A data-driven peridynamic continuum model for upscaling molecular dynamics

Huaiqian You[a,c], Yue Yu[a,*], Stewart Silling[b], Marta D'Elia[c]

[a] *Department of Mathematics, Lehigh University, Bethlehem, PA*
[b] *Center for Computing Research, Sandia National Laboratories, Albuquerque, NM*
[c] *Computational Science and Analysis, Sandia National Laboratories, Livermore, CA*

Abstract

Nonlocal models, including peridynamics, often use integral operators that embed length-scales in their definition. However, the integrands in these operators are difficult to define from the data that are typically available for a given physical system, such as laboratory mechanical property tests. In contrast, molecular dynamics (MD) does not require these integrands, but it suffers from computational limitations in the length and time scales it can address. To combine the strengths of both methods and to obtain a coarse-grained, homogenized continuum model that efficiently and accurately captures materials' behavior, we propose a learning framework to extract, from MD data, an optimal Linear Peridynamic Solid (LPS) model as a surrogate for MD displacements. To maximize the accuracy of the learnt model we allow the peridynamic influence function to be partially negative, while preserving the well-posedness of the resulting model. To achieve this, we provide sufficient well-posedness conditions for discretized LPS models with sign-changing influence functions and develop a constrained optimization algorithm that minimizes the equation residual while enforcing such solvability conditions. This framework guarantees that the resulting model is mathematically well-posed, physically consistent, and that it generalizes well to settings that are different from the ones used during training. We illustrate the efficacy of the proposed approach with several numerical tests for single layer graphene. Our two-dimensional tests show the robustness of the proposed algorithm on validation data sets that include thermal noise, different domain shapes and external loadings, and discretizations substantially different from the ones used for training.

Keywords: nonlocal models, data-driven learning, machine learning, optimization, homogenization, peridynamics

*Corresponding author.
Email addresses: huy316@lehigh.edu (Huaiqian You), yuy214@lehigh.edu (Yue Yu), sasilli@sandia.gov (Stewart Silling), mdelia@sandia.gov (Marta D'Elia)

Contents

1. Introduction

Complex systems where small-scale dynamics and interactions affect the global behavior are ubiquitous in scientific and engineering applications. In disciplines ranging from climate forecasts to material design, heterogeneities in materials and media at the micro or molecular scales need to be accurately captured to guarantee reliable and trustworthy predictions. However, higher degrees of complexity and heterogeneity require numerical simulations of classical mathematical models at small scales that cannot be afforded, despite recent advances in computational power. This fact creates the need for new mathematical models that act at larger scales and that, combined with new advanced architectures, allow for fast predictions [1–11]. The process of upscaling models or data hides several pitfalls that may compromise the reliability of the resulting surrogates.

In the presence of heterogeneities, it is often the case that to adequately reproduce the large-scale behavior of a system, a homogenized model must follow different governing

laws, as well as different constitutive properties, than the ones that apply at the small scale. Homogenization theory addresses the approximate treatment of a partial differential equation (PDE) that contains small-scale oscillations in its coefficients [12]. It seeks to replace these coefficients with constant or slowly-varying coefficients such that the resulting solutions closely approximate solutions to the original problem in an averaged sense. The resulting parameters are called *effective properties*. In many theoretical treatments, the effective properties are valid only in the limiting case of a very small length scale in the oscillatory behavior of the original parameters. The classical notion of effective properties therefore "washes out" the length scale in the original problem, causing *important information to be lost*.

Nonlocality in the spatial dependence of a continuum model has long been recognized as a consequence of homogenization [13, 14]. For example, in continuum mechanics, nonlocality arises from taking the ensemble average of the displacement field in a family of random linear elastic heterogeneous materials [15–20]. To some extent, this nonlocality can be incorporated in homogenized *weakly nonlocal* PDEs that embed length scales in their coefficients. However, weakly nonlocal PDEs are generally insufficient to fully reproduce coarse-grained data because of the limited spectrum of processes that they can describe [21]. In general, increasing the accuracy of weakly nonlocal PDEs to match small-scale data involves using higher and higher order partial derivatives, resulting in practical challenges in numerical implementations.

As pointed out in [21], *nonlocal operators* [22, 23] are among the best candidates as model descriptions that can circumvent these limitations. Theoretical and numerical techniques for nonlocal models are not as advanced as for classical PDEs. This is one of the main reasons why integral operators historically have not received a broader adoption in the context of numerical homogenization. However, current advances in nonlocal theory, computer power, and solution algorithms are making nonlocal equations viable as practical modeling tools. While integral operators have proved to be successful in several contexts such as mechanics [24, 25] and turbulence [26, 27], they have not been systematically explored for coarse graining, or upscaling, of molecular dynamics (MD) models, for which we propose a new rigorous modeling paradigm.

We stress the fact that with nonlocal operators, constitutive laws take the form of kernels (integrand functions), whose functional form cannot be established a priori. Although the integral constitutive laws must be consistent with the classical effective properties, they contain information about the small-scale response of the system and must be chosen to reproduce this response with the greatest fidelity. In a few cases, certain forms of nonlocal kernels have been adopted in the engineering community because experimental evidence

confirms the efficacy of the model, or because a closed form of integrand that matches desired physical properties can be analytically determined. An example of the former situation is fracture mechanics, where peridynamic models have been demonstrated to be accurate [24, 25, 28]. Examples of the latter case include those diffusion processes in which the mean square displacement does not exhibit the linear, classical behavior, but instead exhibits an anomalous fractional behavior [29]. However, at present, only a few preliminary works address the problem of finding an optimal form for the kernel function [30–33].

In view of the growing importance of MD as a tool for designing materials with reduced reliance on laboratory testing, we propose to use nonlocal operators as upscaled continuous models for MD displacements. We seek nonlocal models that capture important aspects of the small-scale behavior better than classical homogenization theory. Building on our previous works [32, 33] we address the question of how to obtain large-scale nonlocal descriptions that capture MD behavior that would remain hidden in classical approaches to homogenization. To accomplish this, we use machine learning to identify optimal nonlocal kernel functions. The machine learning method is required to perform well with small datasets that may include thermal noise.

We summarize below our main contributions.

- We identify the *best upscaled nonlocal model*, without prior knowledge of the material properties, that accurately describes the material's global behavior based on a small set of possibly noisy data.

- The optimal nonlocal model is *guaranteed to be well-posed* and *generalizes well* to settings that are substantially different from the ones used for training. The optimal model is equally accurate for different sources and geometries, so that it enables generalization.

While our ultimate goal is to learn a general integrand for the nonlocal operator, in this work we consider a specific nonlocal model, the Linear Peridynamic Solid (LPS) model [34] as a first step towards a more general learning tool. We focus our experiments on single layered graphene for which we identify optimal two-dimensional nonlocal models.

Outline of the paper. Section 2 shows how to obtain, via smoothing functions, a nonlocal model for MD displacements. In Section 3 summarizes the peridynamic theory, the LPS model, and the discretization technique used in this work. Section 4 presents our learning approach including the well-posedness of the learned model by construction. It also provides the algorithmic workflow and implementation details. Section 5 illustrates the consistency of the proposed method on manufactured solutions. Section 6 demonstrates the effectiveness

of the learning technique for MD displacements. We illustrate several properties including generalization with respect to loadings, domain size and shape. The effect of thermal noise and the sensitivity of the algorithm to noise intensity are considered. Section 7 summarizes our contributions and provides future research ideas.

2. Coarse-Graining of Molecular Dynamics Displacements

In this section it is shown how to define an integral, continuous model for a system of particles. More details can be found in [35]. Similar results obtained with statistical mechanics can be found in [36].

Consider an assembly of mutually interacting particles in a crystal with particle mass M_ε, $\varepsilon = 1, 2, \ldots, N$. Let the reference positions of these particles be $\mathbf{X}_\varepsilon$ and their displacement vectors $\mathbf{U}_\varepsilon(t)$.

Suppose that any particle γ exerts a force $\mathbf{F}_{\varepsilon\gamma}(t)$ on particle ε, and set $\mathbf{F}_{\varepsilon\varepsilon} = \mathbf{0}$. These forces are assumed to be antisymmetric: $\mathbf{F}_{\gamma\varepsilon}(t) = -\mathbf{F}_{\varepsilon\gamma}(t)$, for all ε, γ, and t. It is also assumed that there is a cutoff distance d for the atomic interactions such that $\mathbf{F}_{\varepsilon\gamma} = \mathbf{0}$ if $|\mathbf{X}_\gamma - \mathbf{X}_\varepsilon| > d$. Each particle ε is subjected to a prescribed external force $\mathbf{B}_\varepsilon(t)$.

For any continuum material point $\mathbf{x} \in \mathbb{R}^n$, define a *smoothing* function $\omega(\mathbf{x}, \cdot)$ such that the following normalization holds:

$$\int_{\mathbb{R}^n} \omega(\mathbf{x}, \mathbf{X}_\varepsilon)\, d\mathbf{x} = 1 \tag{2.1}$$

for any ε. For convenience, assume that at any $\mathbf{x}$, $\omega(\mathbf{x}, \cdot)$ has compact support over the ball $B_R(\mathbf{x})$, for $R > 0$. Define the smoothed mass density and body force density fields by

$$\rho(\mathbf{x}) = \sum_{\varepsilon=1}^{N} \omega(\mathbf{x}, \mathbf{X}_\varepsilon) M_\varepsilon, \qquad \mathbf{b}(\mathbf{x}, t) = \sum_{\varepsilon=1}^{N} \omega(\mathbf{x}, \mathbf{X}_\varepsilon) \mathbf{B}_\varepsilon(t), \tag{2.2}$$

and the smoothed displacement field by

$$\mathbf{u}(\mathbf{x}, t) = \frac{1}{\rho(\mathbf{x})} \sum_{\varepsilon=1}^{N} \omega(\mathbf{x}, \mathbf{X}_\varepsilon) M_\varepsilon \mathbf{U}_\varepsilon(t). \tag{2.3}$$

The evolution equation for the smoothed displacements will now be derived. Newton's second law for the particles has the following form: for any ε

$$M_\varepsilon \ddot{\mathbf{U}}_\varepsilon(t) = \sum_{\gamma=1}^{N} \mathbf{F}_{\varepsilon\gamma}(t) + \mathbf{B}_\varepsilon(t). \tag{2.4}$$

Differentiating (2.3) twice with respect to time yields

$$\rho(\mathbf{x})\ddot{\mathbf{u}}(\mathbf{x},t) = \sum_{\varepsilon=1}^{N} \omega(\mathbf{x},\mathbf{X}_\varepsilon) M_\varepsilon \ddot{\mathbf{U}}_\varepsilon(t). \tag{2.5}$$

From (2.2), (2.4), and (2.5),

$$\begin{aligned} \rho(\mathbf{x})\ddot{\mathbf{u}}(\mathbf{x},t) &= \sum_{\varepsilon=1}^{N} \omega(\mathbf{x},\mathbf{X}_\varepsilon) \left[\sum_{\gamma=1}^{N} \mathbf{F}_{\varepsilon\gamma}(t) + \mathbf{B}_\varepsilon(t) \right] \\ &= \sum_{\varepsilon=1}^{N} \sum_{\gamma=1}^{N} \omega(\mathbf{x},\mathbf{X}_\varepsilon) \mathbf{F}_{\varepsilon\gamma}(t) + \mathbf{b}(\mathbf{x},t). \end{aligned} \tag{2.6}$$

From (2.1) and (2.6), for any $\mathbf{x}$,

$$\rho(\mathbf{x})\ddot{\mathbf{u}}(\mathbf{x},t) = \sum_{\varepsilon=1}^{N} \sum_{\gamma=1}^{N} \omega(\mathbf{x},\mathbf{X}_\varepsilon) \mathbf{F}_{\varepsilon\gamma}(t) \left[\int \omega(\mathbf{y},\mathbf{X}_\gamma)\, d\mathbf{y} \right] + \mathbf{b}(\mathbf{x},t),$$

or, equivalently,

$$\rho(\mathbf{x})\ddot{\mathbf{u}}(\mathbf{x},t) = \int \mathbf{f}(\mathbf{y},\mathbf{x},t)\, d\mathbf{y} + \mathbf{b}(\mathbf{x},t) \tag{2.7}$$

where

$$\mathbf{f}(\mathbf{y},\mathbf{x},t) = \sum_{\varepsilon=1}^{N} \sum_{\gamma=1}^{N} \omega(\mathbf{x},\mathbf{X}_\varepsilon) \omega(\mathbf{y},\mathbf{X}_\gamma) \mathbf{F}_{\varepsilon\gamma}(t) \tag{2.8}$$

and $\mathbf{b}$ is given by (2.2). The properties of $\mathbf{F}$ guarantee that the integrand $\mathbf{f}$ is antisymmetric. Since, by assumption, the smoothing functions have support radius R and the interatomic forces have cutoff distance d, it follows that

$$|\mathbf{y}-\mathbf{x}| > \delta \quad \Longrightarrow \quad \mathbf{f}(\mathbf{y},\mathbf{x},t) = \mathbf{0}$$

for all t, where the *horizon* δ is given by

$$\delta = 2R + d. \tag{2.9}$$

In summary, defining the displacements and other fields in the continuum description using the smoothing function ω, leads directly the nonlocal (or integral) equation of motion (2.7). However, the derivation does not provide a material model, that is, the dependence of $\mathbf{f}$ on the deformation in terms of the continuum displacement field $\mathbf{u}$ is not yet determined. The goal of the present work is to identify an optimal form of the integrand function $\mathbf{f}$ in (2.7)

such that the corresponding nonlocal model faithfully represents given MD displacements under a given set of loading conditions on the MD grid.

