communications engineering

A Nature Portfolio journal

Article

https://doi.org/10.1038/s44172-025-00549-5

NeuberNet: a neural operator solving elastic-plastic partial differential equations at V-notches from low-fidelity elastic simulations

Tommaso Grossi [1] ✉, Marco Beghini [2] & Matteo Benedetti [3]

Stress concentrations at geometric irregularities such as reentrant corners make it challenging to efficiently simulate localized plastic deformation in engineering materials. Fully nonlinear models capture these effects accurately but are computationally costly, whereas simplified elastic analyses neglect important nonlinearities. Here, we present NeuberNet, a Multi-Task Nonlinear Manifold Decoder that learns mappings between far-field displacement boundary conditions from low-fidelity elastic simulations and the corresponding high-resolution stress and strain fields derived from elastic-plastic axisymmetric solid mechanics, under assumptions of small-scale plasticity and bilinear isotropic hardening. NeuberNet serves as a data-driven implementation of the substructuring principle, designed to model complex geometries by activating plastic behavior only near stress raisers where nonlinearities arise. We provide guidelines for mesh resolution in low-fidelity simulations, demonstrate NeuberNet's ability to identify violations of the small-scale plasticity assumption, and assess its robustness to nonlinear hardening laws. We also show that NeuberNet generalizes to 3D problems with axisymmetric geometries and non-symmetric boundary conditions. Overall, NeuberNet provides a reliable and computationally efficient framework for small-scale plasticity analysis.

The legacy of the German mechanical engineer Heinz August Paul Neuber endures today, primarily due to his 1937 book *Kernspannungslehre—Grundlagen für genaue Spannungsberechnung*[1] (namely, "Notch stress theory—Fundamentals for Accurate Stress Calculation"), whose impact still permeates solid mechanics curricula and the user manuals of many finite element analysis software packages. Although many structures were being analyzed using simplified theories such as Saint-Venant's[2], photoelastic methods had revealed that stress fields diverge significantly from these simplified theories at points of domain irregularities, commonly referred to as "notches"[3], which are often *inevitable* and *dictated by design*, such as bearing shoulders and keyseats. Among the most common notches are the *reentrant corners* of the structure with a fillet radius at the apex; it is often at these points that component failure is initiated[4–12].

If the material is not brittle—i.e., the most common engineering case—and the notch effect is substantial, the linear elastic prediction of local stress often results in a peak exceeding the material's yield strength, as shown in Fig. 1. In such cases, a localized plastic region forms. Its onset complicates stress field prediction by introducing nonlinearities that are typically intractable analytically. Perhaps the most significant contribution of the German engineer is indeed the so-called Neuber's rule[13], formulated originally for a symmetric reentrant corner in a prismatic solid subjected to monotonic proportional transverse shear, although over the years it has also been adapted to other types of loading (multi-axial[14], cyclic[15]) and validated for other types of notches[16]. The rule unfolds in two phases: first, the stress peak is evaluated under the assumption of purely elastic behavior; then, this peak is adjusted to account for the redistribution of stresses induced by the onset of plasticity near the notch tip. In other words, Neuber's rule aims to establish a (somewhat primordial) mapping between the stress peaks predicted by linear elasticity and the actual peaks arising under more realistic elastic-plastic conditions. This work directly addresses Neuber's research question.

With the development of finite element analysis software, we are now able to overcome the limitations of analytical approaches by numerically simulating the equations of plasticity and obtaining the strain and stress

[1]TeCIP Institute, Scuola Superiore Sant'Anna, Pisa, Italy. [2]Dipartimento di Ingegneria Civile e Industriale, Università di Pisa, Pisa, Italy. [3]Dipartimento di Ingegneria Industriale, Università di Trento, Trento, Italy. ✉e-mail: tommaso.grossi@santannapisa.it

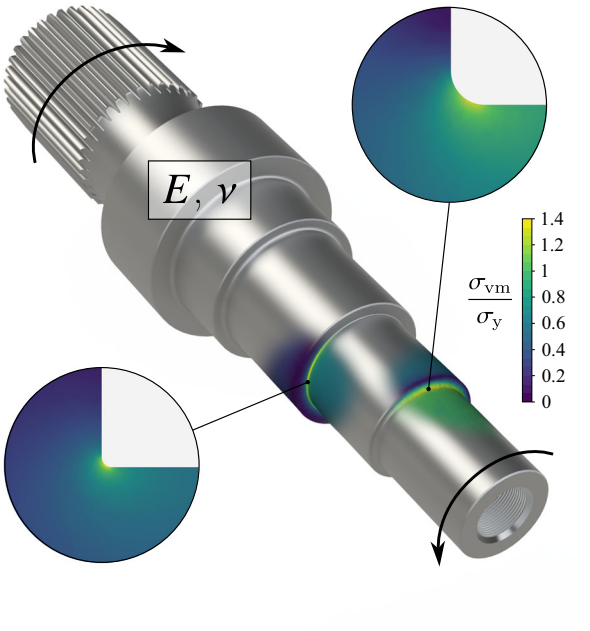

Fig. 1 | Reentrant corners in engineering domains act as local stress raisers, intensifying the stress fields obtained from linear solutions, as seen in a 3D elastic analysis of a shaft. This often leads to the onset of localized plasticity at the notch tip, which can only be assessed through costly nonlinear analyses. The situation depicted in the figure is fairly typical: while the material stays in the elastic regime throughout most of the domain, the von Mises stress σ_{vm} locally exceeds the elastic limit σ_y at some reentrant corners.

fields at any point of a component. Yet, the problem becomes computational in nature: nonlinear FE models are particularly expensive to run, since they are typically solved using Newton-Raphson schemes, where each iteration involves solving a linear model[17]. On top of that, to predict the highly localized stress fields near the notch tip, it is necessary to significantly refine the mesh in those areas, further increasing the required computational load to impractical levels, especially for industrial environments. This defines a particularly intriguing problem: while most mechanical components are designed to fulfill their structural function with a *macroscopically elastic* behavior, failure is triggered by the peak values of stress and strain fields—often exceeding the material's elastic limits—which are reached in *highly localized* regions. However, refining the FE model to account for these phenomena often leads to an unacceptable increase in the computational cost of the model. A possible solution lies in a multiscale substructuring approach[18–20], wherein one first solves the problem at the larger scale and subsequently uses that solution as the boundary condition for a more refined model at the scale of the notch. While substructuring undeniably offers computational advantages and, in effect, addresses the same problem considered in this work, it requires a nontrivial amount of manual pre-processing by the analyst, which often discourages its widespread adoption —at least in industrial practice. From a practical perspective, the analyst must build two separate models and their corresponding meshes, solve the first, and then transfer its results as boundary conditions to the second (see Fig. 2).

As alternatives to FE solvers, many recent machine learning research works have focused on supervised learning of operator mappings between *functional* spaces, such as those that act on the boundary conditions (BCs) of a partial differential equation (PDE) to yield the solution of the equation. To this aim, various architectures have been proposed. Following from the universal approximation result of Chen and Chen[21], the DeepONet

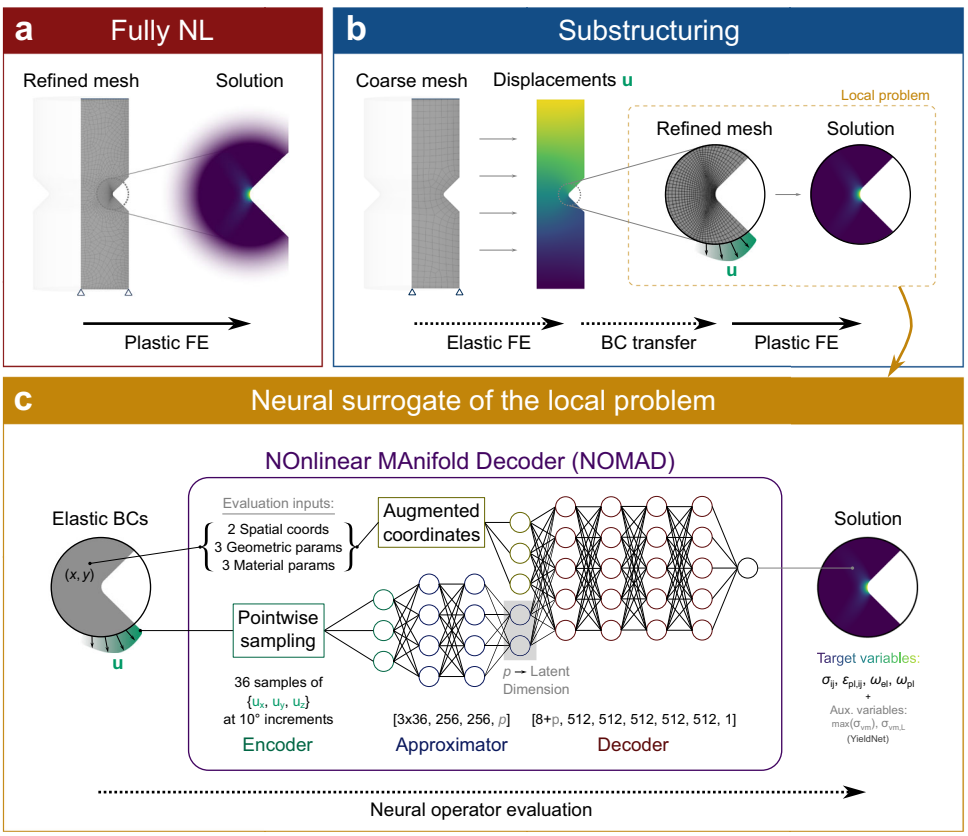

Fig. 2 | Strategies to solve the elastic-plastic PDEs in domains with reentrant corners. The bottleneck process is indicated with a solid arrow. Without loss of generality, the plastic work per unit volume is taken as target variable in plots. **a** To directly obtain the solution fields, a refined mesh is required to capture the steep stress and strain gradients at the notch tip; however, this often leads to an unacceptable increase in computational time. **b** Using a substructuring approach, one first solves a linear elastic FE analysis on a coarse mesh; then, the obtained displacements are applied as boundary conditions to a local problem (substructure) with a refined mesh, which is solved with plasticity activated. Since most elements lie precisely in the notch zone, the computational advantage may not justify the pre-processing overhead. **c** In the proposed approach, we train a Multi-Task NOMAD to act as a neural surrogate for the local problem operator, mapping elastic BCs to a desired solution variable of the elastic-plastic PDEs for notches of *variable* geometry and material. By exploiting an offline-trained model, this approach can solve elastic-plastic problems on domains of arbitrary shape, provided that the elastic limit is exceeded only in localized geometric features, as is often the case in practical applications. This latter assumption is verified by YieldNet---an auxiliary network with the same architecture---that learns the maximum von Mises stress within the domain (to confirm the actual occurrence of plasticity) as well as the stress level at which plastic strains would propagate beyond its border.

