditioning strategy is therefore introduced to reshape the energy landscape seen by the optimizer. After enforcing the essential boundary conditions and removing the constrained degrees of freedom, the reduced stiffness matrix is symmetric positive definite, which makes a Cholesky-based approach particularly appealing. Related matrix-based preconditioning ideas have been explored in the context of physics-informed neural networks, where they have been shown to improve training stability and convergence [46]; however, to the best of the authors' knowledge, this type of strategy has not yet been applied within variational neural operator frameworks.

In the present work, preconditioning is introduced through a change of variables based on the Cholesky factorization of the reduced stiffness matrix. A mild diagonal regularization is first applied to improve numerical robustness,

$$\tilde{\mathbf{K}} = \mathbf{K} + \alpha\,\mathrm{diag}(\mathbf{K}), \tag{19}$$

where $\alpha \ll 1$ is a small positive scalar. The Cholesky factorization of the regularized matrix is then computed once before training begins,

$$\tilde{\mathbf{K}} = \mathbf{L}\mathbf{L}^T, \tag{20}$$

As illustrated in Fig. 5, rather than predicting the physical displacement field $\mathbf{d}$ directly from the FNO output, an intermediate step is introduced.

In particular, the neural operator is trained to approximate an auxiliary variable $\mathbf{z}$, from which the displacement field is recovered through

$$\mathbf{d} = \mathbf{L}^{-T}\mathbf{z}, \tag{21}$$

after which the loss functional $\Pi_h(\mathbf{d})$ is evaluated in its original form given in Eq. (5). Following this procedure the curvature of Π_h with respect to $\mathbf{z}$ is governed by the Hessian

$$\frac{\partial^2 \Pi_h}{\partial \mathbf{z}^2} = \mathbf{L}^{-1}\mathbf{K}\mathbf{L}^{-T}, \tag{22}$$

which is better conditioned than $\mathbf{K}$, so that gradient-based optimization proceeds more uniformly across all directions. As demonstrated in the section on results, this enables convergence in cases where the unmodified framework fails to progress, accelerates training, and allows the use of simpler neural operator architectures with fewer trainable parameters.

5. P-VINO for parametric studies

The P-VINO framework just introduced is best suited to be deployed in parametric studies and multi-query settings, where the same governing problem must be solved repeatedly as one or more parameters vary. In such contexts, the ability of the neural operator to learn the solution operator mapping the parameter space to the solution space can be fully exploited. In principle, once training is completed, the inference cost for a new parameter instance is significantly lower than that of a full finite element solve [36], so the computational burden is concentrated in a single offline training phase, after which the trained operator can be queried at negligible cost for any new instance within the parameter space.

However, realizing this potential involves several challenges that emerge in the training procedure. The latter closely mirrors that of VINO, summarized in Fig. 2, with the key difference that the neural operator now predicts an auxiliary field from which the physical displacement is recovered via the Cholesky factor $\mathbf{L}$. In particular, two critical issues are here discussed: the construction of a training set that adequately covers the parameter space, and the definition of an effective preconditioning strategy when the stiffness matrix varies across training samples.

5.1. Training set construction and sample selection

Training a neural operator requires careful selection of the training samples, as this step is pivotal for ensuring sufficient coverage of the parameter space. Inadequate coverage risks producing an operator that accurately minimizes the energy functional only for the training instances, failing to generalize to unseen parameter configurations.

Two key decisions govern this selection process. The first concerns the choice of an effective sampling strategy, such as maximin or Latin Hypercube Sampling, depending on the structure of the parameter space under consideration. The second, and more challenging, concerns the total number of samples required to achieve sufficiently dense coverage. Unlike the former, no principled method exists to determine this number a priori, which may necessitate an iterative refinement of the training set: the operator is retrained repeatedly until satisfactory generalization on held-out samples is observed. Since each retraining cycle entails a full optimization run, this iterative procedure can be computationally prohibitive, particularly for high-dimensional parameter spaces or expensive loss evaluations. Determining an adequate sample size a priori therefore remains an open challenge, and no general solution is addressed in this work.

5.2. Preconditioning strategy for parametric training

Once the training samples are selected, the parametric nature of the problem implies that the stiffness matrix may vary across samples. In principle, the Cholesky factorization used as a preconditioning strategy in P-VINO should therefore be recomputed for each sample individually, a choice referred to here as the *per-*$\mathbf{L}$ strategy. However, this approach would introduce a prohibitive computational and memory overhead, since a distinct factor $\mathbf{L}^{(k)}$ would need to be stored and applied for every sample at each training epoch.

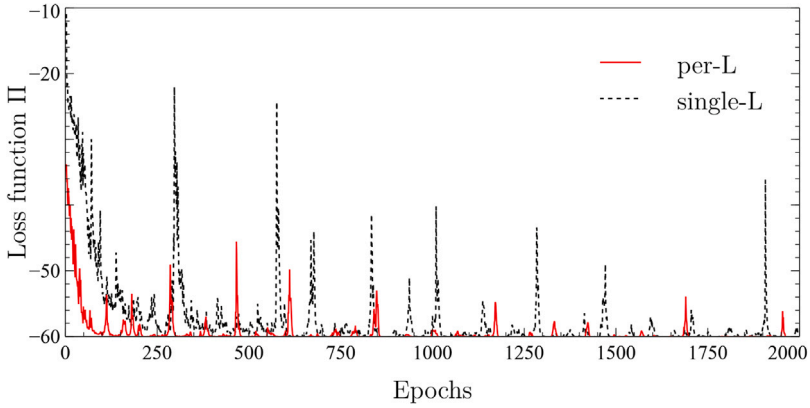


Fig. 6. Convergence history: single-**L** vs. per-**L** preconditioning strategy.

To address this issue, a *single*-**L** preconditioning strategy is adopted in this work, in which a single representative stiffness matrix is selected from the training set and its Cholesky factor **L** is computed once offline and reused to precondition the variational loss across all training samples. This approximate preconditioning enables P-VINO to capture the total strain energy accurately for both training and unseen test samples. Nevertheless, it introduces a residual difficulty in resolving displacement components associated with energetically minor contributions, as the fixed preconditioner cannot adapt to the varying conditioning of each sample's stiffness matrix. However, in standard design practice structural stiffness requirements are usually prescribed. Accordingly, the parameters' space of practical interest is bounded. Hence, the *single*-**L** preconditioning strategy could be used effectively in many practical design scenarios. In this case the preconditioner would be capable of more correctly reshaping the energetic landscape for configuration near the representative one and more accurately predict the response components with relatively small energetic contributions.

As illustrated in Fig. 6, which compares the loss convergence of the *single*-**L** strategy against a *per*-**L** reference in which each training sample receives its own exact Cholesky factorization, the *single*-**L** strategy exhibits slower and less stable convergence. These observations highlight the need for alternative preconditioning strategies that remain effective across variable-stiffness parametric settings without incurring prohibitive computational costs, and represent an open research direction.

6. Results

The numerical results obtained with the P-VINO framework are here presented, aiming at assessing its specific features. In particular, the FEM-based conforming discretization is compared against the immersed Cartesian approach, and the internal energy evaluation strategies are benchmarked against each other and against the original VINO formulation [36]. The benefits of the preconditioning strategy are then demonstrated on an ill-conditioned problem. The individual contributions of the conforming discretization and of the preconditioning strategy are subsequently isolated and assessed through a dedicated ablation study. Finally the framework is assessed in two parametric settings where lamination parameters and applied load, respectively, are treated as variable inputs.

In all test cases, the P-VINO results are compared against solutions obtained from an in-house FEM solver that adopts the same kinematic assumptions, mesh, and integration strategies as the P-VINO framework. Under these conditions, the P-VINO solution is expected to converge to the FEM solution upon training, and any residual discrepancy can be attributed exclusively to the optimizer rather than to discretization or modeling differences. The in-house solver has been validated against Abaqus across a range of problem configurations, and the use of a purpose-built implementation eliminates any differences in the reference solution that might otherwise arise from specific algorithmic choices in commercial codes. To support the reproducibility of the results, all the relevant data, i.e. hyperparameters and training data, are summarized in the Appendix for each single example.

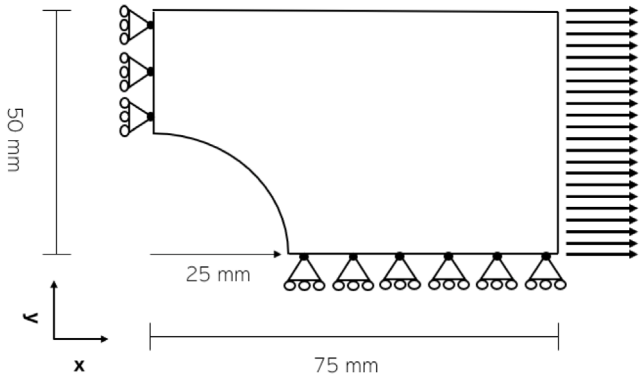


Fig. 7. Quarter plate with hole.

