

Approximating Fracture Paths in Random Heterogeneous Materials: A Probabilistic Learning Perspective

Ariana Quek¹; Jin Yi Yong²; and Johann Guilleminot, M.ASCE³

Abstract: Approximation frameworks for phase-field models of brittle fracture are presented and compared in this work. Such methods aim to address the computational cost associated with conducting full-scale simulations of brittle fracture in heterogeneous materials where material parameters, such as fracture toughness, can vary spatially. They proceed by combining a dimension reduction with learning between function spaces. Two classes of approximations are considered. In the first class, deep learning models are used to perform regression in ad hoc latent spaces. PCA-Net and Fourier neural operators are specifically presented for the sake of comparison. In the second class of techniques, statistical sampling is used to approximate the forward map in latent space, using conditioning. To ensure proper measure concentration, a reduced-order Hamiltonian Monte Carlo technique (namely, probabilistic learning on manifold) is employed. The accuracy of these methods is then investigated on a proxy application where the fracture toughness is modeled as a non-Gaussian random field. It is shown that the probabilistic framework achieves comparable performance in the L^2 sense while enabling the end-user to bypass the art of defining and training deep learning models. DOI: 10.1061/JENMDT.EMENG-7617. © 2024 American Society of Civil Engineers.

Introduction

Predicting failure mechanisms caused by crack initiation and propagation is critical in most engineering applications. In realistic structures, damage development is a complicated phenomenon, the modeling of which requires high-fidelity simulations. The phase-field approach has grown prevalent in the field due to its well-proven modeling capabilities in various settings Kuhn and Müller (2010), Borden et al. (2012), Nguyen et al. (2015), Borden et al. (2014), Bourdin (2007), Eastgate et al. (2001), Clément (2000), and Hofacker and Miehe (2013). This approach employs a regularized formulation of a sharp crack description via a modified variational principle Francfort and Marigo (1998) and Bourdin et al. (2000). While effective in delivering accurate predictions in controlled environments, the phase-field method remains computationally expensive when the complexity of the model increases. This limitation prevents extensive state space exploration, which is critical in the case of heterogeneous materials where a large number of instantiations is required to perform probabilistic analyses.

In this context, the capacity of surrogate models to approximate flow/solution maps by learning core structural characteristics, while boosting online computational efficiency, has attracted a great deal of interest in the recent years. Following the notation in Bhattacharya et al. (2021), consider a nonlinear input–output map $\Psi: \mathcal{X} \rightarrow \mathcal{Y}$ between Hilbert spaces $\mathcal{X}$ and $\mathcal{Y}$. A first set of methods involves image-to-image regression where discretizations of the input and output fields are first considered to formulate the learning task between Euclidean spaces. The finite-dimensional input

and output data thus obtained are then potentially reduced and subsequently fed into a neural network that performs regression on the input-to-output map; see, e.g., Adler and Öktem (2017), Bhatnagar et al. (2019), Geist et al. (2020), Holland et al. (2019), Zhu and Zabaras (2018), and Sepasdar et al. (2022) for methodological developments, as well as Mohammadzadeh and Lejeune (2021) for an application to fracture problems. As discussed elsewhere Bhattacharya et al. (2021), these approaches are sensitive to mesh refinement: by construction, the discretization (or pixelization) affects both the accuracy and convergence rate, and a new neural network architecture must be determined as the resolution varies.

In a slightly different setup, $x \in \mathcal{X}$ can be seen as a parameterization of the solution $y \in \mathcal{Y}$. In this case, a deep neural network can then be used to approximate the mapping between, e.g., space or space–time variables and the output—for a fixed value of x in $\mathcal{X}$; see Dockhorn (2019), Weinan and Yu (2017), Hsieh et al. (2019), Lagaris et al. (1998), Raissi et al. (2019), and Shin (2020) among others. Such methods share similarities with standard collocation methods for the numerical solution to PDEs. The major drawback of this setting is that the approximated map is intrinsically indexed by x . This leads to a computational burden when parametric dependence must be explored, and knowledge of the underlying PDEs is needed.

More recently, new families of neural network-based models, termed neural operators, were developed. Such functional regressors aim to directly map between function spaces. Once constructed, they can be used in conjunction with a number of discretization techniques and resolution levels to generate families of approximations over predefined grids. These models possess the desirable attribute of being discretization invariant: it is possible to utilize the same architecture and hyperparameters, regardless of the discretization of the underpinning functional data Kovachki et al. (2023).

PCA-Net is one form of model reduction operator regression Bhattacharya et al. (2021). The approach takes advantage of a classical finite-dimensionalizing reduction technique (namely, PCA) and develops a neural-network-based approximation between the two resulting finite-dimensional latent spaces. Similar ideas were pursued in, e.g., Hesthaven and Ubbiali (2018) and Wang et al. (2019). Other papers, such as Benner et al. (2020), McQuarrie et al. (2021),

¹Dept. of Mechanical Engineering and Materials Science, Duke Univ., Durham, NC 27708. Email: yepeng.quek@duke.edu

²Dept. of Civil and Environmental Engineering, Duke Univ., Durham, NC 27708. Email: jinyi.yong@duke.edu

³Professor, Dept. of Civil and Environmental Engineering, Duke Univ., Durham, NC 27708 (corresponding author). Email: johann.guilleminot@duke.edu

Note. This manuscript was submitted on October 13, 2023; approved on March 15, 2024; published online on June 6, 2024. Discussion period open until November 6, 2024; separate discussions must be submitted for individual papers. This paper is part of the *Journal of Engineering Mechanics*, © ASCE, ISSN 0733-9399.

Peherstorfer (2019), Peherstorfer and Willcox (2016), and Qian et al. (2020), have combined ideas from model reduction with data-driven learning to find a low-dimensional latent space and learn a system of ordinary differential equations (ODEs).

Another prevailing class of neural operators in the literature, DeepONet Lu et al. (2019, 2021) (and variations thereof) constitute another approach where two subnetworks (namely, the branch and trunk nets) are used to encode the input function and index (e.g., location) variables for the output function. While the branch net requires knowledge of the input over a fixed grid in its original version, the DeepONet framework was later extended and, in particular, made discretization-invariant by utilizing the PCA-based approach introduced in de Hoop et al. (2022); see Kovachki et al. (2023). There exists a vast literature reporting on the efficiency of this framework in a wide range of applications, and a variational energy-based architecture of DeepONet specifically devised to predict crack paths in brittle homogeneous materials can be found in Goswami et al. (2021).

Inspired by preliminary work on the graph kernel network (GKN) Li et al. (2020b), the Fourier neural operator (FNO) framework was proposed in Li et al. (2021a). These methods similarly rely on neural networks to lift to, and project from, the latent space, and involve a parameterization in Fourier space. The approach essentially proceeds by defining a sequence of functions in the latent space, through layers of integral operators. To circumvent the aliasing-generated bias reported in Fanaskov and Oseledets (2022) and expand applicability to arbitrary geometries and various input formats (point clouds, nonuniform meshes, etc.), the authors proposed the geometry-aware FNO in Li et al. (2022).

The paper Kovachki et al. (2023) provides a comparative study of commonly employed neural operators, including PCA-Net, DeepONet, GKN, and FNO. An extensive comparison between DeepONet and FNO can be found in Lu et al. (2022). In addition, hybrid paradigms have recently emerged that combine neural operators with physics-based constraints and differentiable physics Ramsundar et al. (2021), Shankar et al. (2023), and Li et al. (2021b). Alternative approaches include kernel-based frameworks for operator learning operator Battie et al. (2023) and variational autoencoding neural operators Seidman et al. (2023).

The above techniques all involve neural networks as regressors and mostly differ in the way, and at which level, functional dependencies are encoded and decoded. This contribution aims to explore an alternative path relying on a generative model, with the aim of bypassing the complexity of neural network training—which remains a limiting factor in most applications of practical interest,

especially when methods are deployed on small data sets. Our contributions are as follows:

- We formulate a statistical operator learning approach, using a regularization and a generative model (here, probabilistic learning on manifolds), in lieu of a deterministic regressor.
- We provide a comparative study between the proposed approach and some state-of-the-art techniques, including PCA-Net and geometry-aware FNO (geo-FNO).
- We show that the statistical approach can achieve similar accuracy to geo-FNO and PCA-Net, while exhibiting a minimal parameterization.

This paper is organized as follows. We first present the phase-field approach to brittle fracture and formulate the resulting learning problem. We then introduce the probabilistic framework, PCA-GEM (where the abbreviation GEM stands for Generative Model), and provide an overview of PCA-Net and geo-FNO, considered for the sake of comparison. Finally, we assess the performance of the operator learning techniques on two-dimensional fracture simulations involving a spatially varying stochastic toughness.

Forward Problem Description

Phase-Field Model

We consider an arbitrary heterogeneous body $\Omega \subset \mathbb{R}^n$ (with $n \in \{1, 2, 3\}$) with external boundary $\partial\Omega$, containing an internal crack surface Γ ; see Fig. 1(b). The boundary $\partial\Omega$ is decomposed as $\partial\Omega = \partial\Omega_u \cup \partial\Omega_t$, where $\partial\Omega_u$ and $\partial\Omega_t$ are subjected to Dirichlet and Neumann boundary conditions, respectively, with $\partial\Omega_u \cap \partial\Omega_t = \emptyset$. Two variables, namely the displacement field $\mathbf{u}$ and the phase-field variable d , are used to describe the state of the system.

In the phase-field approach, a regularized counterpart for Γ is introduced as

$$\Gamma_\ell(d) = \int_{\Omega} \gamma(d, \nabla d) dV$$

where γ denotes the crack surface density function and ℓ is a regularization length. In this work, we use the so-called AT2 model, defined as

$$\gamma(d, \nabla d) = \frac{1}{2\ell} d^2 + \frac{\ell}{2} |\nabla d|^2$$

which coincides with the form derived in the gamma-convergent regularization of free discontinuity problems Ambrosio and Tortorelli (1990). The phase-field damage variable takes values in the interval

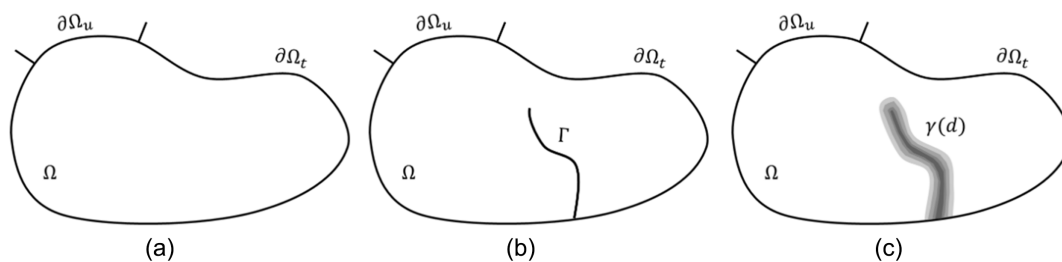


Fig. 1. Three subimages demonstrating the procedure of depicting a smoothed crack surface: (a) intact solid object with Dirichlet boundary condition $\partial\Omega_u$ and $\partial\Omega_t$ indicated; (b) solid object with a crack identified as the crack set Γ ; and (c) solid object with a crack represented using the crack-density function γ . [Reprinted from Computer Methods in Applied Mechanics and Engineering, Vol. 368, T. Hu, J. Guilleminot, and J. E. Dolbow, “A phase-field model of fracture with frictionless contact and random fracture properties: Application to thin-film fracture and soil desiccation,” 113106, © 2020, with permission from Elsevier.]

$[0, 1]$, with $d = 0$ corresponding to intact material and $d = 1$ for fully damaged material. Small deformations are assumed throughout, with the infinitesimal strain tensor defined as

$$\boldsymbol{\varepsilon}(\nabla \mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T) \quad (1)$$

While many variations of the phase-field approach were proposed in the past two decades, we rely on the formulation proposed in the seminal work Miehe et al. (2010) for the sake of illustration; see Francfort and Marigo (1998) and Bourdin et al. (2000, 2008) for reviews. In order to restrict stress degradation to tension, the strain tensor $\boldsymbol{\varepsilon}$ is split into positive and negative components, denoted by $\boldsymbol{\varepsilon}_+$ and $\boldsymbol{\varepsilon}_-$, respectively:

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_+ + \boldsymbol{\varepsilon}_-$$

This is realized by applying the spectral decomposition of the strain tensor via

$$\boldsymbol{\varepsilon}_\pm = \sum_{i=1}^n \langle \varepsilon_i \rangle_\pm \mathbf{n}_i \otimes \mathbf{n}_i$$

where $\{\varepsilon_i\}_{i=1}^n$ are the principal strains, $\{\mathbf{n}_i\}_{i=1}^n$ are the associated principal strain directions, and bracket operators are defined as

$$\langle a \rangle_+ = \begin{cases} a & \text{if } a \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \langle a \rangle_- = \begin{cases} a & \text{if } a \leq 0 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

The internal energy is given by

$$\mathcal{E}_{\text{internal}}(\mathbf{u}, d) = \int_{\Omega} \psi(\boldsymbol{\varepsilon}(\mathbf{u}), d) dV \quad (3)$$

with ψ the elastic energy density. In the case of a linear isotropic material, this density can be written as

$$\psi(\boldsymbol{\varepsilon}, d) = g(d)\psi_0^+(\boldsymbol{\varepsilon}) + \psi_0^-(\boldsymbol{\varepsilon}) \quad (4)$$

where g is the degradation function. A quadratic degradation function is used in this work: $g(d) = (1 - d)^2$ [see Miehe et al. (2010) for discussion]. In Eq. (4), the terms ψ_0^+ and ψ_0^- correspond to the split of a reference (isotropic) elastic energy density ψ_0 (associated with undamaged material), induced by the aforementioned strain split:

$$\begin{aligned} \psi_0^+(\boldsymbol{\varepsilon}) &= \frac{1}{2} \lambda \langle \text{Tr}(\boldsymbol{\varepsilon}) \rangle_+^2 + \mu \boldsymbol{\varepsilon}_+ : \boldsymbol{\varepsilon}_+, \\ \psi_0^-(\boldsymbol{\varepsilon}) &= \frac{1}{2} \lambda \langle \text{Tr}(\boldsymbol{\varepsilon}) \rangle_-^2 + \mu \boldsymbol{\varepsilon}_- : \boldsymbol{\varepsilon}_- \end{aligned} \quad (5)$$

where $\lambda > 0$ and $\mu > 0$ are the Lamé coefficients of the healthy material. The fracture energy associated with the crack set Γ reads

$$\mathcal{E}_{\text{fracture}}|_{\Gamma} = \int_{\Gamma} \mathcal{G}_c dA \quad (6)$$

with $\mathcal{G}_c$ the fracture toughness, and is approximated as

$$\mathcal{E}_{\text{fracture}}|_{\Gamma} \approx \tilde{\mathcal{E}}_{\text{fracture}}|_{\Omega} = \int_{\Omega} \mathcal{G}_c \gamma(d, \nabla d; l) dV \quad (7)$$