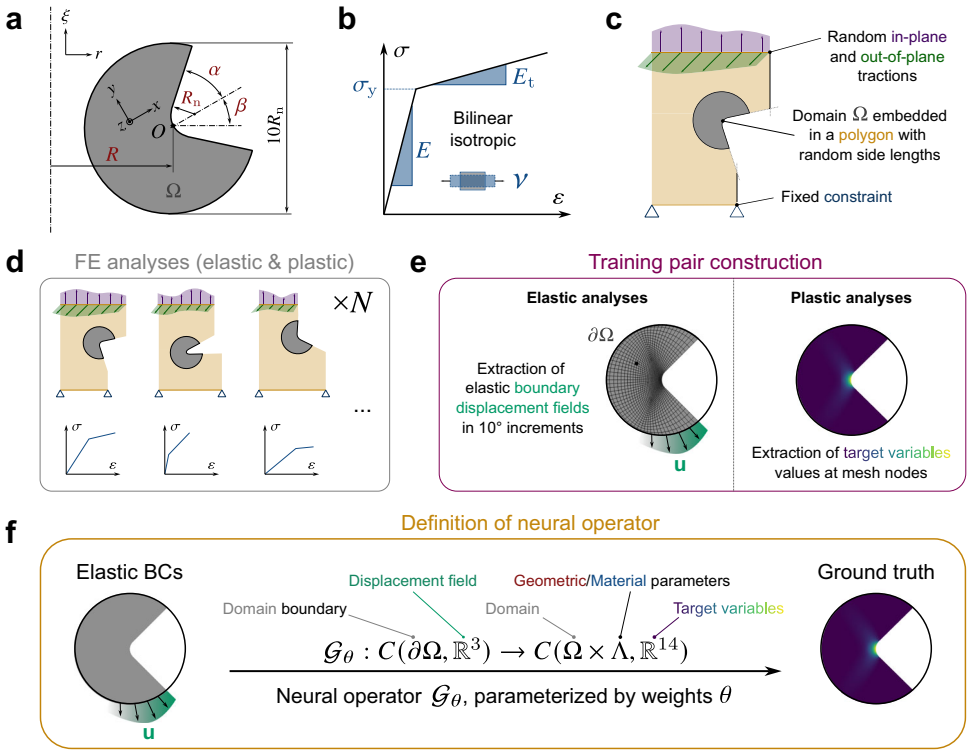

Fig. 3 | Data pipeline for training NeuberNet. **a** Geometric variables that parameterize a reentrant corner in an axisymmetric domain, and definition of the notch local reference system *Oxyz*. The diameter of the reentrant corner subdomain Ω is set to $10R_n$. **b** Material variables that parameterize the bilinear isotropic plastic constitutive law, expressed on the (σ, ε) plane. **c** Generation of realistic boundary elastic displacements on ∂Ω. We embed Ω in a polygon whose sides have random lengths, and subject it to arbitrary distributions of in-plane and out-of-plane loads, in order to generate real-world cases for the notch domain. **d** Finite Element (FE) dataset construction. For each of the $N = 10^4$ generated problems—spanning the range of geometric and constitutive parameters and various boundary conditions (BCs)—we first solve it elastically, storing the boundary displacements on ∂Ω; then, we include plasticity and record the corresponding values of the target variables within Ω. **e** Each training pair consists of the boundary displacements $\boldsymbol{u}$ obtained from linear elastic analyses and the corresponding solution fields derived from plastic material behavior. **f** The defined neural operator is aimed at learning the mapping between the two.

architecture[22] emerged, which has generated a number of variants in the following years[23–29]. DeepONets have also been interpreted as special cases of a more general Neural Operator[30–34], in which the matrix multiplications of the linear layers are replaced by kernel integral operators. When the latter are replaced by convolution operators and evaluated in Fourier space, a Fourier Neural Operator is obtained[35–37]; when the convolution is evaluated through Laplace transforms, a Laplace Neural Operator is attained[38]. Architectures based on the attention mechanisms have also been proposed[39].

As noted by Seidman et al.[40], many of these approaches can be framed as approximating an operator between functional spaces through the composition of three operators: an encoder $\mathcal{E}$, which maps the input functions to a finite-dimensional space; an approximator $\mathcal{A}$, which acts as a projection of the operator to be learned over the space of operators between finite-dimensional manifolds and returns a latent representation of the outputs; and a decoder $\mathcal{D}$, which reconstructs the output functions from their latent representation. An implicit assumption of this architecture is that the set of target functions can be well approximated by a low-dimensional submanifold of the output functional space, known in the literature as the *Operator Learning Manifold Hypothesis*. As a result, the finite-dimensional latent representations of the operator to be learned show good convergence properties with respect to the latent dimension. Restricting the set of parameterizable submanifolds to linear ones, as DeepONets do, can be disadvantageous, as it may lead to inefficient representations of nonlinear manifolds, necessitating an increase in the latent dimension to compensate. To address this limitation, NOnlinear MAnifold Decoders (NOMADs) have been proposed[40]. Following the terminology proposed by Kumar et al.[24] for DeepONets, when a Neural Operator learns not only the solution operator of a single PDE but rather that of a *class* of PDEs—such as those corresponding to different domains or varying values of parameters that influence the PDE itself—the Neural Operator is referred to as *Multi-Task*. A more detailed overview of these architectures can be found in the Methods.

Machine learning already plays an important role in building approximations in the context of solid mechanics[41–50]. The recent advancements in operator learning architectures might lead us to address the computational complexity of elastic-plastic FE analyses by replacing them with neural operator approximations of the mapping between the applied boundary conditions and the target fields. Actually, training a *general* surrogate operator for elastic-plastic PDEs on domains of *arbitrary shape* remains an open problem to this day. The vast variety of geometric domains on which solid mechanics equations are solved in engineering practice threatens the practicality of the operator learning approach, as it requires a substantial initial training cost to achieve the desired reductions in simulation time during inference. In fact, restricting the operator's training to a specific geometry-loadings pair (perhaps with some variations[28,51–53]) has proven successful but greatly limits its applicability, often making the initial investment in training unjustified during inference. This trade-off turns out to be unfavorable even for physics-informed neural networks[54]. On the other hand, transfer learning of the operator across different geometries remains an open problem with no turnkey solutions; pixel-based approaches have been attempted[55].

At least in the case of 2D geometries—such as those analyzed by Neuber—reentrant corner notches (also known as *V-notches*) possess two important characteristics: they are easily parameterizable[56] (see Fig. 3a), and their geometry influences the solution at a smaller dimensional scale compared to the general solution over the entire geometric domain. Inspired by Neuber's work and by stress intensity factors in fracture mechanics, we propose tackling elastic-plastic PDEs with a substructuring approach, organized into two phases (see Fig. 2). In the first phase, the problem is solved using the equations of linear elastic solid mechanics on a coarse mesh, obtaining the solution in terms of displacement fields. In the second phase, we leverage a Multi-Task NOMAD-based neural approximation of the operator that maps the elastic displacement fields on the boundary of a circular domain centered on a notch to the solutions of the elastic-plastic PDE therein under the only assumptions of *small-scale plasticity* and *bilinear isotropic hardening*, and we apply it to the notches present in the simulated structure. As clarified in the Methods, we translate the condition of small-scale plasticity by requiring that the plastic zone does not extend beyond a ball centered at the notch tip with a diameter equal to 10 times the

notch radius. This entails the additional assumption that such a ball indeed contains material, thereby excluding thin-walled structures, where "thinness" is always to be understood relative to the notch tip radius.

Intuitively, we leverage the speed of linear analyses to infer the loading conditions of notches embedded within arbitrarily complex domains from their elastic displacement fields. Thanks to the strong ellipticity of the elastic problem, these fields are particularly smooth and can be obtained with high accuracy even with coarse discretizations. We then use the displacement fields obtained from a coarse elastic model as inputs to solve the more complex elastic-plastic PDEs on localized geometric features, which, in contrast, are easily parameterizable domains—and the most critical zones of the structure in which they are embedded. A *single offline training* is performed (see Fig. 3); then, the resulting neural operator can be applied to all geometries featuring reentrant corners as well as to a vast class of materials. Since the inference time for a neural network is essentially negligible compared to FE analyses, we argue that the marginal cost of this approach is practically zero, as a linear FE analysis would have been performed anyway.

The idea of coupling a numerical solver with a machine learning model at different spatial scales is not new in the literature[57,58]. Krokos et al.[57] propose replacing traditional local microscale solutions with an encoder-decoder Convolutional Neural Network (CNN) to generate fine-scale stress corrections for coarse predictions around unresolved microscale features such as voids. However, they report substantial generalization errors for macroscale geometries and microscale material distributions not included in the training dataset. The method relies on a pixelization procedure that transforms snapshots of the macroscopic fields into images processable by the CNN, which distinguishes it from our approach that instead operates between functional spaces. On the other hand, the application by Yin et al.[58] precisely involves the coupling of FEM models with DeepONets, although it requires an *iterative* update of the solution values at the interface between the two methods, using them alternately as Dirichlet-to-Neumann maps. In contrast, we train the neural operator to learn a *direct mapping* between the cheap FE solution and the desired fields, which are thus obtained in a single step. This also distinguishes our model from other machine learning approaches to solving the plasticity PDEs, in which attempts have been made to capture the temporal evolution of the stress fields using sequential learning models such as recurrent neural networks[43,52,53,55].

Our work is closely related to that of Palchoudhary et al.[59], in its aim of obtaining solutions to the elastic-plastic mechanical problem using the elastostatic solution as input, following an approach inspired by Neuber. In their study, assuming local proportionality between the deviatoric stress and strain tensors, they define a plastic corrector which, through a time integration—performed at each mesh quadrature point—implements what is, in our view, the most comprehensive version of Neuber's rule currently available, valid both for monotonic and cyclic loadings. Despite the undeniable elegance and generality of the proposed approach, the reported relative errors remain substantial in regions experiencing substantial plasticity. Given the non-negligible computational time required to obtain the solution, they also propose an efficient surrogate modeling of the plastic corrector using Gaussian Process Regression. Additionally, they explore the performance of a CNN as an alternative to the proposed algorithm, albeit with high generalization errors. While sharing the same objectives, our approach achieves a solution that is orders of magnitude more accurate, albeit for a less general problem—namely, reentrant corner notches on 2D axisymmetric domains.

We allow the notch geometry and the material constitutive model to vary freely; we parameterize them—according to Fig. 3a, b—and feed them as additional inputs *to the decoder network only*. This contrasts with other approaches in the literature addressing the problem of Multi-Task learning[24], where the parameters are stacked with the functional inputs of the problem—in this case, the elastic boundary displacements—and fed to the approximator network. Some theoretical implications of this choice are reported in the Methods.