At finite temperature, thermal oscillations in displacement are present in any MD simulation. The details of these oscillations are of no interest for purposes of continuum modeling. However, their net effect on the bulk material properties must be included. To smooth out the thermal motions while retaining their net effect, a time-smoothing method is applied. In this method, the following expression is applied to obtain the time-smoothed displacement $\mathbf{U}_\varepsilon(t^n)$ of atom ε:

$$\mathbf{U}_\varepsilon(0) = 0, \qquad \mathbf{U}_\varepsilon(t^n) = (1-\iota)\mathbf{U}_\varepsilon(t^{n-1}) + \iota\widetilde{\mathbf{U}}_\varepsilon(t^n), \quad n > 0 \tag{2.10}$$

where $\widetilde{\mathbf{U}}_\varepsilon$ is the unsmoothed displacement of atom ε and ι is a positive constant, typically on the order of 0.01. The value of ι is chosen so that in effect the time-smoothing has a time scale (100 time steps) that is much larger than that of the thermal oscillations of the atoms (<10 time steps). The value of ι does not depend on the size scale h of the coarse-grained mesh. To obtain displacements that are smoothed in both space and time, the displacement given by (2.10) is used in (2.3):

$$\mathbf{u}(\mathbf{x}) = \frac{1}{\rho(\mathbf{x})}\sum_{\varepsilon=1}^{N}\omega(\mathbf{x},\mathbf{X}_\varepsilon)M_\varepsilon\mathbf{U}_\varepsilon(t^F), \tag{2.11}$$

where t^F is the final time of the MD simulation. The displacements $\mathbf{u}(\mathbf{x})$ contain noise in the form of spatial fluctuations due to the impossibility of completely smoothing out all of the thermal oscillations in an MD simulation within a finite simulation time t^F, regardless of the value of ι. The machine learning algorithm described below attempts to extract continuum material properties from the training data even in the presence of this noise. In Section 6, we will present results on the effectiveness of this machine learning algorithm in treating this type of noisy training data.

Our present objective is to derive the static continuum properties of a crystal from MD. This entails the assumption that the time scale of motions in the MD mesh are much smaller than any intended application of the continuum model. This assumption is justified in the case of the bulk deformation of a perfect graphene sheet, since the relevant time scale of the atomic motions in this material is on the order of femtoseconds. However, if the MD system involves more slowly evolving events such as the motion of dislocations and voids, an acceleration technique that accounts for this longer time scale would need to be used [37]. Even with this modification to the MD simulation, the coarse graining strategy would be essentially the same as described above.

3. Peridynamics

3.1. Peridynamics Background

In the previous section, a coarse-grained continuum momentum balance was derived, given by (2.7). This momentum balance has a fundamentally nonlocal character, since the *pairwise bond force densities* given by (2.8) can be nonzero whenever the material points in the continuum $\mathbf{x}$ and $\mathbf{y}$ are separated by a finite distance up to the horizon δ (Figure 1). This form of the momentum balance is known as the *peridynamic* equation of motion [38]. In peridynamics, each $\mathbf{x}$ interacts through bond forces with other material points $\mathbf{y}$ within a neighborhood with radius δ known as the *family* of $\mathbf{x}$, denoted by $B_\delta(\mathbf{x})$. The equation of motion for material point $\mathbf{x}$ is then

$$\rho(\mathbf{x})\ddot{\mathbf{u}}(\mathbf{x},t) = \int_{B_\delta(\mathbf{x})} \mathbf{f}(\mathbf{y},\mathbf{x},t)\, d\mathbf{y} + \mathbf{b}(\mathbf{x},t). \tag{3.1}$$

A *material model* in peridynamics supplies values of $\mathbf{f}(\mathbf{y},\mathbf{x},t)$ in terms of the deformations of the families of $\mathbf{x}$ and $\mathbf{y}$ and any other relevant variables such as temperature. In general, material models in peridynamics are specified using operators called *states* that are nonlocal analogues of second order tensors [34]. Many material models have been developed for peridynamics, and any material model from the local theory can be translated into peridynamic form [39]. 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. Peridynamics can model fracture because the equation of motion (3.1) is an integro-differential equation that does not involve the partial derivatives of displacement with respect to position. However, the present paper concerns only small deformations in the linear regime of material response and does not address fracture. The extension of the methods described here to determine a linear peridynamic material model to the nonlinear regime, including fracture, is under investigation in separate work. The remainder of this paper deals with a specific material model described in the next section. Note that even though MD displacements are dynamic, we smooth them in time as described in (2.10) and (2.11) so that the time-space smoothed MD data can be described by the static counterpart of the nonlocal equation (2.7),

$$-\int_{B_\delta(\mathbf{x})} \mathbf{f}(\mathbf{y},\mathbf{x},\mathbf{u})\, d\mathbf{y} = \mathbf{b}(\mathbf{x}), \tag{3.2}$$

where $\mathbf{f}$ is to be determined (see the following section) and where we introduced the nonlocal interaction region $B_\delta(\mathbf{x})$. The *horizon* δ determines the extent of the nonlocal interactions. Although, according to Section 2, δ could be determined by the cutoff distance d associated with $\mathbf{F}$ and the radius of the smoothing function R, in the following discussion it is treated as

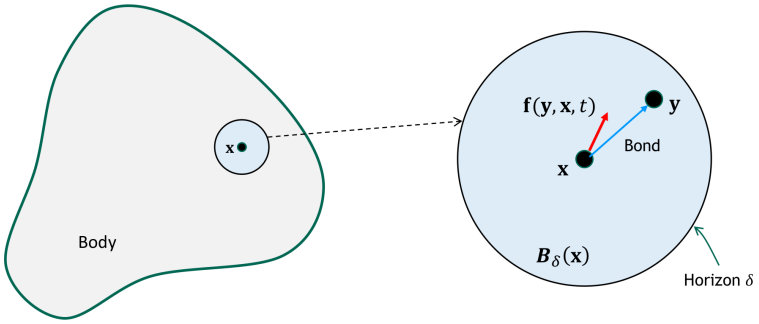

Figure 1: Left: the family of a point $\mathbf{x}$ in a peridynamic body. Right: typical bond and bond force vector.

a learned parameter (Section 6.2). This approach allows for a finite value of δ to be obtained even if $d = \infty$, as would be the case with the Lennard-Jones potential.

3.2. The Linear Peridynamic Solid (LPS) Model

The main application considered in this work is the simulation of displacements in single-layered graphene. The graphene sheet is treated using a two-dimensional nonlocal model under the assumption of plane stress, which is appropriate for a thin sheet. The pairwise bond force density $\mathbf{f}$ is determined using the state-based linear peridynamic solid (LPS) model. The LPS model is a prototypical state-based model appropriate for isotropic elastic materials. It may be regarded as a nonlocal generalization of the local model for an isotropic solid, which contains contributions from shear and dilatation. The LPS model has advantages over the previously developed bond-based peridynamic models in that it is not restricted to a Poisson's ratio of 1/4. The LPS model has known well-posedness properties under certain assumptions [24]. This section summarizes the mathematical formulation for the LPS model and illustrates a meshfree discretization [40–45].

Consider a 2D body occupying the domain $\Omega \subset \mathbb{R}^2$, and let θ be the nonlocal dilatation, generalizing the local divergence of the displacement. Let $K(r)$ be the influence function [46] which modulates nonlocal effects within a peridynamic model. In this work, we assume K to be a radial function compactly supported on the δ-ball $B_\delta(\mathbf{x})$ with α-th order singularity:

$$K(\mathbf{x}, \mathbf{y}) := K(|\mathbf{x} - \mathbf{y}|) = \frac{P(|\mathbf{x} - \mathbf{y}|)}{|\mathbf{x} - \mathbf{y}|^\alpha} \tag{3.3}$$

where $P(r)$ is a bounded function in $[0, \delta]$. The momentum balance and nonlocal dilatation

are given by

$$\begin{aligned}\mathcal{L}_K\mathbf{u}(\mathbf{x}) := &-\frac{C_1}{m(\delta)}\int_{B_\delta(\mathbf{x})}(\lambda-\mu)\,K(|\mathbf{y}-\mathbf{x}|)\,(\mathbf{y}-\mathbf{x})\,(\theta(\mathbf{x})+\theta(\mathbf{y}))\,d\mathbf{y}\\ &-\frac{C_2}{m(\delta)}\int_{B_\delta(\mathbf{x})}\mu K(|\mathbf{y}-\mathbf{x}|)\frac{(\mathbf{y}-\mathbf{x})\otimes(\mathbf{y}-\mathbf{x})}{|\mathbf{y}-\mathbf{x}|^2}\,(\mathbf{u}(\mathbf{y})-\mathbf{u}(\mathbf{x}))\,d\mathbf{y}=\mathbf{b}(\mathbf{x}),\end{aligned} \tag{3.4}$$

$$\theta(\mathbf{x}) := \frac{2}{m(\delta)}\int_{B_\delta(\mathbf{x})} K(|\mathbf{y}-\mathbf{x}|)(\mathbf{y}-\mathbf{x})\cdot(\mathbf{u}(\mathbf{y})-\mathbf{u}(\mathbf{x}))\,d\mathbf{y},$$

where $\mathbf{u}\in\mathbb{R}^2$ denotes the displacement, $\mathbf{b}\in\mathbb{R}^2$ the prescribed body force density, and $m(\delta) := \int_{B_\delta(\mathbf{0})} K(|\mathbf{z}|)|\mathbf{z}|^2 d\mathbf{z}$ the weighted volume. Here we note that $m(\delta)$ is independent of $\mathbf{x}$, and it is determined by the horizon size δ and the influence function K. In the present notation, the nonlocal operator $\mathcal{L}_K[\mathbf{u}](\mathbf{x})$ in (3.4) has the subscript K to emphasize its dependence on the influence function K. This operator corresponds to the integral term $-\int \mathbf{f}(\mathbf{y},\mathbf{x},\mathbf{u})d\mathbf{y}$ in (2.7), or, equivalently, (3.2).

Given a forcing term $\mathbf{b}$, in order to guarantee the existence of a unique solution $\mathbf{u}$, "nonlocal boundary conditions", or volume constraints, must be prescribed on an appropriate *interaction* domain Ω_I, so that the LPS problem becomes

$$\begin{cases}\mathcal{L}_K[\mathbf{u}](\mathbf{x}) = \mathbf{b}(\mathbf{x}) & \mathbf{x}\in\Omega,\\ \mathcal{B}_I\mathbf{u}(\mathbf{x}) = \mathbf{q}(\mathbf{x}) & \mathbf{x}\in\Omega_I.\end{cases} \tag{3.5}$$

Here, $\mathcal{B}_I$ is a nonlocal interaction operator specifying a volume constraint. In this work, without loss of generality, we consider the Dirichlet condition $\mathcal{B}_I = \mathcal{I}$, where $\mathcal{I}$ is the identity operator. Other types of conditions, e.g., Neumann [41, 43, 47, 48], Robin [44, 49] or periodic [32], are also compatible with our learning algorithm.

A meshfree discretization of the LPS model. Given a collection of material points $\mathcal{X} = \{\mathbf{x}_i\}_{i=1,2,\dots,N_p}$, we numerically evaluate $\mathcal{L}_K(\mathbf{u})$ by employing the meshfree, particle discretization introduced in [41], which features ease of implementation and efficiency. At each material point $\mathbf{x}_i$, we adopt the following quadrature rule to approximate the integral in $\mathcal{L}_K$ in (3.4), which we now denote by $\mathcal{L}_K^h$.

$$\begin{aligned}\mathcal{L}_K^h\mathbf{u}(\mathbf{x}_i) := &-\frac{C_1}{m_i(\delta)}\sum_{\mathbf{x}_j\in B_\delta(\mathbf{x_i})}(\lambda-\mu)\,K_{ij}\,(\mathbf{x}_j-\mathbf{x}_i)\left(\theta^h(\mathbf{x}_i)+\theta^h(\mathbf{x}_j)\right)W_{j,i}\\ &-\frac{C_2}{m_i(\delta)}\sum_{\mathbf{x}_j\in B_\delta(\mathbf{x}_i)}\mu K_{ij}\frac{(\mathbf{x}_j-\mathbf{x}_i)\otimes(\mathbf{x}_j-\mathbf{x}_i)}{|\mathbf{x}_j-\mathbf{x}_i|^2}\,(\mathbf{u}(\mathbf{x}_j)-\mathbf{u}(\mathbf{x}_i))\,W_{j,i}=\mathbf{b}(\mathbf{x}_i),\end{aligned} \tag{3.6}$$

$$\theta^h(\mathbf{x}_i) := \frac{2}{m_i(\delta)} \sum_{\mathbf{x}_j \in B_\delta(\mathbf{x}_i)} K_{ij}(\mathbf{x}_j - \mathbf{x}_i) \cdot (\mathbf{u}(\mathbf{x}_j) - \mathbf{u}(\mathbf{x}_i))\, W_{j,i}, \tag{3.7}$$

where $K_{ij} := K(\mathbf{x}_j, \mathbf{x}_i)$ and $m_i(\delta) := \sum_{\mathbf{x}_j \in B_\delta(\mathbf{x}_i)} K_{ij}|\mathbf{x}_j - \mathbf{x}_i|^2 W_{j,i}$. The quadrature weights $W_{j,i}$ are obtained for material points on different subdomains $B_\delta(\mathbf{x}_i)$, from the following optimization problem

$$\underset{\{\omega_{j,i}\}}{\operatorname{argmin}} \sum_{\mathbf{x}_j \in \mathcal{X} \cap B_\delta(\mathbf{x}_i) \setminus \{\mathbf{x}_i\}} W_{j,i}^2 \quad \text{s. t.,} \sum_{\mathbf{x}_j \in B_\delta(\mathbf{x}_i)} q(\mathbf{x}_i, \mathbf{x}_j) W_{j,i} = \int_{B_\delta(\mathbf{x}_i)} q(\mathbf{x}_i, \mathbf{y}) d\mathbf{y} \quad \forall\, q \in \boldsymbol{V},$$

where $\boldsymbol{V}$ denotes the space of functions which should be integrated exactly. Following [41], in this work we take $\boldsymbol{V} := \left\{ q(\mathbf{y}) = \frac{p(\mathbf{y})}{|\mathbf{y}-\mathbf{x}|^3} \,|\, p \in \boldsymbol{P}_5(\mathbb{R}^2) \text{ such that } \int_{B_\delta(\mathbf{x})} q(\mathbf{y}) d\mathbf{y} < \infty \right\}$ and $\boldsymbol{P}_5(\mathbb{R}^2)$ denotes the space of quintic polynomials .

Although the developed learning approach as well as the meshfree quadrature rule can be applied to the general collection of material points $\mathcal{X}$, in this work we consider the uniform Cartesian grid for simplicity:

$$\mathcal{X} := \{(p_1 h, p_2 h) | \boldsymbol{p} = (p_1, p_2) \in \mathbb{Z}^2\} \cap (\Omega \cap \Omega_I),$$

where h is the spatial grid size. Note that, as we discuss in the next section and in Section 6.4, training and validation samples may be generated on different grids. Furthermore, to emphasize the fact that each sample may be generated on a different grid, we denote the grid corresponding to the s-th sample by $\mathcal{X}^s$.

Remark 1. Using the same arguments as in [42], it can be seen that the chosen quadrature rule provides a consistent approximation of $\mathcal{L}_K(\mathbf{u})$ when $\alpha < 3$. Therefore, in the learning algorithm, we require the fractional order α to be bounded by 3, and we note that this requirement may be further relaxed by considering other discretization methods.

