



Peridynamic neural operators: A data-driven nonlocal constitutive model for complex material responses

Siavash Jafarzadeh ^a, Stewart Silling ^b, Ning Liu ^c, Zhongqiang Zhang ^d, Yue Yu ^{a,*}

^a Department of Mathematics, Lehigh University, Bethlehem, PA, USA

^b Center for Computing Research, Sandia National Laboratories, Albuquerque, NM, USA

^c Global Engineering and Materials, Inc., Princeton, NJ, USA

^d Department of Mathematical Sciences, Worcester Polytechnic Institute, Worcester, MA, USA



ARTICLE INFO

Keywords:

Data-driven physics modeling
Neural operators
Scientific machine learning
Ordinary state-based peridynamics
Nonlocal models

ABSTRACT

Neural operators, which can act as implicit solution operators of hidden governing equations, have recently become popular tools for learning the responses of complex real-world physical systems. Nevertheless, most neural operator applications have thus far been data-driven and neglect the intrinsic preservation of fundamental physical laws in data. In this work, we introduce a novel integral neural operator architecture called the Peridynamic Neural Operator (PNO) that learns a nonlocal constitutive law from data. This neural operator provides a forward model in the form of state-based peridynamics, with objectivity and momentum balance laws automatically guaranteed. As applications, we demonstrate the expressivity and efficacy of our model in learning complex material behaviors from both synthetic and experimental data sets. We also compare the performances with baseline models that use predefined constitutive laws. We show that, owing to its ability to capture complex responses, our learned neural operator achieves improved accuracy and efficiency. Moreover, by preserving the essential physical laws within the neural network architecture, the PNO is robust in treating noisy data. The method shows generalizability to different domain configurations, external loadings, and discretizations.

1. Introduction

The detection and prediction of complex material responses from experimental spatial measurements are often needed in applications of interest to the broad scientific and engineering community [1–6]. In disciplines ranging from structural health prognosis to non-destructive evaluation, material characteristics need to be accurately captured to guarantee reliable and trustworthy predictions. For many decades, constitutive models within PDE-based continuum mechanics have been commonly employed to characterize material response. Typically in such characterization, a strain energy density function is predefined with a specific functional form. Then, the material parameters are calibrated through an inverse method or analytical stress-strain fitting to test data. The outcome is a predefined model with fitted material parameters that can be employed as a homogenized surrogate for a real specimen in applications, such as the prediction of material deformation under a broad range of loading scenarios.

The parameter-fitting PDE approach faces three challenges that can affect the reliability of the resulting surrogates. First, the descriptive power of predefined constitutive models is often restricted to certain deformation modes and ranges of strain. This tends to limit the predictivity and generalizability of the model [7–9]. Second, the required functional form of the expert-constructed model is not always available. For example, in spite of many years of research, there is not a definitive material model for many

* Corresponding author.

E-mail address: yuy214@lehigh.edu (Y. Yu).

Table 1
Nomenclature.

Symbol	Description
$\Omega \subseteq \mathbb{R}^d$	physical domain of dimension d
\mathbf{x}	material point position vector at reference configuration
\mathbf{y}	material point position vector at deformed configuration
\mathbf{u}	displacement
t	time
\mathbf{q}	position vector of for neighbors of \mathbf{x}
δ	peridynamic horizon
ρ	mass density
$B_\delta(\mathbf{x})$	neighborhood of \mathbf{x} with radius δ
$f(\mathbf{q}, \mathbf{x}, t)$	pairwise bond force density
$\mathbf{b}(\mathbf{x}, t)$	external force density
ξ	peridynamic bond
η	relative displacement of a bond
$\underline{\mathbf{A}}[\mathbf{x}, t](\xi)$	a peridynamic state at \mathbf{x} and t acting on bond ξ
$\underline{\mathbf{x}}$	identity state
$\underline{\mathbf{Y}}$	deformation state
$\underline{\mathbf{T}}$	force state
$\underline{\mathbf{t}}$	scalar force state
$\underline{\mathbf{M}}$	state for deformed bond direction unit vector
$\underline{\mathbf{e}}$	extension state
\mathbf{e}	bond extension
C_1, C_2	peridynamic elasticity constants
$\underline{\vartheta}$	influence state
\mathcal{G}	nonlocal dilatation
σ^{NN}	peridynamic neural operator
ω^{NN}	neural network function for scalar force state
\mathbf{v}, \mathbf{w}	neural network function for influence state
Ω_I	neural network trainable parameters
χ	interaction domain
D_{tr}, D_{val}	data spatial coordinate set
S_r	training and validation data pairs
$loss_b$	number of training samples
$loss_e$	external force loss
$loss_u$	displacement loss

biotissues such as skin [10,11]. Third, if the fitted material model is a homogenized surrogate for a heterogeneous material, it inevitably approximates the original problem only in an averaged sense [12]. As a result, it may “wash out” the length scale in the original material microstructure, causing information to be lost that can be important in multiscale material modeling problems.

To address the first of these challenges and obtain more flexible predictive models, machine learning (ML) based approaches have emerged as potential solutions. Many of the successful prior attempts on data-driven material modeling can be classified as one of the following two types: (1) approaches that learn a constitutive law [11,13–17], and (2) model-free learning approaches [18–27]. The first approach aims to obtain the strain energy potentials, often in terms of the sums of convex non-decreasing functions of invariants and linear combinations of them, by fitting stress–strain curves. Additional physical requirements, such as objectivity and polyconvexity, are imposed in the design. Recent examples include input convex neural networks (ICNN) [28], constitutive artificial neural networks (CANN) [29], and neural ordinary differential equations (NODE) [30]. However, since these approaches learn from stress–strain curves, they face challenges in capturing material heterogeneity. It can also be difficult for these approaches to handle more general types of training data beyond stress–strain curves, such as displacement measurements from mesoscale simulations [31] and full-field Digital Image Correlation (DIC) [32] measurements.

The second ML approach, model-free learning, offers more flexibility, because the material response can be modeled as a mapping between loading conditions and the resulting displacement and damage fields [19,22–25,27]. In this approach, the constitutive law can be seen as a hidden PDE, and the target mapping serves as the solution operator for this PDE. The goal is then to train a neural network based on spatial measurements as a surrogate for the implicit solution operator. Different neural network architectures have been proposed within this general approach, including the deep neural networks [33–35], graph neural networks [36–40], and convolutional neural networks [41–43], and others. However, the model-free approach models generally do not guarantee fundamental physical laws. As a result, their performances highly rely on the quantity and coverage of available data. Moreover, the learnt network is not generalizable to different grids and domain shapes. That means, modeling the material response for a new instance of new domain shape with a different discretization often requires a new neural network model and correspondingly training from scratch.

Recently, the nonlocal neural operator was proposed as a way to learn a surrogate mapping between function spaces [44,45]. With this type of operator, the increment between layers is modeled as a nonlocal (integral) operator to capture the long-range dependencies in the feature space. The nonlocal neural operator can therefore be seen as a stack of discretized integral equations. When the number of discretization nodes changes, the discretization still corresponds to the same continuous integral equation, and therefore the optimal network parameters can be reused. As a result, the operators feature resolution independence. They also

serve as universal approximators to any function-to-function mapping operator [45]. These advantages make them good candidates for discovering models for complex material response directly from spatial measurements. In [22], neural operators were employed to construct a surrogate mapping from each loading function to its corresponding material response function directly from digital image correlation (DIC) measurements. However, although they provide generalizability to different resolutions, the vanilla neural operators are not generalizable to different domain shapes [25,46]. On the other hand, same as other model-free neural networks, nonlocal neural operators rely on data to obtain the intrinsic physical laws, and hence they are prone to performance deterioration due to overfitting and measurement noises, especially in the small and noisy data regime [47,48].

Beyond their advantages in neural network designs, *nonlocal* operators are also currently receiving much attention as homogenized surrogate models that efficiently and accurately capture small-scale effects [49]. In comparison with classical PDE models, these operators can better incorporate the coupling between constituent materials in a composite microstructure, taking into account the length scales in the microstructure (see Section 4 later in this paper for examples of this). The presence of such length scales in a heterogeneous material leads to technologically important observable effects such as wave dispersion in real materials. Nonlocality also emerges from certain approaches to homogenization. Since nonlocal models usually include a finite cut-off distance for interactions (the *horizon*), an intrinsic length scale such as the length of polymer chains can be introduced into the model in a natural way [50]. Nonlocal models have therefore been proposed as homogenized surrogates for molecular dynamics, providing efficient upscaled models that bridge nanoscale material structures and homogenized continuum responses [51–55].

Nevertheless, in previous work, the form of the nonlocal constitutive model has been adopted on a case-by-case basis, informed by either experimental evidence [56,57] or analytical derivations that help match the desired physical properties [58–60]. In prior work [61–65], our team has developed the nonlocal operator regression approach, which learns the nonlocal kernel of an integral equation from data. This approach has been effective in providing homogenized coarse grained models for graphene sheets from molecular dynamics simulations [31,66]. It has also successfully reproduced features of stress wave propagation in composites and metamaterials such as wave dispersion [61,63–65]. However, the above work relies on a predefined nonlocal model form and has only been applied to linear response prediction. Although there has been rapid progress on nonlocal models, to the best of our knowledge, there is no general procedure for determining the form of a nonlocal constitutive law for complex material behaviors, such as anisotropy, nonlinearity, etc. The present work provides a novel method for addressing this.

In this paper, we aim to address the question of how to obtain from full-field spatial measurements a homogenized surrogate model that captures complex material responses with minimum model-form error. To mitigate the three challenges discussed above encountered in the classical PDE parameter-fitting, and to exploit the advantages of nonlocal operators, we develop a novel nonlocal neural operator architecture. In contrast to existing model-free nonlocal neural operator-based material modeling approaches [23,44,48,67,68] where the neural operator is constructed to find a solution operator mapping the body loads/boundary conditions to the corresponding displacement fields, our model uses nonlocal neural operators to find constitutive laws from data. To accomplish this, we leverage the nonlocal operator regression approach, to *identify both the optimal nonlocal kernel function and the model form*. As summarized in Section 2.1, a main innovation is that we parameterize the nonlocal neural operator as a peridynamic material model [69–72]. As such, the model exploits the expressivity of neural networks while it inherits from peridynamics adherence to the basic physics requirements such as the balance laws for linear and angular momentum, as well as objectivity. This architecture helps to capture the nonlinear homogenized material response as a function-to-function mapping. In Table 2 we summarize relevant properties of the proposed nonlocal constitutive law model in comparison with other state-of-the-art data-driven material models.

We summarize below our main contributions:

- We propose the Peridynamic Neural Operator (PNO), a new nonlocal neural operator-based formulation for capturing complex material responses. The PNO learns an optimal nonlocal constitutive law from full-field, spatial measurements of displacement and loading. In contrast to classical PDE-based models, our architecture learns the nonlocal bond forces as neural networks. This captures nonlinear and anisotropic material responses without prior expert-constructed knowledge on the functional form of the strain energy density.
- Because the neural operator design is an implementation of a special class of peridynamic material models (“mobile” materials), the PNO automatically satisfies the physically required balance laws and material objectivity. This implies that any simulation using PNO is compliant with these fundamental laws, even on samples outside the training range. This is a significant improvement over generic free-form neural networks.
- By learning a constitutive model instead of a solution operator, our learned model is equally accurate for test samples with various resolutions, loading scenarios, and domain settings, even if they are substantially different from the ones used for training.
- Our model learns directly from displacement field–loading field data pairs, which makes it particularly promising for learning complex material responses without explicit constitutive models and microstructure measurements. To demonstrate this capability, we provide in Section 4 examples of learning complex mechanical responses of materials directly from molecular dynamics and digital image correlation (DIC) tracking measurements.

Outline of the paper. Section 2 provides a brief introduction to the peridynamic nonlocal mechanical theory, including the class of material models provided by the PNO, and its connection with the graph-based nonlocal neural operators. In Section 3 we describe the PNO architecture, together with theoretical results on its guaranteed physical consistency. To provide a condensed overview of the learning procedure for practitioners, a summary of the key ingredients and steps can be found in Algorithm 1. Section 4 demonstrates the effectiveness of the learning technique on both synthetic and experimental datasets. We illustrate the model’s

Table 2

List of properties for different experimental data-driven material-modeling approaches.

Model	Generalizable in resolution	Generalizable in domain	Flexible model forms	Homogenized effects	Basic physical laws
Pre-defined models	✓	✓	—	—	✓
NODE-based constitutive laws	✓	✓	✓ ^a	—	✓
Model-free neural networks	✓ ^b	—	✓	—	—
Nonlocal operator regression	✓	✓	—	✓	✓
Peridynamic neural operators	✓	✓	✓	✓	✓

^a Since the strain energy density function is formulated as a neural network of invariants of the right Cauchy–Green deformation tensor [30], the model form depends on the pre-defined choice of invariants.

^b When using nonlocal neural operator architectures, the model can be resolution-invariant [44,67]. Otherwise, the generic vector-to-vector mapping neural networks are not generalizable in measurement grids.

generalizability with respect to loading, domain size, and domain shape, along with its expressivity with respect to nonlinear and anisotropic material responses. It is shown that in comparison with available conventional constitutive models, our method reduces the prediction error by around 30% on a soft biological tissue data set based on DIC-tracked displacement measurements. Section 5 summarizes our contributions and suggests future research directions. Additionally, all major notations are listed in Table 1.

2. Background

Throughout this paper, we use unbolted case letters to denote scalars/scalar-valued functions, bold letters to denote vectors/vector-valued functions, underlined unbolted letters for scalar-valued state functions, underlined bold letters for vector-valued state functions, and calligraphic letters for operators. For any vector ν , we use $|\nu|$ to denote its l^2 norm. For any function $f(x)$, $x \in \Omega \subseteq \mathbb{R}^d$, taking values at nodes $\chi := \{x_1, x_2, \dots, x_M\}$, $\|f\|$ denotes its l^2 norm, i.e., $\|f\| := \sqrt{\sum_{i=1}^M |f(x_i)|^2/M}$, which can be seen as an approximation to the $L^2(\Omega)$ norm of f (up to a constant). \mathbb{R}^d represents the dimension- d Euclidean space.

In this section, we briefly introduce the concept of peridynamic theory and nonlocal neural operators. Moreover, we review relevant concepts of conservation laws and two types of invariances, which will later become complementary to the proposed PNO definition.