We argue that, for the practical solution of this problem, employing a *random sampling* of boundary elastic displacements modeled by Gaussian

Table 1 | Range of normalized geometric and constitutive parameters employed in the FE database

	Geometric parameters			Material parameters		
	R/R_n	α	β	ν	σ_y/E	E_t/E
max	100	75°	50°	0.45	10^{-2}	10^{-1}
min	10	0°	−50°	0.05	10^{-3}	10^{-3}

Processes to learn the mapping defined in Fig. 2c—as is common in the operator learning literature[60]—is an inefficient use of computational resources. This is because, for a notch embedded in an encompassing domain—as would be the case in a mechanical component—the set of admissible boundary conditions at the border of the considered domain is, in fact, an *extremely small subset* of the functional space to which they formally belong; this fact should be employed as an inductive bias. In the limiting case of sharp corners, the BCs must converge to the three eigenmodes of Williams' well-known solution[61]. Indeed, we show that for notches embedded in larger domains, the possible boundary elastic displacements lie on a low-dimensional manifold.

Given the industrial importance of axially symmetric components—such as shafts, axles, rods, and rotors—we showcase our approach by training the neural operator on the elastic-plastic PDEs in their axisymmetric formulation. In honor of Neuber's pioneering work on elastic-to-plastic mappings, we name the neural operator `NeuberNet`.

Results

NeuberNet training and error metrics

We train NeuberNet over $N_{\text{train}} = 8000$ combinations of notch geometry, constitutive model, and applied loads, while we test it over $N_{\text{test}} = 2000$ unseen combinations. Given the input space dimensionality (an $\mathbb{R}^3$-valued functional input plus 3 geometric and 3 constitutive dimensionless parameters obtained through Buckingham's π-theorem[62]), the training dataset is arguably quite sparse. The training FE analyses are purposely designed to exhibit small-scale plastic behavior. As target variables, we consider the six components of the stress tensor normalized by the yield stress σ_y, the six components of the plastic strain tensor, and two energy-related scalar variables—elastic strain energy density and plastic work per unit volume—both normalized by σ_y. The training details are available in the Methods. The variation ranges of the scalar parameters are provided in Table 1.

For each unseen analysis i in the test dataset and for each target variable, we compare across the entire domain Ω the predictions of NeuberNet with the ground truth values obtained from the nonlinear FE analyses. To quantify the accuracy of the proposed method, we define the following error metrics for each test case i:

- $\texttt{RMSE}_i$. Root-mean squared (RMS) error over Ω; simply a measure of the overall error on test case i, as commonly employed in the operator learning literature, evaluated as an L^2 norm on the domain Ω.
- $\texttt{Err(Peak)}_i$. In the context of structural integrity, the RMS error over Ω is often not a sufficient indicator of a method's reliability. Consider, for instance, an approximator that delivers highly accurate predictions over most of the domain but fails to reproduce the stress or strain peak, which in our case is often highly localized. When peak values governs structural strength, such an approximator would be potentially dangerous. Hence, for each analysis i, we report the absolute error in predicting the peak value within the domain.
- $\texttt{MaxErr}_i$. This metric follows the same rationale as above but is defined as the maximum error over the entire domain, not necessarily coinciding with the peak value. We report this metric as well to provide a sense of the worst-case error committed by NeuberNet.

Additionally, we define the following *global* metrics to quantify the prediction error across the entire test dataset:

- `RMSE`. Simply the global RMS error over the entire dataset, i.e., the square root of the classical MSE loss.

- `RMSE(Peak)`. Computed by taking the distribution of absolute errors on the peak values $\texttt{Err(Peak)}_i$ across all test cases and evaluating its RMS value.
- `MaxErr(Peak)`. Computed by taking the distribution of absolute errors on the peak values $\texttt{Err(Peak)}_i$ across all test cases and taking the *maximum* value.

Effect of latent dimension

We train NeuberNet for increasing values of the latent dimension p and evaluate the corresponding RMSE. We observe that a value of $p = 4$ is already sufficient to achieve excellent performance on the test dataset, comparable to higher values of p (see Fig. 4c). We take this observation as evidence that the space of boundary elastic displacements $\boldsymbol{u}$—corresponding to *various* geometries and materials—is in fact of very low dimensionality. We further investigate this hypothesis by performing a PCA of this space, the results of which are reported in Fig. 4a, b. Indeed, the first four Principal Components already account for more than 95% of the variance in the input dataset. In the remainder of the text, we set $p = 64$, since this value minimizes the test RMSE across all target variables.

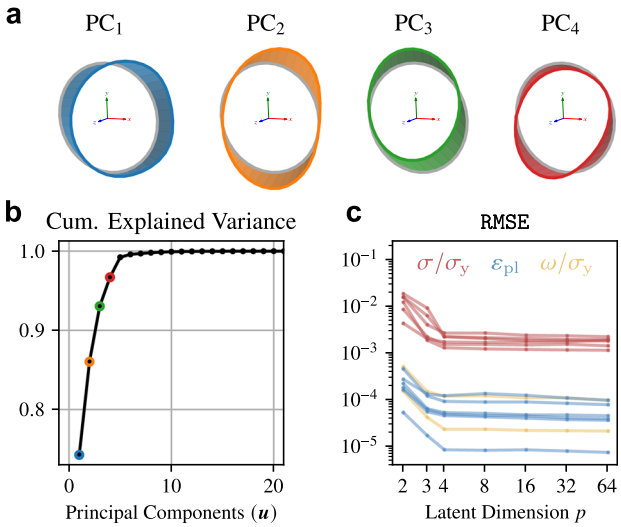

Fig. 4 | Principal Components of boundary displacements and latent-dimension effects in NeuberNet. a Representation of the first four Principal Components of the boundary elastic displacements $\boldsymbol{u}$ in the local reference frame shown in Fig. 3a. Although a clear separation—such as that found in K_I, K_{II} and K_{III} modes in fracture mechanics[76]—is not present, it is still possible to discern deformation modes associated with circumferential extensions, in-plane elongations, and out-of-plane torsions. **b** Cumulative explained variance plot as a function of the number of Principal Components included. **c** Root-mean-squared error on the test dataset as a function of the latent dimension p employed in the architecture of NeuberNet. The trends of the 14 target variables are overlaid on the plot.

NeuberNet predictive performance

In Fig. 5, we report the distribution of the selected error metrics across the test dataset. Figure 6 presents the worst cases in terms of $\texttt{MaxErr}_i$ for the six normalized stress components, expressed in the local reference system *Oxyz* shown in Fig. 3a. Supplementary Figs. S1–S5 provide the worst-case results for the other target variables, as well as the 50$^{\text{th}}$ percentile cases for all NeuberNet outputs.

We also train an auxiliary machine learning model, called `YieldNet`, to aid NeuberNet during inference. YieldNet is a neural operator that takes NeuberNet's functional input and, given the geometric and material parameters, predicts the maximum value of the stress tensor's J_2 invariant over Ω as a scalar output—i.e., the maximum equivalent stress $\sigma^{\Omega}_{\text{vm}}$—and the limit stress $\sigma^{\Omega}_{\text{vm,L}}$ at which the small-scale plasticity hypothesis is violated. This allows us to determine in advance whether the solution is entirely within the elastic range (allowing us to set all plastic variables to zero) or if the loading level is too high to satisfy the small-scale plasticity condition. In the latter case, we notify the user of the need for a full-scale nonlinear model. Details of YieldNet's training procedure and its achieved performance are provided in Supplementary Section S4.

Performance on low-fidelity data

The errors achieved on the test dataset actually represent a theoretical lower bound on those attainable in practice, since the BC displacements $\boldsymbol{u}$ were obtained by solving the FE model using the same fine mesh employed to generate the ground truths for both the target variables and the input displacements themselves. In contrast, the computational advantage of the proposed approach would lie precisely in its ability to exploit the high regularity of the elastic displacements—a characteristic that stems from the ellipticity of the governing PDEs—to compute the BCs on much coarser meshes—see Fig. 7a.

A research question arises: if the elastic analyses used to compute the inputs to NeuberNet were performed on a coarse mesh, how much would the prediction accuracy degrade? This question holds substantial practical relevance, as the required FE analyses are not only linear but also may not require a refined mesh near the notch.

To investigate this, we repeated all analyses in the test dataset using free unmapped meshing with the default ANSYS mesher, varying the characteristic element size l_e from 1 to 10 times the notch radius R_n. The results are promising: even with very coarse meshes, the predictions maintain an acceptable level of accuracy. As shown in Fig. 7b, meshes with elements up to 5 times the notch root radius can be used—essentially reducing the notch to little more than a corner defined by two solid elements—while maintaining

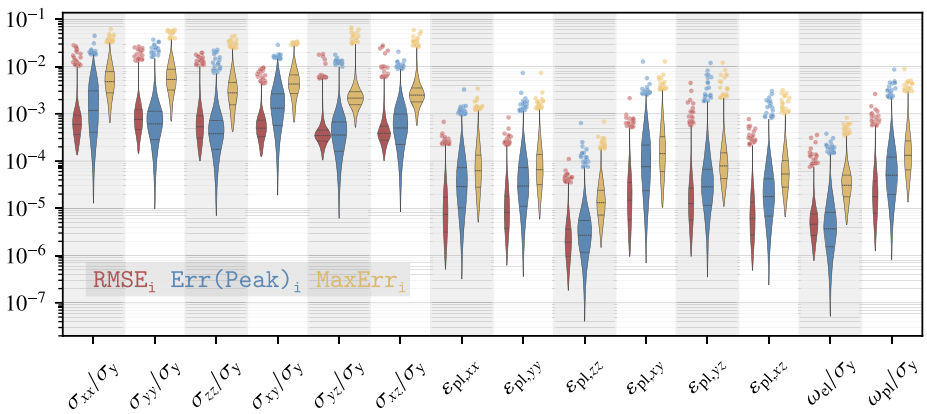

Fig. 5 | Distribution of error statistics across the entire test database for all target variables, expressed in the local reference system *Oxyz* of Fig. 3a. For visual clarity, the lowest 10 percentiles (closest to zero) have been removed from the graph. For each target variable, the 20 most unfavorable cases are shown as data points. A point corresponds to a solution of the elastic-plastic PDEs on Ω, as in typical applications. The chosen metrics are useful for assessing the practical relevance of the results; for example, in terms of peak stress prediction, the worst performance is observed in an analysis where the error on σ_{xx} reaches 4% of the yield stress σ_y, compared to an average error of approximately 0.1% of σ_y.