The solution to the coupled problem is then obtained as the minimizer of the total energy

$$\tilde{\mathcal{E}}_{\text{total}} = -\tilde{\mathcal{E}}_{\text{external}}|_{\Omega, \partial\Omega} + \tilde{\mathcal{E}}_{\text{internal}}|_{\Omega} + \tilde{\mathcal{E}}_{\text{fracture}}|_{\Omega} \quad (8)$$

subject to the irreversibility constraint $\dot{d} \geq 0$, where $\tilde{\mathcal{E}}_{\text{external}}$ is the external energy generated by surface traction and body force. In the above, tilde notation is used to emphasize approximations due to the regularization of the crack set. Following (Miehe et al. 2010), the governing equations are given by

$$\nabla \cdot \tilde{\boldsymbol{\sigma}} + \mathbf{b} = \mathbf{0}, \quad \text{in } \Omega \quad (9)$$

and

$$\mathcal{G}_c \left(\frac{d}{\ell} - \ell \Delta d \right) - 2(1 - d)\mathcal{H}(\boldsymbol{\varepsilon}) = 0, \quad \text{in } \Omega \quad (10)$$

where $\mathcal{H}$ is the monotonic driving force

$$\mathcal{H}(\boldsymbol{\varepsilon}) = \max_{\tau \in [0, t]} \psi_0^+(\boldsymbol{\varepsilon}(\mathbf{x}, \tau)) \quad (11)$$

introduced to enforce the monotonicity constraint $\dot{d} > 0$ (Miehe et al. 2010). Boundary conditions are given by

$$\tilde{\boldsymbol{\sigma}} \cdot \mathbf{n} = \boldsymbol{\tau}, \quad \text{on } \partial\Omega \quad (12)$$

and

$$\nabla d \cdot \mathbf{n} = 0, \quad \text{on } \partial\Omega \quad (13)$$

The stress deteriorates with damage [following the hybrid formulation by Ambati et al. (2014)] according to

$$\tilde{\boldsymbol{\sigma}} = g(d) \frac{\partial \psi_0}{\partial \boldsymbol{\varepsilon}}$$

Trial and weight spaces for the displacement field are respectively defined as

$$\begin{aligned} \mathcal{U}_u &= \{\mathbf{u} \in H^1(\Omega), \mathbf{u} = \bar{\mathbf{u}} \quad \text{on } \partial\Omega_u\}, \\ \mathcal{V}_u &= \{\mathbf{v} \in H^1(\Omega), \mathbf{v} = 0 \quad \text{on } \partial\Omega_u\} \end{aligned} \quad (14)$$

in which $H^1(\Omega)$ = Sobolev space of functions on Ω with derivatives in $L^2(\Omega)$; and $\bar{\mathbf{u}}$ = displacement prescribed on $\partial\Omega_u$. The trial and weight spaces are given by

$$\mathcal{U}_d = \{d \in H^1(\Omega)\}, \quad \mathcal{V}_d = \{w \in H^1(\Omega)\} \quad (15)$$

for the phase-field variable, respectively.

The system of coupled equations is solved using an alternating minimization approach in the FEniCS computing platform [adopted from Hirshikesh et al. (2019)].

Learning Framework for Phase-Field Predictions

In the context of phase-field simulations, we specifically consider the mapping between a material parameter, chosen as the fracture toughness $\mathcal{G}_c$, and the damage field d . The goal is to enable statistical analysis for random instantiations of the toughness, without necessitating the recourse to expensive high-fidelity simulations. For the purpose of building a data set, the (positive) fracture toughness is modeled as a second-order, translation random field indexed by Ω and defined on a probability space $(\Theta, \mathcal{F}, P)$. This toughness random field is denoted by $\{\mathcal{G}_c(s), s \in \Omega\}$. The information-theoretic probabilistic model used to generate realizations of this material parameter is described in the section titled ‘‘Stochastic Model for the Fracture Toughness.’’ The damage random field $\{d(s), s \in \Omega\}$ is then defined through the pushforward action of Ψ , symbolically written as $d = \Psi(\mathcal{G}_c)$ (in the almost sure sense), and is of second-order (since the support of the associated measure

is bounded). Hence we take $\mathcal{X} = L^\infty(\Omega, \mathbb{R}_+^*)$ and $\mathcal{Y} = H^1(\Omega, [0, 1])$, and consider a data set composed of realizations $\{s \mapsto \mathcal{G}_c(s, \theta_i)\}_{i \geq 1}$ of the fracture toughness, and the associated realizations $\{s \mapsto d(s, \theta_i)\}_{i \geq 1}$ of the damage field, with $\theta_i \in \Theta$.

Operator Learning Methods

In this section, we first introduce the operator learning method based on the generative model. PCA-Net and geo-FNO are then briefly reviewed as baseline techniques.

PCA-GEM

Overview

To position the proposed approach and clarify the encoding–decoding strategy, consider the formulation introduced in Bhattacharya et al. (2021). Denote by $F_{\mathcal{X}}: \mathcal{X} \rightarrow \mathbb{R}^{d_{\mathcal{X}}}$ and $G_{\mathcal{X}}: \mathbb{R}^{d_{\mathcal{X}}} \rightarrow \mathcal{X}$ the encoder and decoder associated with the input space, where $d_{\mathcal{X}}$ denotes the reduced input dimension. Similarly, $F_{\mathcal{Y}}: \mathcal{Y} \rightarrow \mathbb{R}^{d_{\mathcal{Y}}}$ and $G_{\mathcal{Y}}: \mathbb{R}^{d_{\mathcal{Y}}} \rightarrow \mathcal{Y}$ are the encoder and decoder for the output space $\mathcal{Y}$, with $d_{\mathcal{Y}}$ the reduced output dimension. Following Bhattacharya et al. (2021), the map Ψ can be approximated as $\Psi \approx G_{\mathcal{Y}} \circ \varphi F_{\mathcal{X}}$, where errors arise from the use of reduced orders in the encoder $F_{\mathcal{X}}$ and decoder $G_{\mathcal{Y}}$. This construction requires (1) the definition of the encoder and decoder; as well as (2) the construction of the mapping $\varphi: \mathbb{R}^{d_{\mathcal{X}}} \rightarrow \mathbb{R}^{d_{\mathcal{Y}}}$.

A convenient way to encode and decode the input and output fields in the considered probabilistic setting is to use Karhunen–Loève expansions. The latter are widely employed, in the field of uncertainty quantification, as a means to perform statistical dimension reduction and devise stochastic solvers (e.g., stochastic collocation methods) in the infinite-dimensional setting Ghanem et al. (2017). We then consider

$$x(s) \approx x^{(d_{\mathcal{X}})}(s) = \underline{x}(s) + \sum_{i=1}^{d_{\mathcal{X}}} \sqrt{\lambda_{x,i}} \eta_{x,i} \phi_{x,i}(s) \quad (16)$$

and

$$y(s) \approx y^{(d_{\mathcal{Y}})}(s) = \underline{y}(s) + \sum_{i=1}^{d_{\mathcal{Y}}} \sqrt{\lambda_{y,i}} \eta_{y,i} \phi_{y,i}(s) \quad (17)$$

where $\underline{x}(s) = \mathbb{E}\{x(s)\}$ and $\{\lambda_{x,i}, \phi_{x,i}\}_{i=1}^{d_{\mathcal{X}}}$ = pairs of associated eigenvalues and eigenfunctions satisfying the Fredholm equation:

$$\int_{\Omega} C_x(s, t) \phi_{x,i}(t) dt = \lambda_{x,i} \phi_{x,i}(s), \quad \forall s \in \Omega \quad (18)$$

where C_x = covariance function of x ; and $\{\eta_{x,i}\}_{i=1}^{d_{\mathcal{X}}}$, where

$$\eta_{x,i} = \frac{1}{\sqrt{\lambda_{x,i}}} \int_{\Omega} (x(s) - \underline{x}(s)) \phi_{x,i}(s) ds \quad (19)$$

is a family of centered, pairwise-uncorrelated random variables (similar equations hold for y , with obvious notation). Note that $x^{(d_{\mathcal{X}})} \rightarrow x$ and $y^{(d_{\mathcal{Y}})} \rightarrow y$ as $d_{\mathcal{X}} \rightarrow +\infty$ and $d_{\mathcal{Y}} \rightarrow +\infty$, respectively, in the mean-square sense. The encoder and decoder are then defined as Bhattacharya et al. (2021)

$$F_{\mathcal{X}}: x^{(d_{\mathcal{X}})} \mapsto \boldsymbol{\eta}_x = (\eta_{x,1}, \dots, \eta_{x,d_{\mathcal{X}}})^T \quad (20)$$

and

$$G_{\mathcal{Y}}: \boldsymbol{\eta}_y = (\eta_{y,1}, \dots, \eta_{y,d_{\mathcal{Y}}})^T \mapsto y^{(d_{\mathcal{Y}})} \quad (21)$$

It remains to define the mapping φ between $F_{\mathcal{X}}(x^{(d_{\mathcal{X}})})$ and $F_{\mathcal{Y}}(y^{(d_{\mathcal{Y}})})$. One natural way to construct this mapping is to use a neural network regressor, denoted by φ_{NN} . This corresponds to the choice made in the seminal contribution Bhattacharya et al. (2021), where the composite map $\tilde{\Psi} = G_{\mathcal{Y}} \circ \varphi_{NN} \circ F_{\mathcal{X}} \approx \Psi$ thus obtained is referred to as PCA-Net (see section titled “PCA-Net”). In this work, we explore an alternative approximation, obtained through statistical sampling rather than regression. Specifically, we consider the random vector $\boldsymbol{\eta} = (\boldsymbol{\eta}_x^T, \boldsymbol{\eta}_y^T)^T$ in $\mathbb{R}^{d_{\mathcal{X}}+d_{\mathcal{Y}}}$ and assume that a generative model enables sampling from the joint probability distribution $P_{\boldsymbol{\eta}}$ of $\boldsymbol{\eta}$. In the context of surrogate modeling, the conditional distribution $P_{\boldsymbol{\eta}_y|\boldsymbol{\eta}_x}$ reads as

$$P_{\boldsymbol{\eta}_y|\boldsymbol{\eta}_x=\mathbf{e}_x}(d\mathbf{e}_y) = \delta_0(\mathbf{e}_y - \Psi(\mathbf{e}_x)) \quad (22)$$

where δ_0 = Dirac measure centered at the origin. An approximation can be introduced as

$$\tilde{\Psi} = G_{\mathcal{Y}} \circ \varphi_{\text{GEM}} \circ F_{\mathcal{X}} \quad (23)$$

where

$$\varphi_{\text{GEM}}(\mathbf{e}_x) = \mathbb{E}_{\hat{P}_{\boldsymbol{\eta}_y|\boldsymbol{\eta}_x}} \{\boldsymbol{\eta}_y | \boldsymbol{\eta}_x = \mathbf{e}_x\} \quad (24)$$

and the notation $\hat{P}_{\boldsymbol{\eta}_y|\boldsymbol{\eta}_x}$ in the right-hand side = regularization obtained through a kernel density estimation, applied to the augmented (generated) data set. Any appropriate generative model can be used to draw samples from the above conditional distribution. In what follows, we consider the formulation proposed in Soize and Ghanem (2016), using a reduced-order underdamped Langevin equation (i.e., a reduced-order Itô stochastic differential equation); see Soize and Ghanem (2020a), Soize and Ghanem (2022), and Soize (2022) for additional methodological developments, as well as Ghanem and Soize (2018), Soize and Ghanem (2020b) and Ghanem et al. (2021) for various applications. This framework is chosen due to its simplicity of implementation, minimal parameterization, and proven efficiency to sample concentrated measures. It is briefly recalled in the next section for the sake of completeness.

Generative Model

To construct the data set, the high-fidelity problem described in the section titled “Forward Problem Description” is solved with N independent realizations of the input field (i.e., the fracture toughness). The realizations of the input and output fields are then encoded to yield samples of the latent variables $\boldsymbol{\eta}_x$ and $\boldsymbol{\eta}_y$ (and the variable $\boldsymbol{\eta}$ obtained by concatenation), following the strategy exposed in the section titled “Overview.” These samples are collected in the matrix $[\boldsymbol{\eta}(\theta)] = [\boldsymbol{\eta}(\theta_1), \dots, \boldsymbol{\eta}(\theta_N)]$ with values in $\mathbb{R}^{n \times N}$, with $n = d_{\mathcal{X}} + d_{\mathcal{Y}}$.