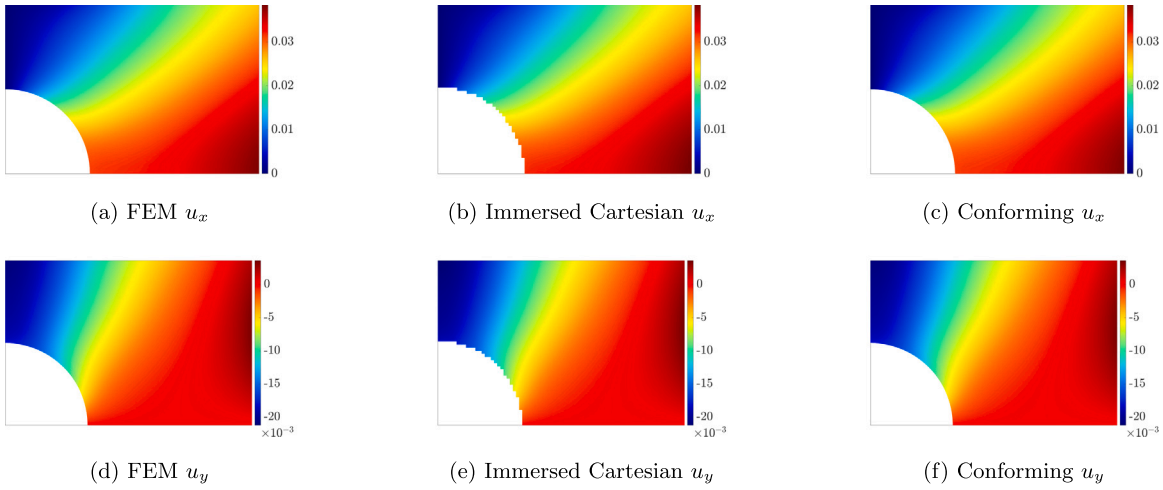
6.1. Geometric discretization: conforming versus immersed strategies

The FEM-based conforming discretization strategy is compared against the immersed Cartesian approach through the following benchmark problem. In particular, the structure under investigation is a quarter of a perforated plate. Plane-stress conditions are assumed, and the load is introduced in the form of a uniform resultant traction of 1000 N applied along the loaded edge; the geometry is shown in Fig. 7.

The material is isotropic linear elastic with Young's modulus $E = 72000$ MPa and Poisson's ratio $\nu = 0.3$. The presence of a curved boundary makes this problem well-suited for assessing both discretization strategies. At the same time, the essential boundary conditions are prescribed along straight edges aligned with the Cartesian grid, so that the immersed strategy of the original VINO framework [36] can be applied without ambiguity.

The two strategies are compared against a reference FEM solution computed on a mesh with a total of 570 Q4 elements. To ensure a fair comparison, both strategies are trained using the same FNO architecture, with 16 modes in each spatial direction, width 50, depth 7, and 64 channels in the last projection layer, and the same learning rate of 0.001.

Table 1 summarizes the total potential energy Π_h, the internal strain energy U_h, and the displacement extrema for both strategies and the FEM reference.



(a) FEM u_x (b) Immersed Cartesian u_x (c) Conforming u_x

(d) FEM u_y (e) Immersed Cartesian u_y (f) Conforming u_y

Fig. 8. Comparison of displacement fields obtained with FEM, immersed Cartesian discretization, and FEM-based conforming discretization.

Table 1
Total potential energy Π_h, internal strain energy U_h, Clapeyron residual $|U_h - \frac{1}{2}W_h|$, and displacement extrema for the quarter plate benchmark.

Method:	FEM (Q4)	P-VINO	
Mesh type:	FE mesh	Immersed	Conforming
Π_h [Nmm]	-1.655×10^{1}	-1.661×10^{1}	-1.655×10^{1}
U_h [Nmm]	1.655×10^{1}	1.673×10^{1}	1.655×10^{1}
$\lvert U_h - \frac{1}{2}W_h \rvert$ [Nmm]	≈ 0	6×10^{-3}	≈ 0
$u^{\max}$ [mm]	3.82×10^{-2}	3.86×10^{-2}	3.82×10^{-2}
$v^{\max}$ [mm]	-2.09×10^{-2}	-2.12×10^{-2}	-2.09×10^{-2}

The conforming strategy leads P-VINO to converge to the FEM solution, as confirmed by the energetic quantities reported in Table 1: the residual $|U_h - \frac{1}{2}W_h|$, which is used as a data-free convergence indicator, approaches zero upon convergence, consistently with the equivalence between the P-VINO minimizer and the FEM solution.

The immersed Cartesian strategy, by contrast, approximates the displacement extrema with reasonable accuracy, but does not achieve full energetic consistency: the absolute value of the total potential energy Π_h differs from the internal strain energy U_h, reflecting the intrinsic limitation of the non-conforming discretization.

To further investigate the field accuracy, the relative L^2-errors with respect the FEM solution are evaluated for both the displacement fields u and v, referring to this definition:

$$\|e_q\|_{L^2} = \frac{\left(\int_\Omega \left(q^{\text{pred}} - q^{\text{FEM}}\right)^2 \, \mathrm{d}\Omega\right)^{1/2}}{\left(\int_\Omega \left(q^{\text{FEM}}\right)^2 \, \mathrm{d}\Omega\right)^{1/2}}, \qquad q \in \{u, v\}. \tag{23}$$

For the conforming strategy, the errors are almost negligible, with $\|e_u\|_{L^2} = 3.48 \times 10^{-6}$ and $\|e_v\|_{L^2} = 3.87 \times 10^{-6}$. By contrast, for the immersed Cartesian mesh, the errors $\|e_u\|_{L^2} = 9.40 \times 10^{-3}$ and $\|e_v\|_{L^2} = 1.83 \times 10^{-2}$ are still small, but no longer negligible. These results confirm the higher accuracy of the FEM-based strategy, which can be attributed to its geometrically conforming nature and, consequently, to a more accurate representation of the physical domain.

The numerical solutions are also compared through the displacement and stress contours reported in Figs. 8 and 9, respectively. As shown in Fig. 8, the displacement fields obtained with both P-VINO discretizations are almost indistinguishable from the FEM reference. This confirms that, at the global displacement level, the immersed Cartesian strategy is still able to provide an accurate approximation of the structural response for domains where the boundary conditions can be imposed on edges parallel or perpendicular to the Cartesian grid.

A different behavior is observed when the stress fields are considered. In Fig. 9, the immersed Cartesian discretization produces less regular stress contours, especially in the neighborhood of the curved boundary associated with the hole. This effect is a direct consequence of the non-conforming nature of the immersed mesh, which does not provide an exact geometrical representation of the physical domain boundary. As a result, the stress distributions obtained with the immersed Cartesian strategy deviate from the FEM reference solution, with the discrepancy being particularly evident in the stress components near the hole boundary, as shown in Figs. 9(b), 9(e), and 9(h). Conversely, the FEM-based conforming discretization accurately reproduces the FEM stress fields throughout the domain, including the region close to the curved boundary. This confirms that the conforming strategy not only restores energetic consistency, but also improves the local quality of the recovered stress fields.

To complete the assessment, the convergence history of the loss functions is reported in Fig. 10 for the two different meshes under investigation. The plot reports the value of the FEM energy as a horizontal line to clarify the converging behavior of the two proposed solution strategies.

In spite of a convergent behavior for both cases, Fig. 10 reveals the significantly faster convergence of the conforming strategy with respect to the immersed approach. With as few as 350 iterations, the total energy deviates by 0.3% with respect to the FE reference. On the contrary, for the immersed mesh, the difference measured at the same epoch is equal to 16.104%. A spike is observed at mid-training for both strategies and is attributed to an overshooting effect induced by the constant learning rate of the Adam optimizer [44]. The total training budget is fixed at 2000 epochs for both strategies to allow the immersed approach to reach approximate convergence; under the conforming strategy, however, training could have been terminated considerably earlier by adopting the Clapeyron residual $|U_h - \frac{1}{2}W_h|$ as a data-free stopping criterion. Taken together, these results confirm that the FEM-based conforming discretization simultaneously improves solution accuracy, restores discrete energetic consistency, and accelerates convergence, while providing a data-free criterion to assess whether the solution has effectively converged to the FEM reference.

6.2. Internal energy evaluation strategies

The internal energy evaluation strategies are now investigated by examination of the corresponding computational costs. For this purpose, a rectangular plate is considered with dimensions 75 mm × 50 mm and linear elastic isotropic constitutive law under plane stress conditions. A uniform traction is applied along the short edge with resultant

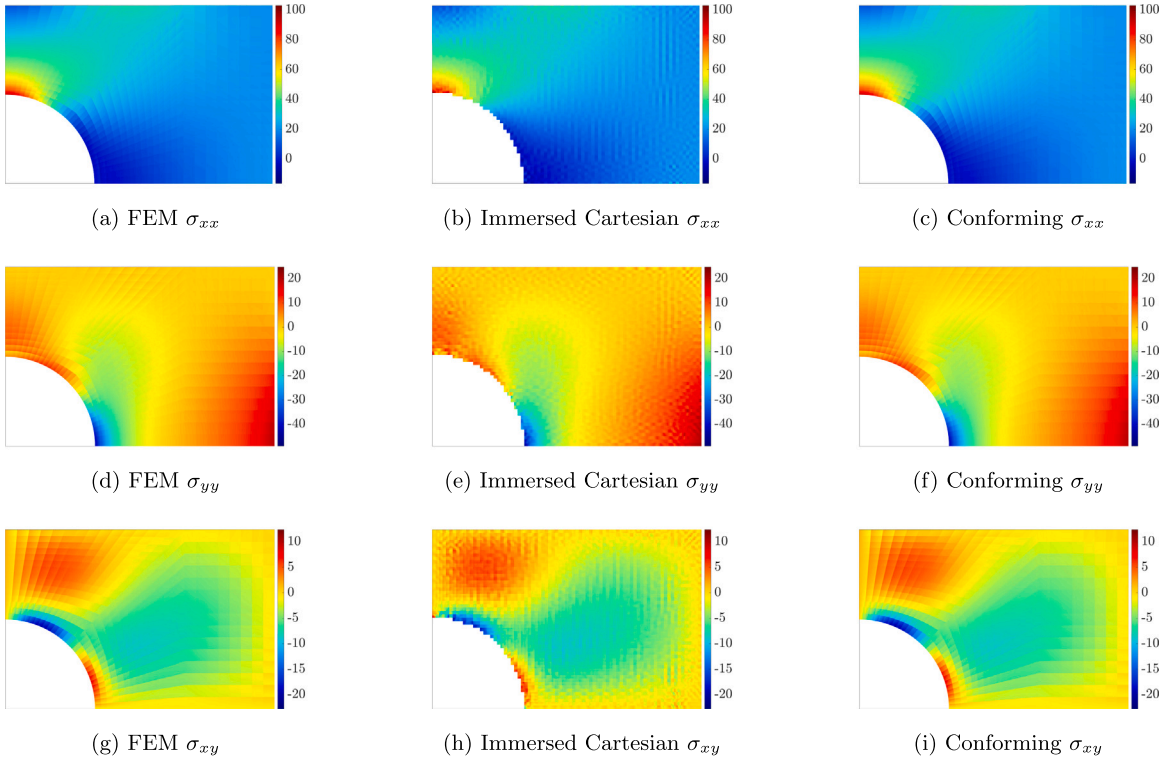


Fig. 9. Comparison of stress fields obtained with FEM, immersed Cartesian discretization, and FEM-based conforming discretization.