2.1. Peridynamic theory

The peridynamic theory of solid mechanics (“peridynamics”) is an extension of the classical PDE theory that is compatible with discontinuous deformations, especially fractures. In peridynamics, considering a domain of interest, $\Omega \subset \mathbb{R}^d$, the equation of motion is given by

$$\rho(x)\ddot{y}(x, t) = \int_{B_\delta(x)} f(q, x, t) dq + b(x, t), \quad (x, t) \in \Omega \times [0, T], \quad (2.1)$$

where x and q are material points in the reference (undeformed) configuration of the body and y is the deformation map. $\rho(x)$ is the mass density function. $B_\delta(x)$ is a ball centered at x of radius (horizon) δ that determines the extent of the long-range interactions. $b(x, t)$ is the body force density (external loading), which is assumed to be prescribed. $f(q, x, t)$ is the pairwise bond force density¹ (dimensions of force per unit volume squared in 3D) that q exerts on x . For any admissible pairwise bond force density field, it is required that

$$f(q, x, t) = -f(x, q, t) \quad (2.2)$$

for all x , q , and t , which ensures that linear momentum is balanced globally, as discussed further below.

The material model determines $f(q, x, t)$ for every x and q , for all t , and for every possible deformation. To specify a material model, objects called *states* are used. A state is a mapping from a bond $q - x$ to some other quantity, usually either a vector or a scalar. The notation for states is as follows:

$$\underline{A}[x, t](\xi) = \nu, \quad (2.3)$$

where ν is the value of the state evaluated at the bond $\xi = q - x$. The square brackets indicate that the state itself is defined at material point x and time t . Two useful examples of scalar-valued states are given by:

$$\underline{x}(\xi) = |\xi|, \quad \underline{1}(\xi) = 1, \quad \text{for all bonds } \xi. \quad (2.4)$$

¹ There is not necessarily a direct physical interaction mechanism, like an electrostatic field, since the interaction in peridynamics can be indirect.

Two vector-valued states that are fundamental to material modeling are the *deformation state* \underline{Y} and the *force state* \underline{T} . The deformation state maps bonds (which are defined in the reference configuration) to their images under the deformation:

$$\underline{Y}[\mathbf{x}, t](\mathbf{q} - \mathbf{x}) = \mathbf{y}(\mathbf{q}, t) - \mathbf{y}(\mathbf{x}, t). \quad (2.5)$$

The force state contains the contribution of the material model at a point to the bond force density:

$$\mathbf{f}(\mathbf{q}, \mathbf{x}, t) = \underline{T}[\mathbf{x}, t](\mathbf{q} - \mathbf{x}) - \underline{T}[\mathbf{q}, t](\mathbf{x} - \mathbf{q}). \quad (2.6)$$

The two terms in right hand side of (2.6) reflect the fact that the force states at both endpoints \mathbf{x} and \mathbf{q} of a bond contribute to the pairwise force density $\mathbf{f}(\mathbf{q}, \mathbf{x}, t)$. A material model $\hat{\underline{T}}$ supplies the force state \underline{T} at \mathbf{x} through a dependence on the deformation of the family of \mathbf{x} , as characterized by the deformation state:

$$\underline{T}[\mathbf{x}, t] = \hat{\underline{T}}(\underline{Y}[\mathbf{x}, t]).$$

If the body is heterogeneous, the material model depends explicitly on position:

$$\underline{T}[\mathbf{x}, t] = \hat{\underline{T}}(\underline{Y}[\mathbf{x}, t], \mathbf{x}).$$

Material models can also depend on temperature, rate of deformation, and other variables, but these are not needed for present purposes. In (2.6), the terms on the right hand side are provided by the material model $\hat{\underline{T}}$ applied at the points \mathbf{x} and at \mathbf{q} :

$$\underline{T}[\mathbf{x}, t] = \hat{\underline{T}}(\underline{Y}[\mathbf{x}, t], \mathbf{x}), \quad \underline{T}[\mathbf{q}, t] = \hat{\underline{T}}(\underline{Y}[\mathbf{q}, t], \mathbf{q}). \quad (2.7)$$

The next task is to define specific material models. It is helpful to define the *point product* of a scalar-valued state \underline{a} with another state \underline{A} by

$$(\underline{a} \underline{A})(\xi) = \underline{a}(\xi) \underline{A}(\xi) \quad \text{for all bonds } \xi.$$

In all previous applications of peridynamics known to the authors, the material model is given by an algebraic expression, for example

$$\hat{\underline{T}}(\underline{Y}) = 3\underline{e} \underline{M}, \quad (2.8)$$

where \underline{e} is the *extension state*, which contains the change in length of each bond as it deforms:

$$\underline{e}(\xi) = |\underline{Y}(\xi)| - |\xi|. \quad (2.9)$$

\underline{M} is a state that contains unit vectors in the direction of the deformed bonds:

$$\underline{M}(\xi) = \frac{\underline{Y}(\xi)}{|\underline{Y}(\xi)|}. \quad (2.10)$$

In the following discussion, it is convenient to introduce the generic relative displacement vector between the endpoints of a bond, defined by $\boldsymbol{\eta} := \mathbf{u}(\mathbf{q}, t) - \mathbf{u}(\mathbf{x}, t)$. In terms of $\boldsymbol{\eta}$ the deformation state \underline{Y} can be written as

$$\underline{Y}[\mathbf{x}, t](\mathbf{q} - \mathbf{x}) = \xi + \boldsymbol{\eta}, \quad (2.11)$$

while \underline{e} and \underline{M} can be expressed in the form

$$\underline{e}(\xi) = |\xi + \boldsymbol{\eta}| - |\xi|, \quad \underline{M}(\xi) = \frac{\xi + \boldsymbol{\eta}}{|\xi + \boldsymbol{\eta}|}. \quad (2.12)$$

All the material models developed in the present work are *ordinary*, meaning that the bond force vectors in the force state are always parallel to the deformed bonds. Thus, for any \underline{Y} , there is a scalar-valued state $\underline{t}(\underline{Y})$ such that

$$\hat{\underline{T}}(\underline{Y}) = \underline{t}(\underline{Y}) \underline{M} \quad \text{for all bonds } \xi. \quad (2.13)$$

All the material models in this paper are also *mobile*, which means that the scalar force state in (2.13) depends only on the length changes of the bonds²:

$$\hat{\underline{T}}(\underline{Y}) = \underline{t}(\underline{e}) \underline{M}, \quad (2.14)$$

where \underline{e} is given by (2.9). Clearly, mobile implies ordinary.

Combining (2.1), (2.6), and (2.13), we obtain the following governing equation

$$\rho(\mathbf{x}) \ddot{\mathbf{y}}(\mathbf{x}, t) = \int_{B_\delta(0)} (\underline{t}[\mathbf{x}, t](\xi) + \underline{t}[\mathbf{x} + \xi, t](-\xi)) \underline{M}[\mathbf{x}, t](\xi) d\xi + \mathbf{b}(\mathbf{x}, t), \quad (2.15)$$

where the identity $\underline{M}[\mathbf{x} + \xi, t](-\xi) = -\underline{M}[\mathbf{x}, t](\xi)$ has been used.

Now consider the status of peridynamics with respect to the basic laws of physics.

² The term “mobile” was coined because of the slight resemblance of these materials to works of art that are suspended from the ceiling and have different parts that rotate freely, independent of each other.

- *Conservation of linear momentum.* First, observe that a pairwise force density f of the form (2.6) necessarily satisfies the antisymmetry requirement (2.2). It is shown (Proposition 7.1 of [70]) that if this antisymmetry condition holds, then the global balance of linear momentum holds in a bounded body B that is free of constraints such as Dirichlet boundary conditions:

$$(2.2) \implies \frac{d}{dt} \int_B \rho(\mathbf{x}) \dot{\mathbf{y}}(\mathbf{x}, t) d\mathbf{x} = \int_B \mathbf{b}(\mathbf{x}, t) d\mathbf{x}. \quad (2.16)$$

Mechanically, the antisymmetry condition means that each endpoint of any bond experiences a force vector that is equal in magnitude and opposite in direction to the other, so the result (2.16) is very reasonable.

- *Conservation of angular momentum.* It is shown that a body composed of an ordinary material necessarily satisfies the global balance of angular momentum (Proposition 8.2 of [70]). That is,

$$(2.13) \implies \frac{d}{dt} \int_B \rho(\mathbf{x}) \mathbf{y}(\mathbf{x}, t) \times \dot{\mathbf{y}}(\mathbf{x}, t) d\mathbf{x} = \int_B \mathbf{y}(\mathbf{x}, t) \times \mathbf{b}(\mathbf{x}, t) d\mathbf{x}. \quad (2.17)$$

The mechanical picture in an ordinary material is that since the bond force is parallel to the bond direction, each endpoint of a bond exerts no net moment on the other, thus ensuring the balance of angular momentum.

The following two invariances do not have the same status as the conservation laws but are required for physical reasonableness:

- *Galilean invariance.* This requirement, also called translational invariance, means that if a body is rigidly translated, its response to deformation remains unchanged. This invariance is built into the basic structure of peridynamic material modeling. To see this, recall that according to (2.7), the bond forces depend only on \underline{Y} . If the body is translated through a constant vector $\mathbf{C} \in \mathbb{R}^d$, then from (2.5),

$$\underline{Y}[\mathbf{x}, t](\mathbf{q} - \mathbf{x}) = (\mathbf{C} + \mathbf{y}(\mathbf{q}, t)) - (\mathbf{C} + \mathbf{y}(\mathbf{x}, t)) = \mathbf{y}(\mathbf{q}, t) - \mathbf{y}(\mathbf{x}, t), \quad (2.18)$$

leaving \underline{Y} unchanged. The translation does not change \mathbf{x} and \mathbf{q} , which are defined in the reference configuration. Thus, any peridynamic material model necessarily satisfies Galilean invariance without additional conditions.

- *Objectivity.* Objectivity, also called material frame invariance, is satisfied by a material model if for any proper orthogonal tensor $\mathbf{R} \in \mathbb{R}^{d \times d}$,

$$\hat{\mathbf{T}}(\underline{Y}') = \mathbf{R} \hat{\mathbf{T}}(\underline{Y}), \quad \text{where} \quad \underline{Y}'(\xi) = \mathbf{R} \underline{Y}(\xi). \quad (2.19)$$

Mechanically, objectivity means that if a body is rigidly rotated *after* it is deformed, then the bond force vectors can be found by rotating the bond force vectors that would be present in the unrotated deformation. Objectivity should not be confused with isotropy, which pertains to what happens if the body is rotated *before* deforming it. It is proved in Proposition 9.3 of [73] that every mobile material model satisfies objectivity, that is, (2.14) \implies (2.19).

Many material models have been developed for peridynamics, and any material model from the local theory can be translated into peridynamic form [73–76]. The most widely used capability that peridynamics offers that is not available in the local theory is the direct modeling of fracture within the basic field equations, since the equation of motion (2.1) is an integro-differential equation that does not involve the partial derivatives of displacement with respect to position [56,77–80]. However, although the present paper concerns both infinitesimal and finite deformations of material responses, we do not address fracture. The extension of the methods described here to include fracture modeling is under investigation in separate work.

In contrast to the usual practice of using an algebraic expression like (2.8) for the material model, in the present work, the material model is provided by a *nonlocal neural operator*. As discussed further below, the mathematical structure of the implementation ensures its physical admissibility. The peridynamic neural operator, described in the next Section, provides an ordinary, mobile material model. It is helpful to first introduce the general structure of this type of model. A typical ordinary material model is the linear peridynamic solid (LPS) [70], which is similar to an isotropic linear elastic solid in the PDE theory. The LPS can be written in the following form:

$$\underline{t}(\xi) = C_1 \vartheta \underline{\omega}(\xi) \underline{x}(\xi) + C_2 \underline{\omega}(\xi) \underline{e}(\xi) = \underline{\omega}(\xi) (C_1 \vartheta \underline{x}(\xi) + C_2 \underline{e}(\xi)). \quad (2.20)$$

Here the *nonlocal dilatation* is defined by

$$\vartheta = \frac{d \int_{B_\delta(0)} \underline{\omega}(\xi) \underline{e}(\xi) |\xi| d\xi}{\int_{B_\delta(0)} \underline{\omega}(\xi) |\xi|^2 d\xi}. \quad (2.21)$$

In (2.20), C_1 and C_2 are constants that are chosen so that the model reproduces the correct bulk modulus and shear modulus, and $\underline{\omega}$ is a weighting function called the *influence state*. In an isotropic expansion of a body with a small volume change, the nonlocal dilatation defined by (2.21) closely approximates the usual dilatation from the PDE theory (the trace of the strain tensor). An important point is that in the LPS, although ϑ depends on the deformation of all the bonds, this dependence appears only through the changes in bond length. Thus the LPS is consistent with (2.14) as well as (2.13), so it is both ordinary and mobile.

In general, the functional form of \underline{t} and $\underline{\omega}$ as they depend on ξ cannot be established a priori for a given material. The integral constitutive laws incorporate the nonlinear and heterogeneous responses of the system and must be chosen to reproduce these possibly complex responses separately for each material. To accomplish this, our proposed method learns \underline{t} and $\underline{\omega}$ from data using neural networks. As such, we obtain a data-driven nonlocal constitutive law in the form of a neural operator architecture, as described in Section 3.

2.2. Nonlocal neural operators

We now introduce the basic concepts and architectures of the general nonlocal neural operators [23,44,67,68,81], which were developed for scientific computing applications entailing the learning of solution operators. A prototypical instance is the case of solving the governing PDE in material modeling problems, where the initial input field (body load/boundary load) is mapped to the corresponding displacement field via a nonlinear parameterized mapping. The demand for operator learning has sparked the development of several architectures in neural operator based methods [19,22–24,44,67,68,81–85,85–87]. Contrary to classical NNs that operate between finite-dimensional Euclidean spaces, neural operators are designed to learn mappings between infinite-dimensional function spaces [19,44,67,68,81–83,88–90]. A remarkable advantage of neural operators lies in their resolution independence, which implies that the prediction accuracy is invariant to the resolution of input functions. Furthermore, in contrast to classical PDE-based approaches, neural operators can be trained directly from data, and hence requires no domain knowledge nor pre-assumed PDEs. All these advantages make neural operators a promising tool for learning complex material responses [19–22,44,47,67,81].