4. Operator Regression for the LPS Model

In this section we illustrate how to extend the data-driven approach developed in [32, 33], known as *nonlocal operator regression*, to the LPS model. Section 4.1 first describes the regression algorithm under the assumption that coarse-grained MD displacements are available at any material point. Then, Section 4.2 introduces solvability constraints that guarantee that the optimal nonlocal model is well-posed by construction[1]. Lastly, Section 4.3

[1]The solvability conditions derived in [32] for 1D nonlocal diffusion problems are not applicable to the more complex LPS model considered in the current work.

summarizes the complete workflow of the data-driven nonlocal operator regression algorithm, which is the process of going from high fidelity MD simulations to data-driven optimal kernels via coarse graining and operator regression.

4.1. Operator Regression Algorithm

The foundation of the nonlocal operator regression algorithm is the fact that a coarse-grained displacement $\mathbf{u}$ follows a nonlocal evolution law of the form (3.5). For this nonlocal model, we seek to identify an optimal constitutive relation on the basis of MD data sets.

Let $\{\mathbf{u}^s(\mathbf{x}_{i,s}), \mathbf{b}^s(\mathbf{x}_{i,s})\}$, $s = 1, \cdots, S$, be given pairs of displacement and body force fields available at $\mathbf{x}_{i,s} \in \mathcal{X}^s$, and let $\mathcal{L}_K$ be the LPS operator defined in (3.4) parametrized by the material properties λ and μ and by the influence function K. We aim to learn an optimal nonlocal operator $\mathcal{L}_K$. This optimal operator consists of the influence function K, which may be sign-changing, and parameters λ and μ, such that the action of $\mathcal{L}_K$ most closely maps $\mathbf{u}^s(\mathbf{x})$ to $\mathbf{b}^s(\mathbf{x})$ for all s. Formally, the optimal influence function and parameters, (λ^*, μ^*, K^*), are the solution of the following optimization problem:

$$(\lambda^*, \mu^*, K^*) = \underset{\lambda,\mu,K}{\operatorname{argmin}} \frac{1}{S} \sum_{s=1}^{S} \left\| \mathcal{L}_K^h[\mathbf{u}^s](\mathbf{x}_{i,s}) - \mathbf{b}^s(\mathbf{x}_{i,s}) \right\|^2_{\ell^2(\mathcal{X}^s)}.$$

To increase the flexibility of the algorithm, each sample can be available on different point sets $\mathcal{X}^s$.

The influence function $K(\mathbf{x}, \mathbf{y})$ will now be parameterized. Following [32], assume that K has the form of (3.3), and represent its numerator P as a linear combination of Bernstein polynomials evaluated at $|\mathbf{x} - \mathbf{y}|$:

$$K(\mathbf{x}, \mathbf{y}) = \sum_{k=0}^{M} \frac{D_k}{|\mathbf{x} - \mathbf{y}|^\alpha} B_{k,M}\left(\frac{|\mathbf{x} - \mathbf{y}|}{\delta}\right).$$

Here the Bernstein polynomials are defined as

$$B_{k,M}(r) = \binom{M}{k} r^k (1 - r)^{M-k}, \qquad \text{for } 0 \le r \le 1.$$

To allow the learning of nonlocal models whose kernels may be partially negative, we allow $D_k \in \mathbb{R}$, for all k. This generality, however, might compromise the well-posedness of the resulting optimal model, since known well-posedness results on LPS models only apply to positive kernels [50]. To guarantee that the LPS model associated with (λ^*, μ^*, K^*) is solvable by construction, we embed in our algorithm sufficient well-posedness conditions for the

discretized operator; these are described in detail in the next section.

The formulation of the constrained optimization problem is as follows. Given a collection of training samples $\{\mathbf{u}^s(\mathbf{x}_{i,s}), \mathbf{b}^s(\mathbf{x}_{i,s})\}$, $s = 1, \cdots, S$, we seek to learn the parameters λ and μ, the Bernstein polynomial coefficients $\mathbf{D} = [D_0, \cdots, D_M] \in \mathbb{R}^{M+1}$, the order α, the horizon δ, and the polynomial order M by minimizing the mean square loss (MSL) of the LPS equation:

$$\begin{cases} (\lambda^*, \mu^*, \mathbf{D}^*, \alpha^*, \delta^*, M^*) = \underset{\lambda,\mu,\mathbf{D},\alpha,\delta,M}{\operatorname{argmin}} \dfrac{1}{S}\displaystyle\sum_{s=1}^{S} ||\mathcal{L}_K^h \mathbf{u}^s(\mathbf{x}_{i,s}) - \mathbf{b}^s(\mathbf{x}_{i,s})||^2_{\ell^2(\mathcal{X}^s)} \\ \\ \text{subject to solvability constraints.} \end{cases} \tag{4.1}$$

4.2. Solvability Constraints for the Discretized LPS Model

When the influence function K as described in (3.3) is nonnegative and $\alpha < 4$, the LPS model is well-posed, as shown, for example, in [50]. However, several works have indicated the practical need for sign-changing kernels [30, 31, 33, 51, 52]. While it is unclear whether multiscale physics inherently leads to sign-changing kernels or if equally descriptive positive kernels could be derived, in [32] the authors found that allowing for sign-changing kernels provides a significant increase in accuracy when modeling high-frequency material response. Therefore, in this work we seek a well-posed LPS model with possibly sign-changing influence functions K. As there is no available theory on sufficient conditions for the well-posedness of LPS models with sign-changing K, we impose well-posedness conditions on the discretized system directly; as a result, well-posedness of the learnt model is guaranteed for the discretization method used during training.

An inequality constraint for the well-posedness of the meshfree discretization approach (3.6)-(3.7) that allows for sign-changing influence functions will now be derived. For simplicity of analysis, and without loss of generality, assume homogeneous Dirichlet-type boundary conditions: $\mathbf{u}(\mathbf{x}) = 0$ in Ω_I.

For the derivation of the solvability constraints that will be employed in our algorithm to ensure the well-posedness of the discretized LPS model, first write the discretized LPS model (3.6)-(3.7) as the following linear system:

$$\begin{pmatrix} \mu\Gamma & (\Phi)^{\mathrm{t}} \\ \Phi & -\frac{1}{\lambda-\mu} I \end{pmatrix} \begin{pmatrix} \mathbf{U} \\ \mathbf{\Theta} \end{pmatrix} = \begin{pmatrix} -\mathbf{B} \\ \mathbf{0} \end{pmatrix}. \tag{4.2}$$

Here, $\mathbf{U} \in \mathbb{R}^{2N_p}$ and $\frac{1}{\lambda-\mu}\mathbf{\Theta} \in \mathbb{R}^{N_p}$ are the vectors of the degrees of freedom (DOFs) of the displacement $\mathbf{u}$ and the nonlocal dilatation θ:

$$\mathbf{U} = [(\mathbf{u}(\mathbf{x}_1))^t, \cdots, (\mathbf{u}(\mathbf{x}_{N_p}))^t]^t, \quad \mathbf{\Theta} = [(\lambda-\mu)\theta(\mathbf{x}_1), \cdots, (\lambda-\mu)\theta(\mathbf{x}_{N_p})]^t.$$

$\mathbf{B}$ is the vector of DOFs of the body load and has the same length and ordering of indices as $\mathbf{U}$. I is an $N_p \times N_p$ identity matrix, and Γ and Φ are the matrices that correspond to the deviatoric and dilatation contributions of the deformation:

$$\Gamma\mathbf{U} = [(\mathbf{f}_{dev}(\mathbf{x}_1))^t, \cdots, (\mathbf{f}_{dev}(\mathbf{x}_{N_p}))^t]^t, \quad \Phi^t\mathbf{U} = [f_{dil}(\mathbf{x}_1), \cdots, f_{dil}(\mathbf{x}_{N_p})]^t,$$

where

$$\mathbf{f}_{dev}(\mathbf{x}_i) = -\frac{C_2}{m_i(\delta)} \sum_{\mathbf{x}_j \in B_\delta(\mathbf{x}_i)} K_{ij} \frac{(\mathbf{x}_j - \mathbf{x}_i) \otimes (\mathbf{x}_j - \mathbf{x}_i)}{|\mathbf{x}_j - \mathbf{x}_i|^2} (\mathbf{u}(\mathbf{x}_j) - \mathbf{u}(\mathbf{x}_i)) W_{j,i}$$
$$f_{dil}(\mathbf{x}_i) = \frac{2}{m_i(\delta)} \sum_{\mathbf{x}_j \in B_\delta(\mathbf{x}_i)} K_{ij} (\mathbf{x}_j - \mathbf{x}_i) \cdot (\mathbf{u}(\mathbf{x}_j) - \mathbf{u}(\mathbf{x}_i)) W_{j,i}.$$

In what follows, for each vector $\mathbf{V} \in \mathbb{R}^{2N_p}$, write $\mathbf{V} = [\mathbf{v}_1^t, \cdots, \mathbf{v}_{N_p}^t]^t$ with each $\mathbf{v}_i \in \mathbb{R}^2$. C denotes a generic constant. The following theorem provides sufficient conditions that guarantee the solvability of (4.2). Let the energy "norm" be defined as $||\mathbf{V}||_E^2 := \mathbf{V}^t\Gamma\mathbf{V}$ for all $\mathbf{V} \in \mathbb{E} \setminus \mathbf{0}$, where $\mathbb{E}$ denotes the quotient space of $\mathbb{R}^{2N_p}$ by the discrete space of infinitesimally rigid displacements:

$$\Pi_{\mathcal{X}} = \{[(\mathbb{Q}\mathbf{x}_1 + \mathbf{d})^t, \cdots, (\mathbb{Q}\mathbf{x}_{N_p} + \mathbf{d})^t]^t,\ \mathbb{Q} \in \mathbb{R}^{2\times 2},\ \mathbb{Q}^T = -\mathbb{Q},\ \mathbf{d} \in \mathbb{R}^2\}.$$

Note that although we use the norm notation, $||\cdot||$, in general, $||\cdot||_E$ does not define a norm unless Γ is symmetric and positive definite. In fact, $||\cdot||_E$ is indeed a norm only when the discrete coercivity condition, reported in the following theorem, is satisfied.

Theorem 4.1 (Sufficient Conditions for Solvability)**.** *The discretized LPS formulation* (4.2) *is solvable for any values of* $\lambda + \mu > 0$ *and* $\mu > 0$*, provided that the following conditions hold:*

i. Discrete Continuity and Coercivity: $\exists\, \underline{C}_\Gamma > 0, \overline{C}_\Gamma < \infty$ *s.t.,*

$$||\mathbf{V}||_E^2 \geq \underline{C}_\Gamma ||\mathbf{V}||_{\ell^2}^2 \ \text{ and } \ ||\mathbf{V}||_E^2 \leq \overline{C}_\Gamma ||\mathbf{V}||_{\ell^2}^2 \quad \forall\, \mathbf{V} \in \mathbb{E}\backslash\{\mathbf{0}\} \tag{4.3}$$

ii. Discrete Inf-Sup: $\exists\, C_\Phi > 0$, *s.t.,*

$$\inf_{\mathbf{P} \in \mathbb{R}^{N_p}\backslash\{\mathbf{0}\}} \sup_{\mathbf{V} \in \mathbb{E}\backslash\{\mathbf{0}\}} \frac{\mathbf{V}^t\Phi^t\mathbf{P}}{||\mathbf{V}||_E ||\mathbf{P}||_{\ell^2}} \geq C_\Phi, \tag{4.4}$$

iii. Discrete, generalized Cauchy-Schwarz: $||\mathbf{V}||_E^2 \geq 2||\Phi\mathbf{V}||_{\ell^2}^2,\ \forall\, \mathbf{V} \in \mathbb{E}.$ (4.5)

Remark 2. In the continuous case with $K \geq 0$, the property (4.5) is an immediate result from the Cauchy-Schwarz inequality, see, for example, [50]. This property yields the equivalence of the semi-norm from the deviatoric part of the deformation and the full strain

energy.

Proof. Inequality (4.3) implies that the energy norm is a norm in $\mathbb{E} \setminus \{\mathbf{0}\}$. We consider two scenarios: $\lambda - \mu \geq 0$ and $\lambda - \mu < 0$.

Case 1: $\lambda - \mu \geq 0$. The symmetry property of the Cartesian grids implies that $m_i(\delta) = m_j(\delta) := m$ and $W_{j,i} = W_{i,j}$ for all $i, j \in \{1, \cdots, N_p\}$. By taking the inner product of (4.2) with $(\mathbf{V}^t, \mathbf{\Xi}^t)$, component-wise, we reformulate the system as a general mixed formulation: find $\mathbf{U} \in \mathbb{R}^{2N_p}$ and $\mathbf{\Theta} \in \mathbb{R}^{N_p}$ such that

$$\begin{aligned} a(\mathbf{U}, \mathbf{V}) + b(\mathbf{V}, \mathbf{\Theta}) &= (-\mathbf{B}, \mathbf{V}), & \forall\, \mathbf{V} \in \mathbb{R}^{2N_p} \\ b(\mathbf{U}, \mathbf{\Xi}) - c(\mathbf{\Theta}, \mathbf{\Xi}) &= 0, & \forall\, \mathbf{\Xi} \in \mathbb{R}^{N_p}. \end{aligned}$$

Here $(\cdot, \cdot)$ denotes the inner product, $a(\mathbf{U}, \mathbf{V}) := \mu \mathbf{V}^t \Gamma \mathbf{U}$, $b(\mathbf{V}, \mathbf{\Xi}) := \mathbf{\Xi}^t \Phi \mathbf{V}$, and $c(\mathbf{\Theta}, \mathbf{\Xi}) := \frac{1}{\lambda - \mu} \mathbf{\Theta}^t \mathbf{\Xi}$. Firstly, note that when (4.3) is satisfied, the symmetric bilinear form $a(\cdot, \cdot)$ is continuous and coercive. Similarly,

$$\begin{aligned} b(\mathbf{V}, \mathbf{\Xi}) =& -\frac{C_1}{m} \sum_{i=1}^{N_p} \left(\sum_{\mathbf{x}_j \in B_\delta(\mathbf{x_i})} (\lambda - \mu)\, K_{ij} \mathbf{v}_i^t\, (\mathbf{x}_j - \mathbf{x}_i)\, (\xi_i + \xi_j)\, W_{j,i} \right) \\ =& \frac{C_1}{2m} \sum_{i=1}^{N_p} \left(\sum_{\mathbf{x}_j \in B_\delta(\mathbf{x_i})} (\lambda - \mu)\, K_{ij} (\mathbf{v}_j - \mathbf{v}_i)^t\, (\mathbf{x}_j - \mathbf{x}_i)\, (\xi_i + \xi_j)\, W_{j,i} \right) \\ \leq& C ||\mathbf{V}||_{\ell^2} ||\mathbf{\Xi}||_{\ell^2} \leq C ||\mathbf{V}||_E ||\mathbf{\Xi}||_{\ell^2} \end{aligned}$$

so that $b(\cdot, \cdot)$ is also a continuous bilinear form. By combining (4.3) and (4.4) with the fact that $c(\mathbf{\Theta}, \mathbf{\Xi}) = \frac{1}{\lambda - \mu} \mathbf{\Theta}^t \mathbf{\Xi} \leq C ||\mathbf{\Theta}||_{\ell^2} ||\mathbf{\Xi}||_{\ell^2}$ and $c(\mathbf{\Xi}, \mathbf{\Xi}) = \frac{1}{\lambda - \mu} ||\mathbf{\Xi}||_{\ell^2}^2 \geq 0$, the well-posedness of (4.2) follows using the same arguments of [53, Section 2.2].