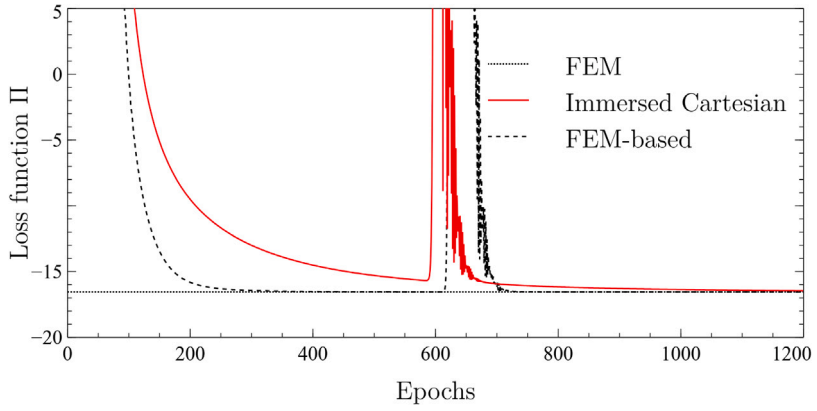


Fig. 10. Convergence history: Immersed Cartesian Mesh vs FEM-based conforming discretization.

1000 N. Symmetry boundary conditions are applied along the left and bottom edges, constraining the displacement in the x-direction on the left edge and in the y-direction on the bottom edge. The mesh consists of 1521 rectangular elements fully contained within the physical domain. As a consequence, the original VINO closed-form expression U_{VINO} is directly applicable, hence enabling a controlled comparison in which the effect of the integration strategy is isolated from that of the discretization. The same FNO architecture and learning rate adopted in the previous section are retained throughout.

Four strategies are investigated here and the corresponding internal energy evaluation reports a subscript to clarify the adopted approach. In particular:

- $VINO$ denotes the use of the closed-form expression, Eq. (14), derived analytically for rectangular elements as in the original VINO framework [36]
- $opt1$ refers to the use of the closed-form expression derived symbolically for generic Q4 elements through the isoparametric formulation, Eq. (16), and made JIT-compilable by splitting the expression into separate algebraic blocks.
- $opt2$ corresponds to a further optimized version of $opt1$ in which the element-geometry-dependent terms, which remain constant throughout training, are precomputed just once, before the optimization begins in order to reduce the JIT-compilation overhead.
- K is the stiffness-based approach in which the elemental stiffness matrices are assembled once by standard numerical quadrature and the internal energy is evaluated during training through the quadratic form in Eq. (17).

The corresponding strain energies are denoted hereafter as U_{VINO}, U_{opt1}, U_{opt2} and U_{K}. Their computational performance is summarized in Table 2, which reports the per-epoch training time $\bar{t}_{\mathrm{ep}}$, the total training time T_{train}, and the JIT compilation overhead T_{JIT} required to construct the computational graph prior to training. All timings are normalized

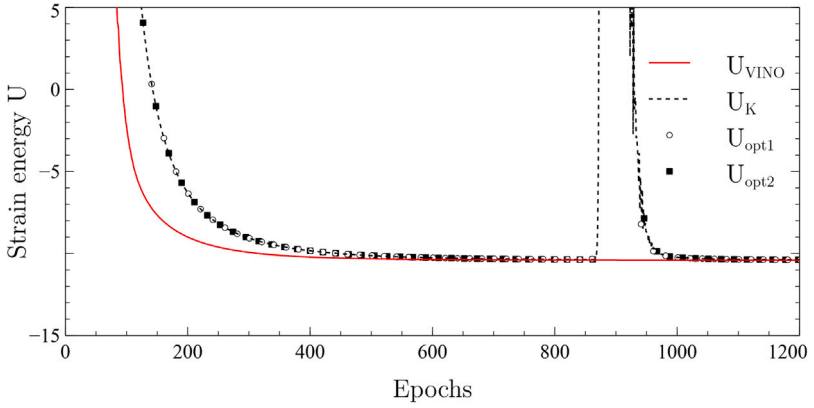


Fig. 11. Convergence history for internal energy evaluation strategies.

Table 2
Computational cost comparison for internal-energy strategies, normalized by the U_{VINO} reference timings t_{ref}.

	U_{opt1}	U_{opt2}	U_K	U_{VINO}
$\bar{t}_{\text{ep}}/t_{\text{ref}}$	1.050	1.029	1.000	1.000
$T_{\text{train}}/t_{\text{ref}}$	1.216	1.107	1.000	1.000
$T_{\text{JIT}}/t_{\text{ref}}$	19.459	9.425	1.111	1.000

by the corresponding values of the U_{VINO} strategy taken as reference (t_{ref}).

Among the strategies applicable to generic element shapes, the U_K-approach achieves per-epoch training times comparable to those of U_{VINO}, since the energy evaluation reduces to a single matrix–vector product per element, as shown in Eq. (17), at the cost of a memory requirement that grows with the number of elements and may become a limiting factor for fine meshes or large parametric training sets. This memory overhead is not present in U_{opt1} and U_{opt2}, which avoid storing the stiffness matrices entirely. Despite the larger JIT compilation overhead shared by both symbolic strategies, the precomputation of geometry-dependent terms introduced in U_{opt2} reduces both the per-epoch training time and the compilation overhead with respect to U_{opt1}, thereby justifying this further optimization.

The convergence histories are reported in Fig. 11 to assess whether the choice of integration strategy affects the training dynamics, beyond the computational cost already discussed.

The curves in the figure confirm that the analytically integrated expression U_{VINO} yields faster and more stable convergence than the strategies based on numerical quadrature. This is consistent with the findings of [36]. On the contrary, the three numerically integrated strategies, i.e. U_{opt1}, U_{opt2}, and $\mathbf{K}_e$, produce nearly indistinguishable convergence curves, confirming that the choice among them is governed primarily by implementation cost and memory requirements rather than by accuracy.

This comparison establishes that the accuracy advantage of analytical integration is restricted to rectangular-element meshes in which all elements are fully contained within the domain, and that for non-rectangular geometries the gain from geometrically exact boundary representation dominates over the choice of integration scheme.

6.3. Ill-conditioning and pre-conditioning

The ill-conditioning of the stiffness matrix arising from the Mindlin model is a critical aspect that can prevent the VINO framework from converging. To demonstrate this issue, a benchmark is proposed here consisting of a rectangular laminated plate of dimensions 75 mm × 50 mm, fully clamped along the short edge. For generality, the stiffness-based strategy U_K is considered for the evaluation of the internal strain energy. The FNO architectures adopted in VINO and P-VINO are summarized in Table 3.

Table 3
FNO hyperparameters for VINO and P-VINO.

	Mode 1	Mode 2	Width	Depth	Channels last proj.
VINO	16	16	110	13	80
P-VINO	16	16	40	7	35

Table 4
Conditioning numbers of the individual stiffness contributions and of the assembled global stiffness matrix for the two laminates under investigation.

	$\kappa(\mathbf{K}_m)$	$\kappa(\mathbf{K}_b)$	$\kappa(\mathbf{K}_s)$	$\kappa(\mathbf{K})$
$[0, 90]_s$	1.82×10^5	1.14×10^5	7.11×10^{22}	2.93×10^8
$[0, 45, 90, -45]_s$	1.12×10^5	7.83×10^4	1.35×10^{22}	3.53×10^8

As seen, VINO requires a significantly more complex architecture than P-VINO, which instead, owing to the improved conditioning of the energy landscape achieved by the preconditioning strategy introduced earlier, is able to reach convergence with a substantially lighter network.

The conditioning number is reported in Table 4 for the individual stiffness contributions entering Eq. (11) and the assembled global stiffness matrix $\mathbf{K}$ for the $[0, 90]_s$ and $[0, 45, 90, -45]_s$ stacking sequences. The membrane–bending coupling contribution $\mathbf{K}_{mb}$ is not reported, as it vanishes identically for both laminates owing to their symmetric stacking sequence.

The results reveal that, although the problem is ill-conditioned, the individual contributions present a markedly different behavior when examined separately. The membrane and bending contributions share similar conditioning, whereas the transverse shear contribution $\mathbf{K}_s$ is by far the most severely ill-conditioned, with $\kappa(\mathbf{K}_s)$ of order 10^{22}, a value directly attributable to the use of Selective Reduced Integration for its evaluation. The practical consequences of this ill-conditioning, however, depend on the nature of the loading. Under membrane-dominated loading, the response is governed almost entirely by the in-plane degrees of freedom, and the ill-conditioning of $\mathbf{K}_s$ has essentially no impact. Under bending-dominated loading, by contrast, the transverse degrees of freedom w, θ_x, θ_y are essential and appear simultaneously in both the shear and bending energy contributions. The severe ill-conditioning of $\mathbf{K}_s$ therefore degrades the approximation of precisely those kinematic components that also control the bending response, compromising not only the shear energy contribution but indirectly the bending one as well, despite $\kappa(\mathbf{K}_b)$ being several orders of magnitude lower. This mechanism, which directly governs the curvature of the energy landscape as shown by Eq. (18), is at the root of the convergence difficulties experienced by VINO in plate bending problems.