The method builds upon three ingredients. First, the data set is normalized using principal component analysis. To this aim, consider the empirical mean

$$\underline{\boldsymbol{\eta}} = \frac{1}{N} \sum_{i=1}^N \boldsymbol{\eta}(\theta_i)$$

and covariance matrix

$$[C] = \frac{1}{N-1} \sum_{i=1}^N (\boldsymbol{\eta}(\theta_i) - \underline{\boldsymbol{\eta}})(\boldsymbol{\eta}(\theta_i) - \underline{\boldsymbol{\eta}})^T$$

and let

$$[\eta] = [\eta, \dots, \eta]$$

The data set is then normalized according to

$$[H] = [\Lambda]^{-1/2} [\Phi]^T ([\eta(\theta)] - [\eta]) \quad (25)$$

where $[\Lambda]$ and $[\Phi]$ = diagonal matrix of eigenvalues, sorted in non-increasing order, and matrix of eigenvectors of the covariance matrix $[C]$, respectively. Here ν dominant eigenmodes are retained using a standard mean-square error criterion, leading in effect to a dimension reduction: $[H] = [\mathbf{h}(\theta_1), \dots, \mathbf{h}(\theta_N)] \in \mathbb{R}^{\nu \times N}$. The matrix $[H]$ is interpreted as the realization of a random matrix $[\mathbf{H}]$ that takes values in $\mathbb{R}^{\nu \times N}$.

Second, a diffusion map basis is constructed as follows Coifman and Lafon (2006). Let k_ϵ be the symmetric, positivity-preserving and positive semidefinite kernel given by

$$k_\epsilon(\mathbf{h}, \mathbf{h}') = \exp(-\|\mathbf{h} - \mathbf{h}'\|^2 / (4\epsilon)), \quad \mathbf{h}, \mathbf{h}' \in \mathbb{R}^\nu \quad (26)$$

where $\|\cdot\|$ = Euclidean norm; and ϵ = positive scaling parameter. Using the normalized data set defined in Eq. (25) and the above kernel, a matrix $[K] \in \mathbb{R}^{N \times N}$ is constructed component-wise as

$$K_{ij} = k_\epsilon(\mathbf{h}(\theta_i), \mathbf{h}(\theta_j)), \quad 1 \leq i, j \leq N \quad (27)$$

Let $[B]$ be defined through $B_{ij} = \delta_{ij} \sum_{k=1}^N K_{ik}$, with δ the Kronecker delta, and consider the matrix

$$[P] = [B]^{-1} [K] \quad (28)$$

The above normalization makes $[P]$ a stochastic matrix that can accordingly be interpreted as the transition matrix of a random walk on the graph that corresponds to the data Coifman et al. (2005). Next, introduce the symmetric positive-definite matrix $[P_S] = [B]^{1/2} [P] [B]^{-1/2}$, and denote by $\{\lambda_{D,i}\}_{i=1}^N$ and $\{\phi_D^{(i)}\}_{i=1}^N$ its eigenvalues (ordered in a non-increasing sequence) and eigenvectors. The matrix of diffusion map basis vectors $[G]$ is then defined as $[G] = [\mathbf{g}^{(1)}, \dots, \mathbf{g}^{(m)}]$ with

$$\mathbf{g}^{(i)} = \lambda_{D,i}^s [B]^{-1/2} \phi_D^{(i)}, \quad 1 \leq i \leq m \quad (29)$$

where $m \leq N$ = parameter enabling dimension reduction; and $s \in \mathbb{N}_{>0}$ = scale parameter [Coifman et al. (2005)].

Third, a Langevin equation is introduced as a means to draw new samples of $[\mathbf{H}]$ (Hamiltonian Monte Carlo). When considered in conjunction with data, this equation typically involves minus the logarithm of the empirical probability density function (built on the data set) as potential (in the drift term). However this classical setup does not ensure proper concentration when samples exhibit some structure Soize and Ghanem (2016). To circumvent this limitation, the diffusion map basis $[G]$ is used to project $[\mathbf{H}]$ as

$$[\mathbf{H}] = [\mathbf{Z}][G]^T \quad (30)$$

where $[\mathbf{Z}]$ = auxiliary random matrix with values in $\mathbb{R}^{\nu \times m}$, defined on the probability space $(\Theta, \mathcal{T}, P)$, and

$$[\mathbf{Z}] = [\mathbf{H}][A], \quad [A] = [G]([G]^T [G])^{-1} \quad (31)$$

This projection enables the definition of the reduced-order Langevin equation

$$\begin{aligned} d[\mathcal{Z}_t] &= [\mathcal{Y}_t] dt \\ d[\mathcal{Y}_t] &= [L([\mathcal{Z}_t])] dt - \gamma[\mathcal{Y}_t] dt + \sqrt{2\gamma} [d\mathcal{W}_t], \quad t > 0 \end{aligned} \quad (32)$$

with the initial condition $[\mathcal{Z}_0] = [\mathbf{H}][A]$ and $[\mathcal{Y}_0] = [\mathbf{N}][A]$ almost surely, where $[\mathbf{N}]$ is a $(\nu \times N)$ matrix, the columns of which are independent copies of the standard Gaussian random vectors in $\mathbb{R}^\nu$; and $\gamma > 0$ = tunable damping parameter. The matrix $[L]$ is defined as $[L([\mathcal{Z}_t])] = [L([\mathcal{Z}_t][G]^T)][A]$, where

$$L([U])_{ij} = \frac{1}{\hat{p}(\mathbf{u}^j)} \{\nabla_{\mathbf{u}^j} \hat{p}(\mathbf{u}^j)\}_i, \quad [U] = [\mathbf{u}^1, \dots, \mathbf{u}^N] \quad (33)$$

and $\hat{p}$ = kernel density estimator constructed with the normalized data set [following Soize (2015)]. The $\mathbb{R}^{\nu \times N}$ -valued stochastic process $\{[d\mathcal{W}_t], t \geq 0\}$ is defined by $[d\mathcal{W}_t] = [d\mathbf{W}_t][A]$, where $[d\mathbf{W}_t] = [d\mathbf{W}_t^1, \dots, d\mathbf{W}_t^N]$ and $\{\mathbf{W}^i\}_{i=1}^N$ are independent copies of the normalized Wiener process in $\mathbb{R}^\nu$. It can be proven that $\lim_{t \rightarrow +\infty} [\mathcal{Z}_t] = [\mathbf{Z}]$ in the probability distribution, which enables the generation of new samples of $[\mathbf{Z}]$, and ultimately of $[\eta]$ through back substitution in Eq. (30) and using [Eq. (25)]:

$$[\eta] = [\eta] + [\Phi][\Lambda]^{1/2} [\mathbf{H}] \quad (34)$$

Various time integrators can be used to solve the Langevin dynamics, including the standard Euler–Maruyama and Verlet schemes. In this work, the latter is deployed for the sake of illustration.

Deep Learning Strategies

PCA-Net

As previously indicated, PCA-Net is an operator approximation method that utilizes PCA to finite-dimensionalize function spaces in the span of PCA bases; see Bhattacharya et al. (2021). Finite-dimensionalization is performed in both the input and output spaces, and a standard fully connected neural network (NN) is used to interpolate between the finite-dimensional latent spaces; see Kovachki et al. (2023) for a review and comparison with other existing PCA-based techniques, including a variant of DeepONet (POD-DeepONet) Lu et al. (2022). Here, we employ PCA-Net for the sake of comparison, in which case the approximation reads as

$$\tilde{\Psi} = G_{\mathcal{Y}} \circ \varphi_{NN} \circ F_{\mathcal{X}} \quad (35)$$

Following Bhattacharya et al. (2021), a zero-extended stacked neural network model φ_{NN} is used to approximate φ to a desired level of accuracy. In this construction, the neural network mapping from $\mathbb{R}^{d_x}$ to $\mathbb{R}^{d_y}$ is set to zero when the input latent variable lies outside the hypercube $[-M, M]^{d_x}$ (where $M > 0$ is a parameter), which allows one to theoretically handle the noncompactness of the latent spaces (see the above reference for results in approximation theory).

To optimize the neural network architecture for the presented application, a comprehensive parametric analysis was performed (see section titled “PCA-Net Results” for details). During the training process, the loss function

$$\mathcal{L}_{\text{train}} = \mathbb{E} \frac{\|\tilde{\Psi} - \Psi\|_2^2}{\|\Psi\|_2^2} \quad (36)$$

was used, where the mathematical expectation is evaluated through a standard statistical estimator and norms are squared for algorithmic efficiency in the retained computing setup.

(Geo-)FNO

The graph neural operator technique [also known as graph kernel network (GKN)] proposed in Li et al. (2020b) seeks to represent the

mapping Ψ through the composition of nonlinear activation functions and a class of integral operators, with kernel integration realized through message passing on graph networks. While GKN is able to achieve crucial properties of a neural operator such as input resolution independence and consideration of long-range interactions in the feature space, specifically node-to-node interactions, this architecture may exhibit instability when the number of hidden layers increases You et al. (2022). Nevertheless, GKNs have demonstrated successful applications in the context of Darcy's flow and Navier–Stokes equations for learning tasks related to partial differential equations (PDEs) Li et al. (2020b, a). The FNO model Li et al. (2021a) was subsequently proposed and relies on the parameterization of the kernel functions in Fourier space, resulting in a more compact, expressive, and resolution-independent representation, as well as in improved computational efficiency through the use of fast Fourier transform (FFT).

In its original form, FNO was limited to rectangular domains with uniform meshes, due to reliance on FFT. This limitation was circumvented in Li et al. (2022) with the development of geo-FNO, an extension of FNO where a homeomorphism between an arbitrary irregular input mesh and a uniform latent mesh is introduced, hence enabling the use of FFT. The deformation of the input domain can be learned in an end-to-end manner using the FNO architecture or predetermined analytically. Consequently, geo-FNO performs end-to-end learning of a deformed uniform latent mesh alongside the solution operator. In the present study, a deformation neural network is employed to learn the coordinate map (i.e., the deformation of the input domain). The deformation network takes the input coordinates, representing the coordinates of the input mesh, and produces the deformed input coordinates. The same sinusoidal features and network configuration as described by the authors in Li et al. (2022) are adopted. The training loss is chosen to be the same as in Eq. (36) for consistency. Readers are referred to Li et al. (2021a, 2022) and Kovachki et al. (2023) for more detailed information.

Numerical Results

In this section, we assess the accuracy of the learning methods presented in the section titled “Operator Learning Methods.” The probabilistic model used to generate samples of the input field is first introduced in the section titled “Stochastic Model for the Fracture Toughness.” Illustrative results related to material uncertainty propagation (i.e., to the generation of the output field) are next shown in the section titled “Forward Simulations and Reduction.” The performance of each learning technique is then discussed in the section titled “Learning Results.”

Stochastic Model for the Fracture Toughness

The fracture toughness is modeled as random field denoted by $\{\mathcal{G}_c(s), s \in \Omega\}$ and is defined on the probability space, with $\Omega = (0, 1]^2$ (unit of length is millimeter). Since this physical parameter is positive almost surely, a non-Gaussian model must be constructed. A convenient way to proceed is to have recourse to a translation model Grigoriu (1984), where the field of interest is expressed as

$$\mathcal{G}_c(s) = \mathcal{T}(\Xi(s)), \quad \forall s \in \Omega \quad (37)$$

where $\{\Xi(s), s \in \Omega\}$ = latent centered Gaussian random field, specified by its covariance function C_Ξ ; and $\mathcal{T}$ = measurable non-linear mapping that pushes forward the Gaussian measure to a non-Gaussian target measure. There exist various strategies to define the latter, ranging from the use of an empirical distribution (estimated on a data set) to *data-free* model construction. Here we invoke information theory Jaynes (1957a, b) and the principle of maximum entropy Shannon (1948) as a rationale to build $\mathcal{T}$ with minimal modeling bias. In this case, and accounting for the aforementioned positivity constraint, the first-order marginal probability distribution induced by entropy maximization corresponds to the Gamma law Hu et al. (2020b). Hence the toughness random field can be defined as

$$\mathcal{G}_c(s) = \left(F_{\mathcal{G}(1/\delta^2, \underline{\mathcal{G}}_c \delta^2)}^{-1} \right) \circ F_{\mathcal{N}(0,1)}(\Xi(s)), \quad \forall s \in \Omega \quad (38)$$

where $F_{\mathcal{G}}^{-1}$ = inverse cumulative distribution of the Gamma law with shape and scale parameters given by $1/\delta^2$ and $\underline{\mathcal{G}}_c \delta^2$, respectively; $\underline{\mathcal{G}}_c$ = mean value of the toughness; and δ = its coefficient of variation. Similarly, $F_{\mathcal{N}}$ is the cumulative distribution of the (standard) Gaussian law. The values $\underline{\mathcal{G}}_c = 2.7 \text{ [MPa} \cdot (\text{mm})^{-2}]$ and $\delta = 0.3$ are used in the simulations presented below.

The latent Gaussian random field $\{\Xi(s), s \in \Omega\}$ is defined by a separable squared-exponential covariance function, with a spatial correlation length L_c set to 0.2 mm. Since the phase-field simulations require a mesh that is much finer than the resolution necessary to resolve the statistical fluctuations, the random field of fracture toughness is generated on a coarse and structured mesh, via its Karhunen–Loève expansion, and then interpolated on the fine unstructured mesh used in the phase-field simulations. Realizations of the input random field are displayed in Fig. 2. Such realizations are subsequently fed into the phase-field framework presented in the section titled “Forward Problem Description” to produce the data set for the operator learning frameworks.

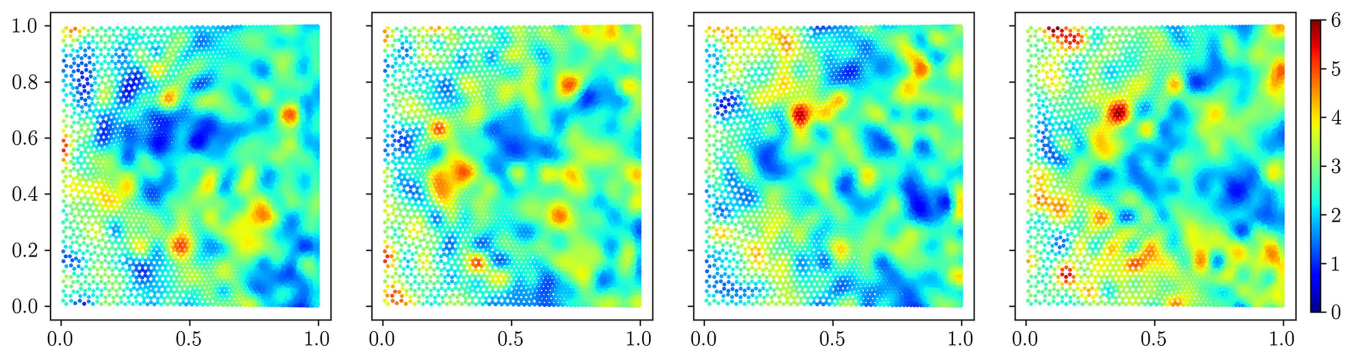


Fig. 2. Independent realizations of the input fracture toughness random field $\{\mathcal{G}_c(s), s \in \Omega\}$.