Case 2: $\lambda - \mu < 0$. In this case $c(\mathbf{\Xi}, \mathbf{\Xi})$ is not coercive with respect to the ℓ^2 norm and therefore the theory in [53] does not apply. By reducing the discrete system (4.2) to $((\lambda - \mu)\Phi^t \Phi + \mu \Gamma)\mathbf{U} = -\mathbf{B}$, we note that $(\lambda - \mu)\Phi^t \Phi + \mu \Gamma$ is not solvable, or, equivalently, non-invertible, if and only if there exists $\mathbf{V} \in \mathbb{E} \setminus \{\mathbf{0}\}$ such that $((\lambda - \mu)\Phi^t \Phi + \mu \Gamma)\mathbf{V} = \mathbf{0}$. We prove that this is not possible by showing that

$$\mathbf{V}^t ((\lambda - \mu)\Phi^t \Phi + \mu \Gamma)\mathbf{V} = 0 \Leftrightarrow \mathbf{V} = \mathbf{0}.$$

We split the proof in two parts: $\lambda \geq 0$ and $\lambda < 0$, respectively. First, for $\lambda \geq 0$, (4.3) and

(4.5) yield

$$\begin{aligned} 0 =& \mathbf{V}^t((\lambda-\mu)\Phi^t\Phi+\mu\Gamma)\mathbf{V} = \lambda\mathbf{V}^t\Phi^t\Phi\mathbf{V}+\mu(||\mathbf{V}||_E^2-||\Phi\mathbf{V}||_{\ell^2}^2) \\ \geq& \lambda\mathbf{V}^t\Phi^t\Phi\mathbf{V}+\frac{\mu}{2}||\mathbf{V}||_E^2 \geq \frac{\mu}{2}||\mathbf{V}||_{\ell^2}^2. \end{aligned}$$

Hence $\mathbf{V}=\mathbf{0}$, which contradicts our assumption. Similarly, when $\lambda<0$, we assume that $||\mathbf{V}||_E^2>0$. From the assumptions $\lambda-\mu<0$ and $\lambda+\mu>0$ we have

$$0 =\mathbf{V}^t((\lambda-\mu)\Phi^t\Phi+\mu\Gamma)\mathbf{V} \geq \frac{\lambda-\mu}{2}\mathbf{V}^t\Gamma\mathbf{V}+\mu||\mathbf{V}||_E^2 = \frac{\lambda+\mu}{2}||\mathbf{V}||_E^2,$$

which, again, implies $\mathbf{V}=\mathbf{0}$, contradicting our assumption. Therefore, $(\lambda-\mu)\Phi^t\Phi+\mu\Gamma$ is invertible and (4.2) is solvable. □

Remark 3. When the influence function K is nonnegative, the continuous LPS model satisfies the ellipticity condition and the inf-sup condition. Moreover, the energy density associated with the dilatation part is bounded by energy density associated with the deviatoric part of the deformation, as shown in [50]. These facts, intuitively, support our well-posedness result, where the constraints (4.3), (4.4) and (4.5) are equivalent to requiring that the discretized LPS model associated with a sign-changing K satisfies the same three properties as its nonnegative-kernel counterpart.

Therefore, we augment our learning problem (4.1) with the inequality (solvability) constraints corresponding to (4.3), (4.4) and (4.5). The constants $\underline{C}_\Gamma$, $\overline{C}_\Gamma$, and C_Φ are only related to the influence function K and are independent of the shear and Lamé parameters λ and μ. We calculate the inf-sup constant C_Φ in (4.4) by solving an eigenvalue problem, as indicated by the following theorem.

Theorem 4.2. *The inf-sup constant in (4.4) in Theorem 4.1 can be expressed as*

$$C_\Phi = \Lambda_{min}(\Phi\Gamma^\dagger\Phi^t)$$

where, for a given square matrix M, $M^\dagger$ denotes its pseudoinverse and $\Lambda_{min}(M)$ denotes its smallest nonzero eigenvalue.

Proof. The proof can be obtained following the same arguments as in [54, Proposition 3.1]. □

For any given K as in (3.3) and a fixed discretization method, the largest eigenvalue of Γ is bounded by construction. Therefore $\overline{C}_\Gamma$ is finite and, in practice, (4.3), (4.4) and (4.5)

can be imposed as:

$$\begin{aligned} &\Lambda_{min}(\Gamma) \geq \zeta, \\ &\Lambda_{min}(\Phi\Gamma^{\dagger}\Phi^{t}) \geq \zeta, \\ &\Lambda_{min}(\Gamma - 2\Phi^{t}\Phi) \geq 0, \end{aligned} \tag{4.6}$$

where $\zeta > 0$ is a given, small number.

We can now state the solvability-constrained optimization problem. Rename the matrices Γ and Φ in (4.2) as $\Gamma_{(\alpha,\mathbf{D},\delta,M)}$ and $\Phi_{(\alpha,\mathbf{D},\delta,M)}$ indicating that they are parameterized with the Bernstein polynomial coefficients $\mathbf{D}$, the fractional order α, the horizon δ, and the highest Bernstein polynomial order M. Given the tolerance parameter $\zeta > 0$, the learning problem is stated as

$$\begin{cases} (\lambda^*, \mu^*, \mathbf{D}^*, \alpha^*, \delta^*, M^*) = \underset{\lambda,\mu,\mathbf{D},\alpha,\delta,M}{\operatorname{argmin}} \dfrac{1}{S}\displaystyle\sum_{s=1}^{S} ||\mathcal{L}_K^h \mathbf{u}^s(\mathbf{x}_{i,s}) - \mathbf{b}^s(\mathbf{x}_{i,s})||^2_{\ell^2(\mathcal{X}^s)} \\ \qquad \text{subject to:} \quad \lambda + \mu > 0,\ \mu > 0,\ \alpha < 3,\ \Lambda_{min}(\Gamma_{(\alpha,\mathbf{D},\delta,M)}) \geq \zeta, \\ \qquad\qquad \Lambda_{min}(\Phi_{(\alpha,\mathbf{D},\delta,M)}\Gamma^{\dagger}_{(\alpha,\mathbf{D},\delta,M)}\Phi^{t}_{(\alpha,\mathbf{D},\delta,M)}) \geq \zeta, \\ \qquad\qquad \Lambda_{min}(\Gamma_{(\alpha,\mathbf{D},\delta,M)} - 2\Phi^{t}_{(\alpha,\mathbf{D},\delta,M)}\Phi_{(\alpha,\mathbf{D},\delta,M)}) \geq 0. \end{cases} \tag{4.7}$$

Remark 4. The constraints in (4.6) are sufficient to guarantee that the model associated with the optimal influence function K is well-posed only when discretized with the same technique and resolution utilized during training. Being only sufficient, these conditions may yield an optimal influence function whose associated model is still well-posed when discretized with different schemes or resolutions. More discussion and numerical tests on this topic can be found in Section 6.4.

4.3. Algorithm and Workflow

In this section we describe the algorithmic details of our learning approach and describe the learning workflow that, starting with MD displacements, delivers the optimal influence function K and the material parameters.

Numerically, the constrained optimization problem (4.7) poses several challenges. Firstly, the quadrature weights $W_{j,i}$ generated in the preprocessing generally depend on the horizon size δ, which hinders the application of a suite of continuous optimization techniques such as gradient descent or Adam. A similar issue applies to M. Moreover, due to the solvability constraints, (4.7) is expected to be nonconvex and likely to exhibit local minima. Lastly, the numerical evaluations of the eigenvalues are time-consuming. For all these reasons, for the

Algorithm 1 Two-stage strategy to solve (4.7) for $(\lambda^*, \mu^*, \alpha^*, \mathbf{D}^*)$.

With fixed δ and M, initialize λ, μ, α, $D_k^{(0)} \sim \mathcal{U}(0,1)$,where $\mathcal{U}(a,b)$ denotes the uniform distribution on (a,b).
Obtain $(\lambda^{pre}, \mu^{pre}, \alpha^{pre}, \mathbf{D}^{pre})$ as a local minimum of $L^{pre}(\lambda, \mu, \alpha, \mathbf{D})$, using the Adam optimizer.
Update $\lambda^{pre} \leftarrow \mathrm{ReLU}(\lambda^{pre}+\mu^{pre})-\mathrm{ReLU}(\mu^{pre})$, $\mu^{pre} \leftarrow \mathrm{ReLU}(\mu^{pre})$, $\alpha^{pre} \leftarrow 3-\mathrm{ReLU}(3-\alpha^{pre})$ and $\mathbf{D}^{pre} \leftarrow \mathrm{ReLU}(\mathbf{D}^{pre})$.
Initialize $(\lambda^{(0)}, \mu^{(0)}, \alpha^{(0)}, \mathbf{D}^{(0)}) = (\lambda^{pre}, \mu^{pre}, \alpha^{pre}, \mathbf{D}^{pre})$ and $\varpi_1^{(0)} = \varpi_2^{(0)} = \varpi_3^{(0)} = 1$.
Set `STEP_MAX` $= 100$, $\phi_1 = \phi_2 = \phi_3 = 0$, $\psi = 1$, $r_1 = 5$, $r_2 = 1/4$, $\epsilon = 10^{-8}$.
while $j \leq$ `STEP_MAX`: **do** ▷ Perform Augmented Lagrangian Algorithm
    Solve the unconstrained optimization problem

$$(\lambda^{(j)}, \mu^{(j)}, \alpha^{(j)}, \mathbf{D}^{(j)}, \boldsymbol{\varpi}^{(j)}) = \underset{\lambda,\mu,\alpha,\mathbf{D},\boldsymbol{\varpi}}{\operatorname{argmin}} L^{corr}(\lambda, \mu, \alpha, \mathbf{D}, \boldsymbol{\varpi}).$$

    if $H_p(\alpha^{(j)}, \mathbf{D}^{(j)}, \boldsymbol{\varpi}^{(j)}) \leq \epsilon$ for all $p \in \{1,2,3\}$, **then**
        Stop.
    else
        if $\exists\, p \in \{1,2,3\}$ s.t. $H_p(\alpha^{(j)}, \mathbf{D}^{(j)}, \boldsymbol{\varpi}^{(j)}) \geq r_2 H_p(\alpha^{(j-1)}, \mathbf{D}^{(j-1)}, \boldsymbol{\varpi}^{(j-1)})$ **then**
            Update penalty $\psi \leftarrow r_1 \psi$.
            if $\psi \geq 10^{20}$ **then**
                Stop.
        else
            Update Lagrange multiplier $\phi_p \leftarrow \phi_p + \psi H_p(\alpha^{(j)}, \mathbf{D}^{(j)}, \boldsymbol{\varpi}^{(j)})$ for $p = 1,2,3$.
    Update the iteration number $j \leftarrow j+1$.
$(\lambda^*, \mu^*, \alpha^*, \mathbf{D}^*) = (\mathrm{ReLU}(\lambda^{(j)}+\mu^{(j)}) - \mathrm{ReLU}(\mu^{(j)}), \mathrm{ReLU}(\mu^{(j)}), 3 - \mathrm{ReLU}(3-\alpha^{(j)}), \mathbf{D}^{(j)})$.

sake of numerical efficiency, we treat δ and M as hyperparameters to be separately tuned to achieve the best learning accuracy without overfitting. As suggested by [32], the optimization problem (4.7) is split into a *prediction* step (without constraints) and a *correction* step (with constraints), and propose a "two-stage" strategy, whose key components are summarized in Algorithm 1.