Table 5
Energy contributions (in Nmm)– VINO, P-VINO, and FEM.

	$[0,90]_s$			$[0,45,90,-45]_s$		
	VINO	P-VINO	FEM	VINO	P-VINO	FEM
Π_h	-6.13×10^{-2}	-2.08	-2.08	-4.45×10^{-2}	-2.76	-2.76
W_h	7.16×10^{-2}	4.16	4.16	5.26×10^{-2}	5.52	5.52
U_h	1.03×10^{-2}	2.08	2.08	8.07×10^{-3}	2.76	2.76
U_m	1.17×10^{-4}	1.50×10^{-6}	≈ 0	2.00×10^{-4}	1.48×10^{-6}	≈ 0
U_b	9.76×10^{-3}	2.08	2.08	7.52×10^{-3}	2.76	2.76
U_s	4.59×10^{-4}	1.85×10^{-3}	1.85×10^{-3}	3.53×10^{-4}	2.70×10^{-3}	2.70×10^{-3}

To provide a quantitative illustration of this mechanism, the benchmark plate introduced above is subjected to a uniformly distributed transverse pressure applied over the entire top surface, activating a purely bending-dominated response. Two stacking sequences are considered: $[0,90]_s$ and $[0,45,90,-45]_s$. The energy decomposition reported in Table 5 follows the contributions introduced in Eq. (11), reporting separately the membrane (U_m), bending (U_b), and transverse shear (U_s) components of the internal strain energy.

The results confirm the failure mechanism discussed above: VINO fails entirely to capture the correct energy level for both laminates. As a direct consequence of the ill-conditioning of $\mathbf{K}_s$, neither the transverse shear contribution U_s nor the bending contribution U_b are correctly approximated. This confirms the coupling mechanism discussed above: the degraded approximation of the transverse degrees of freedom propagates into the bending energy as well. The total potential energy Π_h and internal energy U_h are consequently two orders of magnitude smaller than the FEM reference, and the displacement and rotation fields are entirely erroneous, as reported in Table 6, with the transverse displacement w underestimated by two orders of magnitude. The in-plane displacements u and v are not reported, as they vanish identically for a plate subjected to pure transverse pressure in the FEM solution; spurious non-zero values nonetheless appear in the VINO prediction as a direct consequence of the incomplete suppression of the membrane energy contribution U_m.

P-VINO, by contrast, recovers the correct energy level for both laminates, with all energy components in full agreement with the FEM reference to the reported significant figures, as shown in Table 5. The displacement and rotation extrema reported in Table 6 are consequently in complete agreement with the FEM solution. The Clapeyron residual, $|U_h - \frac{1}{2}W_h|$ approaches zero at the end of the P-VINO training, confirming its capabilities as a data-free convergence indicator. The accuracy of the recovered fields is also confirmed by the relative L^2-errors of w, θ_x, and θ_y, evaluated according to Eq. (23). For the $[0,90]_s$ laminate, the resulting errors are $\|e_w\|_{L^2} = 3.30\times10^{-6}$, $\|e_{\theta_x}\|_{L^2} = 7.92\times10^{-6}$, and $\|e_{\theta_y}\|_{L^2} = 7.73\times10^{-4}$. For the $[0,45,90,-45]_s$ laminate, the corresponding errors are $\|e_w\|_{L^2} = 7.06\times10^{-5}$, $\|e_{\theta_x}\|_{L^2} = 6.91\times10^{-5}$, and $\|e_{\theta_y}\|_{L^2} = 1.04\times10^{-4}$. These results further corroborate the conclusions achieved by inspection of Tables 5 and 6. To further highlight the difficulties experienced by the VINO framework in handling plate problems, the convergence histories for four stacking sequences, $[0,0]_s$, $[0,90]_s$, $[90,90]_s$, and $[0,45,90,-45]_s$, are reported. Two loading conditions are considered: in-plane traction and transverse pressure, shown in Figs. 12 and 13, respectively.

The former activates the membrane response only. Ill-conditioning of $\mathbf{K}_s$ is not expected to prevent convergence, since the response is governed by the in-plane degrees of freedom alone;

The latter activates exclusively the bending and transverse shear mechanisms and represents the critical case for the framework, as discussed above.

As shown in Fig. 12, under in-plane traction loading, VINO is also able to converge, since the bending and shear mechanisms are not activated. Nonetheless, P-VINO converges both faster and more stably than VINO, as the loss history of the latter exhibits pronounced overshoots that are entirely absent in P-VINO, despite relying on a significantly lighter architecture.

Under bending-dominated loading, Fig. 13, VINO stagnates at a plateau well above the FEM energy level for all four laminates, confirming that the ill-conditioning of $\mathbf{K}_s$ prevents the optimizer from making progress regardless of the stacking sequence. P-VINO, by contrast, converges correctly to the FEM total potential energy in all cases.

These results further demonstrate that the improved conditioning of the energy landscape yields a concrete benefit even in cases where ill-conditioning alone does not prevent convergence.

6.4. Assessment of the individual contributions of P-VINO

To quantify the relative contribution of the FEM-based conforming discretization and of the Cholesky-based preconditioning strategy, a quarter of a perforated plate is considered. The geometry is the same as that reported in Fig. 7; however, in the present case the bending response is investigated, so the structure is modeled as a Mindlin plate. Symmetry boundary conditions are applied on the left and bottom edges: the left edge is constrained on the rotation θ_x, while the bottom edge is constrained on the rotation θ_y. In addition, both right and top edges are clamped. The laminate stacking sequence is $[0,90,90,0]$, and a uniform transverse pressure of 1000 Pa is applied on the upper face of the plate. The purpose of this test is to isolate the effect of the two main components introduced in P-VINO, namely the FEM-based conforming discretization and the Cholesky-based preconditioning. To this end, four neural operators are trained and compared:

- **VINO**: this operator corresponds to the original formulation presented in [36]. It relies on an immersed Cartesian discretization strategy and does not include any preconditioning.
- **FEM-based conforming discretization without preconditioning**: this operator adopts the FEM-based conforming discretization, but the variational loss is minimized without applying the Cholesky-based preconditioning.
- **Immersed Cartesian discretization with preconditioning**: this operator retains the immersed Cartesian discretization of the original VINO framework, while the Cholesky-based preconditioning strategy is applied during training.
- **P-VINO**: this operator corresponds to the full proposed framework, combining the FEM-based conforming discretization with the Cholesky-based preconditioning strategy.

All neural operators are trained using the same optimizer and FNO hyperparameters. These settings coincide with those reported in Table 3 for the P-VINO model adopted earlier. In this way, any difference in the final performance can be isolated and directly attributed to the discretization strategy, to the preconditioning strategy, or to their combined effect.

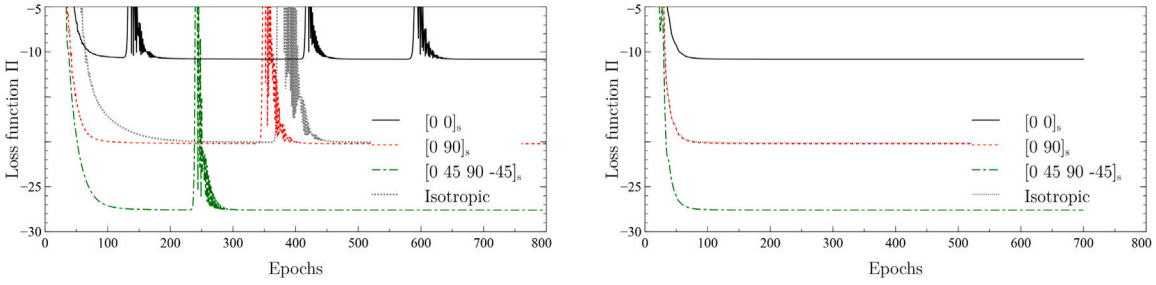
Training is stopped either upon reaching the maximum number of epochs or when the data-free stopping criterion based on the Clapeyron theorem is satisfied, i.e., when the residual $|U_h - \frac{1}{2}W_h|$ falls below the threshold 10^{-4} Nmm.

Fig. 14 illustrates the two discretization strategies. In particular, Fig. 14(a) shows the FEM-based conforming mesh, while Fig. 14(b) presents the immersed Cartesian grid. In this second case, a zoom is provided over the elements which are not fully contained in the domain, which are also highlighted in red, for clarity.

In the FEM-based conforming case, all elements are fully contained within the domain by construction, so the equivalent nodal force at each node a of element e is computed by standard Gauss quadrature. On the other hand, in the immersed Cartesian case, a 3×3 selective Gauss quadrature is applied over each boundary cell where only the integration points falling inside the physical domain – red points in Fig. 14(b) – contribute to the nodal force. This selective integration naturally accounts for the partial overlap between the Cartesian cell

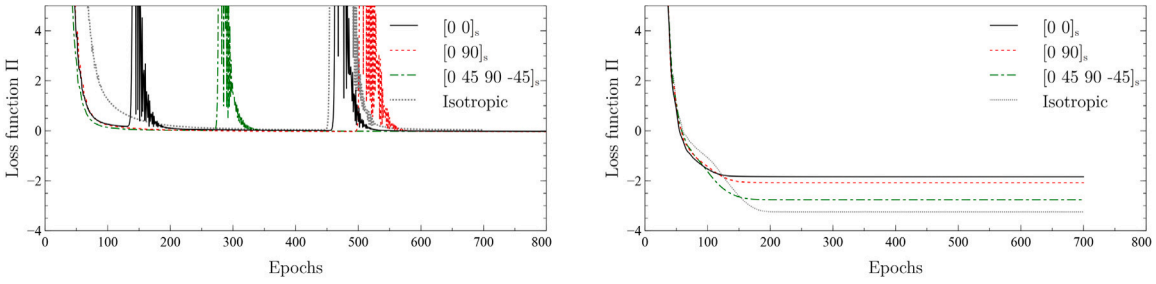
Table 6
Displacement (in mm) and rotations (in rad) – VINO, P-VINO, and FEM.