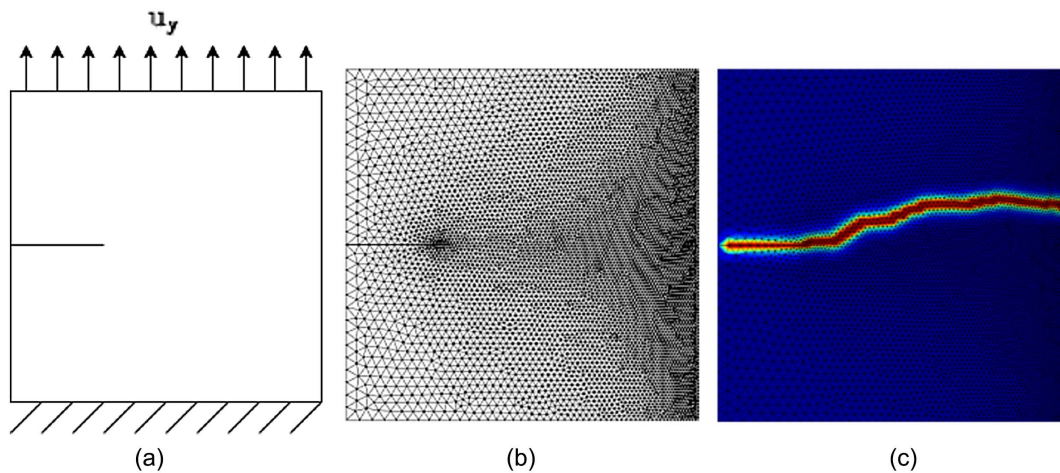


Fig. 3. (a) 2D tension test: a square plate with a preexisting crack and boundary conditions; (b) finite element discretization of the domain Ω ; and (c) example of a damage field.

Forward Simulations and Reduction

To analyze the performance of the operator learning frameworks, we consider an edge-notched unilateral tension test. A $1 \text{ mm} \times 1 \text{ mm}$ plate with a preexisting crack of 0.25 mm on the left side is considered and is loaded in uniaxial tension as illustrated in Fig. 3. The plate is assumed to be composed of a heterogeneous material with known deterministic elastic properties and critical strength, while the fracture toughness is allowed to vary spatially following the model presented in the section titled “Stochastic Model for the Fracture Toughness.” Following Miehe et al. (2010), the values for the Young’s modulus and Poisson ratio were fixed to 210 [GPa] and 0.3 , respectively, and the regularization length is chosen as $\ell = 0.015 \text{ mm}$ (note that a convergence analysis with respect to ℓ was performed). While considering a more complete model accounting for variations in other material parameters is a natural extension [see, e.g., Hu et al. (2020a) and Hun et al. (2019)] for models capturing fluctuations in the critical fracture energy and elastic properties, respectively, such a construction was not pursued given the scope of this work. The plate is fixed at the bottom, and the top surface is allowed to move along the vertical direction (the horizontal displacement is fixed to zero). A vertical displacement u_y is applied at the top of the plate, as illustrated in Fig. 3(a).

The finite element characteristic size h and displacement increments were determined following established practice and convergence analyses Hu et al. (2020a) and Nguyen et al. (2016). Mesh convergence analysis was performed under the constraints

$$\min_e h_e \leq \frac{\ell}{2} \quad (39)$$

and

$$\min_e h_e \leq \frac{L_c}{4} \quad (40)$$

where h_e = characteristic size of the e th finite element and the second inequality ensures that the discretization of the random field is fine enough (recall that L_c is the spatial correlation length of the latent Gaussian random field). Note that the condition in Eq. (39) is generally much stronger than the one expressed in Eq. (40). The final mesh comprises 6,736 linear triangular elements, with

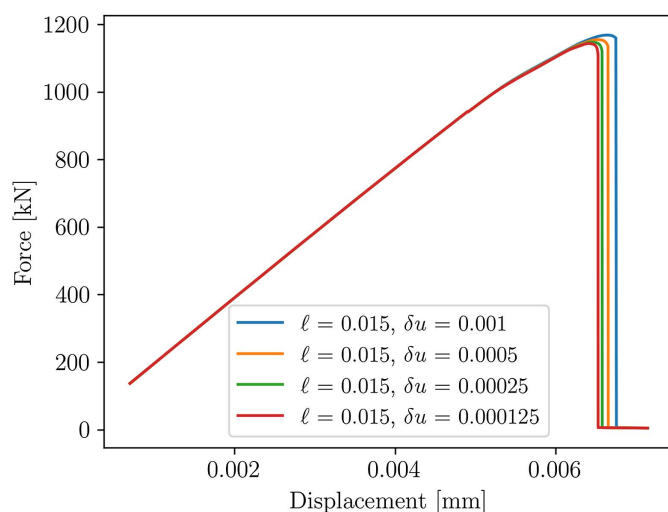


Fig. 4. Convergence of the force-displacement curve obtained for $\ell = 0.0015 \text{ mm}$, with respect to $\delta u \text{ mm}$ for $u_y > 0.7 \times \bar{u} \text{ mm}$.

an element size (in the unstructured mesh) ranging from 0.0075 (near the crack) to 0.0375 (away from the zone of crack propagation).

The convergence of the force-displacement curve in terms of displacement increments is illustrated in Fig. 4. The vertical displacement, denoted by u_y , is monotonically increased with increments $\Delta u_y = \delta u \times \bar{u}$, with $\delta u = 0.1$ for $u_y < 0.7 \times \bar{u}$, and $\delta u = 0.0005$ for $u_y \geq 0.7 \times \bar{u}$, where $\bar{u}$ is a fixed remote displacement set to 0.007 mm . The vertical displacement increases progressively up to a maximum magnitude of 0.00714 mm .

Fig. 5 shows four pairs of stochastic and spatially varying fracture toughness and the corresponding damage fields. High variability is observed in the fracture pattern, due to the choice of the correlation length (which is five times smaller than the size of the domain) and marginal coefficient of variation for the toughness random field.

To determine the dimensions $d_{\mathcal{X}}$ and $d_{\mathcal{Y}}$ of the input and output latent spaces, convergence analyses were performed on the spectrum

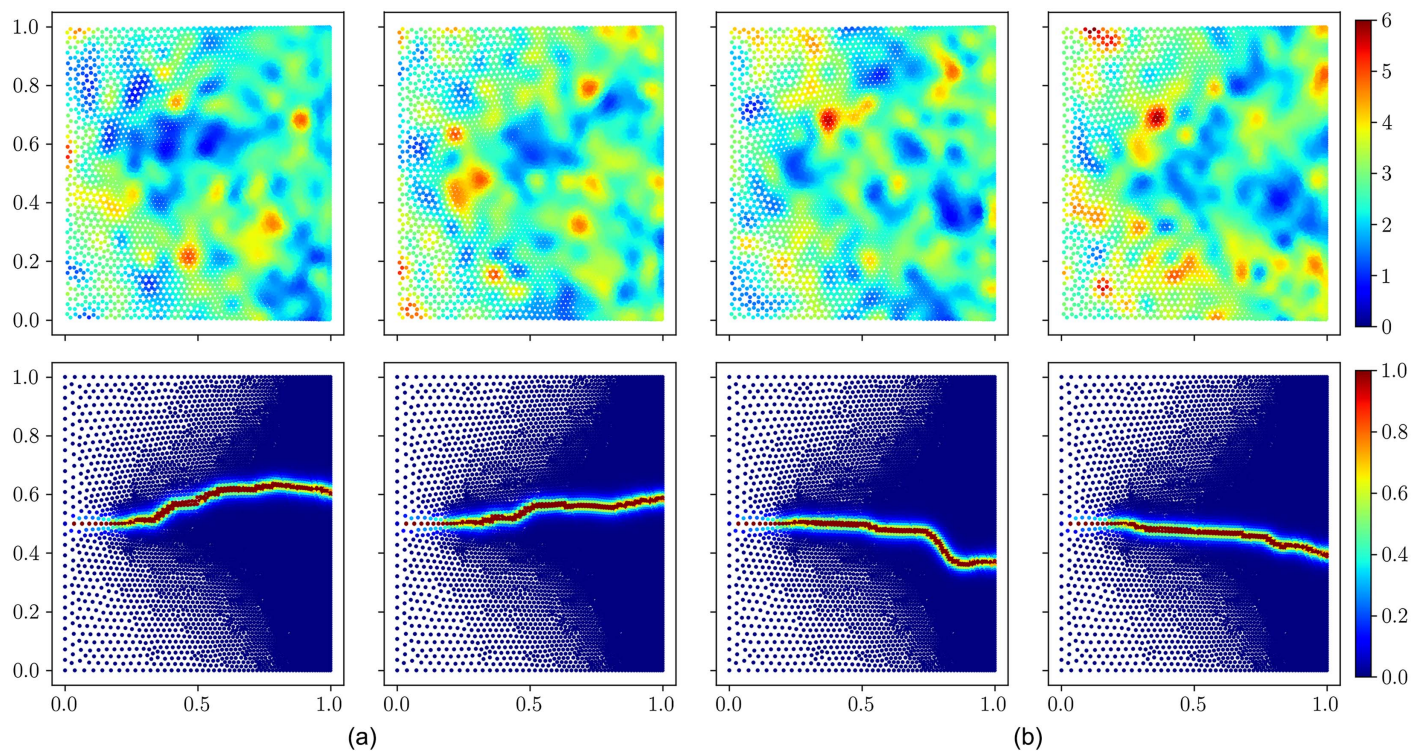


Fig. 5. (a) Realizations of the fracture toughness random field [in $\text{MPa} \cdot (\text{mm})^{-2}$]; and (b) associated realizations of the damage random fields.

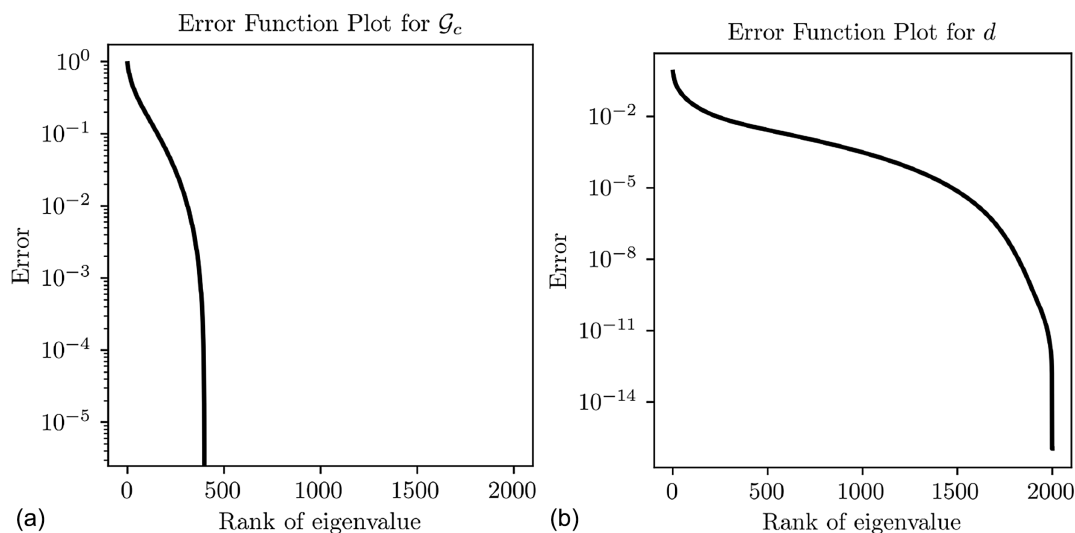


Fig. 6. Graph of the mean-squared error as a function of truncation order for (a) toughness random field; and (b) damage random field.

of the covariance operators for the toughness and damage random fields; see Fig. 6. Based on these results, and selecting the threshold for the errors to 0.01, reduced dimensions were found to be $d_{\chi} = 312$ and $d_{\gamma} = 232$. Fig. 7 provides a qualitative illustration of the impact of the reductions on the input and output fields.

The variability of the crack paths constituting the output in the data set can be seen in Fig. 8 where a set of 2,000 realizations is shown. Note that filtering was applied to enable visualization, with a threshold set to 0.9. These 2,000 samples constitute the data set that will be used in the next section to train (and validate, for the deep learning strategies) the operator learning methods.