The prediction step of the algorithm relies on the fact that, as shown in [50], when $\alpha < 3$, $\lambda + \mu > 0$, $\mu > 0$ and $K \geq 0$, the LPS problem is guaranteed to be well-posed, and therefore no additional solvability constraint of K are required. Therefore, in the prediction step, we find a set of nonnegative Bernstein coefficients that will be used as an initial guess for the second, correction step. The nonnegative coefficients and the corresponding nonnegative influence function are denoted by $\mathbf{D}_k^{pre}$ and K^{pre}, respectively. These are obtained by solving the following, unconstrained problem, whose full solution is denoted by $(\lambda^{pre}, \mu^{pre}, \alpha^{pre}, \mathbf{D}^{pre})$.

$$
\begin{aligned}
L^{pre}(\lambda,\mu,\alpha,\mathbf{D}) := \frac{1}{S}\sum_{s=1}^{S}\sum_{\mathbf{x}_i\in\mathcal{X}^s}\Bigg|\mathbf{b}_i^s + \sum_{k=0}^{M}\frac{\mathrm{ReLU}(D_k)}{m_i(\delta)}\sum_{\mathbf{x}_j\in B_\delta(\mathbf{x_i})}\frac{B_{k,M}\left(\frac{|\mathbf{x}_j-\mathbf{x}_i|}{\delta}\right)W_{j,i}}{|\mathbf{x}_j-\mathbf{x}_i|^{3-\mathrm{ReLU}(3-\alpha)}} \\
\Big[C_1\left(\mathrm{ReLU}(\lambda+\mu)-2\mathrm{ReLU}(\mu)\right)(\mathbf{x}_j-\mathbf{x}_i)\left(\theta_i^s+\theta_j^s\right) \\
+C_2\mathrm{ReLU}(\mu)\frac{(\mathbf{x}_j-\mathbf{x}_i)\otimes(\mathbf{x}_j-\mathbf{x}_i)}{|\mathbf{x}_j-\mathbf{x}_i|^2}\left(\mathbf{u}_j^s-\mathbf{u}_i^s\right)\Big]\Bigg|^2,
\end{aligned}
$$

where

$$
\begin{aligned}
\theta_i^s &:= \sum_{k=0}^{M}\frac{2\mathrm{ReLU}(D_k)}{m_i(\delta)}\sum_{\mathbf{x}_j\in B_\delta(\mathbf{x}_i)}\frac{B_{k,M}\left(\frac{|\mathbf{x}_j-\mathbf{x}_i|}{\delta}\right)W_{j,i}}{|\mathbf{x}_j-\mathbf{x}_i|^{3-\mathrm{ReLU}(3-\alpha)}}(\mathbf{x}_j-\mathbf{x}_i)\cdot\left(\mathbf{u}_j^s-\mathbf{u}_i^s\right), \\
m_i(\delta) &:= \sum_{k=0}^{M}\mathrm{ReLU}(D_k)\sum_{\mathbf{x}_j\in B_\delta(\mathbf{x}_i)}\frac{B_{k,M}\left(\frac{|\mathbf{x}_j-\mathbf{x}_i|}{\delta}\right)W_{j,i}}{|\mathbf{x}_j-\mathbf{x}_i|^{3-\mathrm{ReLU}(3-\alpha)}}|\mathbf{x}_j-\mathbf{x}_i|^2.
\end{aligned}
$$

We solve this unconstrained optimization problem via the Adam optimizer [55]. Here ReLU denotes the rectified linear unit function:

$$
\mathrm{ReLU}(x) := \begin{cases} 0, & \text{for } x \leq 0; \\ x, & \text{for } x > 0. \end{cases}
$$

This operation ensures that $\mathrm{ReLU}(D_k)$ is nonnegative and we replace $\mathbf{D}$ by $\mathrm{ReLU}(\mathbf{D})$ at the end of the prediction step, to guarantee that the learnt $\mathbf{D} \geq 0$. Similar strategies are also

applied to λ, μ and α [2]:

$$\begin{aligned}
&\mu \mapsto \mathrm{ReLU}(\mu),\\
&\lambda \mapsto \mathrm{ReLU}(\lambda+\mu)-\mathrm{ReLU}(\mu),\\
&\alpha \mapsto 3-\mathrm{ReLU}(3-\alpha).
\end{aligned} \tag{4.8}$$

The second step of the algorithm corrects the prediction-step solution by solving for the fully constrained optimization problem. Specifically, by employing $(\lambda^{pre}, \mu^{pre}, \alpha^{pre}, \mathbf{D}^{pre})$ as the initial guess, we apply the augmented Lagrangian method [56, 57] and treat the inequality constraints via slack variables. To do this, introduce the functions

$$\begin{cases}
H_1(\alpha,\mathbf{D},\varpi_1) = \Lambda(\Gamma_{(\alpha,\mathbf{D},\delta,M)}) - \zeta - \varpi_1^2,\\
H_2(\alpha,\mathbf{D},\varpi_2) = \Lambda(\Phi_{(\alpha,\mathbf{D},\delta,M)}\Gamma^{+}_{(\alpha,\mathbf{D},\delta,M)}\Phi^{t}_{(\alpha,\mathbf{D},\delta,M)}) - \zeta - \varpi_2^2,\\
H_3(\alpha,\mathbf{D},\varpi_3) = \Lambda(\Gamma_{(\alpha,\mathbf{D},\delta,M)} - 2\Phi^{t}_{(\alpha,\mathbf{D},\delta,M)}\Phi_{(\alpha,\mathbf{D},\delta,M)}) - \varpi_3^2.
\end{cases}$$

Here, $\boldsymbol{\varpi} = [\varpi_1, \varpi_2, \varpi_3]$ is the vector of slack variables arising from the inequality constraints. Then, minimize the following penalized loss function:

$$\begin{aligned}
L^{corr}(\lambda,\mu,\alpha,\mathbf{D},\boldsymbol{\varpi}) := &\frac{1}{S}\sum_{s=1}^{S}\sum_{\mathbf{x}_i\in\mathcal{X}^s}\Bigg|\mathbf{b}_i^s + \sum_{k=0}^{M}\frac{D_k}{m_i(\delta)}\sum_{\mathbf{x}_j\in B_\delta(\mathbf{x_i})}\frac{B_{k,M}\left(\frac{|\mathbf{x}_j-\mathbf{x}_i|}{\delta}\right)W_{j,i}}{|\mathbf{x}_j-\mathbf{x}_i|^{3-\mathrm{ReLU}(3-\alpha)}}\\
&\Big[C_1\left(\mathrm{ReLU}(\lambda+\mu)-2\mathrm{ReLU}(\mu)\right)(\mathbf{x}_j-\mathbf{x}_i)\left(\theta_i^s+\theta_j^s\right)\\
&+C_2\mathrm{ReLU}(\mu)\frac{(\mathbf{x}_j-\mathbf{x}_i)\otimes(\mathbf{x}_j-\mathbf{x}_i)}{|\mathbf{x}_j-\mathbf{x}_i|^2}\left(\mathbf{u}_j^s-\mathbf{u}_i^s\right)\Big]\Bigg|^2\\
&+\sum_{p=1}^{3}\phi_p H_p(3-\mathrm{ReLU}(3-\alpha),\mathbf{D},\boldsymbol{\varpi}) + \frac{\psi}{2}\sum_{p=1}^{3}H_p^2(3-\mathrm{ReLU}(3-\alpha),\mathbf{D},\boldsymbol{\varpi}),
\end{aligned}$$

where ϕ_p, $p=1,2,3$ are the Lagrange multipliers and ψ is a penalty parameter. At this stage, the Adam optimizer is used. An iterative procedure updates the Lagrange multipliers by 1) solving the unconstrained optimization problem, 2) updating ϕ_p via dual gradient ascent

$$\phi_p \mapsto \phi_p + \psi H_p(\alpha,\mathbf{D},\boldsymbol{\varpi}),$$

[2]Note that although we require $\lambda+\mu>0$, $\mu>0$ and $\alpha<3$ for the well-posedness analysis in Theorem 4.1, for numerical simplicity we employ strategy (4.8) which only guarantees $\lambda+\mu\geq 0$, $\mu\geq 0$ and $\alpha\leq 3$. However, in all numerical tests we observed that the predicted λ, μ, α satisfies the well-posedness conditions, as will be shown in Section 6.

and 3) increasing ψ when the decreasing ratio of $H_p(\alpha, \mathbf{D}, \boldsymbol{\varpi})$ (with respect to the last iteration) does not reach 4 for at least one $p \in \{1, 2, 3\}$. As done in the prediction step, at the end of the correction step, we replace μ by $\mathrm{ReLU}(\mu)$, λ by $\mathrm{ReLU}(\lambda+\mu)-\mathrm{ReLU}(\mu)$, and α by $3-\mathrm{ReLU}(3-\alpha)$. Further details on parameter updates are described in Algorithm 1. The optimal solution is denoted by $(\lambda^*, \mu^*, \alpha^*, \mathbf{D}^*)$.

In applying Algorithm 1, in all our tests, we run the Adam optimizer in PyTorch using a batch size of 70, the inequality constraint tolerance is set to ϵ=1E-5 and the learning rate to 1E-3. The algorithm runs until the loss stagnates, indicating that a stationary point has been reached. Stagnation typically happens between 1000 and 2000 epochs for the first stage of the algorithm and at ∼500 epochs for the second stage at each iteration of the augmented Lagrangian method.

Algorithm 2 Workflow for learning the operator $\mathcal{L}_K$ from MD displacements.

1: Generate MD displacements on fine grids $\{X_\varepsilon^s\}$ using different external forcings and domains configurations and group the samples in three data sets:

$$\mathbb{MD}_{train/val/test} := \{M^s_{\varepsilon,s}, \mathbf{U}^s_{\varepsilon,s}(t), \mathbf{B}^s_{\varepsilon,s}(t)\},\ s = 1, \cdots, S_{train/val/test}.$$

2: Coarse grain in space and smooth in time the data sets $\mathbb{MD}_{train/val/test}$, then evaluate the coarse grained data at coarser grids $\mathcal{X}^s$ to obtain the sets

$$\mathbb{S}_{train/val/test} := \{\mathbf{u}^s(\mathbf{x}_{i,s}), \mathbf{b}^s(\mathbf{x}_{i,s})\},\ s = 1, \cdots, S_{train/val/test}.$$

3: **for** $M \in \mathbb{M}$: **do**
4: **for** $\delta \in \mathbb{D}$: **do**
5: Perform the two-stage optimization strategy for fixed (δ, M) to obtain

$$(\lambda^*_{(\delta,M)}, \mu^*_{(\delta,M)}, \alpha^*_{(\delta,M)}, \mathbf{D}^*_{(\delta,M)}).$$

6: Find δ^*_M that minimizes $\mathrm{Res}(\delta, M; \mathbb{S}_{train})$.
7: Calculate and store $E^{train}_{\mathrm{Res}}(\delta^*_M, M)$, $E^{train}_{\mathbf{u}}(\delta^*_M, M)$, $E^{val}_{\mathrm{Res}}(\delta^*_M, M)$ and $E^{val}_{\mathbf{u}}(\delta^*_M, M)$.
8: Find M^* that minimizes the average of the normalized errors in step 7 and set

$$(\lambda^*, \mu^*, \alpha^*, \mathbf{D}^*, \delta^*, M^*) = (\lambda^*_{(\delta^*_{M^*}, M^*)}, \mu^*_{(\delta^*_{M^*}, M^*)}, \alpha^*_{(\delta^*_{M^*}, M^*)}, \mathbf{D}^*_{(\delta^*_{M^*}, M^*)}, \delta^*_{M^*}, M^*)$$
$$\mathcal{L}^*_K = \mathcal{L}_{K(\delta^*_{M^*}, M^*)}$$

We now discuss how to tune δ and M, which are treated as hyperparameters. Divide the sample set into two sub-sets: the set of training samples $\mathbb{S}_{train} := \{\mathbf{u}^s(\mathbf{x}_{i,s}), \mathbf{b}^s(\mathbf{x}_{i,s})\}$, $s = 1, \cdots, S_{train}$, and the set of validation samples $\mathbb{S}_{val} := \{\widetilde{\mathbf{u}}^s(\mathbf{x}_{i,s}), \widetilde{\mathbf{b}}^s(\mathbf{x}_{i,s})\}$, $s = 1, \cdots, S_{val}$. While all samples correspond to the same material, the problem setting of training and validation samples can be substantially different. For example, they could be generated with

different grids $\mathcal{X}^s$, loading scenarios, boundary conditions, and geometric configuration. For $M \in \mathbb{M}$, we perform Algorithm 1 using the training set $\mathbb{S}_{train}$ with different values of $\delta \in \mathbb{D}$ and denote the corresponding nonlocal operator by $\mathcal{L}_{K(\delta,M)}$. Then, for each M, define the optimal horizon δ_M^* as the one that minimizes the average residual of the LPS equation, denoted by E_{Res}^{train}. Formally,

$$\delta_M^* = \underset{\delta \in \mathbb{D}}{\text{argmin}} \;\; E_{\text{Res}}^{train}(\delta, M) = \underset{\delta \in \mathbb{D}}{\text{argmin}} \;\; \text{Res}(\delta, M; \mathbb{S}_{train}),$$

where

$$\text{Res}(\delta, M; \mathbb{S}_{train}) := \frac{1}{S_{train}} \sum_{\{\mathbf{u}^s(\mathbf{x}_{i,s}), \mathbf{b}^s(\mathbf{x}_{i,s})\} \in \mathbb{S}_{train}} ||\mathcal{L}_{K(\delta,M)}^h \mathbf{u}^s(\mathbf{x}_{i,s}) - \mathbf{b}^s(\mathbf{x}_{i,s})||_{\ell^2(\mathcal{X}^s)}^2.$$

Denote the corresponding nonlocal operator by $\mathcal{L}_{K(\delta_M^*,M)}$. Next, to determine the optimal M that allows for accurate representation of the training samples without overfitting, we test, for each M, the optimal operator $\mathcal{L}_{K(\delta_M^*,M)}$ on the validation set $\mathbb{S}_{val}$ by evaluating the average residual of the LPS equation; the latter, denoted by E_{Res}^{val}, is defined as

$$E_{\text{Res}}^{val}(M) = \text{Res}(\delta_M^*, M; \mathbb{S}_{val}).$$

In principle, small training and validation losses indicate that the model performs well on the training set and generalizes well to other data sets. As an additional metric of accuracy on both the training and validation sets, also consider the displacement mean square error (MSE), denoted by $E_{\mathbf{u}}^{train}$ and $E_{\mathbf{u}}^{val}$, respectively. Formally,

$$E_{\mathbf{u}}^{train}(M) := \frac{1}{S_{train}} \sum_{\{\mathbf{u}^s(\mathbf{x}_{i,s}), \mathbf{b}^s(\mathbf{x}_{i,s})\} \in \mathbb{S}_{train}} ||\mathbf{u}^s(\mathbf{x}_{i,s}) - (\mathcal{L}_{K(\delta_M^*,M)}^h)^{-1} \mathbf{b}^s(\mathbf{x}_{i,s})||_{\ell^2(\mathcal{X}^s)}^2,$$

and $E_{\mathbf{u}}^{val}$ is defined similarly by taking the average over the validation set $\mathbb{S}_{val}$. Based on these metrics, we select M such that the average of the normalized $E_{\text{Res}}^{train}(\delta_M^*, M)$, $E_{\mathbf{u}}^{train}(\delta_M^*, M)$, $E_{\text{Res}}^{val}(\delta_M^*, M)$ and $E_{\mathbf{u}}^{val}(\delta_M^*, M)$ is minimized. Here the normalization is taken with respect to the same quantities evaluated at the baseline, which is the case with $M = 0$ (the constant Bernstein polynomial). In particular, take $M^* = \underset{M}{\text{argmin}}\, \text{AvgE}(M)$ where

$$\text{AvgE}(M) := \frac{1}{4} \left(\frac{E_{\text{Res}}^{train}(\delta_M^*, M)}{E_{\text{Res}}^{train}(\delta_0^*, 0)}, \frac{E_{\mathbf{u}}^{train}(\delta_M^*, M)}{E_{\mathbf{u}}^{train}(\delta_0^*, 0)}, \frac{E_{\text{Res}}^{val}(\delta_M^*, M)}{E_{\text{Res}}^{val}(\delta_0^*, 0)}, \frac{E_{\mathbf{u}}^{val}(\delta_M^*, M)}{E_{\mathbf{u}}^{val}(\delta_0^*, 0)} \right).$$

A summary of the above strategy can be found in Algorithm 2 where we report the overall

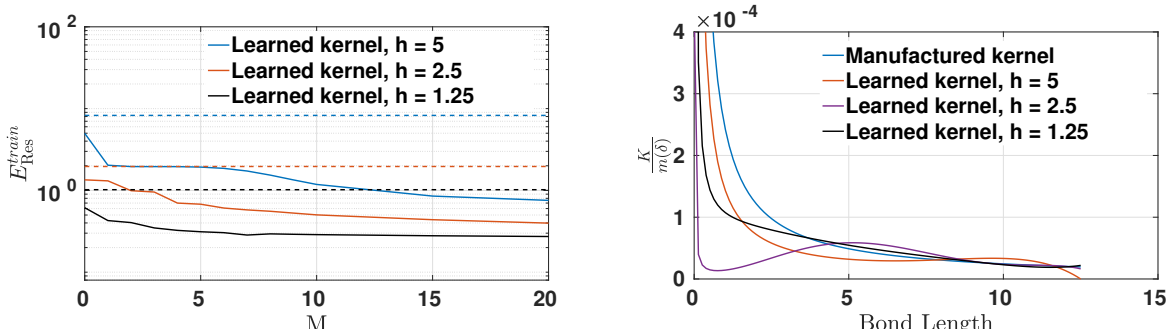

Figure 2: Consistency tests for Algorithm 1 with fixed $\alpha = 1$ and positive influence function. Left: The training loss versus basis order M when using different grid size; the dashed lines indicate the values of the loss functions when the manufactured kernel is used, colors reflect the values of h used for discretization. Right: The comparison of learned influence functions and the manufactured influence function $K_{\text{man}} = \frac{1}{r}$.