	$[0, 90]_s$			$[0, 45, 90, -45]_s$		
	VINO	P-VINO	FEM	VINO	P-VINO	FEM
$w_{\min}$	-2.80×10^{-2}	-2.77	-2.77	-1.96×10^{-2}	-3.92	-3.92
$w_{\max}$	≈ 0	≈ 0	≈ 0	≈ 0	≈ 0	≈ 0
$\theta_{x,\min}$	-1.24×10^{-3}	-4.94×10^{-2}	-4.94×10^{-2}	-1.20×10^{-3}	-6.77×10^{-2}	-6.77×10^{-2}
$\theta_{x,\max}$	8.90×10^{-4}	≈ 0	≈ 0	6.09×10^{-4}	≈ 0	≈ 0
$\theta_{y,\min}$	-5.27×10^{-4}	-1.33×10^{-3}	-1.33×10^{-3}	-2.48×10^{-4}	-3.14×10^{-3}	-3.14×10^{-3}
$\theta_{y,\max}$	5.87×10^{-4}	1.33×10^{-3}	1.33×10^{-3}	6.02×10^{-4}	9.97×10^{-3}	9.97×10^{-3}



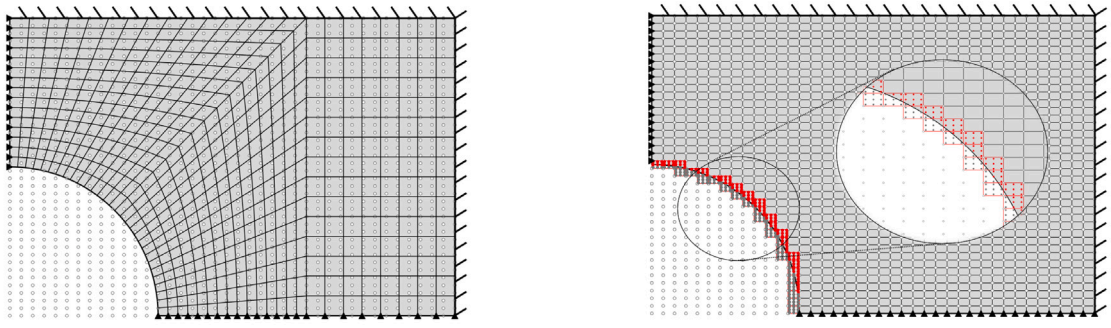
(a) VINO (b) P-VINO

Fig. 12. Loss function for different layups – traction load.



(a) VINO (b) P-VINO

Fig. 13. Loss function for different layups – pressure load.



(a) (b)

Fig. 14. Discretization strategies: (a) FEM-based, (b) Immersed Cartesian.

and the physical boundary. To preliminarily check the effectiveness of this approach, the total load resultant is computed for both strategies, leading to a negligible difference of 0.002% approximately.

To compare the four strategies outlined above, different metrics are used. Specifically, the comparison is conducted by referring to the number of epochs, the Clapeyron residual and L^2 errors according to Eq. (23). In addition, the L^∞- and the energy-norm $\|e\|_E$ are introduced too. The first one is evaluated for the bending deflection w according to the definition:

$$\|e_w\|_{L^\infty} = \frac{\max_i \left| w_i^{\text{pred}} - w_i^{\text{FEM}} \right|}{\max_i \left| w_i^{\text{FEM}} \right|}, \tag{24}$$

Table 7
Relative errors with respect to the FEM reference solution for the four training strategies, together with the number of epochs at which training stopped.

	VINO	Immersed + precond.	FEM-based d.s. + no precond.	P-VINO
Epochs	800	621	800	323
Clapeyron residual ([Nmm])	9.951×10^{-1}	9.968×10^{-5}	1.010	9.948×10^{-5}
$\|e_w\|_{L^2}$	9.809×10^{-1}	1.229×10^{-3}	1.304	1.940×10^{-4}
$\|e_{\theta_x}\|_{L^2}$	1.073	3.211×10^{-3}	1.333	4.904×10^{-4}
$\|e_{\theta_y}\|_{L^2}$	1.196	2.895×10^{-3}	1.503	8.461×10^{-4}
$\|e\|_E$	5.466	6.028×10^{-1}	5.322	2.831×10^{-2}
$\|e_w\|_{L^\infty}$	1.094	1.617×10^{-2}	1.295	2.809×10^{-4}

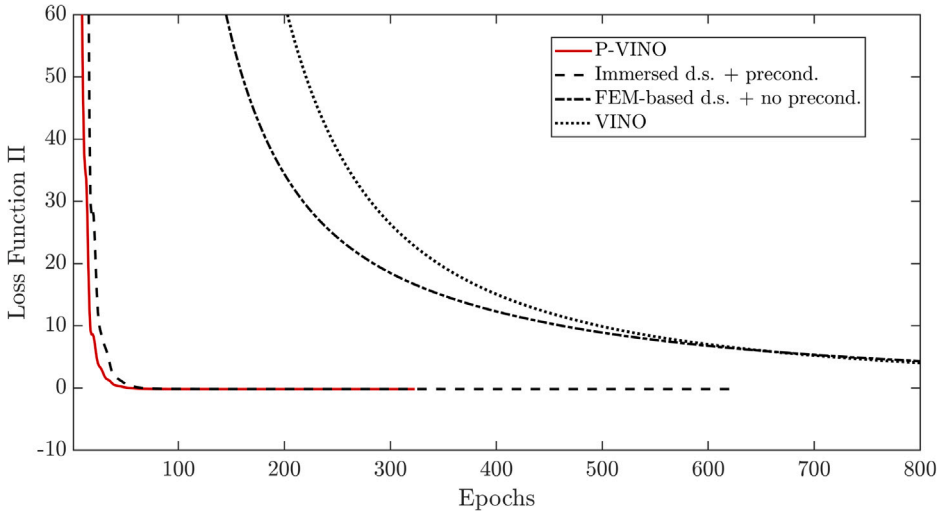


Fig. 15. Convergence history of the four different training strategies.

where i runs over the FEM nodes. The second norm, $\|e\|_E$, is defined as:

$$\|e\|_E = \left(\frac{\mathbf{e}^T \mathbf{K} \mathbf{e}}{\left(\mathbf{d}^{\text{FEM}}\right)^T \mathbf{K} \mathbf{d}^{\text{FEM}}} \right)^{1/2}, \qquad \mathbf{e} = \mathbf{d}^{\text{pred}} - \mathbf{d}^{\text{FEM}}. \tag{25}$$

A summary of the results is provided in Table 7.

The values reported in the first and third columns, corresponding to non-preconditioned strategies, are characterized by large errors. On the contrary, the second and fourth columns, relative to strategies with preconditioning, are in excellent agreement with the FEM reference solution, and the errors are very small. These results illustrate the paramount importance of training the operators with preconditioning. Among the four examined approaches of Table 7, P-VINO outperforms the other ones both in terms of accuracy, as revealed by the smallest observed errors, and convergence speed with as few as 323 epochs. These results confirm that, for bending-dominated problems, a preconditioning strategy is mandatory to recover the solution, while the FEM-based discretization helps to speed up training and achieve overall better accuracy thanks to its conforming nature. The Clapeyron residual also proves to be an effective stopping criterion, as both preconditioning strategies, which are stopped before reaching the maximum number of epochs, recover the solution.

In Fig. 15 the convergence histories of the four operator trainings are presented to illustrate the different observed responses.

As shown, P-VINO displays the fastest convergence and requires only about half the number of epoch compared to the Immersed discretization strategy with preconditioning. Instead, the operators trained without a preconditioning strategy do not converge, even though the FEM-based strategy still accelerates convergence during the early stages of training.

To complete the comparison, a computational analysis of the four operator trainings is presented. All computational times and memory usage are normalized with respect to those of VINO. Table 8 reports the nondimensional data-assembly time, which includes mesh generation and, for the strategies with preconditioning, the Cholesky factorization, together with the JIT compilation time, the mean epoch time, the inference time, and the data memory cost.

The results reveal that, in general, the preconditioning strategy introduces an additional time and memory usage cost. The first one is due to the factorization process in the data-assembly phase and the evaluation of the displacement field d from the auxiliary variable z at each epoch. The second one is due to the extra memory usage for storing L. All of these costs are more relevant for the immersed strategy. Indeed, although both strategies use the same operator grid, the FEM-based conforming discretization employs a smaller number of elements while still achieving more accurate results. The larger number of elements of the immersed strategy translates into a larger stiffness matrix to be assembled and factorized, and into a higher number of element energies to be evaluated at each epoch. In the FEM-based strategy, part of this saving in the per-epoch cost is reduced by the interpolation of the displacement field onto the conforming mesh. Overall, both for the preconditioned and non-preconditioned cases, the FEM-based strategy turns out to be less expensive than the immersed one for the same choice of hyperparameters, while being more accurate. Finally, it should be noted that P-VINO overall requires fewer epochs, further reducing the total training time.