Herein, we focus on the nonlocal neural operator introduced in [44], which has foundation in the representation of the solution of a PDE by the Green's function. For the purpose of introduction, we focus on the static PDE solving scenario, which aims to construct a surrogate operator $\mathcal{G} : \mathbb{F} \rightarrow \mathbb{U}$ that maps the input function $\mathbf{b}(\mathbf{x})$ to the output function $\mathbf{u}(\mathbf{x})$. In the context of material modeling [22,23], \mathbf{b} stands for the boundary and body loads, and \mathbf{u} is the corresponding displacement or damage fields. This architecture is comprised of three building blocks. First, the input function, $\mathbf{b}(\mathbf{x}) \in \mathbb{F}$, is lifted to a higher-dimension representation via a local affine pointwise mapping $\mathbf{h}(\mathbf{x}, 0) = \mathcal{P}[\mathbf{b}](\mathbf{x})$. Next, the feature vector function $\mathbf{h}(\mathbf{x}, 0)$ goes through an iterative layer block where the layer update is defined via nonlinear operator layers: $\mathbf{h}(\cdot, l+1) = \mathcal{J}[\mathbf{h}(\cdot, l)]$, for $l = 0, \dots, L-1$, which will be further discussed in the later contents. Here, $\mathbf{h}(\cdot, l)$ is the l -th layer network feature, which is a dimension- d_h vector-valued function defined on Ω . Finally, the output $\mathbf{u}(\cdot) \in \mathbb{U}$ is obtained through a projection layer: $\mathbf{u}(\mathbf{x}) = \mathcal{Q}[\mathbf{h}(\cdot, L)](\mathbf{x})$.

In nonlocal neural operators, the resolution-independence property is realized by parameterizing the layer update, \mathcal{J} , as an integral operator. In particular, for an L -layer NN, the l -th layer network update is given as:

$$\mathbf{h}(\mathbf{x}, l+1) = \mathcal{J}[\mathbf{h}(\cdot, l)](\mathbf{x}) := \sigma \left(\mathbf{R}\mathbf{h}(\mathbf{x}, l) + \int_{\Omega} \mathbf{K}(\mathbf{x}, \mathbf{q}, \mathbf{b}(\mathbf{x}), \mathbf{b}(\mathbf{q}); \boldsymbol{\nu}) \mathbf{h}(\mathbf{q}, l) d\mathbf{q} + \mathbf{c} \right). \quad (2.22)$$

Here, σ is an activation function, $\mathbf{R} \in \mathbb{R}^{d_h \times d_h}$ is a trainable tensor, $\mathbf{c} \in \mathbb{R}^{d_h}$ is a trainable vector, and $\mathbf{K} \in \mathbb{R}^{d_h \times d_h}$ is a tensor kernel function that takes the form of a (usually shallow) NN whose parameters $\boldsymbol{\nu}$ are to be learned. While in the original version of nonlocal neural operator the integral is extended to the whole set Ω , for efficiency purposes, restrictions to a ball of radius δ centered at \mathbf{x} , i.e. $B_\delta(\mathbf{x})$, can also be considered. However, as expected, this choice might compromise the accuracy since the support of Green's function generally spans the whole domain in PDE solving problems [44].

The implementation of nonlocal neural operators (2.22) resembles graph neural networks in that both follow a message passing framework [91,92], while the discrete aggregation step is substituted by a meshfree approximation of the continuous integral operator [93]. In particular, taking a discretization of \mathbf{u} on a collection of points $\chi = \{\mathbf{x}_j\}_{j=1}^M \subset \Omega$, the kernel integration can be viewed as an aggregation of messages by taking the edge feature:

$$e(\mathbf{x}_i, \mathbf{x}_j) := \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{b}(\mathbf{x}_i), \mathbf{b}(\mathbf{x}_j); \boldsymbol{\nu}), \quad (2.23)$$

and the node feature update:

$$\mathbf{h}(\mathbf{x}_i, l+1) = \sigma \left(\mathbf{R}\mathbf{h}(\mathbf{x}_i, l) + \sum_{\mathbf{x}_j \in N(\mathbf{x}_i)} e(\mathbf{x}_i, \mathbf{x}_j) \mathbf{h}(\mathbf{x}_j, l) \mu(\mathbf{x}_j) + \mathbf{c} \right), \quad (2.24)$$

where $N(\mathbf{x}_i)$ is the neighborhood of \mathbf{x}_i according to the graph on χ and $\mu(\mathbf{x}_j)$ is the Riemannian sum weight. Since this nonlocal layer update is consistent with a continuous integral operator, the learned network parameters are resolution-independent: the learned $\boldsymbol{\nu}$, \mathbf{R} , and \mathbf{c} can be reused even when the grid set, χ , varies, since they correspond to the same integral operator \mathcal{J} . Therefore, the nonlocal architecture provides the key for resolution independence.

Despite the aforementioned advances of neural operators, purely data-driven neural operators need to learn the basic physical laws from data, and therefore suffer from data challenge. In particular, in order to generalize the solution, they require a large corpus of paired datasets, which is prohibitively expensive in many engineering applications. To resolve this challenge, the physics-informed neural operator (PINO) [94] and physics-informed DeepONets [19,95] are introduced, where a PDE-based loss is added to the training loss as a penalization term. However, these approaches still require *a priori* knowledge of the underlying PDEs, which restricts their applications to (known) constitutive law-solving tasks. As another restriction, since most neural operator model learns the material responses as a surrogate solution operator, they are not generalizable to different domain geometries and loading scenarios. That means, an operator learned on square-domain samples cannot be immediately applied to predict the material deformation on a circular domain. To resolve these two limitations, in this work we parameterize the nonlocal neural operator architecture as a nonlocal constitutive law based on the ordinary state-based peridynamics formulation (2.13). As such, the model preserves the fundamental physical laws and is generalizable to different domain geometries and loading scenarios.

3. Peridynamic neural operators

3.1. Proposed architecture

In this Section, we introduce the Peridynamic Neural Operator (PNO) framework, which learns ordinary state-based peridynamic constitutive models from data. It uses the nonlocal neural operator architecture [44,68]. The PNO is used as the surrogate for the integrand functions in (2.1). The model is designed to represent diverse features of material behaviors, such as anisotropy and nonlinearity. By taking the form of (2.13) and (2.14), the proposed operator inherits the physical compatibility of the mobile ordinary state-based materials within the peridynamic theory of mechanics. This guarantees the required invariances under frame translation and rotation, as well as the balances of linear and angular momentum.

In the following, it is convenient to use displacement rather than the deformation map directly. These are related by

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{y}(\mathbf{x}, t) - \mathbf{x} . \quad (3.1)$$

Formally, we attempt the learning of complex physical responses in a mechanical system, based on a set of observations $\mathcal{D} = \{\mathbf{b}^s, \mathbf{u}^s\}_{s=1}^S$ of the loading field $\mathbf{b}^s(\mathbf{x}, t)$ and the corresponding displacement field $\mathbf{u}^s(\mathbf{x}, t)$, where s is the sample index. Inspired by the peridynamic formulation (2.15), we propose to learn a nonlocal neural operator \mathcal{G} , which acts as a surrogate operator for the constitutive law such that:

$$\mathcal{G}[\mathbf{u}^s](\mathbf{x}, t) \approx \rho(\mathbf{x}) \ddot{\mathbf{u}}^s(\mathbf{x}, t) - \mathbf{b}^s(\mathbf{x}, t) , \quad (3.2)$$

where the operator \mathcal{G} is formulated as:

$$\mathcal{G}[\mathbf{u}](\mathbf{x}, t) := \int_{B_\delta(\mathbf{0})} (\mathbf{t}[\mathbf{x}, t]\langle \xi \rangle + \mathbf{t}[\mathbf{x} + \xi, t]\langle -\xi \rangle) \underline{\mathbf{M}}[\mathbf{x}, t]\langle \xi \rangle d\xi .$$

Herein, we parameterize the scalar force state and the influence state with neural networks:

$$\mathbf{t}[\mathbf{x}, t]\langle \xi \rangle := \sigma^{NN}(\omega(\xi), \vartheta(\mathbf{x}, t), e(\xi, \eta), |\xi|; \mathbf{v}) , \quad (3.3)$$

and

$$\omega(\xi) := \omega^{NN}(\xi; \mathbf{w}) , \quad (3.4)$$

where σ^{NN} and ω^{NN} are scalar-valued functions that take the form of a (usually shallow) multi-layer perceptron (MLP) with learnable parameters \mathbf{v} and \mathbf{w} , respectively. $e(\xi, \eta) := |\xi + \eta| - |\xi|$. As a generalization of the nonlocal dilatation defined in (2.21), we set

$$\vartheta(\mathbf{x}, t) := \frac{\int_{B_\delta(\mathbf{0})} \omega^{NN}(\xi; \mathbf{w}) e(\xi, \eta) |\xi| d\xi}{\int_{B_\delta(\mathbf{0})} \omega^{NN}(\xi; \mathbf{w}) |\xi|^2 d\xi} . \quad (3.5)$$

If the material is known to be isotropic a-priori, the following simplification of (3.4) holds:

$$\omega(\xi) := \omega^{NN}(|\xi|; \mathbf{w}) . \quad (3.6)$$

Given a forcing term \mathbf{b}^s , in order to guarantee the existence of a unique solution \mathbf{u}^s , nonlocal boundary conditions (“volume constraints”) must be prescribed on an appropriate *interaction domain* Ω_I . Without loss of generality, the Dirichlet boundary condition is applied, with $\Omega_I = \{\mathbf{x} | \mathbf{x} \in \mathbb{R}^d \setminus \Omega, \text{dist}(\mathbf{x}, \Omega) < 2\delta\}$. As will be elaborated in our experiments of Section 4, the measurements of the displacement field \mathbf{u}^s are defined either with a periodic boundary condition or properly extended using mirror boundary conditions [96], and therefore they are available on $\overline{\Omega} := \Omega \cup \Omega_I$.

Since the layer update in the above PNO is formulated as a continuous integral operator, the learned network parameters are resolution-independent: the learned \mathbf{v} and \mathbf{w} are close to optimal even when used with different resolutions. Importantly, since it is implemented in the form of an ordinary, mobile material model, the PNO preserves the balance of linear and angular momentum and satisfies objectivity, as discussed previously. Similar to the vanilla nonlocal neural operators, PNO can be implemented using a two-layer message passing framework. In particular, for each grid point $\mathbf{x}_i \in \chi$ and time instance t , we calculate the first edge feature in the vanilla nonlocal neural operator as:

$$e^1(\mathbf{x}_i, \mathbf{x}_j) := \omega^{NN}(\mathbf{x}_j - \mathbf{x}_i; \mathbf{w}) , \quad (3.7)$$

and use it to update the first node feature:

$$\mathbf{n}^1(\mathbf{x}_i, t) := \frac{\sum_{N(\mathbf{x}_i)} e^1(\mathbf{x}_i, \mathbf{x}_j) \left(|\mathbf{x}_j + \mathbf{u}(\mathbf{x}_j, t) - \mathbf{x}_i - \mathbf{u}(\mathbf{x}_i, t)| - |\mathbf{x}_j - \mathbf{x}_i| \right) |\mathbf{x}_j - \mathbf{x}_i| \mu(\mathbf{x}_j)}{\sum_{N(\mathbf{x}_i)} e^1(\mathbf{x}_i, \mathbf{x}_j) |\mathbf{x}_j - \mathbf{x}_i|^2 \mu(\mathbf{x}_j)} . \quad (3.8)$$

Then, as the second layer, we update the second edge feature via:

$$e^2(\mathbf{x}_i, \mathbf{x}_j, t) := \sigma^{NN} \left(e^1(\mathbf{x}_i, \mathbf{x}_j), \mathbf{n}^1(\mathbf{x}_i, t), |\mathbf{x}_j + \mathbf{u}(\mathbf{x}_j, t) - \mathbf{x}_i - \mathbf{u}(\mathbf{x}_i, t)| - |\mathbf{x}_j - \mathbf{x}_i|, |\mathbf{x}_j - \mathbf{x}_i|; \mathbf{v} \right) , \quad (3.9)$$

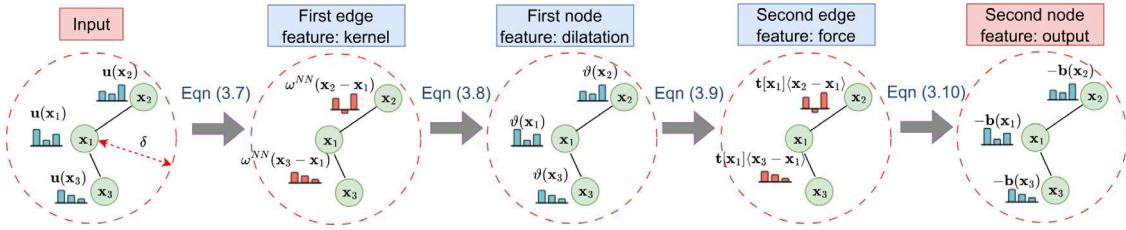


Fig. 1. A graph-based implementation of PNO architecture.

and finally the second node feature as the operator output:

$$\mathcal{G}[\mathbf{u}](\mathbf{x}_i, t) = \mathbf{n}^2(\mathbf{x}_i, t) := \sum_{N(\mathbf{x}_i)} (\mathbf{e}^2(\mathbf{x}_i, \mathbf{x}_j, t) + \mathbf{e}^2(\mathbf{x}_j, \mathbf{x}_i, t)) \frac{\mathbf{x}_j + \mathbf{u}(\mathbf{x}_j, t) - \mathbf{x}_i - \mathbf{u}(\mathbf{x}_i, t)}{|\mathbf{x}_j + \mathbf{u}(\mathbf{x}_j, t) - \mathbf{x}_i - \mathbf{u}(\mathbf{x}_i, t)|} \mu(\mathbf{x}_j). \quad (3.10)$$

Therefore, our PNOs share similar properties to the peridynamic models in its ability to preserve basic physical properties, and can also be seen as a special architecture of nonlocal neural operators.