Learning Results

In this section, we assess the performance of the learning methods on a new testing data set consisting of 200 realizations of η_{χ} . These samples are independent of the ones used for training the generative model and training and validating (with a standard held-out strategy) the deep learning models. Note that since the latent dimension d_{χ} is fairly large, the above realizations of η_{χ} are randomly selected within the 200,000 realizations generated by the PCA-GEM generative model (see below), to avoid extrapolation. Qualitative results are first presented for each approach, in the form of best and worst predicted paths as identified by the error defined as

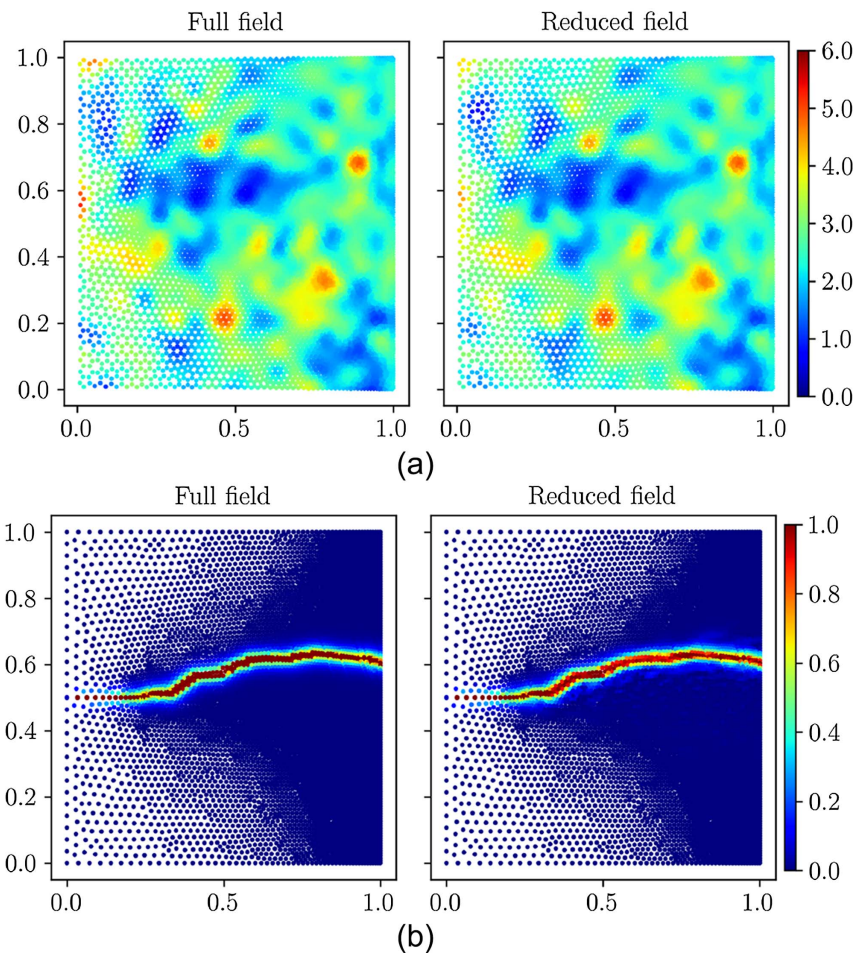


Fig. 7. Illustrative comparison of full-order and reduced-order representations for (a) input random field; and (b) output random field (for a threshold set to 0.01 for the truncation error).

$$\text{Err} = \frac{\|\tilde{\Psi} - \Psi\|_2}{\|\Psi\|_2} \quad (41)$$

Statistical results computed with the error metric are then presented in the section titled “Quantitative Comparison.” Additional

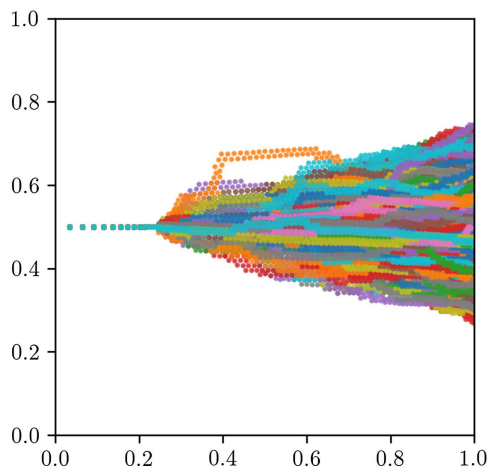


Fig. 8. Realizations of the output damage random field, used in the data set. Here, 2,000 realizations are considered, and a threshold $d > 0.9$ was used for visualization purposes.

examples of qualitative results are provided in the Appendix. In our experiments, we adopted the neural network architecture proposed in Bhattacharya et al. (2021) and Li et al. (2022). Hyperparameter optimization was conducted state-of-the-art Bayesian optimization, using the weights and biases (WANDB) platform (Biewald 2020). Regarding the operator learning methods, Matlab and Python implementations for the generative model can be found in Soize (2021) and Zhong et al. (2021), respectively. PCA-Net was implemented using the PyTorch framework [see Paszke et al. (2019)], and code for geo-FNO can be found in Zongyi et al. (2022).

PCA-GEM Results

To deploy the PCA-GEM strategy with the learning method as a generative model, the scale parameter ε in the kernel function given by Eq. (26) was determined using Lafon’s criterion [see Sec. 2.4 in Lafon (2004)]: $\varepsilon = 206.1960$. Regarding the determination of the dimension m , the graph of the eigenvalues for the transition matrix (Fig. 9) suggests that $m \geq 545$ yields a reasonable approximation. The value $m = 676$ is selected in the following, with a mean-square error in the covariance matrix equal to 0.006. The generator was deployed with $\gamma = 4$ and a time step $\Delta t = 0.2$. Here, 100 additional samples of the random matrix were generated, leading to a total of 200,000 additional samples for the latent variables η_X and η_Y .

Fig. 10 shows some output fields sampled with the PCA-GEM approach, together with the reference results obtained by direct numerical simulations (test data set). Good qualitative agreement

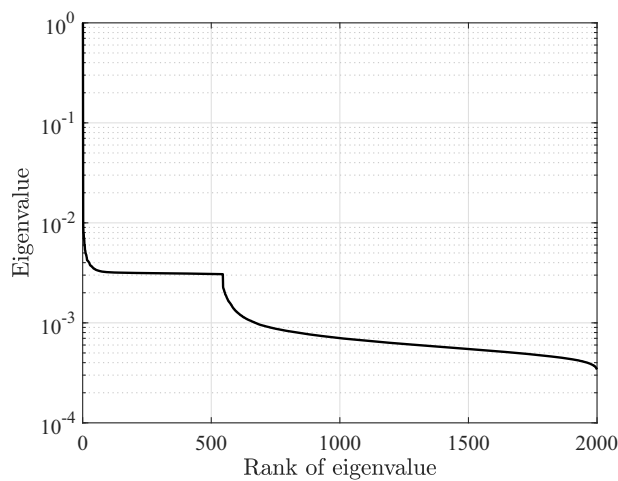


Fig. 9. Plot of the eigenvalues of the transition matrix (in log scale).

is observed in general. For the worst case prediction, the approximation predicts two fracture paths, one corresponding to the path simulated by the reference phase-field approach and another plausible path going through a region of lower fracture toughness.

Geo-FNO Results

Regarding the geo-FNO architecture, we employed 10 Fourier modes with three input channels where the first and second channels correspond to the coordinates of the input fracture toughness field, and the last channel is the fracture toughness. The first input fully connected layer has an output channel size (hidden layer size) of 36 to lift the three input channels to a higher channel dimension of 36. Subsequently, a width of 36 channels is utilized for all the

Fourier integral operators. The final output fully connected neural network does the projection back to the target dimension of 1, and has an input channel size of 36, and hidden layer size of 256. We used the same architecture of deformation neural network as the authors in Li et al. (2022), where the hidden layer sizes are multiples of a width parameter. For our application, the width parameter of the deformation network is 36, the same as that of the FNO operator.

The data set presented at the end of the previous section and containing 2,000 samples was used, with 80% of samples devoted to training and 20% used for validating. As previously mentioned, a new test data set consisting of 200 realizations of η_{χ} is used for comparing the learning methods. A batch size of 64, 100 epochs, and a maximum learning rate of 7.7×10^{-5} were utilized. The training process employed a OneCycleLR scheduler with the AdamW optimizer and a weight decay of 8.49×10^{-4} . Unlike the PCA-Net model, no single parameter holds a dominant influence over the validation loss during the training process. The top-performing and poorest predictions generated by the model are presented in Fig. 11 (test data set). Overall, the predictions are found to be accurate from a qualitative standpoint. While cracks appear to be more pixelated compared to those obtained with PCA-GEM and PCA-Net, geo-FNO exhibits slightly better accuracy in terms of fracture path predictions [in the sense of Eq. (41)].

PCA-Net Results

In the case of PCA-Net, a five-layer dense neural architecture was employed, with hidden layer sizes of 300, 600, 1,200, 600, and 300, respectively. The SELU activation function was utilized. The network was trained using the AdamW optimizer and a weight decay of 2.03×10^{-4} . The same data set as for the geo-FNO model was used (with 80% and 20% of samples used for training and validating, respectively, and 200 realizations of η_{χ} used for testing), and a batch size of 256 was utilized for a total of 100 epochs.

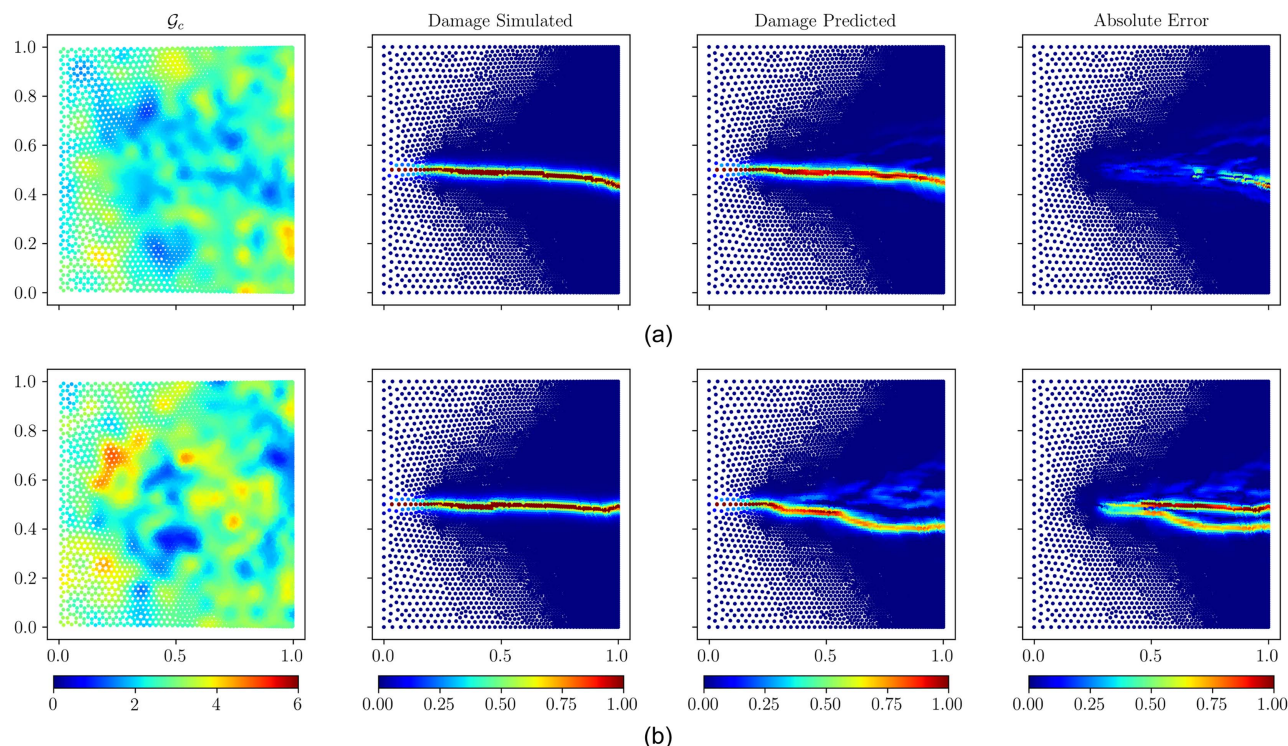


Fig. 10. Best [row (a) with L^2 error of 0.187] and worst [row (b) with L^2 error of 0.942] predictions obtained with PCA-GEM (test data set)

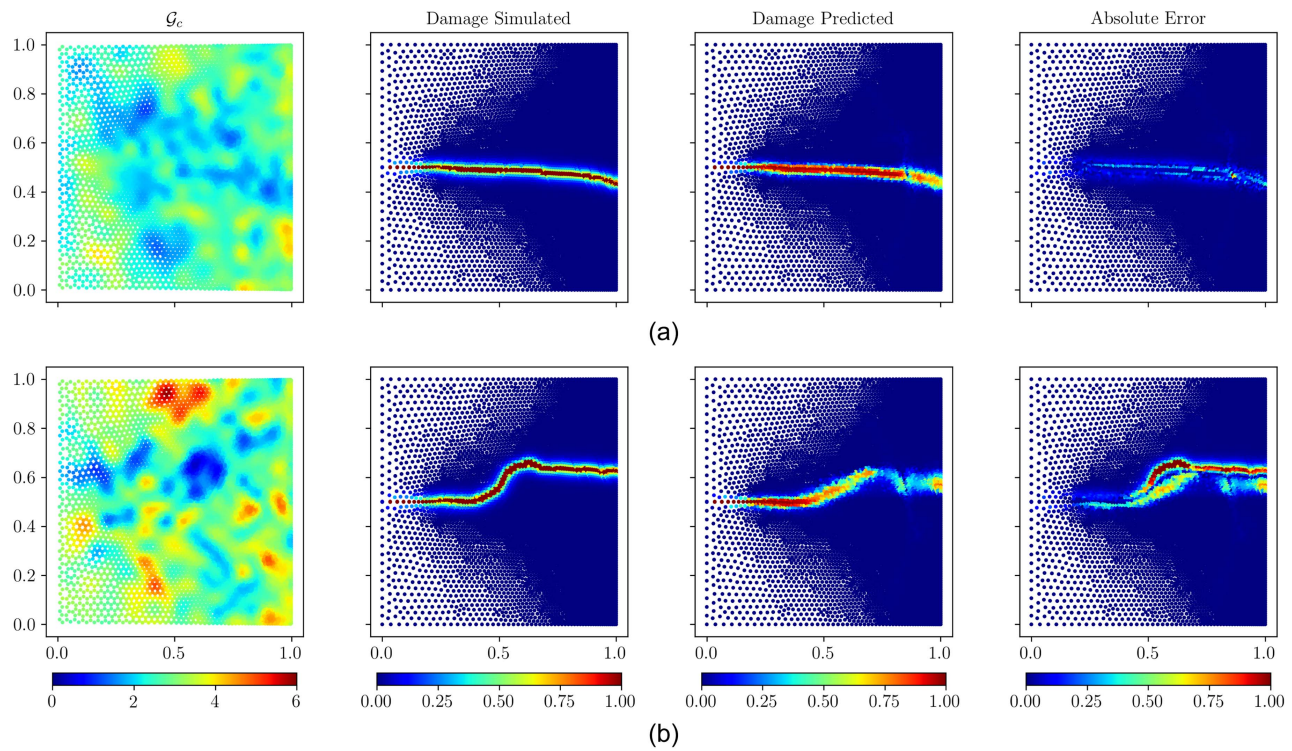


Fig. 11. Best [row (a) with L^2 error of 0.216] and worst [row (b) with L^2 error of 0.829] predictions obtained with geo-FNO (test data set).