6.5. Parametric study: variable stacking sequence

The first parametric study aims at assessing the ability of the neural operator to learn the dependence of the structural response on the laminate stacking sequence. This example also serves as the reference application of the single-**L** preconditioning strategy illustrated earlier. The structure under investigation is the same rectangular plate presented in the previous section and subjected to a uniform in-plane load of 1000 N. Focus of the parametric study is the orientation of the plies of

Table 8
Computational cost of the four training strategies, normalized with respect to VINO.

	VINO	Immersed + precond.	FEM-based d.s. + no precond.	P-VINO
Data assembly	1	5.253	0.271	0.916
JIT compilation	1	2.524	0.991	1.237
Mean epoch time	1	2.259	0.998	1.088
Inference time	1	1.859	0.872	0.969
Data memory	1	2.316	0.197	0.444

Table 9
Mean and maximum relative percentage errors on training ($N_{\text{train}} = 210$) and test ($N_{\text{test}} = 1159$) laminates.

	Training		Test	
Metric	Mean [%]	Max [%]	Mean [%]	Max [%]
Π_h	5.74×10^{-2}	1.81×10^{-1}	3.46×10^{-1}	4.44
U_h	1.35	4.06	4.40	4.00×10^{1}
$\|e_u\|_{L^2}$	7.71×10^{-1}	2.07	2.23	1.79×10^{1}
$\|e_v\|_{L^2}$	3.29	1.45×10^{2}	9.98	3.81×10^{2}
$u^{\max}$	7.51×10^{-1}	2.46	2.22	1.74×10^{1}
$v^{\max}$	2.59×10^{1}	3.49×10^{3}	4.76×10^{1}	1.23×10^{4}
$v^{\min}$	6.31×10^{1}	1.05×10^{2}	3.10×10^{1}	7.50×10^{3}

the symmetric four-layer laminate, with orientations free to vary with steps of 5°. As result, the total number of the combinations amounts to 1369, representing a sufficiently dense and well-populated parametric space to be explored.

The stacking sequence of each laminate is encoded through its lamination parameters. Due to the symmetry of the stack, only $\boldsymbol{\xi}^A$ and $\boldsymbol{\xi}^D$ are required to fully characterize the in-plane and bending stiffness properties, respectively.

Each lamination parameter vector is sampled on an equispaced Cartesian grid and provided as a separate input channel to the neural operator, resulting in eight input channels in total.

Among the 1369 admissible laminates, a training set of $N_{\text{train}} = 210$ laminates, corresponding to approximately 15% of the total, is selected using a maximin design. This strategy selects samples by sequentially maximizing the minimum distance to all previously selected points in the lamination parameter space, thereby promoting a uniform and well-spread coverage of the parameter domain. The remaining 1159 laminates are retained as a test set to assess the generalization capability of the trained operator.

The representative laminate for the single-**L** preconditioning is identified by selecting the training sample whose lamination parameter vector minimizes the sum of pairwise distances to all other training samples,

$$k^\star = \underset{k}{\operatorname{argmin}} \sum_{j=1}^{N_{\text{train}}} \left\| \boldsymbol{\xi}^{(k)} - \boldsymbol{\xi}^{(j)} \right\|_2, \tag{26}$$

where the concatenated vector $\boldsymbol{\xi}^{(k)} = [\boldsymbol{\xi}^{A,(k)}, \boldsymbol{\xi}^{D,(k)}]$ collects the lamination parameters of the kth laminate. The Cholesky factor of the corresponding stiffness matrix is then computed once offline and reused to precondition the variational loss for all training samples. The FNO architecture adopted is the one used for P-VINO in Table 3, while training is stopped after 2000 epochs.

The mean and maximum relative percentage errors for Π_h, U_h, and the displacement extrema, together with the relative L^2-errors for u and v expressed in percentage form are reported in Table 9 for the training and test laminates, respectively. The comparison is presented by inspection of the total potential and internal energies, and the in-plane displacement components.

The errors on the training laminates are generally lower than those observed on the test set. For both sets, the percentage errors on the total potential energy Π_h, the internal strain energy U_h, and the displacement component u remain low, whereas substantially larger errors are observed for the extrema of the transverse in-plane displacement v. This behavior is a direct consequence of the single-**L** preconditioning strategy: since a single Cholesky factor is used to reshape the energy landscape for all laminates, the preconditioning is exact only for the representative laminate and approximate for all others. As a result, the energetically dominant displacement component u is captured accurately, while the minor contribution associated with v, which, for a plate loaded predominantly along x represents only a small fraction of the total internal energy, is not resolved with the same accuracy at the extrema. Nevertheless, the corresponding L^2-errors indicate that the global v field is still reasonably approximated, and that the largest discrepancies are mainly localized in the peak values. Furthermore, laminates with lower stiffness along the loading direction contribute more strongly to the optimization than stiffer ones, introducing an unintended bias in the training process. The significant variability of Π_h across the training laminates, induced solely by the different lamination sequences, is reported for a representative subset in Fig. 16.

The combined effect of the single-**L** approximation and the loss scaling bias is further illustrated in Fig. 17, which reports the signed percentage errors on $u^{\max}$ and $v^{\max}$ evaluated across all 1369 laminate configurations and plotted against the laminate sequence index. Filled circles denote laminates used for testing, while hollow circles denote those used for training. Cases in which $v^{\max}$ is identically zero are excluded from17(b), accounting for the reduced number of points displayed.

As seen, the errors on $u^{\max}$ remain consistently small across the entire configuration space, while those on $v^{\max}$ exhibit a markedly wider spread, confirming that the difficulty in resolving the transverse displacement is systematic and not limited to isolated configurations. The operator approximated by P-VINO therefore yields reliable predictions for $u^{\max}$, whereas the predictive reliability for $v^{\max}$ is undermined by two concurring factors already discussed: the distance of each laminate from the representative one, which governs the quality of the energy landscape reshaping, and its total potential energy Π_h, which controls its weight in the training loss. Laminates far from the representative sample and with low Π_h are consequently the most prone to large errors on $v^{\max}$, suggesting that more advanced preconditioning strategies represent a promising direction for future development.

6.6. Parametric study: variable load

As a second parametric study, the P-VINO framework is applied to investigate a structure, whose configuration is now frozen, but subjected to different loading conditions. In other words, the focus is now on the parametric definition of the external load.

In addition to representing an interesting design scenario, this example serves also to clarify an important computational feature. As the stiffness matrix is shared across all the training instances, the Cholesky factorization is exact for each single sample. The analysis of the results allows then to clarify the role of preconditioning with or without an approximated factorization. The same cantilever plate considered in Section 6.3 is investigated. The loading condition is now introduced as a spatially varying in-plane traction load, modeled as a Gaussian random field with fixed mean $\mu = 25\,\text{N}$, prescribed variance, and a kernel parametrized by a length scale l controlling the smoothness of the realizations. Three representative load realizations are shown in Fig. 18, where the horizontal axis reports the load magnitude in Newton, the vertical axis identifies the 41 nodes distributed along the

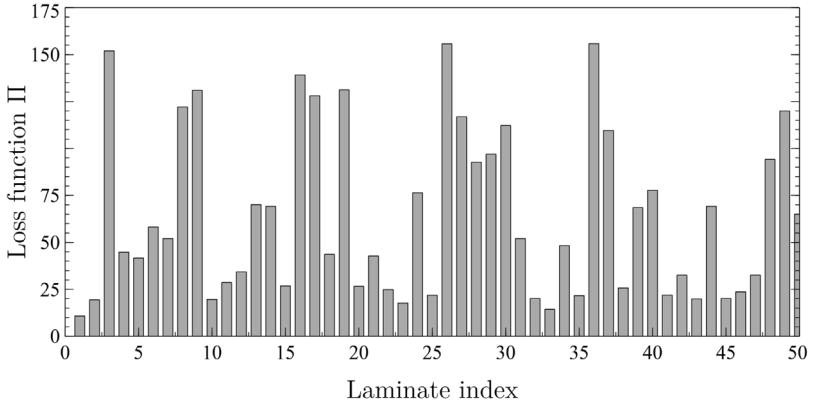


Fig. 16. Total potential energy Π_h for the first 50 laminates used in the training batch.

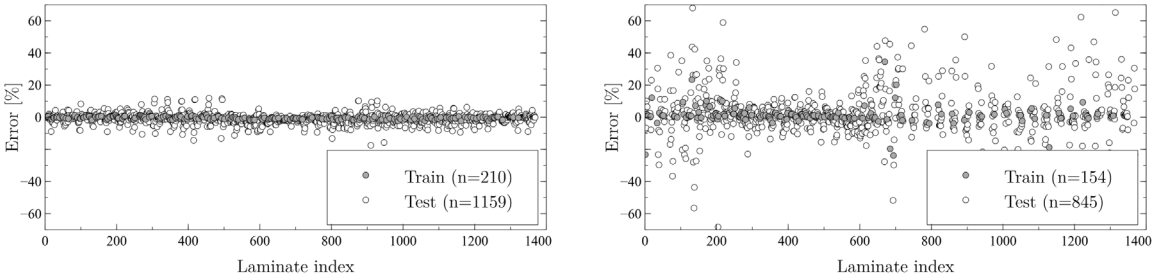


Fig. 17. Percentage errors evaluated across all laminate configurations: (a) $u^{\max}$, (b) $v^{\max}$.

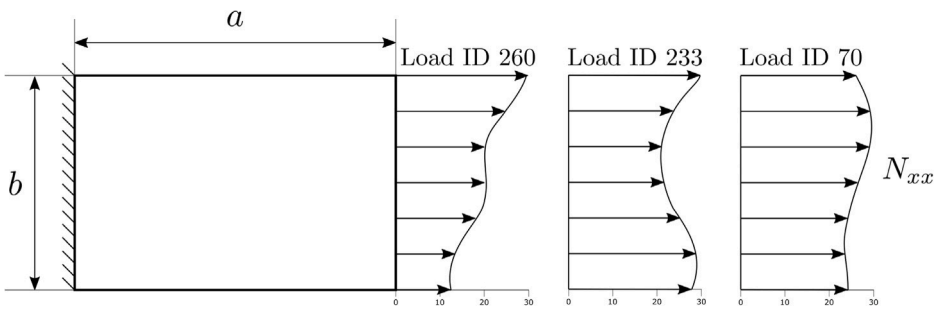


Fig. 18. Three representative realizations of the Gaussian random load field.

loaded edge, and each curve corresponds to a different realization index within the full set of 270 samples.