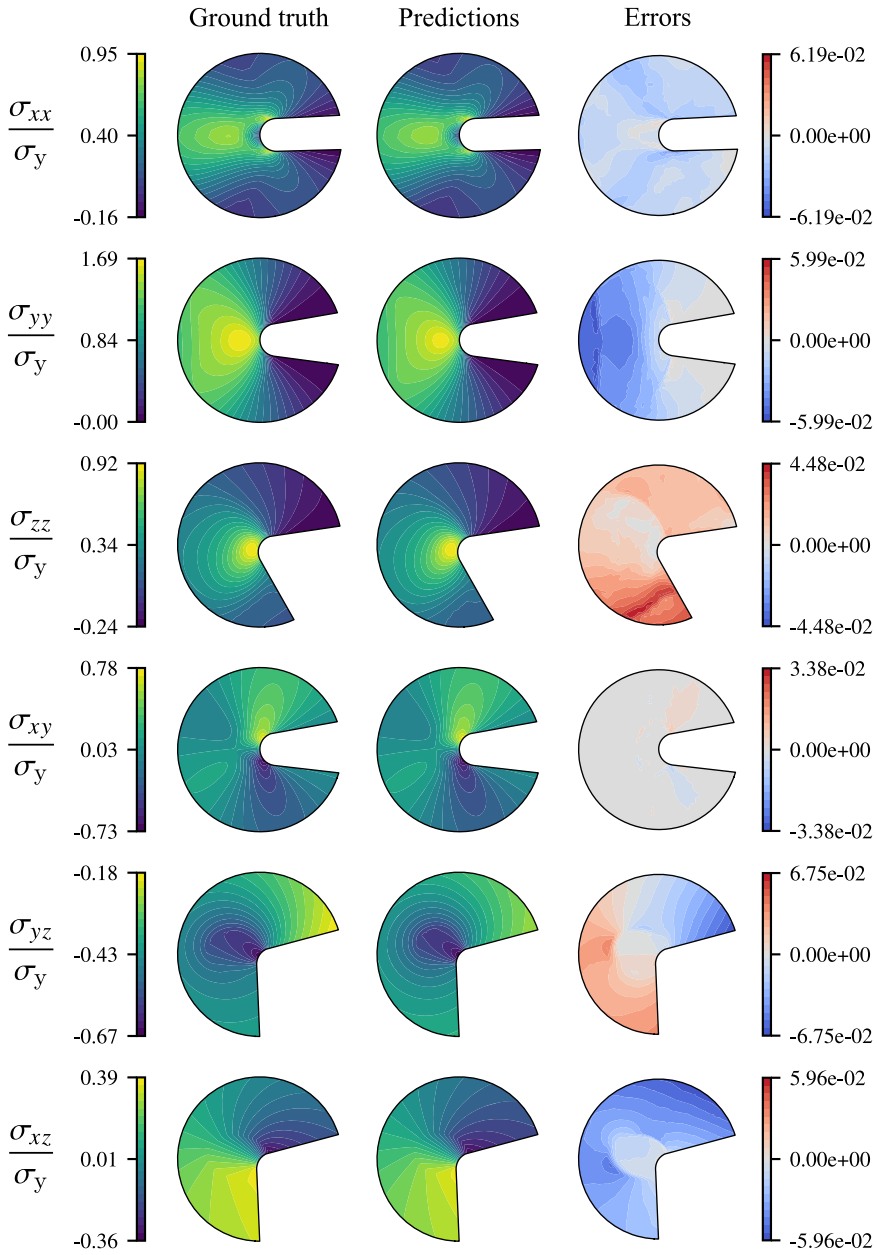

Fig. 6 | Maximum absolute error of normalized stress components over Ω. *Worst-case* performance on the test dataset in terms of maximum absolute error $\texttt{MaxErr}_i$ over the domain Ω for the six normalized stress components, expressed in the local reference system *Oxyz* of Fig. 3a.

an RMS error in predicting the peak stresses below 1% of the material's yield strength. Despite a clear trend, the degradation in performance remains remarkably limited compared to the corresponding computational savings.

Extension to other hardening laws

The adoption of a bilinear plasticity law represents a rather strong assumption regarding the material's constitutive behavior. This naturally raises the research question: what if the material does not follow a bilinear hardening law? We argue that the mechanical response is primarily governed by the material's hardening behavior near the onset of yielding—specifically, by the first-order Taylor expansion of the constitutive law at the yield point. This is justified by the fact that, under typical conditions of small-scale plasticity, the plastic strains developing at the notch tip remain limited in magnitude.

We generate 2000 test FE analyses using a (nonlinear) Voce hardening law[63], with random parameters. Then, we predict the target variables with NeuberNet, using as input to the neural network the bilinear law that approximates the actual law at yielding, through a first-order Taylor approximation. The corresponding RMSE performance is reported in Fig. 8, while the complete distribution of error statistics—analogous to Fig. 5 for the baseline case—is provided in Supplementary Section S7. As expected, the predictive performance degrades, but the errors are still likely to be lower than the intrinsic variability of the material properties themselves in practical applications.

Extension to 3D problems

In solid mechanics, many axisymmetric geometries are subjected to loading conditions that break axial symmetry. A key example of this phenomenon is bending loads in shafts. We observe that NeuberNet is capable of performing zero-shot inference for such loading conditions, owing to the inherent regularity of the solutions to the elastic problem.

The stress fields induced by bending loads vary on a length scale comparable to the diameter of the section, which is substantially larger than the scale at which notch effects and plasticity phenomena occur. Intuitively, at the point experiencing the highest nominal bending stress (in algebraic sense), a notch is subject to locally tensile boundary conditions. The mild

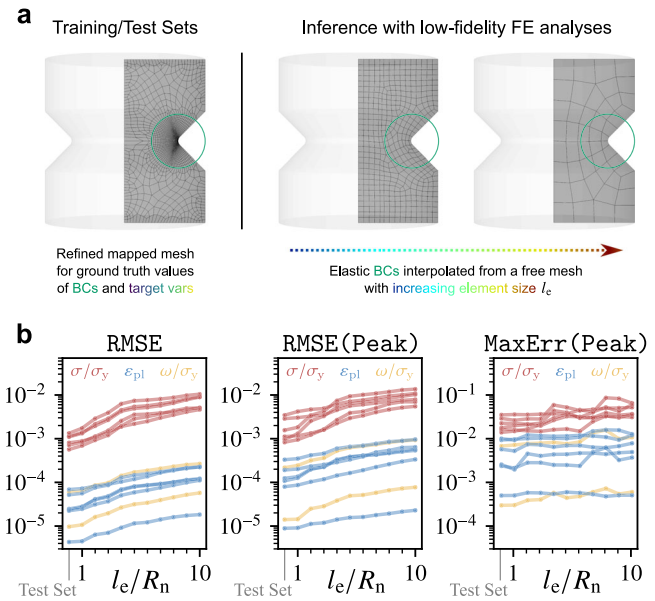

Fig. 7 | Effect of mesh resolution on NeuberNet prediction accuracy. a In order to serve as ground truth for quantifying the performance of NeuberNet, the test dataset is constructed by meshing Ω with a refined mapped mesh, validated for the convergence of both the BC displacements and the target variables. We then ask: in a practical inference scenario, what would happen if the BC displacements were obtained using a relatively coarse mesh? **b** Variation of error statistics with increasing normalized element size l_e/R_n in the notch region. Recall that the diameter of the reentrant corner subdomain Ω is equal to $10R_n$. The trends of the 14 target variables are overlaid on each plot.

variation of these conditions relative to the notch scale causes the solutions to closely resemble those that would result from a similarly distributed load applied uniformly across the entire section.

The only precaution to consider is the removal of the rigid-body motions of the notch in the radial direction—i.e., orthogonal to the axis of symmetry. In axisymmetric elasticity, a radial displacement u_r is uniquely associated with circumferential strains, since $\varepsilon_{\theta\theta} = u_r/r$; however, this no longer holds when axial symmetry is broken. Therefore, we propose eliminating the mentioned rigid-body motions by centering the radial displacement component of $\boldsymbol{u}$ to the mean value of $r\varepsilon_{\theta\theta}$, which can be directly obtained from the linear FE analysis.

We design a 3D geometry of a shaft, apply a combined bending and torsional load, solve the corresponding FE model under linear elastic conditions, and apply NeuberNet to the notch identified by the intersection of the geometry with a plane passing through the axis of symmetry and orthogonal to the bending moment vector—i.e., the region where the maximum equivalent stress is expected. The results in terms of axial stress components are presented in Fig. 9, demonstrating that NeuberNet can also perform zero-shot inference on problems where the axial symmetry is broken. While NeuberNet allows one to obtain the elastic-plastic solution fields in just a few minutes using a linear FE analysis, a corresponding 3D nonlinear FE solution of comparable accuracy would require several *hours* of computation. We report additional results for ten further 3D problems in Supplementary Section S5.

Discussion

We achieve strong predictive performance over a variety of geometries, materials, and loading conditions. The maximum stress errors observed across the entire test dataset are likely smaller than the uncertainty in the yield stress itself[64], rendering any further accuracy practically insignificant.

Although specific benchmarks in solid mechanics are still lacking[65], we argue that the observed computational advantage is fundamentally general. Coarse linear elastic simulations are orders of magnitude cheaper than nonlinear elastic-plastic analyses with refined meshes, even when compared to nonlinear analyses of comparable accuracy to NeuberNet, i.e., those employing slightly coarser meshes[66]. At a minimum, for the same mesh, a nonlinear analysis requires as much time as the linear analysis multiplied by the number of Newton-Raphson iterations, which can easily exceed 10. As a result, there is virtually no trade-off between the neural operator and a standard solver, since no additional training is required for inference.

From the PCA performed on the database of all boundary input displacements on ∂Ω—thus including simulations with different geometries and materials—we observe, interestingly though not unexpectedly, that its dimensionality is markedly reduced, even when projecting onto a linear subspace by retaining only the most significant Principal Components. This observation raises the question of whether the solution space could likewise be projected onto a finite-dimensional space through a Model Order Reduction technique, such as Proper Orthogonal Decomposition[67], and whether a surrogate could then be defined between the two linearly projected input and output spaces in place of a neural operator like NeuberNet. Nonetheless, as we show in Supplementary Section S6, the Kolmogorov *n*-width[68,69] of the output manifold exhibits a slow decay rate. This property prevents achieving errors comparable to those obtained with NeuberNet's nonlinear decoder when using a low-dimensional linear latent space representation.

The demonstrated accuracy and efficiency of NeuberNet pave the way for its integration into CAD and FE software, potentially enabling seamless and rapid prediction of localized plasticity effects within standard engineering design workflows.

Limitations

The restriction to axisymmetric problems is arguably the most substantial limitation of this work's applicability. This choice was primarily motivated by three factors: 1) to minimize the dataset generation time while running FE models on a consumer workstation, enabling efficient exploratory analyses; 2) to represent a substantial number of practical cases of industrial interest; 3) to leverage a vast database of experiments on axisymmetric specimens[70], allowing future applications of NeuberNet to the study of material strength.