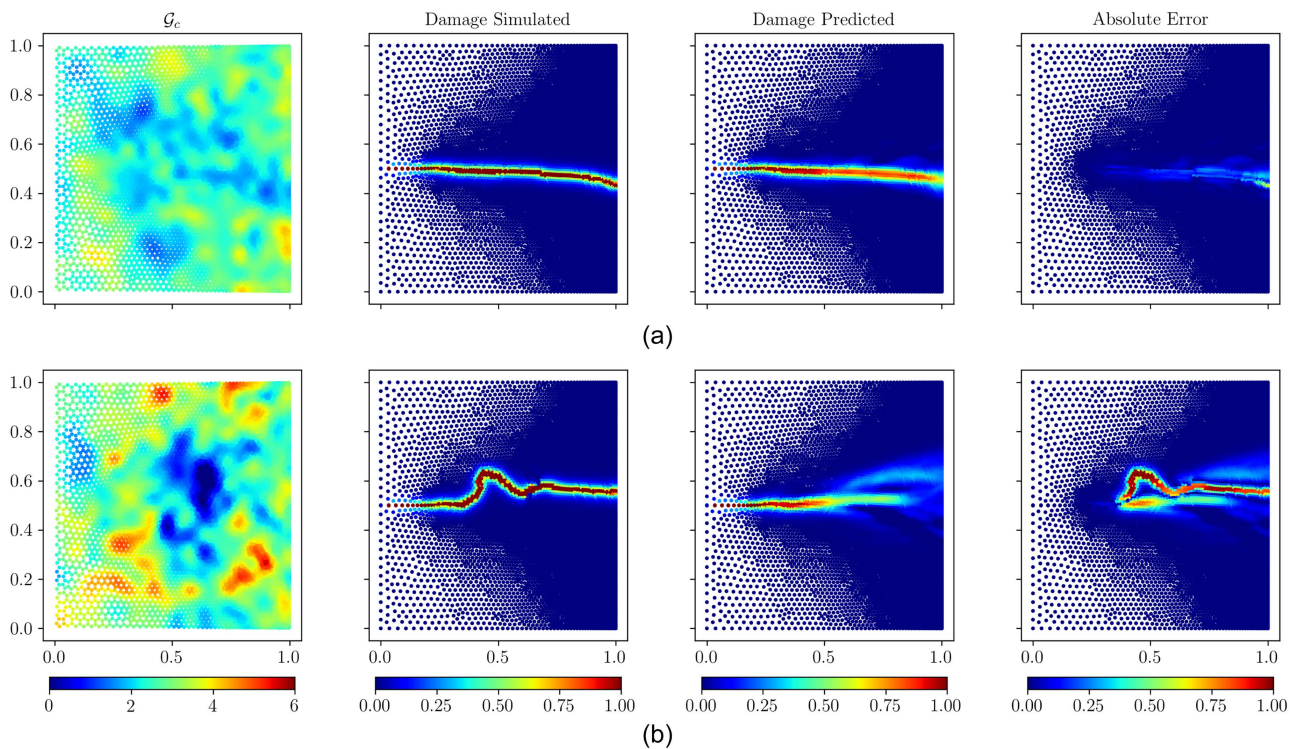


Fig. 12. Best [row (a) with L^2 error of 0.203] and worst [row (b) with L^2 error of 0.862] predictions obtained with PCA-Net.

A OneCycleLR scheduler was implemented, setting the maximum learning rate to 8.89×10^{-3} . It was observed that the width and depth of the neural network had minimal impact on the training loss, whereas the learning rate played a more crucial role—a higher

learning rate resulting in better model performance. Fig. 12 provides a visual representation of the best and worst predictions generated by the model. We found that the majority of the predicted paths exhibit more pronounced diffusivity.

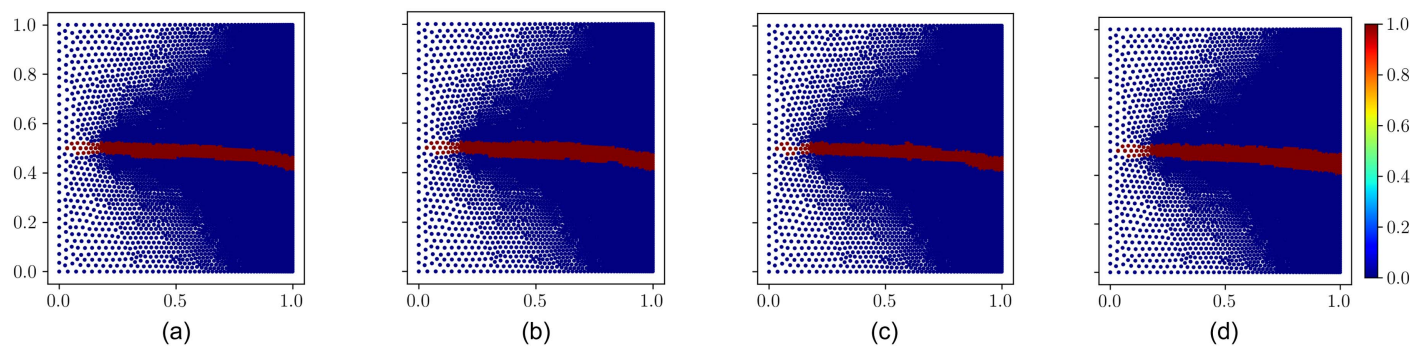


Fig. 13. Binarized damage predictions: (a) reference; (b) PCA-GEM; (c) geo-FNO; and (d) PCA-Net predictions.

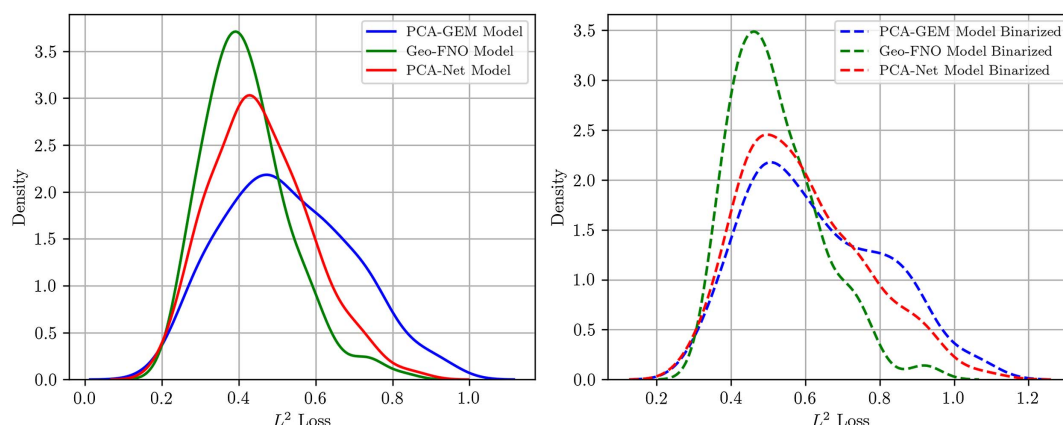


Fig. 14. Model performance evaluation for damage prediction: kernel density plot of L^2 error for PCA-GEM, geo-FNO, and PCA models with testing set of 200 samples.

Quantitative Comparison

Due to the smeared representation introduced by the phase-field approach, output fields may be binarized to represent any nonheal-
thy state in the material. Here, we consider a threshold of 0.2, im-
plying that damage values smaller than or equal to 0.2 are set to 0,
and greater values are set to 1. Examples of binarized representa-
tions are shown in Fig. 13.

Fig. 14 illustrates the distribution of the L^2 error [defined in
Eq. (41)] of nonbinarized and binarized results across the three
models, estimated with a kernel density estimation with 200 inde-
pendent samples. The mean and coefficient of variation of the L^2
errors for the three learning strategies are reported in Table 1, con-
sidering both the binarized and nonbinarized output fields.

Table 1. Mean and coefficient of variation (CoV) of the L^2 error [see
Eq. (41)] obtained with nonbinarized and binarized output fields

Model	Mean of L^2 error (nonbinarized)	CoV of L^2 error (nonbinarized)	Mean of L^2 error (binarized)	CoV of L^2 error (binarized)
PCA-GEM	0.525	0.314	0.626	0.287
geo-FNO	0.421	0.264	0.521	0.231
PCA-Net	0.458	0.279	0.592	0.277

It is seen that the geo-FNO model achieves the lowest mean and
CoV for the loss in all cases, indicating better performance on the
testing data set overall. For binarized representations, PCA-GEM
and PCA-Net perform with similar accuracy in terms of both mean
and CoV, with a relative error in mean of about 17% with respect to
geo-FNO. PCA-Net exhibits slightly better accuracy than PCA-
GEM for nonbinarized models. The offline training time for 100
epochs is 51.53 min for the geo-FNO model, whereas the PCA-Net
model it takes 22.37 min (on an NVIDIA RTX A6000 GPU). Note
that the errors shown in Fig. 14 (and in Table 1) are much larger
than the ones typically reported in operator learning studies. This is
because the output field, while diffused, remains highly localized
(in stark contrast with results involving maps between smooth
fields), so that any deviation from the reference crack path contrib-
utes quite significantly to the error—whether it be computed point-
wise in the sample space, as in Eq. (41), or in mean [Eq. (36)].
Indeed, the chosen L^2 metric evaluates not only accuracy in the pre-
diction of the crack path, but also the capability of the method(s) to
enforce that $d \approx 1$ along the central line of the diffused crack path.
While all techniques perform reasonably well for the former, they
struggle to maintain constant values for the damage field—a feature
that is extremely challenging for the encoding–decoding strategy (in
a function space setting). This is exemplified in the figures below.
Specifically, Fig. 15 demonstrates that the crack path contour is
predicted with fairly good accuracy (for the considered randomly
selected sample), and Fig. 16 illustrates that the operator learning

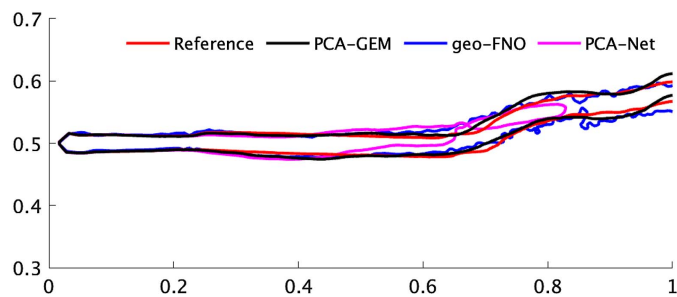


Fig. 15. Contour plot for the reference solution and the operator learning approximations (for a level set to 0.5).

method does not capture the peak of the damage profile very well, hence contributing to the increase of the L^2 error (similar results are obtained with geo-FNO and PCA-Net). The development of *ad hoc* representations ensuring that damage peaks are predicted more accurately is left for future work.

The training and validation losses for the neural networks are shown in Fig. 17 (for geo-FNO and PCA-Net). It is seen that

PCA-Net tends to converge faster during training, but exhibits slightly slower convergence and higher loss on the validation data set.

The above results demonstrate the ability of PCA-GEM to deliver predictions with an accuracy that is comparable to that of some state-of-the-art deep learning techniques. [Note that an exhaustive comparison with other operator learning frameworks is outside the scope of this work, since (1) there exists a myriad of variations on such frameworks, some of which are still being developed or refined; and (2) the definition of methodologies to fairly compare different classes of methods remains an open problem.] While this can be achieved without the burden of architecture optimization, a limitation of this method lies in potentially poor generalization capabilities, due to the use of conditional expectations in a very high dimensional space. Extensions to overcome this issue constitute an interesting topic for future research.

Conclusion

In this work, we considered the construction of approximation frameworks for phase-field models of brittle fracture. Two classes of methods were presented. In the first class, deep learning models

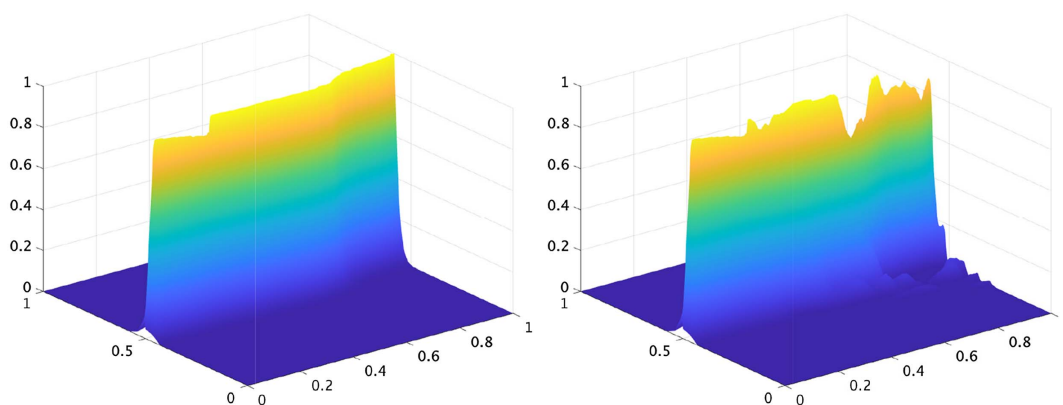


Fig. 16. 3D plots of the reference solution (left) and PCA-GEM prediction (right). Results obtained with geo-FNO and PCA-Net are qualitatively similar.