Of the 270 realizations generated, $N_{\text{train}} = 200$ are used for training and the remaining $N_{\text{test}} = 70$ are retained as a test set to assess generalization. Since all samples share the same laminate and therefore the same stiffness matrix, a single Cholesky factorization is performed once offline and applied exactly to all training and test samples.

The mean and maximum relative percentage errors for Π_h, U_h, and the displacement extrema, together with the relative L^2-errors for u and v expressed in percentage form, are reported in Table 10 for the training and test sets.

The errors on the training and test sets are in close agreement, confirming the generalization capability of the operator across unseen load realizations. Compared to the variable stacking sequence case, a marked reduction in the errors is observed across all metrics, and in particular for the transverse displacement v, consistently with the expectation that an exact Cholesky factorization allows the energy landscape of every sample to be reshaped with equal accuracy. The total potential energy Π_h, the internal strain energy U_h, and the displacement component u are all recovered with very low errors on both sets, while the errors on v, although larger in relative terms due to its smaller contribution to the total energy, are substantially reduced with respect to the approximate preconditioning case.

This behavior is further confirmed by Fig. 19, which reports the distributions of the relative percentage errors on Π_h and $u^{\max}$ for both training and test samples, with the horizontal axis showing the error

Table 10
Mean and maximum relative percentage errors on training ($N_{\text{train}} = 200$) and test ($N_{\text{test}} = 70$) sets.

	Training		Test	
Metric	Mean [%]	Max [%]	Mean [%]	Max [%]
Π_h	3.55×10^{-4}	1.66×10^{-3}	4.54×10^{-4}	2.53×10^{-3}
U_h	2.22×10^{-1}	4.07×10^{-1}	2.25×10^{-1}	6.40×10^{-1}
$\|e_u\|_{L^2}$	1.34×10^{-1}	2.56×10^{-1}	1.38×10^{-1}	3.82×10^{-1}
$\|e_v\|_{L^2}$	6.79×10^{-1}	2.22	8.52×10^{-1}	2.70
$u^{\max}$	1.14×10^{-1}	3.90×10^{-1}	1.21×10^{-1}	3.28×10^{-1}
$v^{\max}$	5.12×10^{-1}	2.84	5.89×10^{-1}	2.59
$v^{\min}$	4.20×10^{-1}	2.60	5.84×10^{-1}	5.54

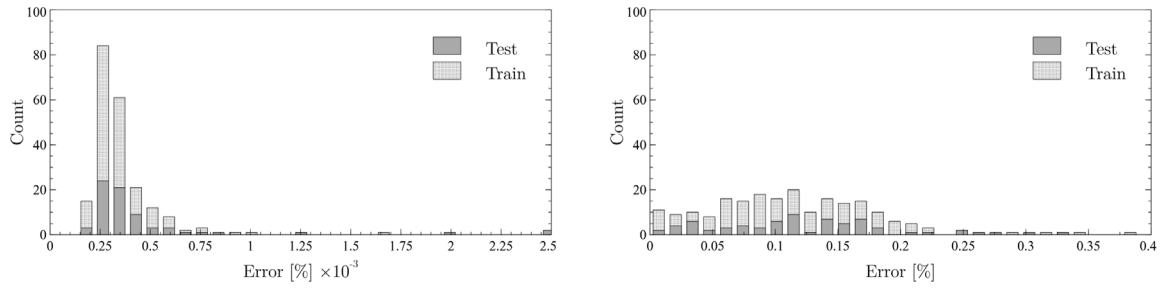


Fig. 19. Distributions of the relative percentage errors: (a) Π_h, (b) $u^{\max}$.

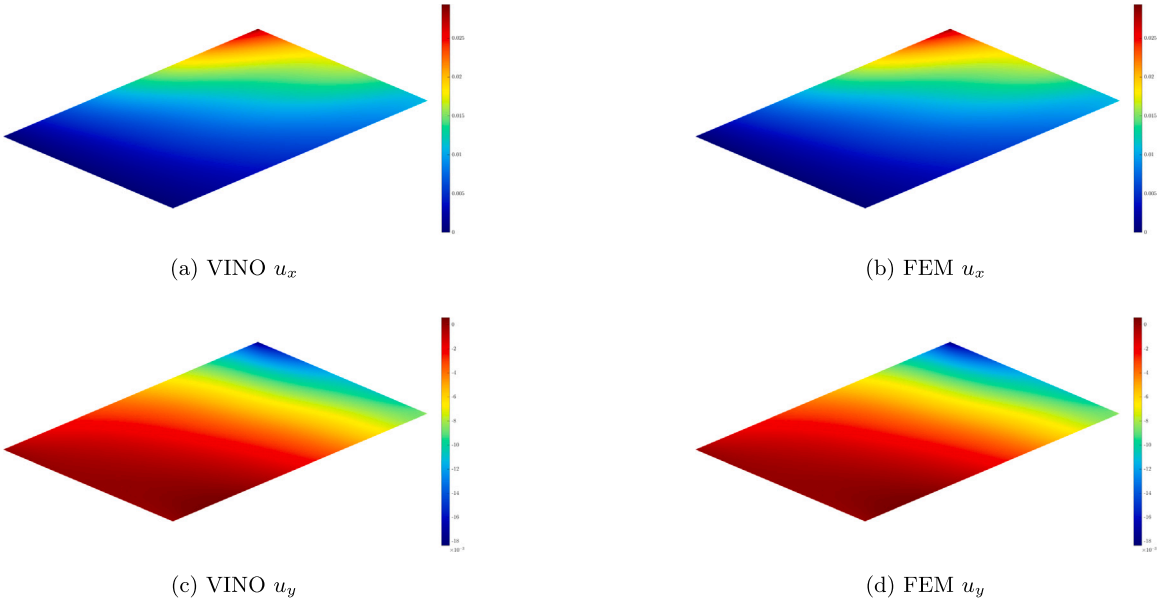


Fig. 20. Comparison of the deformed configurations obtained with VINO and FEM, colored by the displacement components u_x and u_y.

threshold and the vertical axis reporting the number of samples falling below it. The strong overlap between the two distributions confirms the robustness and generalization of the framework in this setting.

As a final comparison, the displacement fields u and v predicted by P-VINO are compared against the reference finite element solution for one of the considered load distributions (load realization 260). The contour plot is reported in 20, showing good pointwise agreement for u and v.

7. Conclusions

This work introduced P-VINO, a physics-informed neural operator framework that combines a FEM-based conforming discretization strategy with a preconditioning mechanism, enabling the accurate approximation of solution fields over non-rectangular geometries. By design, P-VINO establishes a tight link with classical finite element solutions and is equipped to handle severely ill-conditioned problems, which commonly arise in structural mechanics applications. The conforming discretization is restricted here to a preliminary and methodological investigation; the analysis of more complex geometries is among the main potentials of the proposed strategy, and future studies may build on it to progressively shift neural-operator applications toward real-life engineering configurations.

P-VINO was further assessed as a surrogate model for parametric, multi-query structural problems, where it represents a promising alternative to classical solvers. In fixed-stiffness settings, the learned operator accurately recovers the displacement field across the parametric configurations. When the stiffness matrix varies within the parametric space, however, the accuracy is limited to configurations close to the reference one employed in the single-$\mathbf{L}$ preconditioning strategy. For those configurations located farther from the reference, the preconditioner is no longer capable of correctly reshaping the energetic landscape, leading to larger errors in the displacement components associated with smaller energy contributions. For this reason, the development of alternative preconditioning strategies that remain effective in variable-stiffness parametric settings constitutes an open research direction. A future research direction concerns the development of appropriate normalization strategies: in the current setting, the heterogeneity of the total potential energy across training samples introduces an implicit bias in the optimization, as samples with larger absolute energy values contribute disproportionately to the mean loss.

The present work focused on linear static problems. The framework, however, is readily extensible to geometrically or materially nonlinear settings, which constitute a natural direction for future investigations. Finally, it is worth emphasizing that neural operators represent a rapidly evolving class of methods. Beyond the methodological contributions presented here, further advances in computational efficiency are to be expected as training algorithms and architectures continue to mature, potentially enhancing the practical competitiveness of frameworks such as P-VINO in engineering applications.

Table 11
Key FNO training settings across the numerical examples of Section 6.

Example	n_x	n_y	Modes 1	Modes 2	Width	Depth	Last channel	α	Epochs
6.1 Immersed	80	54	16	16	50	7	64	–	2000
6.1 Conforming	80	54	16	16	50	7	64	–	2000
6.2 Energy strategies	40	40	16	16	50	7	64	–	1500
6.3 VINO	40	40	16	16	110	13	80	–	1500
6.3 P-VINO	40	40	16	16	40	7	35	10^{-6}	700
6.4 P-VINO	40	40	16	16	40	7	35	10^{-6}	800
6.5 P-VINO	40	40	16	16	40	7	35	10^{-6}	2000
6.6 P-VINO	40	40	16	16	40	7	35	10^{-6}	1000

CRediT authorship contribution statement

L.M. Boccia: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **R. Vescovini:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Investigation, Data curation, Conceptualization. **N. Fantuzzi:** Writing – review & editing, Validation, Supervision, 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.