Additionally, the assumption of small-scale plasticity—although often verified in practice—constitutes a substantial limitation in the generality of the approach when proposed as an alternative to fully nonlinear finite element analyses. As anticipated in the Introduction, this also restricts the minimum thickness of the component under inference, since the circular domain Ω on which plasticity occurs must be entirely contained within the component; this limitation is shared with Neuber's rule in its original formulation.

NeuberNet is trained on multiaxial loadings that are, however, both proportional and monotonic; it does not incorporate explicit integration of a full time history and, as such, cannot handle non-proportional loadings. Similar to Neuber's rule, our approach could nevertheless be easily extended to proportional cyclic loadings[15]. Specifically, instead of defining a mapping between the elastic boundary conditions and the *static* stress and strain fields, one would define a new mapping where the targets are the *stabilized amplitudes* of the oscillatory stress and strain fields. This problem is formally equivalent to the one proposed here, and we intend to explore it in future work.

We have shown how NeuberNet can already be applied to axisymmetric geometries with non-axisymmetric loading conditions. Extending this analysis to solve the elastic-plastic PDE on 2D domains under plane stress or plane strain would be a relatively straightforward extension, for which we foresee no obstacles. However, extending the work to general 3D notches would entail a considerable increase in the cost of generating the training dataset and a reconsideration of the parameterization of reentrant corners—for instance, deciding whether to represent polyhedral angles or just dihedral ones—which are not so easily parameterizable in the 3D case.

Methods

Theoretical framework

Let $\mathcal{X} \subset \mathbb{R}^n$ be a compact set. We define $C(\mathcal{X}, \mathbb{R}^d)$ as the space of continuous functions from $\mathcal{X}$ to $\mathbb{R}^d$; $L^2(\mathcal{X}, \mathbb{R}^d)$ as the space of square-integrable functions from $\mathcal{X}$ to $\mathbb{R}^d$; $H^1(\mathcal{X}, \mathbb{R}^d)$ as the space of functions from $\mathcal{X}$ to $\mathbb{R}^d$ that are square-integrable up to their first weak derivative.

Let $\Sigma \subset \mathbb{R}^2$ be a compact set, that generates a geometry of interest by revolving it about an axis on the plane. We first use a finite element solver to get a weak solution $\boldsymbol{u}_{\mathrm{el}} \in H^1(\Sigma, \mathbb{R}^3)$ of the linear elasticity PDEs at any $\boldsymbol{x} \in \Sigma$:

$$\begin{aligned} \nabla \cdot \boldsymbol{\sigma} &= 0 \\ \boldsymbol{\varepsilon} &= \frac{1}{2}\left(\nabla_{\boldsymbol{x}}\boldsymbol{u} + (\nabla_{\boldsymbol{x}}\boldsymbol{u})^T\right) \\ \boldsymbol{\sigma} &= \frac{E}{1+\nu}\left(\boldsymbol{\varepsilon} + \frac{\nu}{1-2\nu}\,\mathrm{tr}\,(\boldsymbol{\varepsilon})\boldsymbol{I}\right) \end{aligned} \tag{1}$$

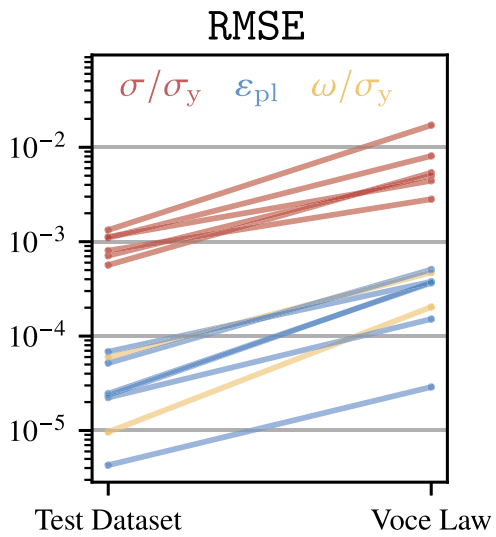

Fig. 8 | Degradation of NeuberNet's predictive performance resulting from the approximation of a nonlinear hardening law by a bilinear law, obtained by constructing a first-order Taylor approximation at the yield point. We build a finite element dataset comprising 2000 analyses with a material following a Voce hardening law and attempt to predict the corresponding target variables. Indeed, the accuracy degrades, but is still likely to be smaller than the inherent uncertainty in the material properties themselves.

with axisymmetric/torsional boundary conditions—i.e., out-of-plane displacements are included in the solution fields. We denote with E and ν, respectively, the Young modulus and Poisson ratio of the material.

Let $\Omega \subset \Sigma$ be a subdomain that includes a reentrant corner and such that $\mathrm{diam}\,\Omega \ll \mathrm{diam}\,\Sigma$. Without loss of generality, we assume that Ω is the intersection of Γ with a ball centered at the reentrant corner tip, parameterized by a notch radius R_{n}, a notch radial position R, a half-opening angle α, and a bisector orientation β, as in Fig. 3a. Assuming bilinear isotropic hardening, we introduce the additional constitutive parameters σ_{y} as the initial yield stress and E_{t} as the plastic tangent modulus. Let $\Lambda \subset \mathbb{R}^8$ be a compact set that contains the range of variation for the geometric and constitutive parameters. Our goal is to learn a continuous operator $\mathcal{G}_\theta : C(\partial\Omega, \mathbb{R}^3) \to C(\Omega \times \Lambda, \mathbb{R}^{14})$, parameterized by θ, to approximate the functional mapping that sends $\boldsymbol{u}_{\mathrm{el}}(\partial\Omega)$ to the solution fields of interest of the J_2-plasticity PDEs with bilinear isotropic hardening:

$$\begin{aligned} \nabla \cdot \boldsymbol{\sigma} &= 0 \\ \boldsymbol{\varepsilon} &= \frac{1}{2}\left(\nabla_{\boldsymbol{x}}\boldsymbol{u} + (\nabla_{\boldsymbol{x}}\boldsymbol{u})^T\right) \\ \boldsymbol{\sigma} &= \frac{E}{1+\nu}\left((\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{\mathrm{p}}) + \frac{\nu}{1-2\nu}\,\mathrm{tr}\left(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{\mathrm{p}}\right)\boldsymbol{I}\right) \\ \dot{\boldsymbol{\varepsilon}}_{\mathrm{p}} &= \lambda \nabla_{\boldsymbol{\sigma}} f(\boldsymbol{\sigma}) \\ f(\boldsymbol{\sigma}) &= \sqrt{\frac{3}{2}}\left\|\boldsymbol{\sigma} - \frac{1}{3}\mathrm{tr}(\boldsymbol{\sigma})\boldsymbol{I}\right\|_F - \sigma_{\mathrm{f}} \\ \sigma_{\mathrm{f}} &= \sigma_{\mathrm{y}} + E_{\mathrm{t}} p \\ \dot{p} &= \sqrt{\frac{2}{3}}\left\|\dot{\boldsymbol{\varepsilon}}_{\mathrm{p}}\right\|_F \end{aligned} \tag{2}$$

with usual conditions $\lambda \geq 0$, $f(\boldsymbol{\sigma}) \leq 0$ and $\lambda f(\boldsymbol{\sigma}) = 0$. The plastic flow stress is denoted with σ_{f}. Being useful technical quantities for material strength evaluations[11], we also define the elastic strain energy density ω_{el} and the plastic work per unit volume ω_{pl}:

$$\omega_{\mathrm{el}} \triangleq \frac{1}{2}\boldsymbol{\sigma} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{\mathrm{p}}) \tag{3}$$

$$\omega_{\mathrm{pl}} \triangleq \int \boldsymbol{\sigma} : \mathrm{d}\boldsymbol{\varepsilon}_{\mathrm{p}} \tag{4}$$

and gather them into the $\mathbb{R}^2$-valued field $\boldsymbol{\omega}$. A solution to (2) is represented by a stacking of $\boldsymbol{\sigma}$ (6 components), $\boldsymbol{\varepsilon}_{\mathrm{p}}$ (6 components), and $\boldsymbol{\omega}$ (2 components), defined as $\boldsymbol{g}$; hence, each solution $\boldsymbol{g}$ is a $\mathbb{R}^{14}$-valued field.

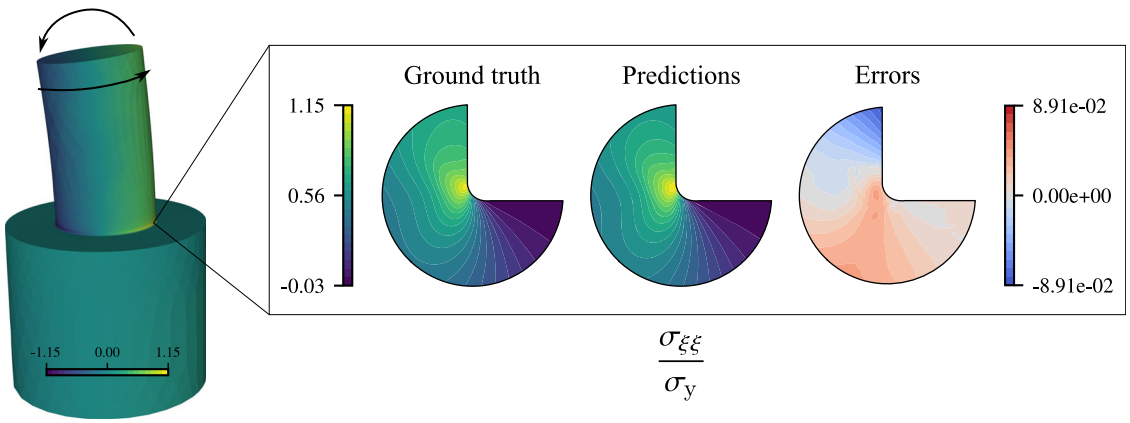

Fig. 9 | Zero-shot inference on a 3D axisymmetric geometry with a non-symmetric bending load applied, superimposed on a torsional load. We solve the problem under linear elasticity, and we get the boundary displacements, which are fed to NeuberNet. In parallel, we solve the nonlinear elastic-plastic FE problem to get the ground truth values. We report inference performance for the normalized axial stress component, obtained by rotating the stress tensors into the global cylindrical system. The error in predicting the peak stress value is less than 3% of the yield stress.