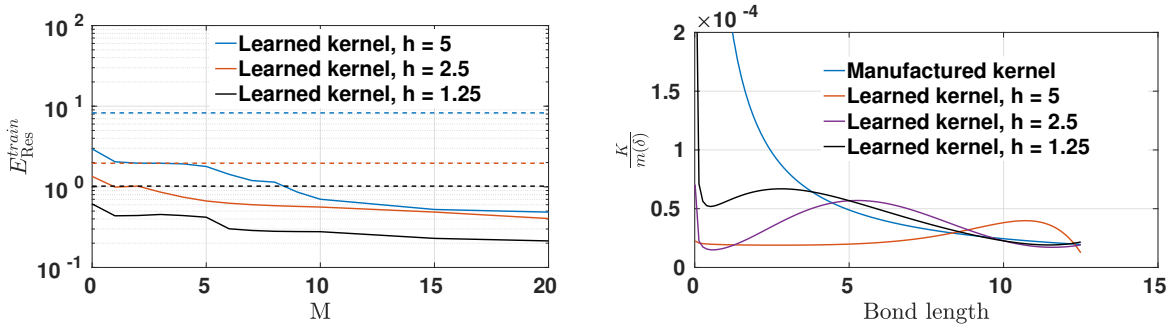

Figure 3: Consistency tests for Algorithm 1 and positive influence function, while learning α. Left: The training loss versus basis order M when using different grid size; the dashed lines indicate the values of the loss functions when the manufactured kernel is used, colors reflect the values of h used for discretization. Right: The comparison of learned influence functions and the manufactured influence function $K_{\text{man}} = \frac{1}{r}$.

workflow of our learning procedure.

5. Consistency Tests for Manufactured Solutions

In this section, we test our operator regression algorithm by considering analytic training pairs $\{\mathbf{u}^s(\mathbf{x}), \mathbf{b}^s(\mathbf{x})\}$ satisfying (3.4) for the manufactured influence function

$$K_{\text{man}}(\mathbf{x}, \mathbf{y}) = \frac{1}{|\mathbf{x} - \mathbf{y}|}.$$

In particular, data sets are considered with different spatial resolutions $\mathcal{X}^s := \{(p_1 h, p_2 h) | \boldsymbol{p} = (p_1, p_2) \in \mathbb{Z}^2\} \cap (\Omega \cap \Omega_I)$ and different Bernstein polynomial orders M with the purpose of validating Algorithm 1 before employing it on MD data sets.

For the computational domain $\Omega = [0, 100]^2$, the training data pairs $\{\mathbf{u}^s(\mathbf{x}), \mathbf{b}^s(\mathbf{x})\}_{s=1}^{70}$

are generated by setting

$$\mathbf{u}(x_1, x_2) = (0.1\cos(2k_1\pi x_1)\cos(2k_2\pi x_2), 0), \text{ or}$$
$$\mathbf{u}(x_1, x_2) = (0, 0.1\cos(2k_1\pi x_1)\cos(2k_2\pi x_2)),$$

with $k_1, k_2 \in \{0, 1, 2, 3, 4, 5\}$. Then, for each displacement field $\mathbf{u}^s(\mathbf{x})$, the corresponding forcing field $\mathbf{b}^s$ is computed from (3.4) with $\lambda = 0.1010$, $\mu = 0.4545$, and $\delta = 0.125$. By evaluating $\mathbf{u}^s$ and $\mathbf{b}^s$ on different grid sets $\mathcal{X}_h$ with $h = 5$, $h = 2.5$ and $h = 1.25$, respectively, three training sets of size 70 are then obtained. These are denoted by $\mathbb{S}_{train}^{h=5}$, $\mathbb{S}_{train}^{h=2.5}$ and $\mathbb{S}_{train}^{h=1.25}$.

To verify the consistency of the prediction step in Algorithm 1 and its behavior with respect to increasing resolution and polynomial order, first consider a positive influence function K with prescribed fractional order $\alpha = 1$. Then choose the Bernstein basis order M in $[0, 20]$. The prediction step involves the solution of a convex optimization problem with a non-empty feasible set. Therefore, every local minimum is a global minimum.

For different training sets, the training losses $E_{\text{Res}}^{val}(M)$ are plotted with respect to increasing polynomial orders M in Figure 2, left. These results suggest that the training loss improves as the basis order M is increased and as the grid size h decreases. Furthermore, the manufactured ground-truth kernels have a higher training losses compared to the learned kernels due to the discretization error, which suggests that the learning algorithm is able to obtain better kernels on each grid set. Figure 2, right, shows a comparison of the learned influence functions for a fixed polynomial order $M = 10$. The learned influence function gets closer to the manufactured influence function K_{man} as $h \to 0$. This illustrates the consistency of the learning algorithm for a given α.

To investigate the effects of the fractional order α, we now consider a positive influence function with unknown fractional order and use, again, the prediction step of Algorithm 1. In this case, the convergence of the optimal kernel to the manufactured one is not guaranteed, since the fractional order is a tunable parameter. The training losses and learnt influence functions are provided in Figure 3. Although the algorithm does not recover exactly the influence function K_{man}, low values of E_{Res}^{train} can be achieved as M increases.

6. Application to graphene using MD

To illustrate the efficacy of our method in obtaining an optimal peridynamic model from coarse-grained MD displacements, we consider graphene sheets as the application. Graphene is a two-dimensional form of carbon with a hexagonal structure. Because of its high stiffness and strength, as well as other unusual physical properties, graphene is being studied for

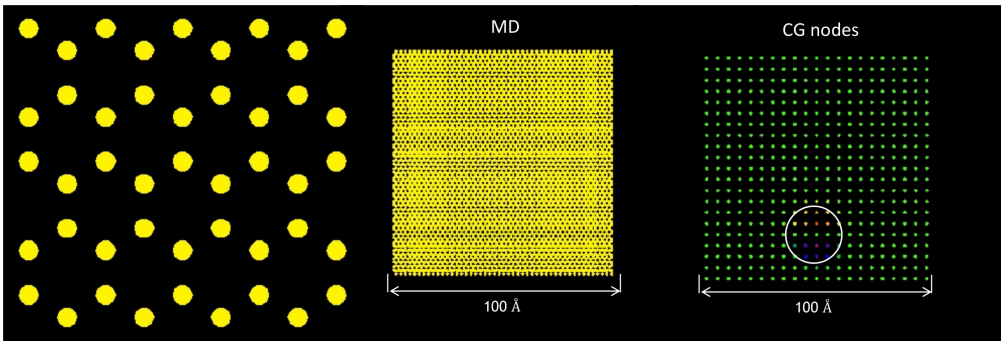

Figure 4: Left: hexagonal graphene atomic structure. Center: full MD grid. Right: coarse grained node positions.

possible use in a number of applications, including as a structural material. For the present study, an MD model was created using the Tersoff interatomic potential [58]. This potential is widely used in the MD community for graphene because it incorporates the relative rotation angle between covalent bonds, which strongly affects the mechanical response. A thermostat is included in the MD model in the present study to control the temperature and periodic boundary conditions are applied. The MD grid is shown in Figure 4, center. The grid has 3588 atoms. The corresponding coarse-grained node positions are shown in Figure 4, right. To simplify the analysis, out-of-plane motions were not considered in this study, although they would occur in a real material.

Unstressed graphene nominally has an interatomic spacing of 1.46Å. In this study, values of the coarse-grained quantities $\mathbf{u}_i$ and $\mathbf{b}_i$ are evaluated on a square lattice of nodes indexed by i with spacing h=5.0Å. We also consider an additional, finer data set generated for validation purposes with spacing 2.5Å.

For any coarse grained node position $\mathbf{x}_i$ and any atomic position $\mathbf{X}_\varepsilon$, define the smoothing function by

$$\omega(\mathbf{x}_i, \mathbf{X}_\varepsilon) = \frac{\tau(\mathbf{x}_i, \mathbf{X}_\varepsilon)}{\sum_j \tau(\mathbf{x}_j, \mathbf{X}_\varepsilon)} \tag{6.1}$$

where the cone-shaped function $\tau(\mathbf{x}, \mathbf{X}) = \max\left\{0,\ R - |\mathbf{X} - \mathbf{x}|\right\}$ induces a coarse-graining radius of R=10.0Å. Note that (6.1) satisfies the normalization requirement (2.1).

In all cases, external loading is applied to the atoms in the MD grid in addition to the random loads applied by the thermostat. The external loading for each atom $\mathbf{B}_\varepsilon$ is constant over time. The magnitude of the loading is chosen so that the bond strains are no larger than 2%, which is less than the strains at which nonlinear effects appear. As described in Section 6.3, the magnitude of the loading is varied relative to the forces on the atoms that sustain the thermal oscillations. This variation helps to test the robustness of the machine learning method in extracting the continuum material properties in the presence of thermal

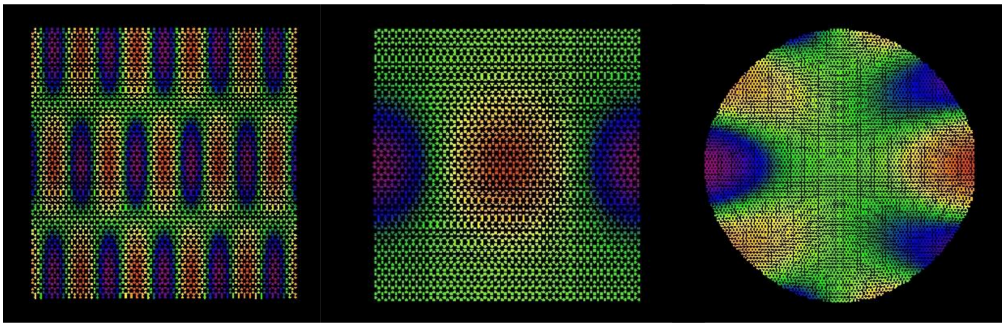

Figure 5: Contours of U_1 displacement in typical MD simulations at zero temperature for the three types of datasets. Left: training. Center: validation. Right: testing.

oscillations that create noise.

6.1. Data Sets and Metrics of Accuracy

Three sets of data are generated from the MD simulations for each of two values of temperature, $0K$ and $300K$. In all MD experiments, the atoms are initialized with positions on a hexagonal lattice in the x_1-x_2 plane with an interatomic spacing of $1.46\mathring{A}$. The mass of each atom is 2.0E-26kg, or 12amu. For purposes of computing stresses, the thickness of the lattice is set to 3.35Å, which is the approximate distance between layers in multilayer graphene. The MD time step size is 5.0E-16s, or 5.0fs.

Two types of samples are generated for each data set: the standard samples with spacing h=5.0Å which will be denoted by $\mathbb{S}^{\text{0K/300K}}_{train/val/test}$, and the samples with finer grids h=2.5Å, denoted by $\mathbb{S}^{\text{0K/300K,fine}}_{train/val/test}$. Unless stated otherwise, the h=5.0Å data sets are used in the learning tasks. The finer grid data sets are employed to assess the generalization properties of the proposed learning approach to different grids (further details and discussions are provided in Section 6.4). Images showing contours of U_1, the component of atomic displacement in the x_1 direction, for the training, validation, and testing samples are provided in Figure 5.

1) Training data set (70 samples): The MD domain is a 100Å×100Å square, and, for $k_1, k_2 \in \{0, \frac{\pi}{50}, \frac{2\pi}{50}, \ldots, \frac{5\pi}{50}\}$, the prescribed external loadings are given by

$$\mathbf{b}(x_1, x_2) = (C^1_{k_1,k_2} \cos(k_1 x_1) \cos(k_2 x_2), 0), \text{ or } \mathbf{b}(x_1, x_2) = (0, C^2_{k_1,k_2} \cos(k_1 x_1) \cos(k_2 x_2)). \tag{6.2}$$

As mentioned above, the constant $C^1_{k_1,k_2}$ and $C^2_{k_1,k_2}$ are adjusted so that the bond strains are no larger than 2%, so the deformation remains in the linear range of material response.

2) Validation data set (10 samples). For the same MD grid and coarse-grained nodes as in

k	C_k^1	C_k^2	p_k	R_k
1	0.001	0	0	25
2	0	0.001	0	25
3	0.001	0	0	15
4	0	0.001	0	15
5	0.001	0	0	10
6	0.001	0	1	25
7	0	0.001	1	25
8	0.001	0	1	15
9	0	0.001	1	15
10	0.001	0	1	10

Table 1: Parameters used in the MD loading in the 10 validation tests.

the training data set, the applied loads in the validation data set are as follows:

$$\mathbf{b}(x_1, x_2) = (C_k^1, C_k^2) \sum_{j=-1}^{1} (-1)^j \cos\left(\frac{\pi}{2} \min\left\{1, \frac{r_{j,k}}{R_k}\right\}\right)$$

where

$$r_{j,k} = \sqrt{(x_1 - (1-p_k)Lj)^2 + (x_2 - p_k Lj)^2}$$

where L=50 and the values of the parameters C_k^1, C_k^2, p_k and R_k are given in Table 1. In each case, loads are applied to the atoms within three disks of radius R_k with centers at the center of the grid and at the left and right boundaries (if $p_k = 0$) or the upper and lower boundaries (if $p_k = 1$). The direction of the load vectors is either in the x_1 or x_2 directions. The loads in all cases are self-equilibrated and periodic.