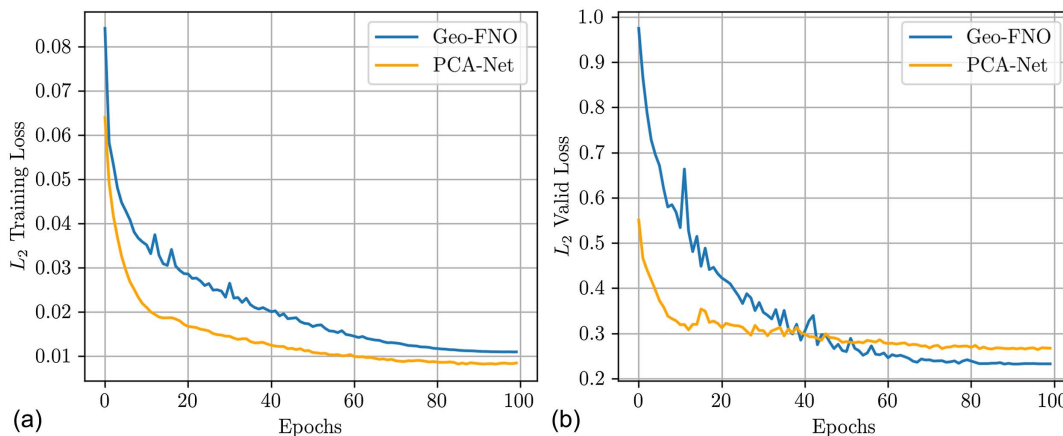


Fig. 17. Comparison of (a) train; and (b) valid losses between geo-FNO and PCA-Net. Losses as a function of epoch are computed according to Eq. (36).

are used to perform *regression* in ad hoc latent spaces, defined through linear and nonlinear reduction techniques. State-of-the-art approaches, namely PCA-Net and Fourier neural operators, are briefly reviewed. We then proposed a second class of approaches where *conditional expectation* is used to approximate the forward map in latent space. To enable proper estimation, the data set is augmented using probabilistic learning on manifold—a Hamiltonian Monte Carlo technique that ensures measure concentration through model reduction. The accuracy of these methods was subsequently investigated on fracture simulations involving a spatially varying stochastic toughness, modeled as a non-Gaussian random field. It was shown that the probabilistic approach achieves comparable performance in the L^2 sense and therefore offers a valuable alternative to deep learning models. Directions for future research include the

analysis of generalization capabilities and the development of representations capturing damage peaks with greater accuracy.

Appendix. Additional Numerical Results

Here, we present additional qualitative examples to illustrate the predictive accuracy of the learning frameworks. Specifically, we showcase three additional instances of improved predictions and three examples of poorer predictions for each model, aiming to offer a more holistic presentation of models' performance. Results obtained using PCA-GEM, geo-FNO, and PCA-Net are provided in Figs. 18–20, respectively. Note that the trends observed in these samples must not be generalized, and that these results should not

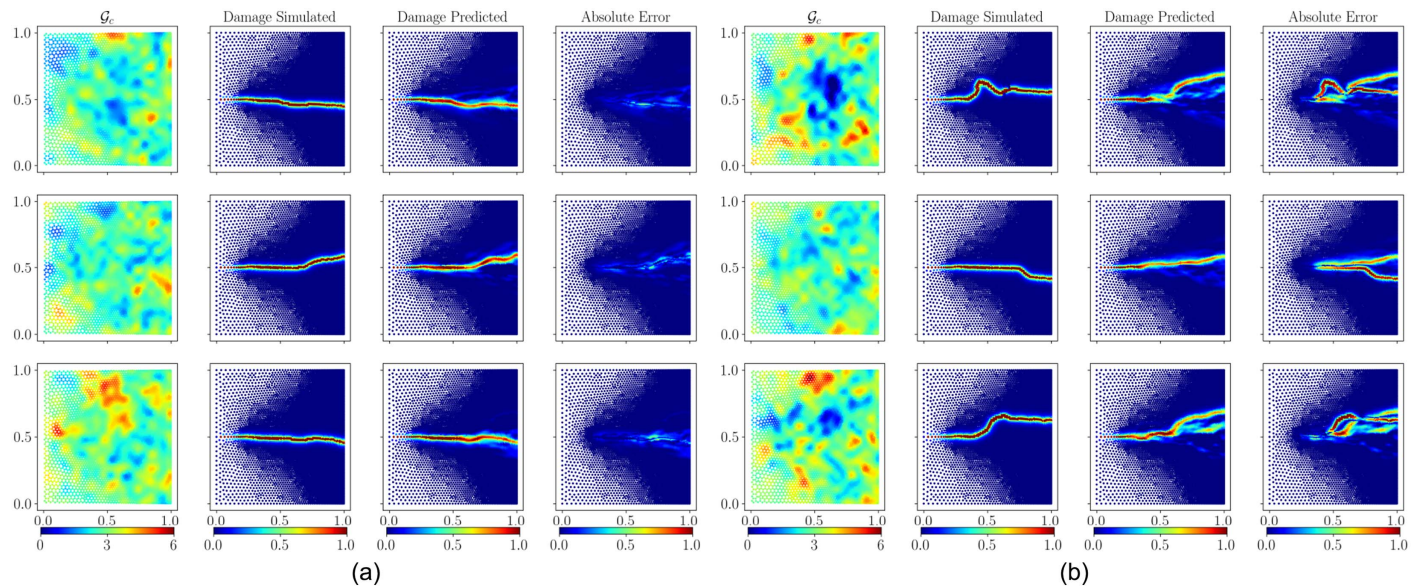


Fig. 18. Additional examples of predictions by PCA-GEM: (a) predictions with higher accuracy, with errors $L^2 = [0.226, 0.231, 0.256]$ from top to bottom; and (b) predictions with lower accuracy, with errors $L^2 = [0.929, 0.917, 0.901]$ from top to bottom.

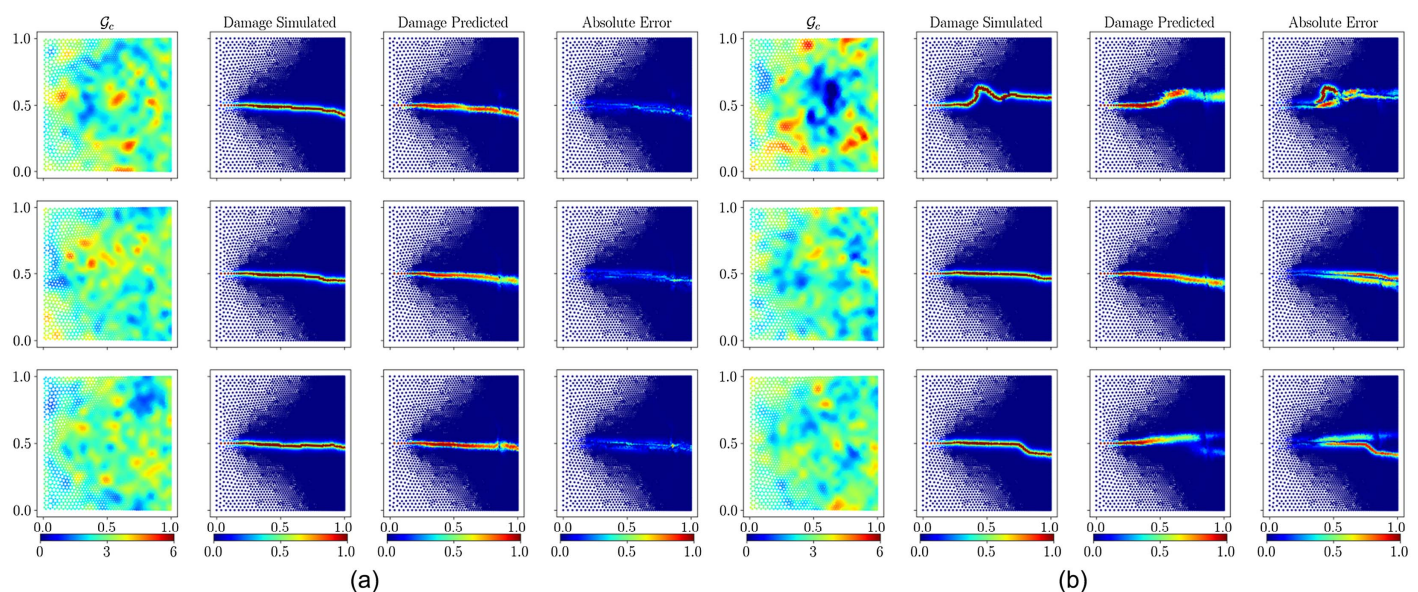


Fig. 19. Additional examples of predictions by geo-FNO: (a) predictions with higher accuracy, with errors $L^2 = [0.228, 0.235, 0.241]$ from top to bottom; and (b) predictions with lower accuracy, with errors $L^2 = [0.765, 0.742, 0.735]$ from top to bottom.

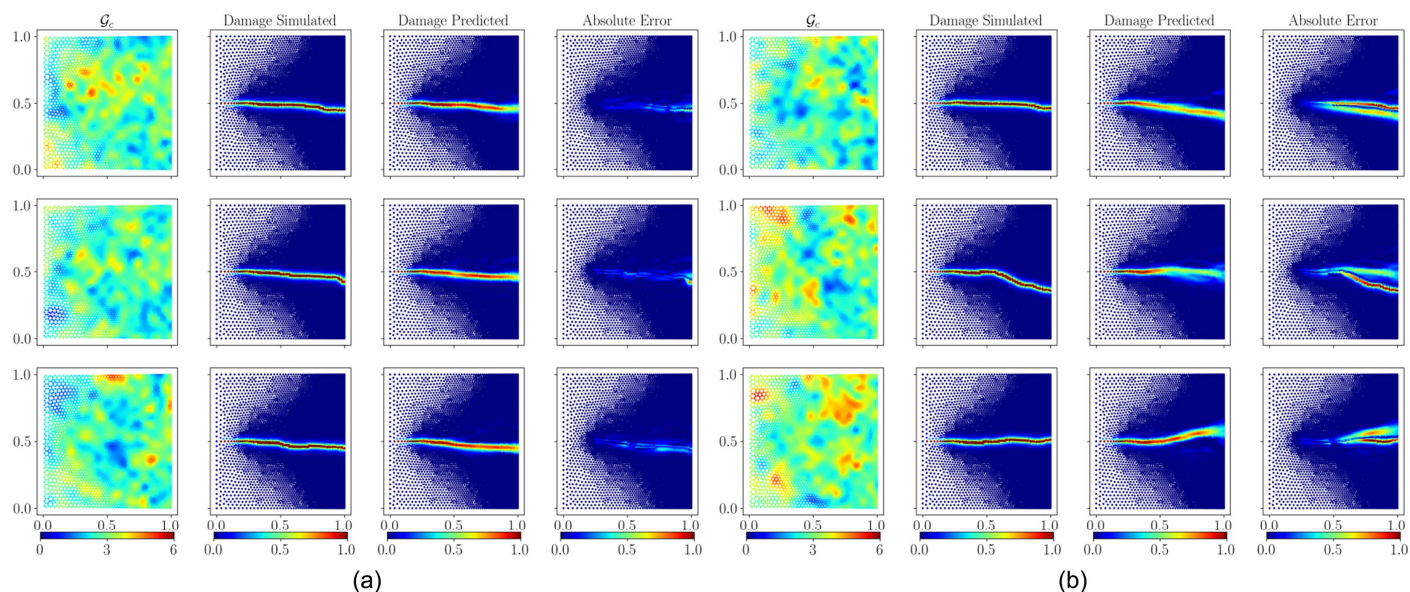


Fig. 20. Additional examples of predictions by PCA-Net: (a) predictions with higher accuracy, with errors $L^2 = [0.217, 0.223, 0.240]$ from top to bottom; and (b) predictions with lower accuracy, with errors $L^2 = [0.821, 0.761, 0.754]$ from top to bottom.

be used to compare the performance of the models against one another.

Data Availability Statement

Some or all data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments

Financial support of the National Science Foundation (NSF) under Grant DGE-2022040 is gratefully acknowledged. The authors thank Prof. John Dolbow for useful discussions regarding the phase-field method and its implementation.