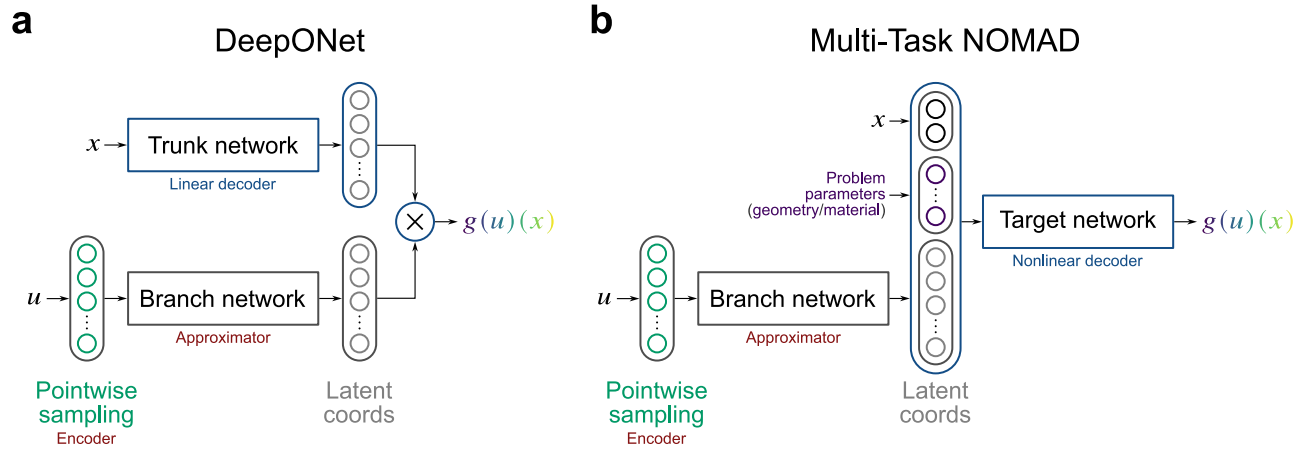

Fig. 10 | Comparison of DeepONet and Multi-Task NOMAD architectures for operator learning. a. Architecture of a classical DeepONet. The input function u is sampled at a finite number of points (encoder) and passed to a neural network (referred to as the *branch network*), which returns p latent coordinates. The spatial coordinates x, at which we aim to evaluate the output function, are instead passed to a separate neural network (referred to as the *trunk network*), which outputs another set of p values. The output function is reconstructed as the dot product (linear decoder) between the p latent coordinates and the p functions of x produced by the trunk network. **b**. A NOMAD is a variant of the DeepONet architecture. As in the latter, the input function u is sampled at a finite number of points (encoder) and passed to a neural network (approximator), which returns p latent coordinates. NOMADs replace the linear decoder of DeepONets with a general nonlinear decoder parameterized by a target network, which takes as input a stack comprising the spatial coordinates x, at which we wish to evaluate the output function, and the latent coordinates. In our case, we require the NOMAD to be Multi-Task; therefore, we additionally include the geometric and material parameters of the problem as inputs to the target network.

Note that we include the geometric and constitutive parameters (belonging to Λ) as *additional dimensions* in an augmented space where the solution fields are defined, leveraging the continuity of these fields with respect to continuous variations in the parameters. In other words, the neural operator learns to generate the solution corresponding to a given $\boldsymbol{u}_{\mathrm{el}}$, expressed as a function of spatial coordinates *and* the problem's parameters. From a practical standpoint, a point evaluation of an output function involves specifying a spatial location *and* the parameters—such as Poisson ratio and yield stress—at which we wish to evaluate it.

To build the neural operator, we take N_{t} training pairs of functions $(\boldsymbol{u}^i_{\mathrm{el}}, (\boldsymbol{\sigma}^i, \boldsymbol{\varepsilon}^i_{\mathrm{p}}, \boldsymbol{\omega}^i)) = (\boldsymbol{u}^i_{\mathrm{el}}, \boldsymbol{g}^i)$, obtained by solving Problem (2) with a nonlinear FE solver over the whole Σ domain, and minimize a mean squared error (MSE) loss over $\theta \in \mathbb{R}^{d_\theta}$:

$$\mathcal{L}(\theta) \triangleq \frac{1}{N_{\mathrm{t}}} \sum_{i=1}^{N_{\mathrm{t}}} \left\| \mathcal{G}_\theta(\boldsymbol{u}^i_{\mathrm{el}}) - \boldsymbol{g}^i \right\|^2_{L^2(\Omega \times \Lambda, \mathbb{R}^{14})} \tag{5}$$

Note that $\boldsymbol{u}_{\mathrm{el}}(\partial\Omega)$ *is not actually* a displacement boundary condition for Problem (2); in fact, even though it is reasonable to assume that the plastic solution $\boldsymbol{u}_{\mathrm{pl}}$ to (2) evaluated at $\partial\Omega$ is approximately equal to $\boldsymbol{u}_{\mathrm{el}}(\partial\Omega)$ under the small-scale plasticity hypothesis, we directly train the operator over $\boldsymbol{u}_{\mathrm{el}}$—the only one available from linear FE analyses—and skip this source of error. In the rest of the paper, we omit the subscript "el" and denote $\boldsymbol{u}_{\mathrm{el}}$ as $\boldsymbol{u}$.

Dataset generation

The dataset of this work is built out of 10^4 distinct reentrant corners having variable material properties, which are embedded into larger domains of different shapes, meshed, and loaded with far-field boundary conditions. Leveraging the incremental nature of plasticity, loads are incrementally scaled from the elastic limit into the plastic domain until the small-scale plasticity hypothesis is violated, i.e., until the plastic zone has reached the boundary of the reentrant corner subdomain Ω; this condition is taken as a threshold for defining small-scale plasticity. As a result, each simulation populates the dataset with the solution fields at the FE nodes for multiple, variably scaled boundary conditions. This approach primarily serves to reduce the number of FE analyses, which constitute a substantial timing bottleneck in the process. All FE analyses are performed in ANSYS using scripts written in APDL; implementation details are provided in Supplementary Section S2. The FE simulations are performed on a workstation equipped with a Ryzen 9 7950X CPU and 128 GB of RAM. For each of the 10^4 geometries, an average of around 10 load increments are applied before violating the small-scale plasticity condition. Solving all the analyses required about 30 h of computational time.

In a later post-processing phase, all outputs are collected by a `Python` script and undergo some additional steps. The displacements fields along the reentrant corner boundary at yield are resampled over 10° steps starting from the $-x$ direction in Fig. 3a, thereby obtaining a fixed number of 36 entries for each component, for a total of 108 sensors; for the polar locations not belonging to the domain, the displacements are set to a linear interpolation of the closest values in the circumferential direction. Then, the axial displacements and rotations have their mean components removed (to eliminate non-strain-producing rigid-body motions) and are transformed with respect to the local reference system *Oxyz* of Fig. 3a.

By leveraging Buckingham's π-theorem[62] to obtain a reduced set of dimensionless parameters, we normalize all lengths and displacements by the notch radius. However, we found that further dividing the latter by the dimensionless factor σ_{y}/E was more beneficial to the training process. As an intuitive explanation, for a given geometry and given elastic properties, the boundary displacements required to yield the material are proportional to σ_{y}/E; then, dividing them by that factor, the input dataset is compressed onto a smaller and less scattered domain, which is known to be beneficial to the training[71]. On the other hand, stress and energy outputs are normalized over the material yield stress σ_{y}, while strain outputs are already dimensionless. This yields a total of three geometric (R/R_{n}, α, β) and three material (ν, σ_{y}/E, E_{t}) dimensionless parameters.

A single training input is obtained by concatenating the normalized boundary displacements of the reentrant corner subdomain Ω, the geometry/material parameters and the spatial coordinates where the target variables are to be evaluated (see Fig. 10b). The corresponding training output consists of the stress, plastic strain, and energy-related fields at that location.

NeuberNet architecture and training

The architecture of a classical DeepONet[22] is shown in Fig. 10a and consists of two distinct neural networks. The first, referred to as the *branch network*,

transforms a pointwise sampling of the input function at p coordinates into a suitable latent space. The second, called *trunk network*, generates p basis functions of the spatial coordinates x, which, when multiplied by the coefficients returned by the branch network, reconstruct the output $g(u)$ evaluated at the coordinates x. Equivalently, a DeepONet projects the solution space onto a linear finite-dimensional latent space, whose basis is learned by the trunk network, while the coefficients—dependent on the specific input—are learned by the branch network. NOMADs[40] (see Fig. 10b) are a straightforward extension of DeepONets, in which the solution space is represented as a general *non-linear manifold* parameterized by a neural network (specifically, a non-linear decoder). This decoder takes as input the latent coordinates returned by the branch network together with the spatial coordinates at which the output is to be evaluated.

After conducting preliminary experiments, we selected an ensemble of Multi-Task NOMADs[40]—one for each solution component, as in Yin et al.[58]—as NeuberNet's main neural architecture (see Fig. 10b), owing to their generality in capturing the low-dimensional latent manifolds on which the output functions lie. The approximators are $[108, 256, 256, p]$ multilayer perceptrons (MLPs), while the decoders are $[8 + p, 512, 512, 512, 512, 512, 1]$ MLPs; all networks share a ReLU activation function. As reported in the Results, we employ $p = 64$ as latent dimension. The decoders take as input a concatenation of the approximator output with the coordinates where the target function must be evaluated. Exploiting the continuity of the target functions with respect to the parameters defining the reentrant corner geometry and the material parameters, we consider them as *additional coordinates* of an augmented space over which the target functions are defined; hence, we concatenate the geometric parameters R/R_n, α, β and the material parameters ν, σ_y/E, E_t/E with the spatial coordinates to form the input that is fed to the decoder, i.e., what would be the trunk input in a DeepONet (see Fig. 10a). As a result, the input dimension of the decoder is $8 + 64 = 72$.

A single training point of NeuberNet consists of a triplet $\left(\boldsymbol{u}, \boldsymbol{x}, g(\boldsymbol{u})\right)$, namely: the value of the target variable $g(\boldsymbol{u})$ evaluated at a given spatial position (FE node) and geometric/material parameters $\boldsymbol{x}$, corresponding to an input BC $\boldsymbol{u}$. There are as many training points as the number of input BCs multiplied by the number of FE nodes at which the target variables are available. As well-known in the context of DeepONets[22,27,72], storing the dataset in that form is particularly memory-inefficient; instead, we keep separate datasets of the 10^4 applied $\boldsymbol{u}$ and, for each load step, of the corresponding target variables, and we dynamically build mini-batches of data triplets at training time.

We employ a 80/20 train-test split of the geometry-materials-loads combinations, so that the training weights are validated against unseen problems. We scale the training pairs to achieve unit variance. We simultaneously train all the NeuberNet components for 1000 epochs with an `AdamW` optimizer, using a batch size of 4096, an initial learning rate of 10^{-4}, a cosine annealing scheduler and a weight decay of 10^{-4}. Just for the experiments varying the latent dimension p, we restrict training to 100 epochs to reduce training time. The training algorithm was implemented in `PyTorch` and was run on a single NVIDIA RTX 4090 GPU. A preliminary hyperparameter search with a tree-structured Parzen estimator[73] had been conducted through the `Optuna` framework[74], to fine-tune the networks architecture and the optimizer. No overfitting has been observed during training. Even though it is theoretically permissible, we do not employ any physical loss to train the network weights, leaving this point for future explorations. Loss histories are presented in Supplementary Section S3, while details on the training of YieldNet are provided in Supplementary Section S4.