3) Test data set (4 samples). To demonstrate that the learned material model applies to geometries different from the original square geometry, four additional test cases are considered. Here, the MD region is a disk of radius 100Å. Within this disk loading is applied as listed in Table 2 to the exterior of a circle with radius 50Å, with the interior unloaded. The equilibrium displacements, with smoothing as described previously, are computed at the coarse-grained nodes, which are spaced 5Å or 2.5Å, apart on a square lattice. All of these cases have loadings that are discontinuous functions of the radius and therefore are more challenging from a modeling perspective than the validation cases described above. In cases 2 and 4 the loading is also a discontinuous function of the angle because of the sign function (sgn).

As metrics of accuracy on the training, validation and test sets, in this section we calculate the averaged mean square loss (MSL) and displacement mean square error (MSE) for each

Case	b_1	b_2
1	$C\cos 4\theta\cos\theta$	$C\cos 4\theta\sin\theta$
2	$C\,\mathrm{sgn}(\cos 4\theta)\cos\theta$	$C\,\mathrm{sgn}(\cos 4\theta)\sin\theta$
3	0	$C\,\mathrm{sgn}(\sin\theta)\sin\theta$
4	$C\sin 3\theta\sin\theta$	$C\sin 3\theta\cos\theta$

Table 2: Loading applied to the exterior of a disk of radius 50Å in the four tests. In all cases, C=0.0005. b_1 and b_2 are components of $\mathbf{b}$ along the x_1 and x_2 directions, respectively, and θ is the polar angle in the plane.

learnt kernel on these three sets, which will be referred to as $E_{Res/\mathbf{u}}^{train/val/test}$, respectively. To provide a fair comparison between different sets, all these accuracy metrics except E_{Res}^{test} are normalized with respect to either the force loading or the displacement fields. Specifically, $E_{Res}^{train/val}$ is normalized by $||\mathbf{b}^s||^2_{l_2(\mathcal{X}^s)}$ and $E_{\mathbf{u}}^{train/val/test}$ is normalized by $||\mathbf{u}^s||^2_{l_2(\mathcal{X}^s)}$. For E_{Res}^{test} we report the absolute value instead since on the test samples the loading is only applied outside the computational domain and we have $\mathbf{b}^s = 0$ on Ω. Therefore, we can not normalize E_{Res}^{test} with respect to the force loading field.

Changing the shape of the smoothing functions, while holding the radius R constant, has only a small effect on the results. For example, replacing the cone-shaped function τ used in (6.1) with a paraboloid changes the coarse-grained displacements by about 0.3% in a typical MD simulation used to generate the training data. Changing the radius R affects the horizon δ and therefore affects the dispersion properties of waves with wavelengths comparable to or less than the horizon given by (2.9) [38].

6.2. Learning Results

We first tune the hyperparameters (δ, M) following the procedure described in Algorithm 2. The optimal δ_M^* for each fixed polynomial order M and the corresponding E_{Res}^{train}, $E_{\mathbf{u}}^{train}$, E_{Res}^{val}, $E_{\mathbf{u}}^{val}$, and AvgE are reported in Table 3 and Figure 6. Based on these results, we set $(\delta_M^*, M) = (20\mathring{A}, 10)$ for the $0K$ tasks, and $(\delta_M^*, M) = (20\mathring{A}, 15)$ for the $300K$ tasks. For both data sets the optimal horizon size is $\delta = 20$Å , which is twice the support radius R. This value is very close to the horizon that would be predicted by (2.9), because the MD cutoff distance $d = 1.46$Å$\ll R$, and there are very few interatomic interactions that connect the smoothing functions with centers separated by $2R$. Compared to the data set at $0K$, the $300K$ data set requires a higher polynomial order M and therefore a more complex influence function. This is possibly due to the occurrence of thermal oscillations in the MD simulations at $300K$. We use these optimal pairs (δ_M^*, M) as the default choices in all tests below (unless stated otherwise).

The learning results are provided in the left plot of Figure 7 and in Table 4. For both the 0K and the 300K data sets, the Young's modulus is estimated to be around 1TPa, which is

data set	M	δ^*_M	E^{train}_{Res}	$E^{train}_{\mathbf{u}}$	E^{val}_{Res}	$E^{val}_{\mathbf{u}}$	AvgE
$0K$	0	12.5Å	13.91%	17.54%	16.31%	14.49%	1
	5	12.5Å	10.42%	12.19%	13.02%	7.69%	0.6933
	10	**20Å**	9.81%	11.72%	13.28%	7.16%	**0.6704**
	15	22.5Å	9.80%	11.61%	13.50%	7.22%	0.6731
	20	25Å	9.75%	11.89%	13.53%	7.00%	0.6729
$300K$	0	12.5Å	13.46%	31.33%	20.15%	14.86%	1
	5	12.5Å	10.50%	13.80%	17.83%	9.66%	0.6784
	10	20Å	9.79%	13.32%	18.11%	9.08%	0.6549
	15	**20Å**	9.82%	13.16%	18.08%	8.88%	**0.6505**
	20	25Å	9.81%	13.36%	18.34%	9.23%	0.6609

Table 3: Losses from the optimal $\delta^*(M)$ for each values of M, where the optimal cases and the corresponding average of the normalized errors are highlighted with bold.

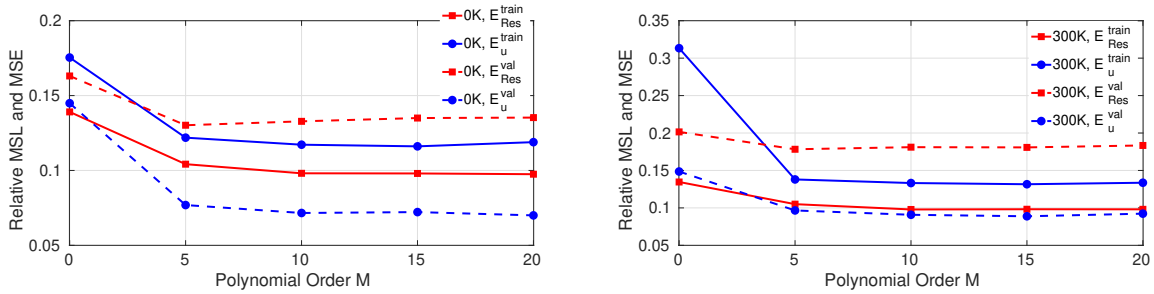

Figure 6: Optimal relative mean square loss (MSL) and relative mean square error (MSE) for each polynomial order M. Left: results at 0K. Right: results at 300K.

consistent with experimental evidence [59, 60] and computations via first principles [61] or MD [62, 63]. The predicted Poisson ratio is negative, which results from graphene's exceptionally high resistance to relative angle changes (shear) between the covalent bonds. The predicted value $\nu = -0.4$ is consistent with other MD and molecular statistics simulations [64, 65]. As expected, the optimal influence functions shown in the left plot of Figure 7 are partially negative for both data sets. This fact highlights the importance of allowing for sign-changing influence functions.

6.3. Sensitivity to Thermal Oscillations

Recall from (2.4) that each atom in the MD simulation experiences *internal* forces $\mathbf{F}_{\gamma\varepsilon}$ due to interaction with other atoms and *external* forces $\mathbf{B}_\varepsilon$ due to prescribed loads. At finite temperature, the internal forces consist largely of the random forces that produce thermal oscillations. A convenient way to characterize the relative magnitude of the random and

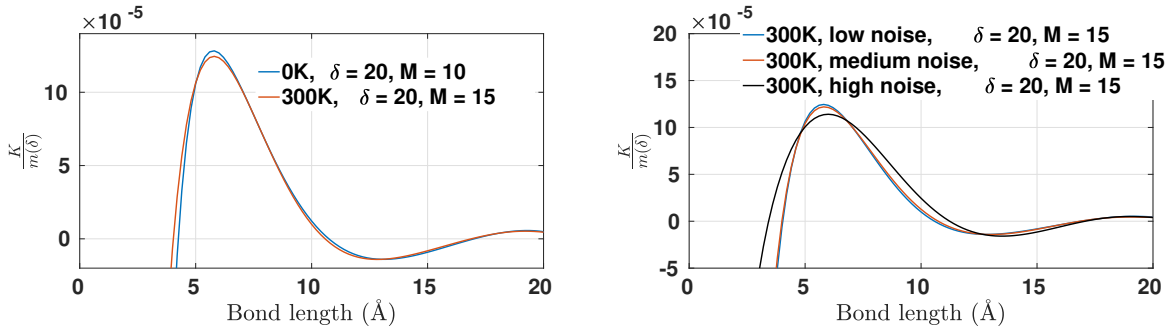

Figure 7: Left: Optimal influence functions K at 0K and 300K. Right: Optimal influence functions K at 300K with different level of noise.

data set	M	α	λ (TPa)	μ (TPa)	E (TPa)	ν
$0K$	10	2.8335	-0.4796	0.7978	0.91	-0.4297
	E_{Res}^{train}	$E_{\mathbf{u}}^{train}$	E_{Res}^{val}	$E_{\mathbf{u}}^{val}$	E_{Res}^{test}	$E_{\mathbf{u}}^{test}$
	9.81%	11.72%	13.28%	7.16%	2.03E-1	6.75%
data set	M	α	λ (TPa)	μ (TPa)	E (TPa)	ν
$300K$	15	2.5946	-0.4583	0.7753	0.90	-0.4196
	E_{Res}^{train}	$E_{\mathbf{u}}^{train}$	E_{Res}^{val}	$E_{\mathbf{u}}^{val}$	E_{Res}^{test}	$E_{\mathbf{u}}^{test}$
	9.82%	13.16%	18.08%	8.88%	2.08E-1	9.21%

Table 4: Optimal material parameters and MSL/MSE on training, validation and testing data sets at $0K$ and $300K$. E_{Res}^{test} shows the absolute ℓ^2 error as the inner circle in the test problem is unloaded.

prescribed forces is the *signal-to-noise ratio* (SNR) defined by

$$SNR = \sqrt{\frac{\sum_{\varepsilon} |\mathbf{B}_{\varepsilon}|^2}{\sum_{\varepsilon} \left| \mathbf{B}_{\varepsilon} + \sum_{\gamma} \mathbf{F}_{\gamma\varepsilon} \right|^2}}.$$

At finite temperature with zero loading, $SNR = 0$, while at zero temperature but finite loading in equilibrium, $SNR = \infty$. Most, but not all, of the random forces and oscillations are smoothed out by the coarse graining prior to application of the algorithm to learn the kernel and material parameters. The effect of the random noise is also reduced by the smoothing of displacements over time using (2.10).

Next, we investigate the robustness of the learning approach by applying Algorithm 2 to training data sets with different signal-to-noise ratios. With temperature $300K$, three training data sets are created by changing the relative magnitude of the loading described in (6.2) to 1, 1/4 and 1/10, respectively. Due to the existence of thermal noise, the smaller the loading magnitude is, the smaller the signal-to-noise ratio will be. Therefore, we denote these three training data sets as the "Low noise" data set, "Med noise" data set and "High noise" data set, respectively.

300K Low Noise	StN ratio	α	λ (TPa)	μ (TPa)	E (TPa)	ν
	0.1556	2.5946	-0.4583	0.7753	0.90	-0.4196
	E_{Res}^{train}	$E_{\mathbf{u}}^{train}$	E_{Res}^{val}	$E_{\mathbf{u}}^{val}$	E_{Res}^{test}	$E_{\mathbf{u}}^{test}$
	9.82%	13.16%	18.08%	8.88%	2.08E-1	9.21%
300K Med Noise	StN ratio	α	λ (TPa)	μ (TPa)	E (TPa)	ν
	0.0543	2.5197	-0.4782	0.7798	0.87	-0.4422
	E_{Res}^{train}	$E_{\mathbf{u}}^{train}$	E_{Res}^{val}	$E_{\mathbf{u}}^{val}$	E_{Res}^{test}	$E_{\mathbf{u}}^{test}$
	14.52%	28.27%	18.34%	9.82%	2.11E-1	9.28%
300K High Noise	StN ratio	α	λ (TPa)	μ (TPa)	E (TPa)	ν
	0.0224	1.9365	-0.3266	0.6890	0.95	-0.3106
	E_{Res}^{train}	$E_{\mathbf{u}}^{train}$	E_{Res}^{val}	$E_{\mathbf{u}}^{val}$	E_{Res}^{test}	$E_{\mathbf{u}}^{test}$
	27.64%	48.86%	23.73%	17.54%	1.84E-1	7.95%

Table 5: Test of algorithm robustness on 300K data set with different noise levels. E_{Res}^{test} shows the absolute ℓ^2 error as the inner circle in the test problem is unloaded.

To study the sensitivity of the learning algorithm with respect to decreasing SNRs, we plot and compare the optimal influence function K in the right plot of Figure 7. It can be seen that while the learnt influence function from the "Med noise" data set is almost the same as the influence function from the "Low noise" data set, the learnt influence function from "High noise" data set slightly differs. To provide a further quantitative comparison, Table 5 provides the estimated material parameters as well as the loss and errors on the validation and test data sets. The learnt influence functions from all training sets achieves a similar level of accuracy on the test data set. On the validation set, the learnt model from "Med noise" set achieves a similar accuracy as the model from "Low noise" set, while the displacement mean square error increases for the learnt model from "High noise" set. Not surprisingly, since the SNR decreases by 7 times in the "High noise" set, $E_{\mathbf{u}}^{val}$ doubles.

6.4. Generalization

This section discusses the generalization of the optimal influence function to different loadings, domains and discretizations.

Different Loadings: The loadings in the validation and test data sets are substantially different from those in training data sets. Therefore, these can be used to assess the performance of the learnt models as reported in Table 4. The validation loss is consistently lower than 20% and the solution error smaller than 10%, illustrating that the optimal models can be generalized to problems with different loadings.

Different Domains: In the test data set the domain is a disk (as opposed to the square domain used for training). The results of applying the optimal learnt model to this test problem are provided in Table 4. Here the loss E_{Res}^{test} is presented as the absolute error

$\mathbb{S}_{train}$	$\mathbb{S}_{val}$ and $\mathbb{S}_{test}$	E_{Res}^{val}	$E_{\mathbf{u}}^{val}$	E_{Res}^{test}	$E_{\mathbf{u}}^{test}$
$h = 5\mathring{A}$ and $h = 2.5\mathring{A}$	$h = 2.5\mathring{A}$	16.19%	8.01%	2.95E-0	8.44%
$h = 5\mathring{A}$ and $h = 2.5\mathring{A}$	$h = 5\mathring{A}$	13.24%	9.29%	1.97E-1	7.80%

Table 6: Learning results from hybrid resolution datasets with fixed $\delta = 12.5\mathring{A}$.

because the interior circle is unloaded. For both 0K and 300K, the solution error is consistently below 10%, showing that the optimal models can perform well with different domain configurations[3].