3.2. Learning algorithm

Algorithm 1 Algorithm for PNOs.

```

1: Pre-processing Phase:
2: Read data and inputs:  $\chi = \{\mathbf{x}_i\}_{i=1}^N$ ,  $D_{tr} := \{\mathbf{u}_i^{s,tr}, \mathbf{b}_i^{s,tr}\}_{i=1,s=1}^{N,S_{tr}}$  for training,  $D_{val} := \{\mathbf{u}_i^{s,val}, \mathbf{b}_i^{s,val}\}_{i=1,s=1}^{N,S_{val}}$  for validation sets.
3: Find and store the peridynamic neighbor node set  $N(\mathbf{x}_i)$  in the form of message passing graph structure:
4: for  $i = 1 : N$  do
5:   Find all  $\mathbf{x}_j$  such that  $|\mathbf{x}_j - \mathbf{x}_i| < \delta$ .
6: Training Phase:
7: for  $ep = 1 : epoch_{max}$  do
8:   Initialize error:  $E_{tr} = 0$ 
9:   for each batch do
10:    Reset batch loss:  $loss = 0$ 
11:    if Case I then
12:      Compute batch loss following (3.11).
13:    else if Case II then
14:      For each training sample in this batch, run nonlinear solver to obtain  $\mathcal{G}^{-1}[\mathbf{b}^s]$ .
15:      Compute batch loss following (3.12).
16:    Update PNO parameters  $\mathbf{w}, \mathbf{v}$  with the Adam optimizer.
17:  Compute training error  $E_{tr}$ .
18: Validation Phase:
19: if training error decreases then
20:   Initialize validation error:  $E_{val} = 0$ 
21:   if Case I then
22:     Compute validation error following (3.11).
23:   else if Case II then
24:     For each validation sample, run nonlinear solver to obtain  $\mathcal{G}^{-1}[\mathbf{b}^s]$ .
25:     Compute validation error following (3.12).
26:   If validation error also decreases, save the current model.

```

In this Section we elaborate the learning algorithm of PNOs, with its architecture and implementation steps demonstrated in Fig. 1. Although the prescribed architecture is readily applicable to higher-dimensional domains and dynamic cases, in this work we focus on (quasi)static and two-dimensional ($d = 2$) tasks. In this case, our goal is to learn the surrogate operator \mathcal{G} such that

$$\mathcal{G}[\mathbf{u}](\mathbf{x}) \approx -\mathbf{b}(\mathbf{x}),$$

Algorithm 1 lays out the structure of the pseudo code used for this work to learn the constitutive model, and we will explain more details below. The program is coded in Python and employs PyTorch libraries for GPU computing. The code uses the Adam optimizer in PyTorch for minimizing the loss function.

Formally, we assume that measurements are available in the following format:

- Coordinates of nodes where measurements are taken: $\chi = \{\mathbf{x}_i\}_{i=1}^N$, where N is the total number of nodes.

- External force–displacement function pairs at each node x_i and for each sample s : $\{\mathbf{u}_i^s, \mathbf{b}_i^s\}_{i=1,s=1}^{N,S}$ where S is the total number of samples.

Moreover, we split the available measurements into three data sets, and let S_{tr} , S_{val} , S_{test} denote the total number of samples in training, validation, and test sets, respectively.

The neural network functions for ω^{NN} and σ^{NN} are multi-layer perceptrons (MLP) with three layers and ReLu as the activation function. The widths of the MLP layers are denoted by $(W_{in}, W_{h1}, W_{h2}, W_{out})$, where W_{in} is the input's dimension, W_{h1}, W_{h2} are the widths of the hidden layers, and W_{out} is the dimension of the output. For ω^{NN} , depending on the assumption of isotropy or anisotropy through (3.6) or (3.4), W_{in} can be 1 or 2. In this study, $W_{out} = 1$ for all examples, although it can take on larger values in other applications. From (3.3), since $|\xi|$ and $e(\xi, \eta)$ are scalars and $\vartheta(\mathbf{x})$ has the same dimension as $\omega(\xi)$ (see (3.5)), it follows that $W_{in} = 4$ and $W_{out} = 1$ in the σ^{NN} MLP. W_{h1}, W_{h2} in both ω^{NN} and σ^{NN} as well as the horizon size δ are hyperparameters of choice to tune. The larger these values are, the more expressive (and more expensive) the PNO becomes.

Depending on whether the data includes nonzero external forces (Case I) or not (Case II), either of the following two different loss functions is used. In Case I, we follow the convention of nonlocal neural operator literatures [44,67] and consider the relative L^2 error of the output function, \mathbf{b} , as the loss function, which is given by

$$\text{loss}_b = \frac{1}{S_{tr}} \sum_{s=1}^{S_{tr}} \frac{\|G[\mathbf{u}^s] + \mathbf{b}^s\|}{\|\mathbf{b}^s\|}. \quad (3.11)$$

Physically, the loss function in (3.11) measures the error in the internal force distribution on the nodes needed to equilibrate the system.

If external forces are zero (Case II), the loss function (3.11) fails in that the denominator becomes zero. If the denominator is removed to avoid division by zero, the program leads to the trivial solution for the model parameters, such that with any deformation the model returns zero forces. To circumvent this problem, in Case II, we define the loss in terms of the displacement field:

$$\text{loss}_u = \frac{1}{S_{tr}} \sum_{s=1}^{S_{tr}} \frac{\|\mathcal{G}^{-1}[\mathbf{b}^s] - \mathbf{u}^s\|}{\|\mathbf{u}^s\|}, \quad (3.12)$$

where $\mathcal{G}^{-1}[\mathbf{b}^s]$ denotes the numerical solution of $\mathcal{G}[\mathbf{u}] = -\mathbf{b}^s$ using an iterative nonlinear static solver (see line 21 in Algorithm 1). In this work we use the Polak–Ribiere conjugate gradient method [97], due to its efficiency and robustness in peridynamics nonlinear problems [98]. Additional inputs needed for the solver are \mathbf{u}_0^s , \mathbf{u}_{BC}^s , tol , itr_{max} , which denote the initialization displacement (which may or may not be zero), Dirichlet boundary condition values on Ω_I , convergence tolerance, and maximum allowed iterations, respectively.

Two difficulties arise in Case II. Firstly, the computational time significantly increases due to the multi-iteration procedure per sample and per epoch. Secondly and more importantly, each static-solver iteration (which includes at least one model evaluation) is technically a part of the loss function in the sense that it participates in auto-differentiation/back-propagation by the optimizer. This procedure requires a relatively large memory allocation compared with the Case I where the loss function is defined for forces and the model is evaluated just once for each sample in each epoch. Consequently, in Case II, the memory allocation strongly depends on the number of iterations the nonlinear solver converges in. In practice, loss in Case I is the preferred option, and Case II is an alternative employed only when there are no external forces, such as in the scenario of our experimental data for the tissue.

To make the Case II algorithm affordable, considering hardware limitations, one can take the following measures to reduce the number of solver's iterations:

1. Avoiding the use of too small tolerances for convergence criterion.
2. Smart initialization of the solution. Examples of smart initialization values include using displacements from another sample with a close measurement sequence, or using some form of interpolation of Dirichlet boundary conditions. For example, in this work we use experimentally measured displacements for the sample at the previous temporal sequence as the solver's initial displacement field for the current instance.
3. Limiting the maximum number of iterations to the lowest possible value at which convergence for most samples is likely to be achieved.

In Case I, a substantial number of different external loading fields can be used to train the PNO without excessive cost. However, in Case II, the training requires a full peridynamic solution for each instance of the training data, which greatly increases the cost. In Case II, we found it helps to modify the architecture of the scalar force state given by (3.3) as follows:

$$f[\mathbf{x}](\xi) := \sigma^{NN}(\omega(\xi), \vartheta(\mathbf{x}), e(\xi, \eta), |\xi|; \nu) e(\xi, \eta). \quad (3.13)$$

The new factor of $e(\xi, \eta)$ guarantees that a small deformation bond will be associated with a small force state, and it causes the larger deformation bonds to weigh more on the overall equilibrium deformation. Thus, the modified form (3.13) tends to improve the training efficiency.

Note that since PNO is a constitutive law, i.e. it is a pointwise mapping between deformation and the internal forces, it is independent of domain topology or geometry. Having that said, similar to loading profile and boundary conditions, topological features also affect the force–displacement patterns in the domain. Since PNO is trained on these data, topological features of the training samples can only in-directly influence the PNO accuracy by the amount of coverage they provide on force–displacement patterns.

In the next section, the PNO is used to learn constitutive models for three mechanically different materials. The first two are examples of Case I, and the third example is of Case II.

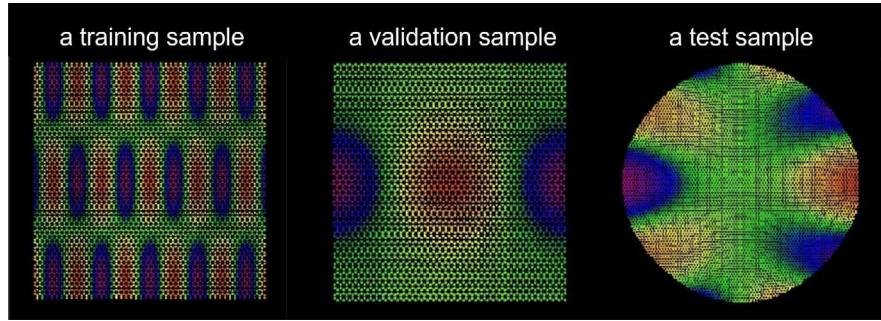


Fig. 2. Demonstration of typical training, validation, and testing samples from MD data sets in Example I: Graphene. Colors show the u_1 displacement component.

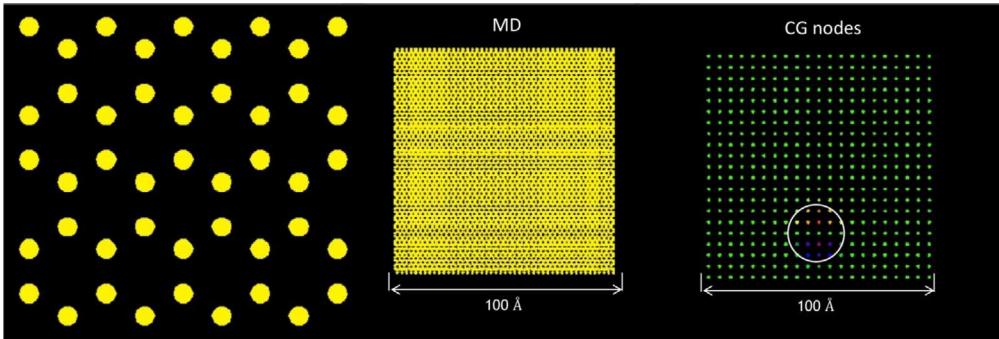


Fig. 3. Demonstration of data generation for Example I: Graphene. Left: hexagonal atomic lattice in MD. Center: full MD mesh. Right: 21×21 nodes coarse-grained nodes. Colors within the small circle show coarse-grained bond forces on the node at the center of the circle.

4. Numerical examples

In this section, the PNO is applied to learn the constitutive laws for three different materials with distinct characteristics:

1. A single layer graphene at zero temperature.
2. An anisotropic hyperelastic material going under large deformation.
3. A heterogeneous anisotropic biological tissue under large deformation.

The first example is chosen to demonstrate that the PNO is capable of learning a continuum constitutive law from microscale data, in this case using molecular dynamics (MD). The second example illustrates the ability of the method to capture anisotropy and material response under large deformation. While the first two examples are based on synthetic data sets, the third example uses real-world experimental data from a tissue specimen with heterogeneity and complex behaviors, and the measurements also contain unavoidable noises [23]. All tests are performed on a NVIDIA RTX 3090 GPU card with 24 GB memory.

4.1. Example I: Graphene

In this example we use PNO to learn a continuum constitutive model for single layer graphene at zero temperature, using synthetic data sets generated by molecular dynamics (MD) simulations.

Graphene is a two-dimensional form of carbon arranged in hexagonal lattice structure, with excellent mechanical properties and with a variety of applications [99]. The MD simulation uses the Tersoff interatomic potential [100] to compute the forces between the atoms. The MD code finds the equilibrium displacements of atoms subjected to predefined external force fields and boundary conditions. The details of the MD data generation can be found in [31,101]. Here, a summary of these data sets is provided with a focus on features relevant to the present study. The training and validation sets, with 70 and 10 samples respectively, model a $100 \text{ \AA} \times 100 \text{ \AA}$ 2D square region, consisting of hexagonally arranged atoms with an interatomic distance of 1.46 \AA and with periodic boundary conditions. The external forces in the training and validation sets are prescribed oscillatory functions of position with different wavelengths for each sample. The validation set uses external forces in both spatial directions that are significantly different from those used in the training set. The test set includes 4 samples on a disk of radius 100 \AA with a traction-free boundary. The external forces are applied on an exterior ring with 50 \AA distance from the outer edge, with zero force on the disk that is inside the ring. Fig. 2 shows examples of the training, validation, and test samples using MD. A coarse-graining method is applied to the MD data to obtain the displacements and forces on a uniform rectangular grid with a nodal spacing of $\Delta x = 5 \text{ \AA}$, which is coarser

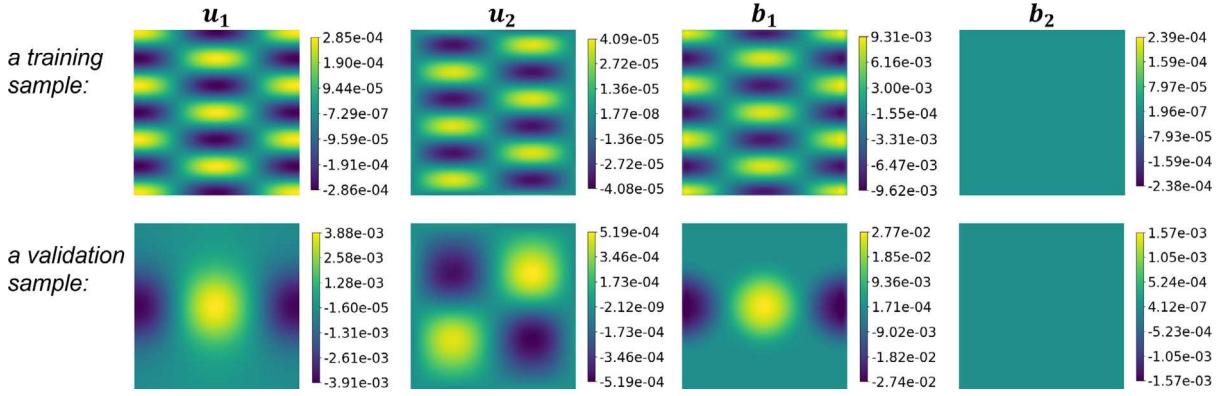


Fig. 4. The body force and the corresponding displacement field components for a training and a validation sample of the coarse-grained graphene data set.