Inference

Having trained all of its sub-networks, NeuberNet is subsequently set in inference mode. The normalized boundary displacements, geometric parameters and material properties are initially fed to YieldNet, which returns the maximum equivalent stress σ^{Ω}_{vm} over Ω and the limit stress $\sigma^{\Omega}_{vm,L}$ for small-scale plasticity. This opens up three alternatives and three corresponding solutions:

- $\sigma^{\Omega}_{vm} < \sigma_y$: no plastic behavior arises. As NeuberNet isn't explicitly trained on purely elastic solutions, we linearly scale the boundary displacements to get $\sigma^{\Omega}_{vm} = \sigma_y$ (i.e., right at the yield limit), we get the stress and elastic strain energy density predictions from NeuberNet (plastic strains and work are left to zero), and we finally scale the results according to $\sigma^{\Omega}_{vm}/\sigma_y$.
- $\sigma_y < \sigma^{\Omega}_{vm} < \sigma^{\Omega}_{vm,L}$: small-scale plasticity conditions are met. We fed the inputs to NeuberNet and get the corresponding results.
- $\sigma^{\Omega}_{vm,L} < \sigma^{\Omega}_{vm}$: small-scale plasticity conditions are violated. We notify the user, indicating that a fully non-linear FE analysis may provide a more reliable assessment of the given structure.

When performing inference with NeuberNet, one needs a sufficiently refined grid of points over the reentrant corner subdomain Ω. For the analyses in the dataset, we evaluate the results at the FE corner node locations to directly compare the surrogate predictions with the ground truths. A preliminary convergence analysis ensures that the refinement of the FE mesh is sufficient to capture the shape of the results and to neglect discretization error.

However, in the intended use case of NeuberNet, such a mesh is not available, as its purpose is precisely to avoid refined meshing of the reentrant corner. Given the extremely low cost of forward evaluation with NeuberNet, we recommend evaluating the target values on a very refined grid—independent of the actual FE nodes—with a characteristic size of $l \approx R_n/10$, which is sufficient to capture the critical values of the target variables. Since the domain Ω is known, we simply probe NeuberNet inside it and ignore values outside the domain; as an alternative, masking approaches have recently been proposed[24].

Low-fidelity analyses

Each of the 2000 geometry-loadings pair in the FE test dataset is reprocessed in the following alternative way. Instead of being meshed with the refined mapped scheme used to obtain the ground truth values of the target variables, each model is remeshed with ANSYS free-mesh tool, imposing a given characteristic element size l_e. Then, a linear elastic solution is pursued, and the corresponding boundary displacement fields $\boldsymbol{u}$ are stored in a `txt` file. Eventually, for a given l_e, one obtains a dataset of boundary displacements generated by a low-fidelity elastic simulation, as opposed to the refined meshes used to run the ground truth plastic analyses. These new boundary displacements are then used to perform inference through NeuberNet, and the results are compared with the ground truths.

Nonlinear hardening analyses

We take the 2000 analyses of the FE test dataset, and for each of them, we replace the constitutive model with a Voce hardening law[63]: $\sigma_f = \sigma_y + R_\infty(1 - e^{-bp})$. We set b to a fairly typical value[75] of 10; we randomly generate a σ_y/E between 10^{-3} and 10^{-2}, and a R_∞/E between 10^{-4} and 10^{-2}.

During inference, we set E_t to the derivative of the Voce law at the yield point—i.e., to $R_\infty b$. Finally, the corresponding predictions are compared with the ground truth values obtained from the FE analyses.

3D shaft analyses

We design a shaft with two radii $r_1 = 15$ mm and $r_2 = 30$ mm, connecting them with a 90° fillet having a tip radius of 0.5 mm. The shaft is made of an aluminum alloy, with $E = 70$ GPa, $\nu = 0.3$, $\sigma_y = 350$ MPa, and $E_t = 3500$ MPa. In ANSYS, we build a FE model of the shaft geometry with 8-node SOLID186 brick elements, refining the mesh size in correspondence of the reentrant corner notch according to a convergence analysis of the result variables. We fully restrain the lower end of the shaft, while the upper end is rigidly constrained using MPC184 elements, with loads applied to the corresponding master node. We apply a linear combination of axial,

bending, and torsional loads. The bending moment violates the axial symmetry of the problem. Plastic behavior is introduced with a bilinear isotropic hardening data table, and a solution is searched through an iterative Newton-Raphson solver.

In parallel, we solve the same problem with a linear elastic material model. Then, we take the symmetry plane perpendicular to the bending moment direction and we define on it a circular path of diameter $10R_n$ centered on the notch tip. We map the displacements over that path; in addition, we also map the product $r\varepsilon_{\theta\theta}$ over the same path, as a surrogate of the radial displacements in axisymmetric conditions; we eventually store all of them on a `txt` file, which is then fed to NeuberNet.

Data availability

The entire dataset used in this work is available in a public Zenodo repository at the following link. The repository includes all the outputs of FE analyses in their raw `txt` form and their post-processed versions as `NumPy` compressed `npz` files; additionally, the corresponding predictions obtained with NeuberNet are included.

Code availability

All the `APDL` and `Python` scripts that are needed to reproduce the findings of this paper are available in the following GitHub repository. For those who wish to directly perform predictions with NeuberNet without undergoing the training process, the final state of the learnable weights used to perform inference throughout the paper is included in the repository.

Received: 26 February 2025; Accepted: 6 November 2025;
Published online: 13 December 2025

References

1. Neuber, H. *Kerbspannungslehre*. https://doi.org/10.1007/978-3-662-36565-6 (Springer Berlin Heidelberg, 1937).
2. Ieşan, D. On Saint-Venant's problem. *Arch. Ration. Mech. Anal.* **91**, 363–373 (1986).
3. Savruk, M. P. & Kazberuk, A.*Stress Concentration at Notches*. https://doi.org/10.1007/978-3-319-44555-7(Springer International Publishing, 2017).
4. Taylor, D. *The Theory of Critical Distances: a New Perspective in Fracture Mechanics* (Elsevier, 2007).
5. Zhu, S.-P., Ye, W.-L., Correia, J. A. F. O., Jesus, A. M. P. & Wang, Q. Stress gradient effect in metal fatigue: review and solutions. *Theor. Appl. Fract. Mech.* **121**, 103513 (2022).
6. Berto, F. & Lazzarin, P. A review of the volume-based strain energy density approach applied to V-notches and welded structures. *Theor. Appl. Fract. Mech.* **52**, 183–194 (2009).
7. Zhu, S.-P., He, J.-C., Liao, D., Wang, Q. & Liu, Y. The effect of notch size on critical distance and fatigue life predictions. *Mater. Des.* **196**, 109095 (2020).
8. Ambati, M., Gerasimov, T. & De Lorenzis, L. Phase-field modeling of ductile fracture. *Comput. Mech.* **55**, 1017–1040 (2015).
9. Evans, A. & Hutchinson, J. A critical assessment of theories of strain gradient plasticity. *Acta Mater.* **57**, 1675–1688 (2009).
10. Park, K. & Paulino, G. H. Cohesive Zone Models: a critical review of traction-separation relationships across fracture surfaces. *Appl. Mech. Rev.* **64**, 060802 (2011).
11. Benedetti, M., Berto, F., Le Bone, L. & Santus, C. A novel strain-energy-density based fatigue criterion accounting for mean stress and plasticity effects on the medium-to-high-cycle uniaxial fatigue strength of plain and notched components. *Int. J. Fatigue* **133**, 105397 (2020).
12. Bertini, L., Le Bone, L., Santus, C., Chiesi, F. & Tognarelli, L. High load ratio fatigue strength and mean stress evolution of quenched and tempered 42CrMo4 steel. *J. Mater. Eng. Perform.* **26**, 3784–3793 (2017).
13. Neuber, H. Theory of stress concentration for shear-strained prismatical bodies with arbitrary nonlinear stress-strain law. *J. Appl. Mech.* **28**, 544–550 (1961).
14. Glinka, G. Energy density approach to calculation of inelastic strain-stress near notches and cracks. *Eng. Fract. Mech.* **22**, 485–508 (1985).
15. Topper, T. H., Wetzel, R. M. & Morrow, J. Neuber's rule applied to fatigue of notched specimens. Tech. Rep. http://www.dtic.mil/docs/citations/AD0659550 (1967).
16. Härkegård, G. & Mann, T. Neuber prediction of elastic-plastic strain concentration in notched tensile specimens under large-scale yielding. *J. Strain Anal. Eng. Des.* **38**, 79–94 (2003).
17. Bathe, K.-J. (ed.) *Finite Element Procedures*, 2nd edn. (Klaus-Jürgen Bathe, 2014).
18. Noor, A. K., Kamel, H. A. & Fulton, R. E. Substructuring techniques—status and projections. *Comput. Struct.* **8**, 621–632 (1978).
19. Pan, W., Leen, S. & Hyde, T. Investigation of an iterative sub-structure method for elastic and elastic–plastic framework analysis. *Comput. Struct.* **84**, 690–698 (2006).
20. Ghosh, S., Lee, K. & Moorthy, S. Multiple scale analysis of heterogeneous elastic structures using homogenization theory and voronoi cell finite element method. *Int. J. Solids Struct.* **32**, 27–62 (1995).
21. Chen, T. & Chen, H. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Trans. Neural Netw.* **6**, 911–917 (1995).
22. Lu, L., Jin, P., Pang, G., Zhang, Z. & Karniadakis, G. E. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nat. Mach. Intell.* **3**, 218–229 (2021).
23. Jin, P., Meng, S. & Lu, L. MIONet: learning multiple-input operators via tensor product. *SIAM J. Sci. Comput.* **44**, A3490–A3514 (2022).
24. Kumar, V., Goswami, S., Kontolati, K., Shields, M. D. & Karniadakis, G. E. Synergistic learning with multi-task DeepONet for efficient PDE problem solving. *Neural Netw.* **184**, 107113 (2025).
25. Lanthaler, S. Operator learning with PCA-Net: upper and lower complexity bounds. *J. Mach. Learn. Res.* **24**, 1–67 (2023).
26. Lee, J. Y., Cho, S. W. & Hwang, H. J. HyperDeepONet: learning operator with complex target function space using the limited resources via hypernetwork. *The Eleventh International Conference on Learning Representations* (ICLR 2023).
27. Lu, L. et al. A comprehensive and fair comparison of two neural operators (with practical extensions) based on FAIR data. *Comput. Methods Appl. Mech. Eng.* **393**, 114778 (2022).
28. Park, J. & Kang, N. Point-DeepONet: a Deep Operator Network Integrating PointNet for nonlinear analysis of non-parametric 3D geometries and load conditions. ArXiv: http://arxiv.org/abs/2412.18362 (2024).
29. Venturi, S. & Casey, T. SVD perspectives for augmenting DeepONet flexibility and interpretability. *Comput. Methods Appl. Mech. Eng.* **403**, 115718 (2023).
30. Kovachki, N. et al. Neural operator: Learning maps between function spaces with applications to pdes. *J. Mach. Learn. Res*. **24**, 1–97 (2023).
31. Li, Z. et al. Neural operator: graph kernel network for partial differential equations. ArXiv: http://arxiv.org/abs/2003.03485 (2020).
32. Bartolucci, F. et al. Representation equivalent neural operators: a framework for alias-free operator learning. *Adv. Neural Inform. Proc. Syst.* **36**. https://proceedings.neurips.cc/paper_files/paper/2023/hash/dc35c593e61f6df62db541b976d09dcf-Abstract-Conference.html (2024).
33. Azizzadenesheli, K. et al. Neural operators for accelerating scientific simulations and design. *Nat. Rev. Phys.* **6**, 320–328 (2024).
34. Zappala, E. et al. Learning integral operators via neural integral equations. *Nat. Mach. Intell.* **6**, 1046–1062 (2024).