Hybrid Discretizations: Since the proposed approach learns a continuous nonlocal operator rather than a discrete surrogate for the solution, the learning approach can naturally handle data sets with different resolutions or even different discretization methods. To provide initial studies on the generalization properties with respect to different resolutions, we consider a hybrid data set with samples of different resolutions and investigate the performance of the learning algorithm. In particular, the training data set is defined as the union of $\mathbb{S}_{train}^{0K}$ and $\mathbb{S}_{train}^{0K,fine}$. Table 6 reports the accuracy of the learnt model on both standard and fine validation and test data sets. From the results in the table, the solution error is consistently below 10%, which highlights the capability of the proposed algorithm to handle data sets with different resolutions.

7. Conclusion

We introduced a new optimization-based, data-driven approach to extract an optimal linear peridynamic solid model from MD data. The peridynamic constitutive law is learned by optimizing the influence function and material parameters. The influence function is allowed to be sign-changing, thus improving the descriptive power of the optimal model. The nontrivial problem of learning well-posed models in the presence of sign-changing influence functions was addressed by deriving new sufficient conditions for the discretized peridynamic model, embedded in the learning procedure as inequality constraints. To assess the performance of the proposed learning algorithm, we tested the robustness with respect to noise and the ability of the optimal model to generalize to different domain configurations, external loadings, and discretizations.

A fundamental aspect of the proposed procedure is the fact that we learn a continuous operator rather than a discrete operator or a surrogate for the solution; this fact guarantees generalization of the optimal model to settings that are different from the ones used

[3]Among all training samples, higher relative solution errors are observed from samples driven by high frequency external loadings. Comparing with the training set, the test samples are all with relatively low spatial frequency and therefore have smaller solution errors.

during training. Furthermore, the continuous setting opens new research direction, such as considering different discretization methods when validating the optimal model.

Although the present work focuses on single layered graphene, the results suggest that this method may impact a broader range of materials and applications. As a follow-up work we plan to extend our algorithm to more complex materials behaviors, such as large deformation and/or damage.

Acknowledgements

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper, SAND2021-9450, describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

The work of S. Silling, and M. D'Elia is supported by by the Sandia National Laboratories Laboratory Directed Research and Development (LDRD) program. M. D'Elia is also partially supported by the U.S. Department of Energy, Office of Advanced Scientific Computing Research under the Collaboratory on Mathematics and Physics-Informed Learning Machines for Multiscale and Multiphysics Problems (PhILMs) project. H. You and Y. Yu are supported by the National Science Foundation under award DMS 1753031. Portions of this research were conducted on Lehigh University's Research Computing infrastructure partially supported by NSF Award 2019035. This work also used the Extreme Science and Engineering Discovery Environment (XSEDE) [66], which is supported by National Science Foundation grant number ACI-1548562.

References

[1] T. I. Zohdi, Homogenization methods and multiscale modeling, Encyclopedia of Computational Mechanics Second Edition (2017) 1–24.

[2] A. Bensoussan, J.-L. Lions, G. Papanicolaou, Asymptotic analysis for periodic structures, Vol. 374, American Mathematical Soc., 2011.

[3] E. Weinan, B. Engquist, Multiscale modeling and computation, Notices of the AMS 50 (9) (2003) 1062–1070.

[4] Y. Efendiev, J. Galvis, T. Y. Hou, Generalized multiscale finite element methods (gmsfem), Journal of computational physics 251 (2013) 116–135.

[5] C. Junghans, M. Praprotnik, K. Kremer, Transport properties controlled by a thermostat: An extended dissipative particle dynamics thermostat, Soft Matter 4 (1) (2008) 156–161.

[6] R. Kubo, The fluctuation-dissipation theorem, Reports on progress in physics 29 (1) (1966) 255.

[7] F. Santosa, W. W. Symes, A dispersive effective medium for wave propagation in periodic composites, SIAM Journal on Applied Mathematics 51 (4) (1991) 984–1005.

[8] M. Dobson, M. Luskin, C. Ortner, Sharp stability estimates for the force-based quasicontinuum approximation of homogeneous tensile deformation, Multiscale Modeling & Simulation 8 (3) (2010) 782–802.

[9] M. Ortiz, A method of homogenization of elastic media, International journal of engineering science 25 (7) (1987) 923–934.

[10] N. Moës, J. T. Oden, K. Vemaganti, J.-F. Remacle, Simplified methods and a posteriori error estimation for the homogenization of representative volume elements (rve), Computer methods in applied mechanics and engineering 176 (1-4) (1999) 265–278.

[11] T. J. Hughes, G. N. Wells, A. A. Wray, Energy transfers and spectral eddy viscosity in large-eddy simulations of homogeneous isotropic turbulence: Comparison of dynamic smagorinsky and multiscale models over a range of discretizations, Physics of Fluids 16 (11) (2004) 4044–4052.

[12] G. W. Milton, The Theory of Composites, Cambridge University Press, 2002.

[13] A. C. Eringen, D. G. B. Edelen, On nonlocal elasticity, International Journal of Engineering Science 10 (3) (1972) 233–248.

[14] F. Bobaru, J. T. Foster, P. H. Geubelle, S. A. Silling, Handbook of peridynamic modeling, CRC press, 2016.

[15] M. Beran, J. McCoy, Mean field variations in a statistical sample of heterogeneous linearly elastic solids, International Journal of Solids and Structures 6 (8) (1970) 1035–1054.

[16] K. Cherednichenko, V. P. Smyshlyaev, V. Zhikov, Non-local homogenised limits for composite media with highly anisotropic periodic fibres, Proceedings of the Royal Society of Edinburgh Section A Mathematics 136 (1) (2006) 87–114.

[17] F. C. Karal Jr, J. B. Keller, Elastic, electromagnetic, and other waves in a random medium, Journal of Mathematical Physics 5 (4) (1964) 537–547.

[18] Y. Rahali, I. Giorgio, J. Ganghoffer, F. dell'Isola, Homogenization à la piola produces second gradient continuum models for linear pantographic lattices, International Journal of Engineering Science 97 (2015) 148–172.

[19] V. P. Smyshlyaev, K. D. Cherednichenko, On rigorous derivation of strain gradient effects in the overall behaviour of periodic heterogeneous media, Journal of the Mechanics and Physics of Solids 48 (6-7) (2000) 1325–1357.

[20] J. R. Willis, The nonlocal influence of density variations in a composite, International Journal of Solids and Structures 21 (7) (1985) 805–817.

[21] Q. Du, B. Engquist, X. Tian, Multiscale modeling, homogenization and nonlocal effects: Mathematical and computational issues, Contemporary Mathematics 754.

[22] Q. Du, K. Zhou, Mathematical analysis for the peridynamic nonlocal continuum theory, ESAIM: Mathematical Modelling and Numerical Analysis 45 (02) (2011) 217–234.

[23] E. Madenci, A. Barut, M. Dorduncu, Peridynamic differential operator for numerical analysis, Springer, 2019.

[24] E. Emmrich, O. Weckner, et al., On the well-posedness of the linear peridynamic model and its convergence towards the navier equation of linear elasticity, Communications in Mathematical Sciences 5 (4) (2007) 851–864.

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

[26] P. Clark Di Leoni, T. A. Zaki, G. Karniadakis, C. Meneveau, Two-point stress–strain-rate correlation structure and non-local eddy viscosity in turbulent flows, Journal of Fluid Mechanics 914 (2021) A6.

[27] G. Pang, M. D'Elia, M. Parks, G. E. Karniadakis, nPINNs: nonlocal Physics-Informed Neural Networks for a parametrized nonlocal universal Laplacian operator. Algorithms and Applications, to appear in Journal of Computational Physics (2020).

[28] P. Diehl, S. Prudhomme, M. Lévesque, A review of benchmark experiments for the validation of peridynamics models, Journal of Peridynamics and Nonlocal Modeling 1 (1) (2019) 14–35.

[29] D. Benson, S. Wheatcraft, M. Meerschaert, Application of a fractional advection-dispersion equation, Water Resources Research 36 (6) (2000) 1403–1412.

[30] X. Xu, J. T. Foster, Deriving peridynamic influence functions for one-dimensional elastic materials with periodic microstructure, arXiv:2003.05520 (2020).

[31] X. Xu, M. D'Elia, J. Foster, Bond-based peridynamic kernel learning with energy constraint, arXiv:2101.01095 (2021).

[32] H. You, Y. Yu, N. Trask, M. Gulian, M. D'Elia, Data-driven learning of robust nonlocal physics from high-fidelity synthetic data, Computer Methods in Applied Mechnics and Engineering 374 (2021) 113553.

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

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

[35] S. Silling, Peridynamic modeling, numerical techniques, and applications, Elsevier, 2021.

[36] R. B. Lehoucq, M. P. Sears, Statistical mechanical foundation of the peridynamic nonlocal continuum theory: Energy and momentum conservation laws, Physical Review E 84 (3) (2011) 031112.

[37] A. F. Voter, Hyperdynamics: Accelerated molecular dynamics of infrequent events, Physical Review Letters 78 (20) (1997) 3908.

[38] S. A. Silling, Reformulation of elasticity theory for discontinuities and long-range forces, Journal of the Mechanics and Physics of Solids 48 (1) (2000) 175–209.

[39] S. A. Silling, R. B. Lehoucq, Peridynamic theory of solid mechanics, Advances in applied mechanics 44 (2010) 73–168.

[40] N. Trask, H. You, Y. Yu, M. L. Parks, An asymptotically compatible meshfree quadrature rule for nonlocal problems with applications to peridynamics, Computer Methods in Applied Mechanics and Engineering 343 (2019) 151–165.

[41] Y. Yu, H. You, N. Trask, An asymptotically compatible treatment of traction loading in linearly elastic peridynamic fracture, Computer Methods in Applied Mechanics and Engineering 377 (2021) 113691.

[42] Y. Fan, X. Tian, X. Yang, X. Li, C. Webster, Y. Yu, An asymptotically compatible probabilistic collocation method for randomly heterogeneous nonlocal problems, In preprint.

[43] H. You, X. Y. Lu, N. Trask, Y. Yu, An asymptotically compatible approach for Neumann-type boundary condition on nonlocal problems, arXiv:1908.03853 (2019).

[44] H. You, Y. Yu, D. Kamensky, An asymptotically compatible formulation for local-to-nonlocal coupling problems without overlapping regions, Computer Methods in Applied Mechanics and Engineering 366 (2020) 113038.

[45] M. Foss, P. Radu, Y. Yu, Convergence analysis and numerical studies for linearly elastic peridynamics with dirichlet-type boundary conditions, arXiv preprint arXiv:2106.13878.

[46] P. Seleson, M. Parks, On the role of the influence function in the peridynamic theory, International Journal for Multiscale Computational Engineering 9 (6).

[47] M. D'Elia, X. Tian, Y. Yu, A physically-consistent, flexible and efficient strategy to convert local boundary conditions into nonlocal volume constraints, SIAM Journal of Scientific Computing 42 (4) (2020) A1935–A1949.

[48] M. D'Elia, Y. Yu, On the prescription of boundary conditions for nonlocal poisson's and peridynamics models, arXiv preprint arXiv:2107.04450.

[49] Y. Yu, F. F. Bargos, H. You, M. L. Parks, M. L. Bittencourt, G. E. Karniadakis, A partitioned coupling framework for peridynamics and classical theory: analysis and simulations, Computer Methods in Applied Mechanics and Engineering 340 (2018) 905–931.

[50] T. Mengesha, Q. Du, Nonlocal constrained value problems for a linear peridynamic navier equation, Journal of Elasticity 116 (1) (2014) 27–51.

[51] O. Weckner, S. A. Silling, Determination of the constitutive model in peridynamics from experimental dispersion data, International Journal of Multiscale Computational Engineering, under review.

[52] T. Mengesha, Q. Du, Analysis of a scalar nonlocal peridynamic model with a sign changing kernel, Discrete & Continuous Dynamical Systems-B 18 (5) (2013) 1415–1437.

[53] K. J. Bathe, The $inf-sup$ condition and its evaluation for mixed finite element methods, Computers & Structures 79 (2) (2001) 243 – 252.

[54] F. Brezzi, M. Fortin, Mixed and hybrid finite element methods, Vol. 15, Springer Science & Business Media, 2012.

[55] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980 (2014).

[56] Y. Yu, J. Chen, T. Gao, M. Yu, DAG-GNN: DAG structure learning with graph neural networks, arXiv:1904.10098 (2019).

[57] J. Nocedal, S. Wright, Numerical Optimization, Springer Science & Business Media, 2006.

[58] J. Tersoff, Empirical interatomic potential for carbon, with applications to amorphous carbon, Physical Review Letters 61 (25) (1988) 2879.

[59] C. Lee, X. Wei, J. W. Kysar, J. Hone, Measurement of the elastic properties and intrinsic strength of monolayer graphene, science 321 (5887) (2008) 385–388.

[60] I. Frank, D. M. Tanenbaum, A. M. van der Zande, P. L. McEuen, Mechanical properties of suspended graphene sheets, Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures Processing, Measurement, and Phenomena 25 (6) (2007) 2558–2561.

[61] F. Liu, P. Ming, J. Li, Ab initio calculation of ideal strength and phonon instability of graphene under tension, Physical Review B 76 (6) (2007) 064120.

[62] T.-w. HAN, P.-f. He, J. Wang, A.-h. Wu, Molecular dynamics simulation of single graphene sheet under tension, New Carbon Materials 25 (4) (2010) 261–266.

[63] T. Han, P. He, Y. Luo, X. Zhang, Research progress in the mechanical properties of graphene, Advances in Mechanics 41 (3) (2011) 279.

[64] H. Qin, Y. Sun, J. Z. Liu, M. Li, Y. Liu, Negative poisson's ratio in rippled graphene, Nanoscale 9 (12) (2017) 4135–4142.

[65] J.-W. Jiang, T. Chang, X. Guo, H. S. Park, Intrinsic negative poisson's ratio for single-layer graphene, Nano letters 16 (8) (2016) 5286–5290.

[66] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, et al., Xsede: accelerating scientific discovery, Computing in science & engineering 16 (5) (2014) 62–74.