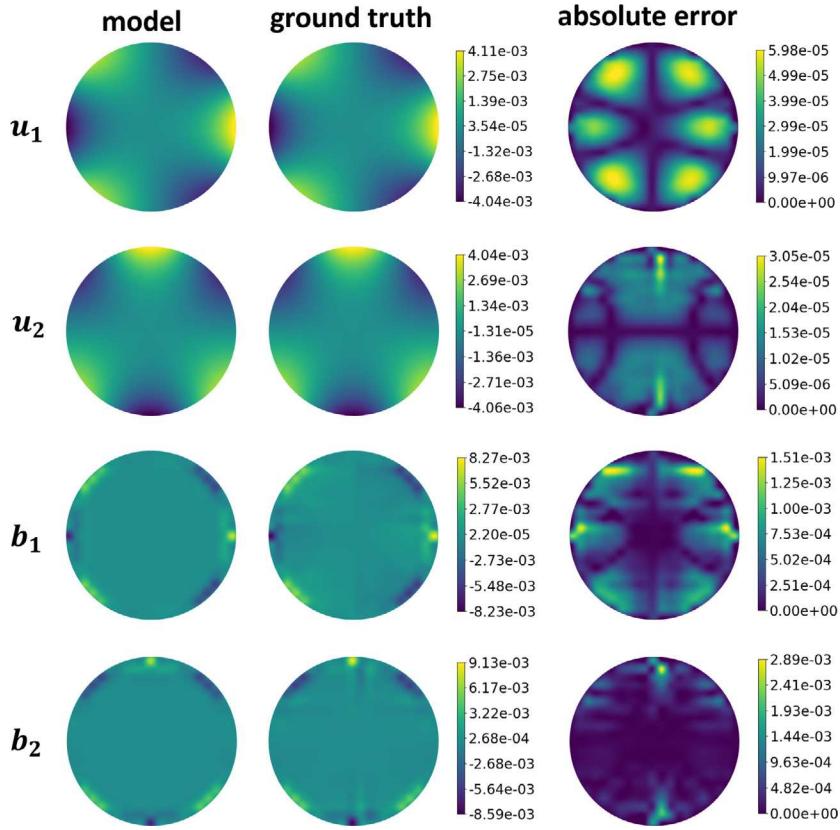


Fig. 5. PNO prediction results of Example I: Graphene. Comparison of the displacement field (given body force and boundary conditions) and the prediction of body force (given displacement field) versus the corresponding ground truth for a test sample.

than the original MD lattice. Fig. 3 shows the hexagonal lattice of atoms, the MD domain, and the resulting 21×21 coarse-grained domain for training and validation sets. The radius for the smoothing functions used in the coarse-graining method is chosen to be 10 Å, that is, twice the coarse grained grid spacing. The smoothing function used in the coarse graining algorithm determines the horizon δ in the coarse grained data. The coarse grained displacements no longer represent individual atoms, but instead represent a weighted average of their motions. More details of the coarse-graining method can be found in [31,101]. The displacement and external forces for typical training and validation samples are shown in Fig. 4.

The training and validation data sets in this problem had nonzero forces and followed the Case I procedure described in Algorithms 1. After training several PNO models with a range of hyperparameters in the optimizer, the best model was selected to minimize the average error (3.11) in the training and validation data sets. After selecting the best model, it was applied to the

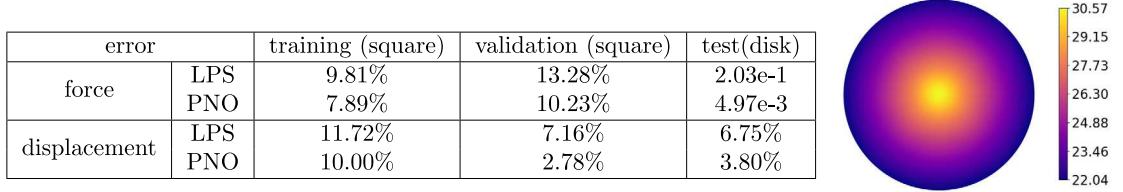


Fig. 6. PNO prediction results of Example I: Graphene. Left: L^2 -errors of the proposed PNO model, in comparison with the optimal LPS model [31]. Right: The learned kernel function $\omega^{NN}(|\xi|/\delta)$ of the PNO model.

test set, which, as noted above, is significantly different from the training set. Since graphene is isotropic at small deformations, the kernel function (3.6) was used. The MLP for ω^{NN} in the trained model had widths (1, 256, 512, 1), while the σ^{NN} MLP widths were (4, 512, 512, 1). The horizon size is taken as $\delta = 4\Delta x = 20 \text{ \AA}$, as suggested in [31]. The errors in both forces and displacement were computed for all three data sets.

As noted in that algorithm, two different errors were reported to show the accuracy of the trained PNO: (1) the error in the predicted internal force for a given the displacement field; and (2) the error in the predicted displacement for a given combination of the boundary conditions and external forces. For training and validation sets, the Dirichlet BC in Ω_I were obtained by extending the prescribed displacement field from the original $100 \text{ \AA} \times 100 \text{ \AA}$ domain into the ring (Ω_I) using the periodic boundary condition. For the test set, since the domain is circular and not periodic, the solver computed the displacements on the interior disk with the radius of 50 \AA , treating the displacement on the exterior ring with the thickness of 2δ as the Dirichlet BC.

During training, we follow the Case I procedure and use the relative L^2 error of forces as the training and validation criteria. In the nonlinear solver, we initialize the displacement field as $u_0^s = 0$ for all cases. Then, we compute the relative L^2 errors for training, validation, and test sets.³ The predicted displacements and predicted forces are compared against ground-truth values for a test sample in Fig. 5. In the left of Fig. 6 we report the computed errors from the trained PNO, compared with those from an optimal LPS model, i.e., a peridynamic model of the form (2.20) with an algebraic kernel function optimized on the same training and validation data sets [31]. One can clearly see the improvement of the present PNO model over the previous data-driven approach where the focus was only on learning a particular kernel function, ω . The PNO allows the force state function to be learned from the data while adhering to the fundamental structure of an ordinary-state-based peridynamics and hence having material objectivity and the balance of angular momentum guaranteed. This comparison highlights the importance of learning the model form in terms of the force state \underline{t} together with the kernel function ω : the test error for forces has been improved by 97.6%, and the error for displacements has been improved by 43.7%. Further more, in the right plot of Fig. 6 we show the learned kernel function for graphene. While for this example ω^{NN} is a 1D function (only a function of bond length), it is shown in a 2D image to demonstrate how the values vary for neighbor points in the physical domain. Interestingly, this learned kernel shares some important features, positive values with the maximum in center and a monotonic decay in radial direction, with qualitative forms for the influence states that are typically assumed in the peridynamics literature [72,102].

4.2. Example II: Anisotropic hyperelastic material

The previous example demonstrated the learning of a continuum constitutive law from microscale simulations. The displacements in that example were in the range of small deformation and the material was known to be isotropic in this deformation regime. In the present example, we use the PNO to learn from the data for a large-deformation, hyperelastic, anisotropic material. As in the graphene example, the data in this example is also synthetic, but is generated with a classical PDE-based constitutive law and finite element method (FEM) code rather than MD. The sample is a 2D sheet using the Holzapfel-Gasser-Ogden (HGO) [103] material model in plane stress. The material parameters given in [22] are used in the HGO model. The PNO for this material uses the kernel in the form of (3.4) which allows anisotropy through dependency on the bond vector, containing information on magnitude and direction. The material contains fibers that are aligned in the vertical (x_2) direction, making the response stiffer in that direction. This is unlike the radially symmetric form that is adequate for the graphene example, since graphene is isotropic.

In the left plot of Fig. 7 we demonstrate the setting of the FEM solver used for the data generation: A 1×1 square domain is uniformly meshed using 21×21 nodes, and hence the grid size $\Delta x = 0.05$. The domain is subjected to external forces generated from a Gaussian random field and zero Dirichlet BC on all the boundaries. The open-source FEM package FENiCS [104] is used to solve the static problem for 250 instances. In each instance, different external loads \mathbf{b} are applied, and the displacement field \mathbf{u} is computed. Fig. 7, on the right, shows one instance of the generated data, and Fig. 8 shows the distribution of the in-plane displacement gradient magnitude, implying a large-deformation regime. The data is then (randomly) divided into training, validation, and testing sets with 200, 25, 25 samples for each, respectively.

In the present example, we consider 1×1 square domains for the training and validation samples (as demonstrated in Fig. 10), all with (local) Dirichlet boundary conditions. To supply the required nonlocal Dirichlet data in the boundary volume Ω_I , a mirror-based fictitious nodes method (FNM) [96,105,106] is used to assign values within the extended boundary. The mirror-based FNM

³ Since the external forces on the interior disk were zero, on the test set we report the absolute error for the forces.

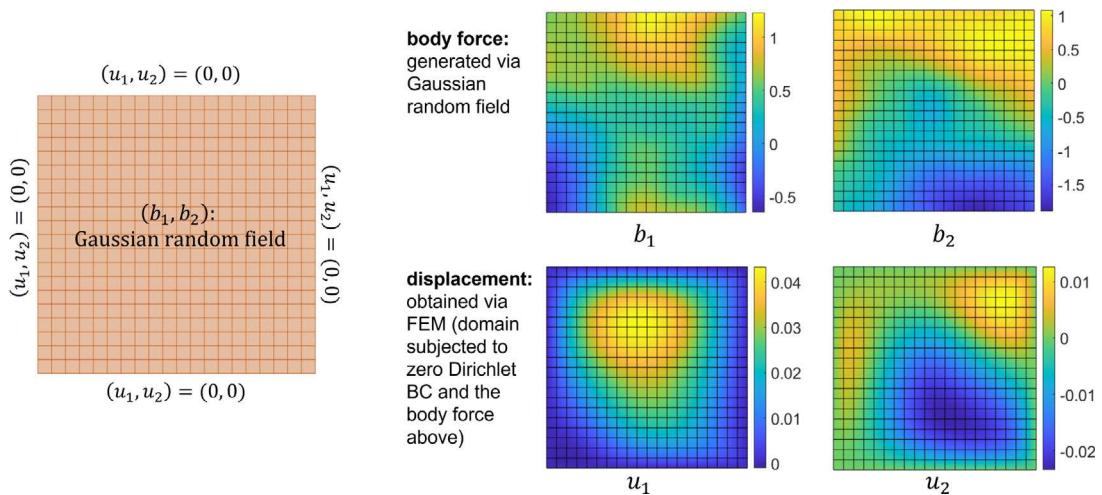


Fig. 7. Settings for data generation in Example II: Anisotropic Hyperelastic Material. Left: The geometry to generate synthetic hyperelastic data using FEM and HGO material model. Right: An example of the hyperelasticity material data generated via FEM and HGO material model.

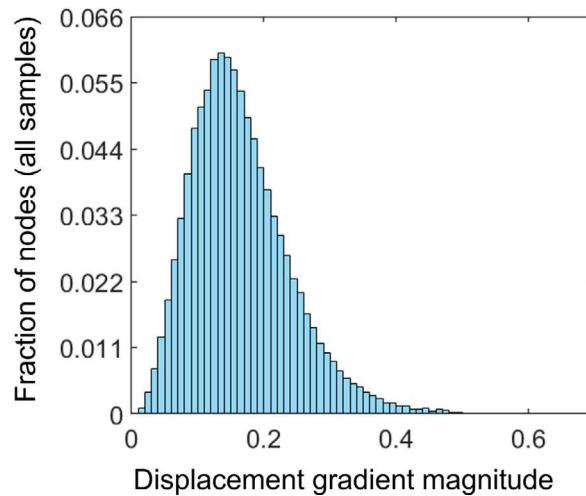


Fig. 8. Distribution of in-plane displacement gradient magnitude in Example II, considering all nodes in all samples.

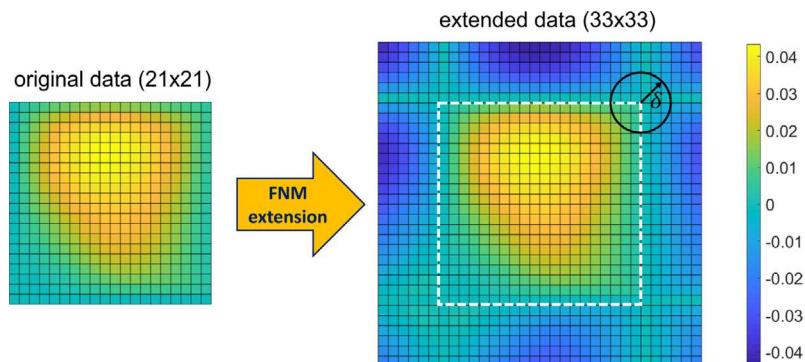


Fig. 9. Demonstration for data generation in Example II: Anisotropic Hyperelastic Material. Domain extension along the boundaries to with values assigned according to the mirror-based fictitious nodes method [105,106] to approximate local Dirichlet BC of zero.

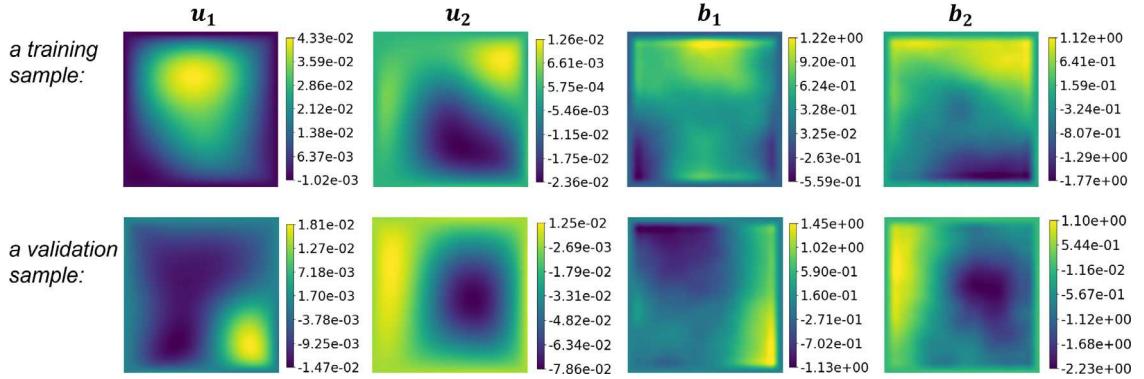


Fig. 10. The external force and the corresponding displacement field components for a training and a validation sample of Example II: Anisotropic Hyperelastic Material.