35. Li, Z. et al. Fourier neural operator for parametric partial differential equations. *The Ninth International Conference on Learning Representations* (ICLR 2021).
36. Kovachki, N., Lanthaler, S. & Mishra, S. On universal approximation and error bounds for Fourier neural operators. *J. Mach. Learn. Res.* **22**, 1–76 (2021).
37. Li, Z., Huang, D. Z., Liu, B. & Anandkumar, A. Fourier neural operator with learned deformations for PDEs on general geometries. *J. Mach. Learn. Res.* **24**, 1–26 (2023).
38. Cao, Q., Goswami, S. & Karniadakis, G. E. Laplace neural operator for solving differential equations. *Nat. Mach. Intell.* **6**, 631–640 (2024).
39. Kissas, G. et al. Learning operators with coupled attention. *J. Mach. Learn. Res.* **23**, 1–63 (2022).
40. Seidman, J., Kissas, G., Perdikaris, P. & Pappas, G. J. NOMAD: nonlinear manifold decoders for operator learning. *Adv. Neural Inf. Process. Syst.* **35**, 5601–5613 (2022).
41. Abueidda, D. W., Koric, S., Sobh, N. A. & Sehitoglu, H. Deep learning for plasticity and thermo-viscoplasticity. *Int. J. Plast.* **136**, 102852 (2021).
42. Ahmed, B. et al. Physics-informed deep operator networks with stiffness-based loss functions for structural response prediction. *Eng. Appl. Artif. Intell*. **144**, 110097 (2025).
43. Bonatti, C. & Mohr, D. On the importance of self-consistency in recurrent neural network models representing elasto-plastic solids. *J. Mech. Phys. Solids* **158**, 104697 (2022).
44. Goswami, S., Yin, M., Yu, Y. & Karniadakis, G. E. A physics-informed variational DeepONet for predicting crack path in quasi-brittle materials. *Comput. Methods Appl. Mech. Eng.* **391**, 114587 (2022).
45. Hildebrand, S. & Klinge, S. Comparison of neural FEM and neural operator methods for applications in solid mechanics. *Neural Comput. Appl.* **36**, 16657–16682 (2024).
46. Huang, D., Fuhg, J. N., Weißenfels, C. & Wriggers, P. A machine learning based plasticity model using proper orthogonal decomposition. *Comput. Methods Appl. Mech. Eng.* **365**, 113008 (2020).
47. Liu, B. et al. A learning-based multiscale method and its application to inelastic impact problems. *J. Mech. Phys. Solids* **158**, 104668 (2022).
48. Pantidis, P. & Mobasher, M. E. Integrated Finite Element Neural Network (I-FENN) for non-local continuum damage mechanics. *Comput. Methods Appl. Mech. Eng.* **404**, 115766 (2023).
49. Settgast, C., Abendroth, M. & Kuna, M. Constitutive modeling of plastic deformation behavior of open-cell foam structures using neural networks. *Mech. Mater.* **131**, 1–10 (2019).
50. Susmel, L. Estimating notch fatigue limits via a machine learning-based approach structured according to the classic Kf formulas. *Int. J. Fatigue* **179**, 108029 (2024).
51. Koric, S., Viswantah, A., Abueidda, D. W., Sobh, N. A. & Khan, K. Deep learning operator network for plastic deformation with variable loads and material properties. *Eng. Comput.* **40**, 917–929 (2024).
52. He, J., Koric, S., Abueidda, D., Najafi, A. & Jasiuk, I. Geom-DeepONet: a point-cloud-based deep operator network for field predictions on 3D parameterized geometries. *Comput. Methods Appl. Mech. Eng.* **429**, 117130 (2024).
53. He, J. et al. Sequential Deep Operator Networks (S-DeepONet) for predicting full-field solutions under time-dependent loads. *Eng. Appl. Artif. Intell.* **127**, 107258 (2024).
54. Grossmann, T. G., Komorowska, U. J., Latz, J. & Schönlieb, C.-B. Can physics-informed neural networks beat the finite element method? *IMA J. Appl. Math.* **89**, 143–174 (2024).
55. He, J. et al. Novel DeepONet architecture to predict stresses in elastoplastic structures with variable complex geometries and loads. *Comput. Methods Appl. Mech. Eng.* **415**, 116277 (2023).
56. Dini, D. & Hills, D. A. Asymptotic characterisation of nearly-sharp notch root stress fields. *Int. J. Fract.* **130**, 651–666 (2004).
57. Krokos, V., Bui Xuan, V., Bordas, S. P. A., Young, P. & Kerfriden, P. A Bayesian multiscale CNN framework to predict local stress fields in structures with microscale features. *Comput. Mech.* **69**, 733–766 (2022).
58. Yin, M., Zhang, E., Yu, Y. & Karniadakis, G. E. Interfacing finite elements with deep neural operators for fast multiscale modeling of mechanics problems. *Comput. Methods Appl. Mech. Eng.* **402**, 115027 (2022).
59. Palchoudhary, A., Peter, S., Maurel, V., Ovalle, C. & Kerfriden, P. A plastic correction algorithm for full-field elasto-plastic finite element simulations: critical assessment of predictive capabilities and improvement by machine learning. *Comput. Mech.* **75**, 1263–1293 (2025).
60. Zhu, M., Zhang, H., Jiao, A., Karniadakis, G. E. & Lu, L. Reliable extrapolation of deep neural operators informed by physics or sparse observations. *Comput. Methods Appl. Mech. Eng.* **412**, 116064 (2023).
61. Williams, M. L. Stress singularities resulting from various boundary conditions in angular corners of plates in extension. *J. Appl. Mech.* **19**, 526–528 (1952).
62. Bertrand, J. Sur l'homogénéité dans les formules de physique. *Cah. de. Rech. de. l'Acad. de. Sci.* **86**, 916–920 (1878).
63. Voce, E. The relationship between stress and strain for homogeneous deformation. *J. Inst. Met.* **74**, 537–562 (1948).
64. Clifton, F. *Strength variability in structural materials. Reports and Memoranda No. 3654, UK Ministry of Aviation Supply—Aeronautical Research Council, Farnborough.* https://reports.aerade.cranfield.ac.uk/handle/1826.2/2925 (1969).
65. Machine learning solutions looking for PDE problems. *Nat. Mach. Intell.* **7**, 1–1 https://doi.org/10.1038/s42256-025-00989-w (2025).
66. McGreivy, N. & Hakim, A. Weak baselines and reporting biases lead to overoptimism in machine learning for fluid-related partial differential equations. *Nat. Mach. Intell.* **6**, 1256–1269 (2024).
67. Model order reduction in fluid dynamics: challenges and perspectives. in *Reduced Order Methods for Modeling and Computational Reduction*, 235–273. https://doi.org/10.1007/978-3-319-02090-7_9 (Springer International Publishing, 2014).
68. Bachmayr, M. & Cohen, A. Kolmogorov widths and low-rank approximations of parametric elliptic PDEs. *Math. Comput.* **86**, 701–724 (2016).
69. Pinkus, A. *N-widths in Approximation Theory*, Vol. 7 (Springer Science & Business Media, 2012).
70. Benedetti, M. et al. Multiaxial plain and notch fatigue strength of thick-walled ductile cast iron EN-GJS-600-3: combining multiaxial fatigue criteria, theory of critical distances, and defect sensitivity. *Int. J. Fatigue* **156**, 106703 (2022).
71. Wang, S., Wang, H. & Perdikaris, P. Improved architectures and training algorithms for deep operator networks. *J. Sci. Comput.* **92**. https://doi.org/10.1007/s10915-022-01881-0 (2022).
72. Karumuri, S., Graham-Brady, L. & Goswami, S. Efficient Training of Deep Neural Operator Networks via Randomized Sampling. *World Scientific Annual Rev. Artif. Intell*. **3**, 2540001 (2025).
73. Bergstra, J., Bardenet, R., Bengio, Y. & Kégl, B. Algorithms for Hyper-Parameter Optimization. In *Advances in Neural Information Processing Systems*, Vol. 24. https://papers.nips.cc/paper_files/paper/2011/hash/86e8f7ab32cfd12577bc2619bc635690-Abstract.html (Curran Associates, Inc., 2011).
74. Akiba, T., Sano, S., Yanase, T., Ohta, T. & Koyama, M. Optuna: a next-generation hyperparameter optimization framework. in *Proc. 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, 2623–2631. https://doi.org/10.1145/3292500.3330701 (Association for Computing Machinery, 2019).
75. Singh, K. Strain hardening behaviour of 316L austenitic stainless steel. *Mater. Sci. Technol.* **20**, 1134–1142 (2004).

76. Anderson, T. L. *Fracture Mechanics: Fundamentals and Applications* (CRC Press, 2017).

Acknowledgements
Author T. Grossi wishes to express his gratitude to Prof. L. Bertini for financial support and for insightful discussions, and to I. Senegaglia for his assistance with the graphical post-processing of FE analyses.

Author contributions
M. Benedetti framed the problem and posed the research question. T. Grossi generated the datasets through FE simulations, designed and trained the neural operator's architecture, conducted data analyses, and wrote the initial draft of the manuscript. M. Benedetti and M. Beghini supervised the study and provided strategic guidance. All authors reviewed the manuscript.

Competing interests
The authors declare no competing interests.

Additional information
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s44172-025-00549-5.

Correspondence and requests for materials should be addressed to Tommaso Grossi.

Peer review information *Communications Engineering* thanks Pierre Kerfriden and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Primary Handling Editors: [Emanuela Bosco] and [Rosamund Daw]. A peer review file is available.