Appendix. Hyperparameters and implementation details

The framework is implemented in Python using the JAX library [47], which provides automatic differentiation and Just-In-Time (JIT) compilation; the latter is used throughout to accelerate the repeated evaluation of the loss function during training. All computations were performed on CPU (3th Gen Intel® Core™ i5-13400, 10 cores, 32 GB RAM) using JAX in float64 precision. In all numerical examples the Adam optimizer is used, and the FNO is trained with a fixed learning rate of 10^{-3} and a weight decay of 10^{-4}. In each run the initial FNO parameters are initialized with seed 0 of the PRNG key. Table 11 reports n_x, n_y the FNO modes, the width and depth, last channel projection, the α parameter and the number of epochs used in each example presented in the previous sections. In examples of Sections 6.2 and 6.4 the network parameters are the same for all the proposed strategies to ensure a fair comparison. The parameter α is chosen to be several orders of magnitude smaller than the smallest non-zero entry of K. As such, it has virtually no effect on the convergence behavior, the conditioning number or the final accuracy; its only purpose is to improve the numerical robustness of the Cholesky factorization. Normalization is applied to the FNO input: the grid coordinates are normalized to the unit square and, in Section 6.5, the lamination-parameter features are normalized channel-wise by their maximum absolute value over the training set.

Data availability

No data was used for the research described in the article.

References

[1] Goodfellow I, Bengio Y, Courville A. Deep learning. Cambridge, MA: MIT Press; 2016.
[2] Hinton GE, Osindero S, Teh Y-W. A fast learning algorithm for deep belief nets. Neural Comput 2006;18(7):1527–54.
[3] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. Neural Netw 1989;2(5):359–66.
[4] Bisagni C, Lanzi L. Post-buckling optimisation of composite stiffened panels using neural networks. Compos Struct 2002;58(2):237–47.
[5] Petrolo M, Carrera E. Selection of element-wise shell kinematics using neural networks. Comput Struct 2021;244:106425.
[6] Yamaguchi T, Okuda H. Zooming method for FEA using a neural network. Comput Struct 2021;247:106480.
[7] Liu X, Tao F, Yu W. A neural network enhanced system for learning nonlinear constitutive law and failure initialization criterion of composites using indirectly measurable data. Compos Struct 2020;252:112658.
[8] Jiang H, Nie Z, Yeo R, Farimani AB, Kara LB. StreeGAN: a generative deep learning model for two-dimensional stress distribution prediction. J Appl Mech 2021;88(5):051005.
[9] Kovachki N, Li Z, Liu B, Azizzadenesheli K, Bhattacharya K, Stuart A, Anandkumar A. Neural operator: Learning maps between function spaces with applications to pdes. J Mach Learn Res 2023;24(89):1–97.
[10] Lu L, Jin P, Pang G, Zhang Z, Karniadakis GE. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. Nat Mach Intell 2021;3(3):218–29.
[11] Li Z, Kovachki N, Azizzadenesheli K, Liu B, Bhattacharya K, Stuart A, Anandkumar A. Fourier neural operator for parametric partial differential equations. In: International conference on learning representations. 2021.
[12] Faroughi SA, Pawar NM, Fernandes C, Raissi M, Das S, Kalantari NK, Mahjour SK. Physics-guided, physics-informed, and physics-encoded neural networks and operators in scientific computing: Fluid and solid mechanics. J Comput Inf Sci Eng 2024;24(4):040802.
[13] Azizzadenesheli K, Kovachki N, Li Z, Liu-Schiaffini M, Kossaifi J, Anandkumar A. Neural operators for accelerating scientific simulations and design. Nat Rev Phys 2024;6(5):320–8.
[14] Kurth T, Subramanian S, Harrington P, Pathak J, Mardani M, Hall D, Miele A, Kashinath K, Anandkumar A. Fourcastnet: Accelerating global high-resolution weather forecasting using adaptive fourier neural operators. In: Proceedings of the platform for advanced scientific computing conference. 2023, p. 13.
[15] Kaewnuratchadasorn C, Wang J, Kim C-W. Physics-informed neural operator solver and super-resolution for solid mechanics. Computer-Aided Civ Infrastruct Eng 2024;39(22):3435–51.
[16] Jawetz CL, Alexeev A. Hydrodynamic performance estimation of undulating elastic plates using the fourier neural operator. Phys Fluids 37(10):2025.
[17] He Z. Application of neural operators to the phase field modeling of brittle fractures. In: Conference on AI, science, engineering, and technology (aIxSET). 2024, p. 311–5.
[18] Faroughi SA, Roriz AI, Fernandes C. A meta-model to predict the drag coefficient of a particle translating in viscoelastic fluids: a machine learning approach. Polymers 2022;14(3):430.
[19] Rao C, Sun H, Liu Y. Hard encoding of physics for learning spatiotemporal dynamics. 2021, arXiv preprint:2105.00557.
[20] Chen RTQ, Rubanova Y, Bettencourt J, Duvenaud DK. Neural ordinary differential equations. In: Advances in Neural Information Processing Systems. vol. 31, 2018.
[21] Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 2019;378:686–707.
[22] Haghighat E, Raissi M, Moure A, Gomez H, Juanes R. A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. Comput Methods Appl Mech Engrg 2021;379:113741.
[23] Yan CA, Vescovini R, Dozio L. A framework based on physics-informed neural networks and extreme learning for the analysis of composite structures. Comput Struct 2022;265:106761.
[24] Yan CA, Vescovini R, Fantuzzi N. A neural network-based approach for bending analysis of strain gradient nanoplates. Eng Anal Bound Elem 2023;146:517–30.
[25] Shi F, Li M, Fantuzzi N, Deng B, Shang D, Lu J, Xie H. Physics-informed neural network for multiscale mechanical behavior of microstructured composite materials as Cosserat continuum. Eng Anal Bound Elem 2026;183:106610.
[26] Kharazmi E, Zhang Z, Karniadakis GEM. hp-VPINNs: Variational physics-informed neural networks with domain decomposition. Comput Methods Appl Mech Engrg 2021;374:113547.

[27] Li W, Bazant MZ, Zhu J. A physics-guided neural network framework for elastic plates: comparison of governing equations-based and energy-based approaches. Comput Methods Appl Mech Engrg 2021;383:113933.
[28] Liu Z, Bai J, Gu Y, Xiang P. Physics-informed neural network based on layerwise theory for bending analysis of laminated plates. Eng Anal Bound Elem 2026;183:106569.
[29] Weinan E, Yu B. The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems. Commun Math Stat 2018;6(1):1–12.
[30] Guo H, Zhuang X, Rabczuk T. A deep collocation method for the bending analysis of kirchhoff plate. Comput Mater Contin 2019;59(2):433–56.
[31] Zhuang X, Guo H, Alajlan N, Zhu H, Rabczuk T. Deep autoencoder based energy method for the bending, vibration, and buckling analysis of kirchhoff plates with transfer learning. Eur J Mech A Solids 2021;87:104225.
[32] Jagtap AD, Karniadakis GE. Extended physics-informed neural networks (XPINNs): a generalized space–time domain decomposition based deep learning framework for nonlinear partial differential equations. Commun Comput Phys 2020;28(5):2002–41.
[33] Dwivedi V, Srinivasan B. Physics informed extreme learning machine (PIELM) - a rapid method for the numerical solution of partial differential equations. Neurocomputing 2020;391:96–118.
[34] Li Z, Zheng H, Kovachki N, Jin D, Chen H, Liu B, Azizzadenesheli K, Anandkumar A. Physics-informed neural operator for learning partial differential equations. ACM/IMS J Data Sci 2024;1(3):1–27.
[35] Wang S, Wang H, Perdikaris P. Learning the solution operator of parametric partial differential equations with physics-informed deeponets. Sci Adv 2021;7(40):eabi8605.
[36] Eshaghi MS, Anitescu C, Thombre M, Wang Y, Zhuang X, Rabczuk T. Variational physics-informed neural operator (vino) for solving partial differential equations. Comput Methods Appl Mech Engrg 2025;437:117785.
[37] Zhong W, Meidani H. Physics-informed geometry-aware neural operator. Comput Methods Appl Mech Engrg 2025;434:117540.
[38] Byskov E. elementary Continuum Mechanics for Everyone: With Applications to Structural Mechanics. New York: Springer; 2013.
[39] Tsai SW, Hahn HT. Introduction to composite materials. Lancaster, PA, USA: Technomic Publishing Company; 1980.
[40] Wei M, Zhang X. Super-resolution neural operator. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023, p. 18247–56.
[41] Li Z, Huang DZ, Liu B, Anandkumar A. Fourier neural operator with learned deformations for PDEs on general geometries. J Mach Learn Res 2023;24(388):1–26.
[42] JAX Developers. JAX documentation: just-in-time compilation. 2023, https://jax.readthedocs.io/en/latest/jax-101/02-jitting.html.
[43] Das R, Agarwal N, Sanghavi S, Dhillon IS. Towards quantifying the preconditioning effect of adam. 2024, arXiv preprint:2402.07114.
[44] Kingma DP, Ba J. Adam: A method for stochastic optimization. In: International conference on learning representations. 2015.
[45] Liu S, Su C, Yao J, Hao Z, Su H, Wu Y, Zhu J. Preconditioning for physics-informed neural networks. 2024, arXiv preprint:2402.00531.
[46] Griese F, Hoppe F, Rüttgers A, Knechtges P. Preconditioned fem-based neural networks for solving incompressible fluid flows and related inverse problems. J Comput Appl Math 2025;469:116663.
[47] Bradbury J, Frostig R, Hawkins P, Johnson MJ, Leary C, Maclaurin D, Necula G, Paszke A, VanderPlas J, Wanderman-Milne S, et al. Jax: Composable transformations of python+ numpy programs, version 0.3. 13. 2018, Online at http://github.com/jax-ml/jax (Accessed 31 October 2025), 2018.