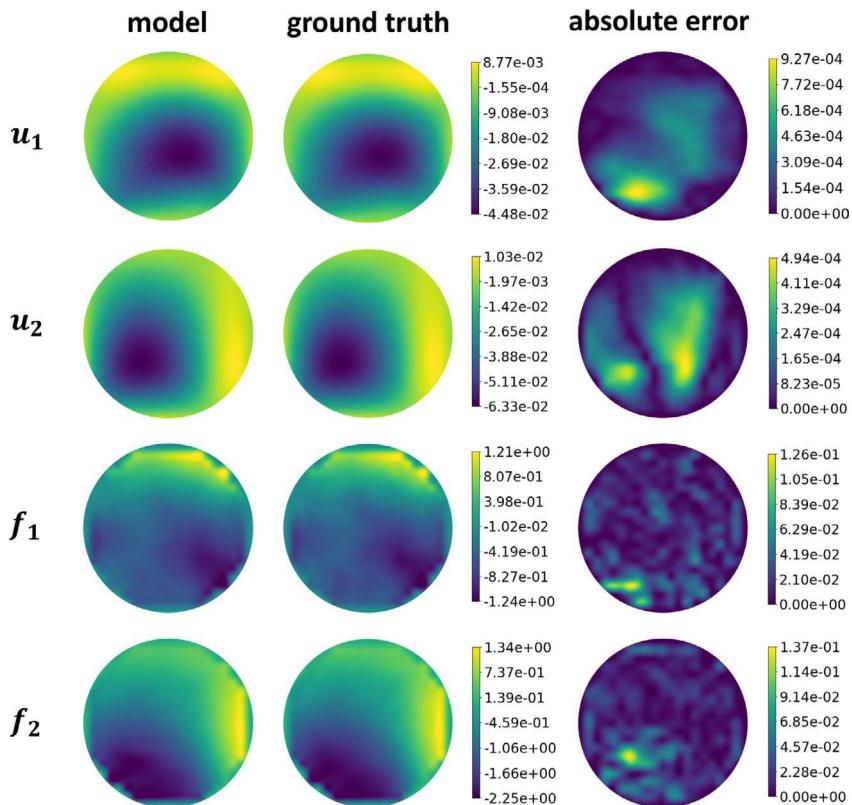


Fig. 11. Demonstration of test samples in Example II: Anisotropic Hyperelastic Material. PNO prediction of the displacement field (given external force and boundary conditions) and the prediction of body force (given displacement field) versus the corresponding ground truth for a test sample.

provides volume constraints on the extensions such that the desired local BC is well approximated on the original local boundary. To do this, the following formula is used:

$$u(\mathbf{x}_{vc}) = 2u(\mathbf{x}_{bc}) - u(\mathbf{x}_{mir}), \quad (4.1)$$

where \mathbf{x}_{vc} is a constrained node in the extension. \mathbf{x}_{bc} denotes a point on the local boundary that is closest to \mathbf{x}_{vc} , and $\mathbf{x}_{mir} := 2\mathbf{x}_{bc} - \mathbf{x}_{vc}$ is the “mirror node” for \mathbf{x}_{vc} inside the domain across the boundary. \mathbf{x}_{vc} , \mathbf{x}_{bc} , and \mathbf{x}_{mir} form a straight line normal to the boundary with \mathbf{x}_{bc} in the middle. More details of this method can be found in [106]. Fig. 9 shows the nonlocal BC in a 2δ -thick extended layer for the u_1 component of the sample shown in Fig. 7. While using an extended region to supply nonlocal boundary data, the

PD horizon size	error	training	validation	test
$\delta = 3\Delta x$	force	3.21%	3.68%	3.51%
	displacement	0.73%	0.77%	0.72%
$\delta = 4\Delta x$	force	3.93%	4.45%	4.37%
	displacement	0.90%	0.90%	1.03%
$\delta = 5\Delta x$	force	3.50%	3.89%	3.82%
	displacement	0.86%	0.97%	0.98%

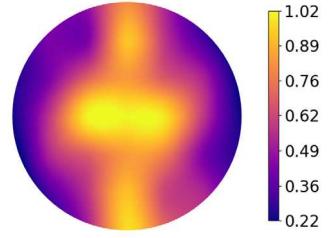


Fig. 12. PNO prediction results of Example II: Anisotropic Hyperelastic Material. Left: L^2 -errors of the proposed PNO model with different horizon sizes δ . Right: The learned kernel function $\omega^{NN}(\xi/\delta)$ of the PNO model.

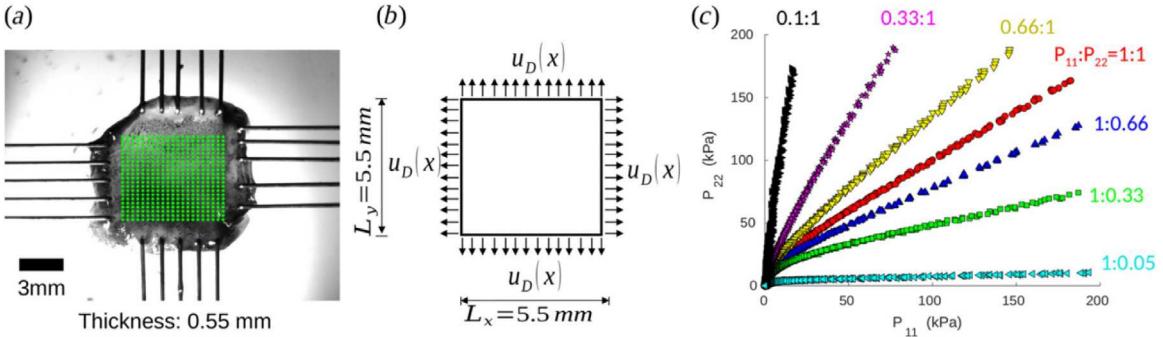


Fig. 13. The experimental set up for biaxial testing of the biological tissue [22] in Example III.

PNO method for purposes of evaluating errors is restricted to the interior. As always with Dirichlet boundary conditions in either a local or nonlocal system, the balances of linear and angular momentum in the form (2.16) and (2.17) do not apply because they do not account for forces and moments artificially created by the constraints.

To demonstrate the generalizability of PNOs to different geometries and BCs, we used a disk-shaped region rather than a square for the test set. The central disk had a radius of 0.5 (Fig. 11). A ring with thickness 2δ -thick was used as an extension of the region for the nonlocal BC, as in the graphene example. Within the ring, displacements from FEM were prescribed, and the PNO solution was then obtained within the disk. The widths of the MLPs for ω^{NN} and σ^{NN} are respectively: (2, 256, 512, 1) and (4, 512, 512, 1). Since the external forces are nonzero, the training and testing follow the algorithm and formulations of Case I described in Section 3.2.

Recall that in the graphene example, the horizon δ is determined by the coarse grained data. In the present example, there is no coarse graining, and δ is an adjustable parameter. To see the effect of δ , the PNO is trained assuming three different horizon sizes: $\delta = 3\Delta x$, $4\Delta x$, and $5\Delta x$, where Δx is the FEM mesh spacing. For each δ , the PNO is trained on the same training set, and the best model is selected based on the minimum validation error. In the left of Fig. 12 we report the relative L^2 -error for forces and displacements. Comparing the results for different δ suggests that the learning results have been compatible when using different horizon sizes, that is, there is no preferred internal length scale for the material. This observation is consistent with the fact that this data is generated using a local theory, that means the ground-truth solution has no length scale. Hence, in the rest of tests we will use $\delta = 3\Delta x$, which has the lowest computational cost. In Fig. 11 we show the ground-truth, PNO predictions, and absolute error fields for displacements and forces for one of the test samples. One can see that the model has successfully capture important solution features visually. This fact is also verified when comparing the test error with training and validation errors: as shown in the left of Fig. 12, all three errors are in a similar scale, highlighting the generalizability of PNOs. To further investigate the physical meaning of the learning results, the learned kernel function, ω , for this example is plotted in the right plot of Fig. 12. The kernel values are generally larger in vertical direction, reflecting the stiffer response in the x_2 -direction due to the anisotropy in the HGO model.

4.3. Example III: A biological tissue

In the present example, the PNO is applied to learn a constitutive model for a real material based on data collected from experimental measurements [22]. In particular, we consider a porcine TVAL specimen subjected to biaxial stretching with digital image correlation (DIC) for recording the displacement field during the tests. The material is heterogeneous and anisotropic and, in these tests, undergoes strains as large as 90%. Fig. 13 shows the experimental setup (a), the corresponding boundary-value problem (b), and the biaxial loading ratios (c). The tissue undergoes seven different protocols (biaxial loading ratios). The original experiments repeated each protocol for three cycles. However, in the present study we aim to compare the PNO against the best

Table 3

Learning results of Example III: A Biological Tissue: average relative error of the displacement field for the biological tissue under seven protocols of biaxial tension test.

Training (100 samples)	Validation (20 samples)	Test (1455 samples)
7.02%	9.47%	7.01%

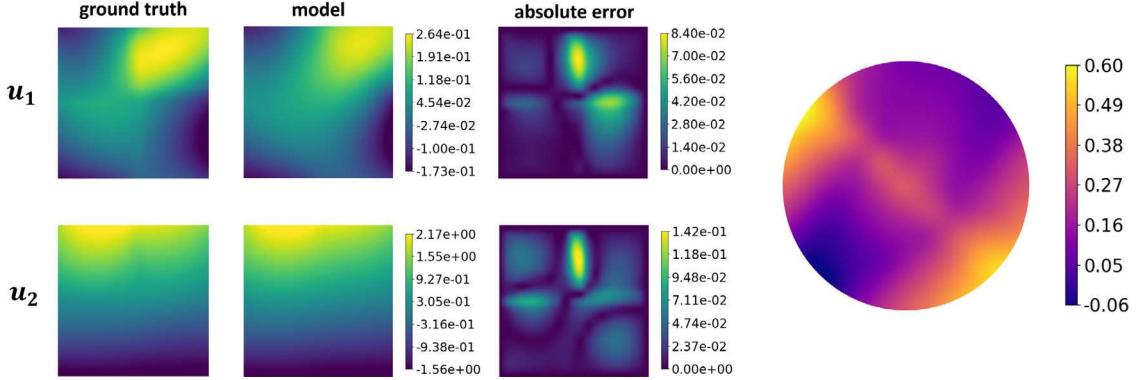


Fig. 14. Learning results of Example III: A Biological Tissue. Left: PNO predictions of the displacement field versus ground truth (experimentally measured) for a test sample. Right: The kernel function $\omega^{NN}(\xi/\delta)$ learned in the PNO model.

available continuum models, and the results of such models are available only for the first cycle of each protocol [22]. The training set used for this problem consists of 100 samples equi-spaced temporally over the total measurement sequence of 7378 samples from all seven protocols. The validation set consists of 20 samples from the remainder of the data, such that samples with high, low, and mid-level stretches from all seven protocols are included. For testing, and to compare the accuracy of the PNO against different classical (local) constitutive models, the same 1455 test samples are employed that were used in [22].

The displacements are measured in each sample with DIC at 441 positions on a 21×21 grid in a square $0.55 \text{ mm} \times 0.55 \text{ mm}$ domain with uniform spacing $\Delta x = 0.0275 \text{ mm}$. The experiments are quasi-static with zero external forces. Therefore, Case II (see Section 3.2) is used for training and testing. Recall that in Case II, a nonlinear solver is required in each sample and each epoch, which can be computationally expensive. To reduce the computational cost, values of $tol=0.001$ and $itr_{max}=15$ in Algorithm 1 are used with the training set, which are relatively relaxed criteria. To make 15 allowed iterations sufficient to converge for most samples, the initial displacements are obtained from other samples that are close in the measurement sequence. Also, smaller neural networks are used for the PNO than in the previous two examples. The MLP widths are set to $(2, 32, 64, 1)$ for ω^{NN} , and $(4, 64, 64, 1)$ for σ^{NN} . To allow for anisotropy, (3.4) is used as the form of ω^{NN} . For the validation set, since they do not go under auto-differentiation, the GPU memory is not affected by the solver's iterations. Hence, we used zero initial displacement and $itr_{max}=100$ for the validation set to filter out models that are overfitted to close initialization during training.

Although the tissue is slightly heterogeneous, the PNO in this study is homogeneous, that is, no part of the operator is explicitly a function of \mathbf{x} . Since the length scale of the heterogeneities in the samples is not known, the largest horizon size that is practical to use in GPU memory was assumed, which is $\delta = 5\Delta x$. As in the previous example, we use the mirror-based FNM [105,106] to supply boundary conditions on a 2δ -extension on the original $0.55 \text{ mm} \times 0.55 \text{ mm}$ domain.

Table 3 shows the average relative L^2 displacement errors for the training, validation and test sets. Then, on the left plot of Fig. 14 we show one instance of the test set. The ground-truth displacement field, the PNO prediction, and the absolute errors are plotted. The PNO model reproduces the material response very well except at specific locations. A possible explanation for this is that heterogeneity in the sample is strongest at these locations, so the homogeneous model is less accurate there. Another is that in the experiment, there is some unavoidable nonuniformity in the way the loads are applied. This can be seen in the left image in Fig. 13, which shows some variation in the spacing of the rod-like structures that load the edges of the sample. To further investigate the physical meaning of our learning results, on the right plot of Fig. 14 the learned kernel is provided. The image shows that this kernel represents an anisotropic material with stiffer responses in $-45^\circ/+135^\circ$ direction. This type of tissue is reported to have two different families of collagen fibers in its microstructure [107]. Along with possible heterogeneity, this may explain the apparently more complex dependence of the kernel on the bond vector, compared with the previous example.

For comparisons with the PNO approach, we employed three models for the planar stress-strain behavior of tissues: (i) a Fung-type model [108], (ii) an invariant-based model [7,109,110], and (iii) a structure-informed model [8,9,111,112]. In all three models we solved for the displacement field using Abaqus [113], and calculated their relative error with respect to the experimentally-retrieved displacements of each node. The optimal model parameters were obtained by minimizing the total relative displacement error on all training samples. That means, the training procedure is more favorable to these three baseline models, since their

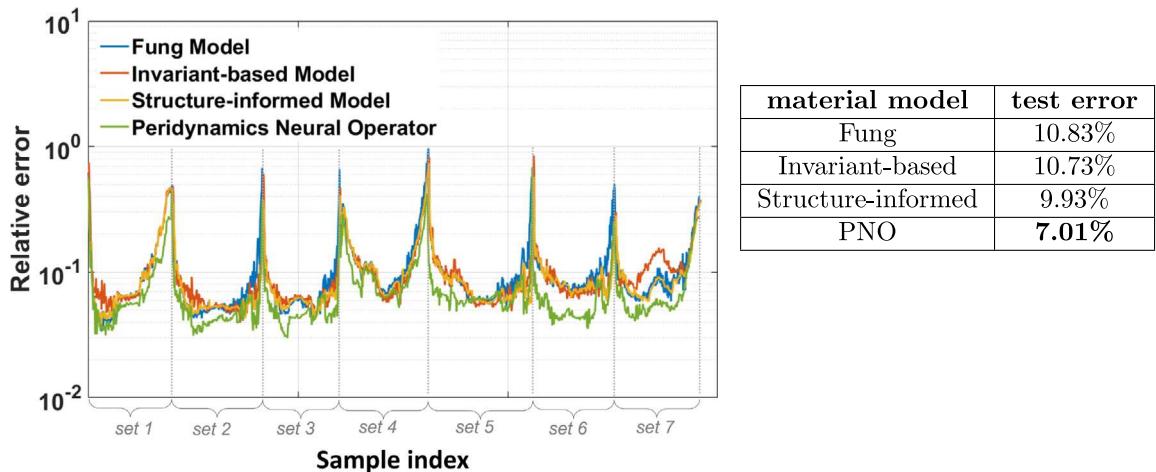


Fig. 15. Learning results of Example III: A Biological Tissue, and comparison with state-of-the-art classical tissue models. Left: Relative error plotted vs sample index for 1455 samples as the test set for three classical models and the learned PNO model. Right: Comparison of the average L^2 -relative error of the displacement field over the 1455 test samples.

parameters are optimized based on all 7378 training samples while our PNO has only considered 120 training and validation samples. Further details of these models are given in [22].

Fig. 15, on the left, shows the per-sample relative L^2 -errors plotted for the PNO and these three classical constitutive models. The corresponding average relative L^2 -errors are provided in the right of Fig. 15. Both results show that PNO outperforms all three of the classical models. To explain the improvement, we note that while the classical models are sophisticated and specifically designed for tissue materials, they suffer from intentional and unintentional simplifications that ignore subtle characteristics hidden in the real material. In contrast, the PNO uses the power of neural networks to learn the material model from experimental data, unconstrained by an assumed algebraic form of the material model. Another advantage of the PNO comes from its nonlocality, which captures some aspects of the δ -scale heterogeneity of the tissue. This example has demonstrated that the PNO can learn a constitutive model for a real, heterogeneous, anisotropic, large-deformation material and that this model can outperform state-of-the-art classical constitutive models.

5. Summary and future directions

In this paper, we introduced a novel nonlocal neural operator architecture for material constitutive modeling that can be trained directly from full-field spatial measurements. The operator is implemented within the framework of an ordinary state-based peridynamic material model. The heart of the architecture is a neural operator with two shallow neural networks: the nonlocal kernel ω^{NN} together with a nonlocal model form σ^{NN} for the peridynamic scalar force state.

The model determines the peridynamic force state vector function for any deformation. Because it is implemented in this way, the model automatically satisfies the conservation laws and required invariances that peridynamics does. It also provides the expressivity inherent in neural networks.

As discussed in the previous sections, extensive numerical experiments were performed, covering a variety of material modeling problems. These include the learning of a material model from molecular dynamics simulations, PDE-based synthetic data sets, and digital image correlation (DIC) measurements. Empirical results show that the PNO provides physically consistent predictions for highly nonlinear and anisotropic materials, even in small and noisy data regimes. Moreover, it was verified that the learned model is generalizable to different loading scenarios and domain settings.

In future work we plan to explore how additional fundamental aspects of continuum mechanics, such as the polyconvexity within the context of hyperelasticity can be incorporated into the PNO framework. We also plan to explore how the expressivity in neural operators can be exploited to better model heterogeneous media. It seems possible to embed the observable or trainable material heterogeneity into the model. Since peridynamics has been widely adopted in the engineering community because of its capability in modeling material fracture, as another natural extension we plan to extend our algorithm to material damage problems.

CRediT authorship contribution statement

Siavash Jafarzadeh: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis. **Stewart Silling:** Writing – review & editing, Writing – original draft, Methodology, Formal analysis, Data curation, Conceptualization. **Ning Liu:** Software, Resources, Methodology, Data curation. **Zhongqiang Zhang:** Writing – review & editing, Methodology, Formal analysis. **Yue Yu:** Writing – original draft, Supervision, Project administration, Methodology, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

S. Jafarzadeh would like to acknowledge support by the AFOSR, USA grant FA9550-22-1-0197, and Y. Yu would like to acknowledge support by the National Science Foundation, USA under award DMS-1753031. Portions of this research were conducted on Lehigh University's Research Computing infrastructure partially supported by National Science Foundation, USA Award 2019035.

This article has been authored by an employee of National Technology and Engineering Solutions of Sandia, LLC under Contract No. DE-NA0003525 with the U.S. Department of Energy (DOE). The employee owns all right, title and interest in and to the article and is solely responsible for its contents. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this article or allow others to do so, for United States Government purposes. The DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan <https://www.energy.gov/downloads/doe-public-access-plan>.

References

- [1] E.E. Prudencio, P.T. Bauman, S. Williams, D. Faghihi, K. Ravi-Chandar, J.T. Oden, A dynamic data driven application system for real-time monitoring of stochastic damage, *Procedia Comput. Sci.* 18 (2013) 2056–2065.
- [2] 2014 Technical Strategic Plan, Tech. rep., The Air Force Office of Scientific Research, 2014.
- [3] R. Talreja, J. Varna, *Modeling Damage, Fatigue and Failure of Composite Materials*, Elsevier, 2015.
- [4] J. Sorić, P. Wriggers, O. Allix, *Multiscale Modeling of Heterogeneous Structures*, Springer, 2018.
- [5] C. Mourlas, G. Markou, M. Papadrakakis, Accurate and computationally efficient nonlinear static and dynamic analysis of reinforced concrete structures considering damage factors, *Eng. Struct.* 178 (2019) 258–285.
- [6] G. Markou, R. Garcia, C. Mourlas, M. Guadagnini, K. Pilakoutas, M. Papadrakakis, A new damage factor for seismic assessment of deficient bare and frp-retrofitted rc structures, *Eng. Struct.* 248 (2021) 113152.
- [7] C.-H. Lee, R. Amini, R.C. Gorman, J.H. Gorman, M.S. Sacks, An inverse modeling approach for stress estimation in mitral valve anterior leaflet valvuloplasty for *in vivo* valvular biomaterial assessment, *J. Biomech.* 47 (9) (2014) 2055–2063.
- [8] Q. He, D.W. Laurence, C.-H. Lee, J.-S. Chen, Manifold learning based data-driven modeling for soft biological tissues, *J. Biomech.* 117 (2021) 110124.
- [9] C.-H. Lee, W. Zhang, K. Feaver, R.C. Gorman, J.H. Gorman, M.S. Sacks, On the *in vivo* function of the mitral heart valve leaflet: Insights into tissue–interstitial cell biomechanical coupling, *Biomech. Model. Mechanobiol.* 16 (5) (2017) 1613–1632.
- [10] G. Limbert, *Skin Biophysics: from Experimental Characterisation to Advanced Modelling*, Vol. 22, Springer, 2019.
- [11] V. Taç, K. Linka, F. Sahli-Costabal, E. Kuhl, A.B. Tepole, Benchmarking physics-informed frameworks for data-driven hyperelasticity, *Comput. Mech.* (2023) 1–17.
- [12] G.W. Milton, *The Theory of Composites*, SIAM, 2022.
- [13] J. Ghaboussi, D.E. Sidarta, New nested adaptive neural networks (nann) for constitutive modeling, *Comput. Geotech.* 22 (1) (1998) 29–52.
- [14] X. Zhang, K. Garikipati, Machine learning materials physics: Multi-resolution neural networks learn the free energy and nonlinear elastic response of evolving microstructures, *Comput. Methods Appl. Mech. Engrg.* 372 (2020) 113362.
- [15] N.N. Vlassis, R. Ma, W. Sun, Geometric deep learning for computational mechanics Part I: Anisotropic hyperelasticity, *Comput. Methods Appl. Mech. Engrg.* 371 (2020) 113299.
- [16] M. Liu, L. Liang, W. Sun, A generic physics-informed neural network-based constitutive model for soft biological tissues, *Comput. Methods Appl. Mech. Engrg.* 372 (2020) 113402.
- [17] J.N. Fuhr, N. Bouklas, R.E. Jones, Learning hyperelastic anisotropy from data via a tensor basis neural network, *J. Mech. Phys. Solids* 168 (2022) 105022.
- [18] Z. Li, D.Z. Huang, B. Liu, A. Anandkumar, Fourier neural operator with learned deformations for pdes on general geometries, arXiv preprint arXiv:2207.05209.
- [19] S. Goswami, M. Yin, Y. Yu, G.E. Karniadakis, A physics-informed variational deepnet for predicting crack path in quasi-brittle materials, *Comput. Methods Appl. Mech. Engrg.* 391 (2022) 114587.
- [20] M. Yin, E. Ban, B.V. Rego, E. Zhang, C. Cavinato, J.D. Humphrey, G. Em Karniadakis, Simulating progressive intramural damage leading to aortic dissection using deepnet: an operator–regression neural network, *J. R. Soc. Interface* 19 (187) (2022) 20210670.
- [21] M. Yin, E. Zhang, Y. Yu, G.E. Karniadakis, Interfacing finite elements with deep neural operators for fast multiscale modeling of mechanics problems, 2022, arXiv:2203.00003.
- [22] H. You, Q. Zhang, C.J. Ross, C.-H. Lee, M.-C. Hsu, Y. Yu, A physics-guided neural operator learning approach to model biological tissues from digital image correlation measurements, *J. Biomech. Eng.* 144 (12) (2022) 121012.
- [23] H. You, Q. Zhang, C.-J. Ross, C.-H. Lee, Y. Yu, Learning Deep Implicit Fourier Neural Operators (IFNOs) with Applications to Heterogeneous Material Modeling, arXiv preprint arXiv:2203.08205.
- [24] N. Liu, Y. Yu, H. You, N. Tatikola, Ino: Invariant neural operators for learning complex physical systems with momentum conservation, in: *International Conference on Artificial Intelligence and Statistics*, PMLR, 2023, pp. 6822–6838.
- [25] N. Liu, S. Jafarzadeh, Y. Yu, Domain agnostic fourier neural operators, arXiv preprint arXiv:2305.00478.
- [26] N. Liu, Y. Fan, X. Zeng, M. Klöwer, Y. Yu, Harnessing the power of neural operators with automatically encoded conservation laws, arXiv preprint arXiv:2312.11176.
- [27] S. Goswami, A. Bora, Y. Yu, G.E. Karniadakis, Physics-informed neural operators, arXiv preprint arXiv:2207.05748.
- [28] D.K. Klein, M. Fernández, R.J. Martin, P. Neff, O. Weeger, Polyconvex anisotropic hyperelasticity with neural networks, *J. Mech. Phys. Solids* 159 (2022) 104703.

[29] K. Linka, E. Kuhl, A new family of constitutive artificial neural networks towards automated model discovery, *Comput. Methods Appl. Mech. Engrg.* 403 (2023) 115731.

[30] V. Tac, F.S. Costabal, A.B. Tepole, Data-driven tissue mechanics with polyconvex neural ordinary differential equations, *Comput. Methods Appl. Mech. Engrg.* 398 (2022) 115248.

[31] H. You, Y. Yu, S. Silling, M. D'Elia, A data-driven peridynamic continuum model for upscaling molecular dynamics, *Comput. Methods Appl. Mech. Engrg.* 389 (2022) 114400.

[32] T. Chu, W. Ranson, M.A. Sutton, Applications of digital-image-correlation techniques to experimental mechanics, *Exp. Mech.* 25 (1985) 232–244.

[33] R. Yang, Y. Li, D. Zeng, P. Guo, Deep dic: Deep learning-based digital image correlation for end-to-end displacement and strain measurement, *J. Mater. Process. Technol.* 302 (2022) 117474.

[34] J.P. de Almeida Martins, D. Topgaard, Multidimensional correlation of nuclear relaxation rates and diffusion tensors for model-free investigations of heterogeneous anisotropic porous materials, *Sci. Rep.* 8 (1) (2018) 2488.

[35] H. Yang, Q. Xiang, S. Tang, X. Guo, Learning material law from displacement fields by artificial neural network, *Theor. Appl. Mech. Lett.* 10 (3) (2020) 202–206.

[36] K. Guo, M.J. Buehler, A semi-supervised approach to architected materials design using graph neural networks, *Extreme Mech. Lett.* 41 (2020) 101029.

[37] T. Xue, S. Adriaenssens, S. Mao, Learning the nonlinear dynamics of mechanical metamaterials with graph networks, *Int. J. Mech. Sci.* 238 (2023) 107835.

[38] J.M. Hestroffer, M.-A. Charpagne, M.I. Latypov, I.J. Beyerlein, Graph neural networks for efficient learning of mechanical properties of polycrystals, *Comput. Mater. Sci.* 217 (2023) 111894.

[39] A. Thomas, A.R. Durmaz, M. Alam, P. Gumbisch, H. Sack, C. Eberl, Materials fatigue prediction using graph neural networks on microstructure representations, *Sci. Rep.* 13 (1) (2023) 12562.

[40] R.J. Gladstone, H. Rahmani, V. Suryakumar, H. Meidani, M. D'Elia, A. Zarei, Gnn-based physics solver for time-independent pdes, arXiv preprint [arXiv:2303.15681](https://arxiv.org/abs/2303.15681).

[41] H. Jin, T. Jiao, R.J. Clifton, K.-S. Kim, Dynamic fracture of a bicontinuously nanostructured copolymer: A deep-learning analysis of big-data-generating experiment, *J. Mech. Phys. Solids* 164 (2022) 104898.

[42] R. Kaviani, J. Kolinski, High resolution interferometric imaging of liquid-solid interfaces with hotnet, *Exp. Mech.* 63 (2) (2023) 309–321.

[43] E.A. Holm, R. Cohn, N. Gao, A.R. Kitahara, T.P. Matson, B. Lei, S.R. Yarasi, Overview: Computer vision and machine learning for microstructural characterization and analysis, *Metall. Mater. Trans. A* 51 (2020) 5985–5999.

[44] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Neural operator: Graph kernel network for partial differential equations, arXiv preprint [arXiv:2003.03485](https://arxiv.org/abs/2003.03485).

[45] S. Lanthaler, Z. Li, A.M. Stuart, The nonlocal neural operator: Universal approximation, arXiv preprint [arXiv:2304.13221](https://arxiv.org/abs/2304.13221).

[46] Z. Li, N.B. Kovachki, C. Choy, B. Li, J. Kossaifi, S.P. Ota, M.A. Nabian, M. Stadler, C. Hundt, K. Azizzadenesheli, et al., Geometry-informed neural operator for large-scale 3d pdes, arXiv preprint [arXiv:2309.00583](https://arxiv.org/abs/2309.00583).

[47] L. Lu, X. Meng, S. Cai, Z. Mao, S. Goswami, Z. Zhang, G.E. Karniadakis, A comprehensive and fair comparison of two neural operators (with practical extensions) based on fair data, arXiv preprint [arXiv:2111.05512](https://arxiv.org/abs/2111.05512).

[48] L. Zhang, H. You, T. Gao, M. Yu, C.-H. Lee, Y. Yu, Metano: How to transfer your knowledge on learning hidden physics, *Comput. Methods Appl. Mech. Engrg.* 417 (2023) 116280, [http://dx.doi.org/10.1016/j.cma.2023.116280](https://dx.doi.org/10.1016/j.cma.2023.116280), A Special Issue in Honor of the Lifetime Achievements of T. J. R. Hughes.

[49] Q. Du, B. Engquist, X. Tian, Multiscale modeling, homogenization and nonlocal effects: Mathematical and computational issues, *Contemp. Math.* 754 (2020) 115–140.

[50] R. Maranganti, P. Sharma, Length scales at which classical elasticity breaks down for various materials, *Phys. Rev. Lett.* 98 (19) (2007) 195504.

[51] M. Beran, J. McCoy, Mean field variations in a statistical sample of heterogeneous linearly elastic solids, *Int. J. Solids Struct.* 6 (8) (1970) 1035–1054.

[52] K. Cherednichenko, V.P. Smyshlyayev, V. Zhikov, Non-local homogenized limits for composite media with highly anisotropic periodic fibres, *Proc. R. Soc. Edinb. Sect. A: Math.* 136 (1) (2006) 87–114.

[53] F.C. Karal Jr., J.B. Keller, Elastic, electromagnetic, and other waves in a random medium, *J. Math. Phys.* 5 (4) (1964) 537–547.

[54] Y. Rahali, I. Giorgio, J. Ganghoffer, F. Dell'Isola, Homogenization à la piola produces second gradient continuum models for linear pantographic lattices, *Internat. J. Engrg. Sci.* 97 (2015) 148–172.

[55] S.P. Neuman, D.M. Tartakovsky, Perspective on theories of non-fickian transport in heterogeneous media, *Adv. Water Resour.* 32 (5) (2009) 670–680.

[56] P. Diehl, S. Prudhomme, M. Lévesque, A review of benchmark experiments for the validation of peridynamics models, *J. Peridyn. Nonlocal Model.* 1 (1) (2019) 14–35.

[57] D.A. Benson, S.W. Wheatcraft, M.M. Meerschaert, Application of a fractional advection–dispersion equation, *Water Resour. Res.* 36 (6) (2000) 1403–1412.

[58] Y. Jiang, L. Li, Y. Hu, A spatiotemporally-nonlocal continuum field theory of polymer networks, *Sci. China Phys., Mech. Astron.* 66 (5) (2023) 254611.

[59] K. Deshmukh, T. Breitzman, K. Dayal, Multiband homogenization of metamaterials in real-space: Higher-order nonlocal models and scattering at external surfaces, *J. Mech. Phys. Solids* 167 (2022) 104992.

[60] F.V. Difonzo, L. Lopez, S.F. Pellegrino, Physics informed neural networks for an inverse problem in peridynamic models, arXiv preprint [arXiv:2312.11316](https://arxiv.org/abs/2312.11316).

[61] H. You, Y. Yu, S. Silling, M. D'Elia, Data-driven learning of nonlocal models: from high-fidelity simulations to constitutive laws, in: Accepted in AAAI Spring Symposium: MLPS, 2021.

[62] H. You, Y. Yu, N. Trask, M. Gulian, M. D'Elia, Data-driven learning of nonlocal physics from high-fidelity synthetic data, *Comput. Methods Appl. Mech. Engrg.* 374 (2021) 113553.

[63] F. Lu, Q. An, Y. Yu, Nonparametric learning of kernels in nonlocal operators, *J. Peridyn. Nonlocal Model.* (2023) 1–24.

[64] Y. Fan, M. D'Elia, Y. Yu, H.N. Najm, S. Silling, Bayesian nonlocal operator regression: A data-driven learning framework of nonlocal models with uncertainty quantification, *J. Eng. Mech.* 149 (8) (2023) 04023049.

[65] L. Zhang, H. You, Y. Yu, Metanor: A meta-learnt nonlocal operator regression approach for metamaterial modeling, *MRS Commun.* 12 (5) (2022) 662–677.

[66] H. You, X. Xu, Y. Yu, S. Silling, M. D'Elia, J. Foster, Towards a unified nonlocal, peridynamics framework for the coarse-graining of molecular dynamics data with fractures, *Appl. Math. Mech.* 44 (2023) 1125–1150.

[67] Z. Li, N.B. Kovachki, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, A. Anandkumar, et al., Fourier neural operator for parametric partial differential equations, in: International Conference on Learning Representations, 2020.

[68] H. You, Y. Yu, M. D'Elia, T. Gao, S. Silling, Nonlocal kernel network (NKN): a stable and resolution-independent deep neural network, arXiv preprint [arXiv:2201.02217](https://arxiv.org/abs/2201.02217).

[69] S.A. Silling, Reformulation of elasticity theory for discontinuities and long-range forces, *J. Mech. Phys. Solids* 48 (1) (2000) 175–209.

[70] S.A. Silling, M. Epton, O. Weckner, J. Xu, E. Askari, Peridynamic states and constitutive modeling, *J. Elasticity* 88 (2) (2007) 151–184.

[71] E. Madenci, S. Oterkus, Ordinary state-based peridynamics for plastic deformation according to von mises yield criteria with isotropic hardening, *J. Mech. Phys. Solids* 86 (2016) 192–219.

[72] F. Bobaru, J.T. Foster, P.H. Geubelle, S.A. Silling, Handbook of Peridynamic Modeling, CRC Press, 2016.

[73] S.A. Silling, R.B. Lehoucq, Peridynamic theory of solid mechanics, *Adv. Appl. Mech.* 44 (2010) 73–168.

[74] X. Tian, Q. Du, Asymptotically compatible schemes for robust discretization of parametrized problems with applications to nonlocal models, *SIAM Rev.* 62 (1) (2020) 199–227.

[75] E. Madenci, A. Barut, M. Futch, Peridynamic differential operator and its applications, *Comput. Methods Appl. Mech. Engrg.* 304 (2016) 408–451.

[76] J.T. Foster, S.A. Silling, W. Chen, An energy based failure criterion for use with peridynamic states, *Int. J. Multiscale Comput. Eng.* 9 (6).

[77] R. Lipton, Dynamic brittle fracture as a small horizon limit of peridynamics, *J. Elasticity* 117 (2014) 21–50.

[78] P. Diehl, R. Lipton, T. Wick, M. Tyagi, A comparative review of peridynamics and phase-field models for engineering fracture mechanics, *Comput. Mech.* 69 (6) (2022) 1259–1293.

[79] Y.D. Ha, F. Bobaru, Studies of dynamic crack propagation and crack branching with peridynamics, *Int. J. Fract.* 162 (2010) 229–244.

[80] E. Madenci, E. Oterkus, Peridynamic theory, in: *Peridynamic Theory and Its Applications*, Springer, 2013, pp. 19–43.

[81] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, A. Stuart, K. Bhattacharya, A. Anandkumar, Multipole graph neural operator for parametric partial differential equations, in: *Advances in Neural Information Processing Systems*, vol. 33.

[82] G. Gupta, X. Xiao, P. Bogdan, Multiwavelet-based operator learning for differential equations, *Adv. Neural Inf. Process. Syst.* 34 (2021) 24048–24062.

[83] L. Lu, P. Jin, G.E. Karniadakis, Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators, *arXiv preprint arXiv:1910.03193*.

[84] S. Cao, Choose a transformer: Fourier or galerkin, in: *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 24924–24940.

[85] Y. Yin, M. Kirchmeyer, J.-Y. Franceschi, A. Rakotomamonjy, et al., Continuous pde dynamics forecasting with implicit neural representations, in: *The Eleventh International Conference on Learning Representations*, 2022.

[86] Z. Hao, Z. Wang, H. Su, C. Ying, Y. Dong, S. Liu, Z. Cheng, J. Song, J. Zhu, Gnot: A general neural operator transformer for operator learning, in: *International Conference on Machine Learning*, PMLR, 2023, pp. 12556–12569.

[87] Z. Li, K. Meidani, A.B. Farimani, Transformer for partial differential equations' operator learning, *arXiv preprint arXiv:2205.13671*.

[88] Y.Z. Ong, Z. Shen, H. Yang, IAE-NET: Integral autoencoders for discretization-invariant learning, <http://dx.doi.org/10.13140/RG.2.2.25120.87047/2>.

[89] G. Gupta, X. Xiao, P. Bogdan, Multiwavelet-based operator learning for differential equations, in: A. Beygelzimer, Y. Dauphin, P. Liang, J.W. Vaughan (Eds.), in: *Advances in Neural Information Processing Systems*, 2021, URL <https://openreview.net/forum?id=LZDiWaC9CGL>.

[90] L. Lu, P. Jin, G. Pang, Z. Zhang, G.E. Karniadakis, Learning nonlinear operators via deeponet based on the universal approximation theorem of operators, *Nat. Mach. Intell.* 3 (3) (2021) 218–229.

[91] J. Gilmer, S.S. Schoenholz, P.F. Riley, O. Vinyals, G.E. Dahl, Neural message passing for quantum chemistry, in: *International Conference on Machine Learning*, PMLR, 2017, pp. 1263–1272.

[92] J. Brandstetter, D.E. Worrall, M. Welling, Message passing neural pde solvers, in: *International Conference on Learning Representations*, 2021.

[93] N. Liu, Y. Fan, X. Zeng, M. Klöwer, Y. Yu, Harnessing the power of neural operators with automatically encoded conservation laws, 2023, *arXiv: 2312.11176*.

[94] Z. Li, H. Zheng, N. Kovachki, D. Jin, H. Chen, B. Liu, K. Azizzadenesheli, A. Anandkumar, Physics-informed neural operator for learning partial differential equations, *arXiv preprint arXiv:2111.03794*.

[95] S. Wang, H. Wang, P. Perdikaris, Learning the solution operator of parametric partial differential equations with physics-informed deeponets, *Sci. Adv.* 7 (40) (2021).

[96] M. Foss, P. Radu, Y. Yu, Convergence analysis and numerical studies for linearly elastic peridynamics with dirichlet-type boundary conditions, *J. Peridyn. Nonlocal Model.* 5 (2) (2023) 275–310.

[97] J.R. Shewchuk, et al., An introduction to the conjugate gradient method without the agonizing pain, 1994.

[98] Q. Van Le, Relationship between Microstructure and Mechanical Properties in Bi 2 Sr 2 CaCu 2 Ox Round Wires Using Peridynamic Simulation, North Carolina State University, 2014.

[99] D. Akinwande, C.J. Brennan, J.S. Bunch, P. Egberts, J.R. Felts, H. Gao, R. Huang, J.-S. Kim, T. Li, Y. Li, et al., A review on mechanics and mechanical properties of 2d materials—graphene and beyond, *Extreme Mech. Lett.* 13 (2017) 42–77.

[100] J. Tersoff, Empirical interatomic potential for carbon, with applications, *Phys. Rev. Lett.* 61 (25).

[101] S.A. Silling, M. D'Elia, Y. Yu, H. You, M. Fermen-Coker, Peridynamic model for single-layer graphene obtained from coarse-grained bond forces, *J. Peridyn. Nonlocal Model.* 5 (2) (2023) 183–204.

[102] P. Seleson, M. Parks, On the role of the influence function in the peridynamic theory, *Int. J. Multiscale Comput. Eng.* 9 (6).

[103] G.A. Holzapfel, T.C. Gasser, R.W. Ogden, A new constitutive framework for arterial wall mechanics and a comparative study of material models, *J. Elast. Phys. Sci. Solids* 61 (1) (2000) 1–48.

[104] M. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M.E. Rognes, G.N. Wells, The fenics project version 1.5, *Arch. Numer. Softw.* 3 (100).

[105] S. Oterkus, E. Madenci, A. Agwai, Peridynamic thermal diffusion, *J. Comput. Phys.* 265 (2014) 71–96.

[106] J. Zhao, S. Jafarzadeh, Z. Chen, F. Bobaru, An algorithm for imposing local boundary conditions in peridynamic models on arbitrary domains.

[107] D.J. Fitzpatrick, K. Pham, C.J. Ross, L.T. Hudson, D.W. Laurence, Y. Yu, C.-H. Lee, Ex vivo experimental characterizations for understanding the interrelationship between tissue mechanics and collagen microstructure of porcine mitral valve leaflets, *J. Mech. Behav. Biomed. Mater.* 134 (2022) 105401.

[108] Y.C. Fung, S.Q. Liu, J.B. Zhou, Remodeling of the constitutive equation while a blood vessel remodels itself under stress, *J. Biomech. Eng.* 115 (4B) (1993) 453–459, <http://dx.doi.org/10.1115/1.2895523>.

[109] E.L. Johnson, D.W. Laurence, F. Xu, C.E. Crisp, A. Mir, H.M. Burkhardt, C.-H. Lee, M.-C. Hsu, And isogeometric analysis of tricuspid valves, *Comput. Methods Appl. Mech. Engrg.* 384 (2021) 113960.

[110] D. Kamensky, F. Xu, C.-H. Lee, J. Yan, Y. Bazilevs, M.-C. Hsu, A contact formulation based on a volumetric potential: Application to isogeometric simulations of atrioventricular valves, *Comput. Methods Appl. Mech. Engrg.* 330 (2018) 522–546.

[111] R. Fan, M.S. Sacks, Simulation of planar soft tissues using a structural constitutive model: finite element implementation and validation, *J. Biomech.* 47 (9) (2014) 2043–2054.

[112] C.-H. Lee, J.-P. Rabbah, A.P. Yoganathan, R.C. Gorman, J.H. Gorman, M.S. Sacks, On the effects of leaflet microstructure and constitutive model on the closing behavior of the mitral valve, *Biomech. Model. Mechanobiol.* 14 (6) (2015) 1281–1302.

[113] G. Abaqus, Abaqus 6.11, Dassault Systemes Simulia Corporation, Providence, RI, USA.