References

- Adler, J., and O. Öktem. 2017. "Solving ill-posed inverse problems using iterative deep neural networks." *Inverse Prob.* 33 (12): 124007. <https://doi.org/10.1088/1361-6420/aa9581>.
- Ambati, M., T. Gerasimov, and L. De Lorenzis. 2014. "A review on phase-field models of brittle fracture and a new fast hybrid formulation." *Comput. Mech.* 55 (Feb): 383–405. <https://doi.org/10.1007/s00466-014-1109-y>.
- Ambrosio, L., and V. M. Tortorelli. 1990. "Approximation of functional depending on jumps by elliptic functional via t-convergence." *Commun. Pure Appl. Math.* 43 (8): 999–1036. <https://doi.org/10.1002/cpa.3160430805>.
- Battle, P., M. Darcy, B. Hosseini, and H. Owghadi. 2023. "Kernel methods are competitive for operator learning." Preprint, submitted April 26, 2023. <https://arxiv.org/abs/1504.06600v1>.
- Benner, P., P. Goyal, B. Kramer, B. Peherstorfer, and K. Willcox. 2020. "Operator inference for non-intrusive model reduction of systems with non-polynomial nonlinear terms." *Comput. Methods Appl. Mech. Eng.* 372 (Dec): 113433. <https://doi.org/10.1016/j.cma.2020.113433>.
- Bhatnagar, S., Y. Afshar, S. Pan, K. Duraisamy, and S. Kaushik. 2019. "Prediction of aerodynamic flow fields using convolutional neural networks." *Comput. Mech.* 64 (2): 525–545. <https://doi.org/10.1007/s00466-019-01740-0>.
- Bhattacharya, K., B. Hosseini, N. B. Kovachki, and A. M. Stuart. 2021. "Model reduction and neural networks for parametric PDEs." *SMAI J. Comput. Math.* 7 (Jun): 121–157. <https://doi.org/10.5802/smai-jcm.74>.
- Biewald, L. 2020. "Experiment tracking with weights and biases." Accessed March 16, 2023. <https://www.wandb.com/>.
- Borden, M. J., T. J. Hughes, C. M. Landis, and C. V. Verhoosel. 2014. "A higher-order phase-field model for brittle fracture: Formulation and analysis within the isogeometric analysis framework." *Comput. Methods Appl. Mech. Eng.* 273 (May): 100–118. <https://doi.org/10.1016/j.cma.2014.01.016>.
- Borden, M. J., C. V. Verhoosel, M. A. Scott, T. J. Hughes, and C. M. Landis. 2012. "A phase-field description of dynamic brittle fracture." *Comput. Methods Appl. Mech. Eng.* 217–220 (Apr): 77–95. <https://doi.org/10.1016/j.cma.2012.01.008>.
- Bourdin, B. 2007. "Numerical implementation of the variational formulation for quasi-static brittle fracture." *Interfaces Free Boundaries* 9 (3): 411–430. <https://doi.org/10.4171/ifb/171>.
- Bourdin, B., G. Francfort, and J.-J. Marigo. 2000. "Numerical experiments in revisited brittle fracture." *J. Mech. Phys. Solids* 48 (4): 797–826. [https://doi.org/10.1016/S0022-5096\(99\)00028-9](https://doi.org/10.1016/S0022-5096(99)00028-9).
- Bourdin, B., G. A. Francfort, and J.-J. Marigo. 2008. "The variational approach to fracture." *J. Elast.* 91 (Apr): 5–148. <https://doi.org/10.1007/s10659-007-9107-3>.
- Clément, A. J.-B. 2000. "Revisiting brittle fracture as an energy minimization problem: Comparisons of grith and barenblatt surface energy models." *J. Mech. Phys. Solids* 46 (8): 1319–1342. [https://doi.org/10.1016/S0022-5096\(98\)00034-9](https://doi.org/10.1016/S0022-5096(98)00034-9).
- Coifman, R. R., and S. Lafon. 2006. "Diffusion maps." *Appl. Comput. Harmon. Anal.* 21 (1): 5–30. <https://doi.org/10.1016/j.acha.2006.04.006>.
- Coifman, R. R., S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. 2005. "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps." *Proc. Natl. Acad. Sci.* 102 (21): 7426–7431. <https://doi.org/10.1073/pnas.0500334102>.
- de Hoop, M. V., D. Z. Huang, E. Qian, and A. M. Stuart. 2022. "The cost-accuracy trade-off in operator learning with neural networks." Preprint, submitted March 24, 2022. <https://arxiv.org/abs/2203.13181>.
- Dockhorn, T. 2019. "A discussion on solving partial differential equations using neural networks." Preprint, submitted January 10, 2023. <https://arxiv.org/abs/1904.07200>.

- Eastgate, L. O., J. P. Sethna, M. Rauscher, T. Cretegnny, C.-S. Chen, and C. R. Myers. 2001. "Fracture in mode I using a conserved phase-field model." *Phys. Rev. E Stat. Nonlinear Soft Matter. Phys.* 65 (3): 036117. <https://doi.org/10.1103/PhysRevE.65.036117>.
- Fanaskov, V., and I. Oseledets. 2022. "Spectral neural operators." Preprint, submitted May 21, 2022. <https://arxiv.org/abs/2205.10573>.
- Francfort, G. A., and J.-J. Marigo. 1998. "Revisiting brittle fracture as an energy minimization problem." *J. Mech. Phys. Solids* 46 (8): 1319–1342. [https://doi.org/10.1016/S0022-5096\(98\)00034-9](https://doi.org/10.1016/S0022-5096(98)00034-9).
- Geist, M., P. Petersen, M. Raslan, R. Schneider, and G. Kutyniok. 2020. "Numerical solution of the parametric diffusion equation by deep neural networks." Preprint, submitted April 25, 2020. <https://arxiv.org/abs/2004.12131>.
- Ghanem, R., D. Higdon, and H. Owadi. 2017. *Handbook of uncertainty quantification*. Berlin: Springer.
- Ghanem, R., and C. Soize. 2018. "Probabilistic nonconvex constrained optimization with fixed number of function evaluations." *Int. J. Numer. Methods Eng.* 113 (4): 719–741. <https://doi.org/10.1002/nme.5632>.
- Ghanem, R., C. Soize, V. Aithataju, and L. Mehrez. 2021. "Probabilistic learning on manifolds for prognosis and characterization of the digital twin." In *Proc., Mechanistic Machine Learning and Digital Twins for Computational Science, Engineering and Technology*. Barcelona, Spain: International Association for Computational Mechanics.
- Goswami, S., M. Yin, Y. Yu, and G. E. Karniadakis. 2021. "A physics-informed variational deepoNet for predicting the crack path in brittle materials." Preprint, submitted August 16, 2021. <https://arxiv.org/abs/2108.06905>.
- Grigoriu, M. 1984. "Crossings of non-gaussian translation processes." *J. Eng. Mech.* 110 (4): 610–620. [https://doi.org/10.1061/\(ASCE\)0733-9399\(1984\)110:4\(610\)](https://doi.org/10.1061/(ASCE)0733-9399(1984)110:4(610)).
- Hesthaven, J., and S. Ubbiali. 2018. "Non-intrusive reduced order modeling of nonlinear problems using neural networks." *J. Comput. Phys.* 363 (Jun): 55–78. <https://doi.org/10.1016/j.jcp.2018.02.037>.
- Hirshikesh, H., S. Natarajan, R. K. Annabattula, and E. Martínez Pañeda. 2019. "Phase field modelling of crack propagation in functionally graded materials." *Composites, Part B* 169 (Jul): 239–248. <https://doi.org/10.1016/j.compositesb.2019.04.003>.
- Hofacker, M., and C. Miehe. 2013. "A phase field model of dynamic fracture: Robust field updates for the analysis of complex crack patterns." *Int. J. Numer. Methods Eng.* 93 (3): 276–301. <https://doi.org/10.1002/nme.4387>.
- Holland, J. R., J. D. Baeder, and K. Duraisamy. 2019. "Field inversion and machine learning with embedded neural networks: Physics-consistent neural network training." In *Proc., AIAA Aviation 2019 Forum*. Reston, VA: American Institute of Aeronautics and Astronautics.
- Hsieh, J.-T., S. Zhao, S. Eismann, L. Mirabella, and S. Ermon. 2019. "Learning neural PDE solvers with convergence guarantees." Preprint, submitted June 4, 2019. <https://arxiv.org/abs/1906.01200>.
- Hu, T., J. Guillemot, and J. Dolbow. 2020a. "A phase-field model of fracture with frictionless contact and random fracture properties: Application to thin-film fracture and soil desiccation." *Comput. Methods Appl. Mech. Eng.* 368 (Aug): 113106. <https://doi.org/10.1016/j.cma.2020.113106>.
- Hu, T., J. Guillemot, and J. E. Dolbow. 2020b. "A phase-field model of fracture with frictionless contact and random fracture properties: Application to thin-film fracture and soil desiccation." *Comput. Methods Appl. Mech. Eng.* 368 (Aug): 113106. <https://doi.org/10.1016/j.cma.2020.113106>.
- Hun, D.-A., J. Guillemot, J. Yvonnet, and M. Bornert. 2019. "Stochastic multiscale modeling of crack propagation in random heterogeneous media." *Int. J. Numer. Methods Eng.* 119 (13): 1325–1344. <https://doi.org/10.1002/nme.6093>.
- Jaynes, E. T. 1957a. "Information theory and statistical mechanics." *Phys. Rev.* 106 (4): 620–630. <https://doi.org/10.1103/PhysRev.106.620>.
- Jaynes, E. T. 1957b. "Information theory and statistical mechanics. II." *Phys. Rev.* 108 (2): 171–190. <https://doi.org/10.1103/PhysRev.108.171>.
- Kovachki, N., Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar. 2023. "Neural operator: Learning maps between function spaces with applications to PDEs." *J. Mach. Learn. Res.* 24 (89): 1–97.
- Kuhn, C., and R. Müller. 2010. "A continuum phase field model for fracture." *Eng. Fract. Mech.* 77 (18): 3625–3634. <https://doi.org/10.1016/j.engfractmech.2010.08.009>.
- Lafon, S. S. 2004. "Diffusion maps and geometric harmonics." Ph.D. thesis, Dept. of Mathematics, Yale Univ.
- Lagaris, I., A. Likas, and D. Fotiadis. 1998. "Artificial neural networks for solving ordinary and partial differential equations." *IEEE Trans. Neural Netw.* 9 (5): 987–1000. <https://doi.org/10.1109/72.712178>.
- Li, Z., D. Z. Huang, B. Liu, and A. Anandkumar. 2022. "Fourier neural operator with learned deformations for PDEs on general geometries." *J. Mach. Learn. Res.* 24 (388): 1–26.
- Li, Z., N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. 2020a. "Multipole graph neural operator for parametric partial differential equations." *Adv. Neural Inf. Process. Syst.* 33 (Jun): 6755–6766.
- Li, Z., N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. 2020b. "Neural operator: Graph kernel network for partial differential equations." Preprint, submitted March 7, 2020. <http://arxiv.org/abs/2003.03485>.
- Li, Z., N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. 2021a. "Fourier neural operator for parametric partial differential equations." Preprint, submitted October 18, 2020. <https://arxiv.org/abs/2010.08895>.
- Li, Z., H. Zheng, N. Kovachki, D. Jin, H. Chen, B. Liu, K. Azizzadenesheli, and A. Anandkumar. 2021b. "Physics-informed neural operator for learning partial differential equations." Preprint, submitted November 6, 2021. <https://arxiv.org/abs/2111.03794>.
- Lu, L., P. Jin, G. Pang, Z. Zhang, and G. Karniadakis. 2021. "Learning nonlinear operators via deepoNet based on the universal approximation theorem of operators." *Nat. Mach. Intell.* 3 (3): 218–229. <https://doi.org/10.1038/s42256-021-00302-5>.
- Lu, L., P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis. 2019. "Learning nonlinear operators via deepoNet based on the universal approximation theorem of operators." Preprint, submitted October 8, 2019. <http://arxiv.org/abs/1910.03193>.
- Lu, L., X. Meng, S. Cai, Z. Mao, S. Goswami, Z. Zhang, and G. E. Karniadakis. 2022. "A comprehensive and fair comparison of two neural operators (with practical extensions) based on FAIR data." *Comput. Methods Appl. Mech. Eng.* 393 (2): 114778. <https://doi.org/10.1016/j.cma.2022.114778>.
- McQuarrie, S. A., C. Huang, and K. E. Willcox. 2021. "Data-driven reduced-order models via regularised operator inference for a single-injector combustion process." *J. R. Soc. New Zealand* 51 (2): 194–211. <https://doi.org/10.1080/03036758.2020.1863237>.
- Miehe, C., M. Hofacker, and F. Welschinger. 2010. "A phase field model for rate-independent crack propagation: Robust algorithmic implementation based on operator splits." *Comput. Methods Appl. Mech. Eng.* 199 (45–48): 2765–2778. <https://doi.org/10.1016/j.cma.2010.04.011>.
- Mohammadzadeh, S., and E. Lejeune. 2021. "Predicting mechanically driven full-field quantities of interest with deep learning-based metamodelling." *Extreme Mech. Lett.* 50 (Aug): 101566. <https://doi.org/10.1016/j.eml.2021.101566>.
- Nguyen, T., J. Yvonnet, M. Bornert, C. Chateau, K. Sab, R. Romani, and R. Le Roy. 2016. "On the choice of parameters in the phase field method for simulating crack initiation with experimental validation." *Int. J. Fract.* 197 (2): 213–226. <https://doi.org/10.1007/s10704-016-0082-1>.
- Nguyen, T., J. Yvonnet, Q.-Z. Zhu, M. Bornert, and C. Chateau. 2015. "A phase field method to simulate crack nucleation and propagation in strongly heterogeneous materials from direct imaging of their microstructure." *Eng. Fract. Mech.* 139 (May): 18–39. <https://doi.org/10.1016/j.engfractmech.2015.03.045>.
- Paszke, A., et al. 2019. "PyTorch: An imperative style, high-performance deep learning library." Preprint, submitted December 3, 2019. <https://arxiv.org/abs/1912.01703>.
- Peherstorfer, B. 2019. "Sampling low-dimensional markovian dynamics for pre-asymptotically recovering reduced models from data with operator inference." Preprint, submitted August 29, 2019. <https://arxiv.org/abs/1908.11233>.

- Peherstorfer, B., and K. Willcox. 2016. "Data-driven operator inference for nonintrusive projection-based model reduction." *Comput. Methods Appl. Mech. Eng.* 306 (Jul): 196–215. <https://doi.org/10.1016/j.cma.2016.03.025>.
- Qian, E., B. Kramer, B. Peherstorfer, and K. Willcox. 2020. "Lift & learn: Physics-informed machine learning for large-scale nonlinear dynamical systems." *Physica D* 406 (May): 132401. <https://doi.org/10.1016/j.physd.2020.132401>.
- Raissi, M., P. Perdikaris, and G. Karniadakis. 2019. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations." *J. Comput. Phys.* 378 (5): 686–707. <https://doi.org/10.1016/j.jcp.2018.10.045>.
- Ramsundar, B., D. Krishnamurthy, and V. Viswanathan. 2021. "Differentiable physics: A position piece." Preprint, submitted September 14, 2021. <https://arxiv.org/abs/2109.07573>.
- Seidman, J. H., G. Kissas, G. J. Pappas, and P. Perdikaris. 2023. "Variational autoencoding neural operators." Preprint, submitted February 20, 2023. <https://arxiv.org/abs/2302.10351>.
- Sepasdar, R., A. Karpatne, and M. Shakiba. 2022. "A data-driven approach to full-field nonlinear stress distribution and failure pattern prediction in composites using deep learning." *Comput. Methods Appl. Mech. Eng.* 397 (Jul): 115126. <https://doi.org/10.1016/j.cma.2022.115126>.
- Shankar, V., V. Puri, R. Balakrishnan, R. Maulik, and V. Viswanathan. 2023. "Differentiable physics-enabled closure modeling for burgers' turbulence." *Mach. Learn. Sci. Technol.* 4 (1): 015017. <https://doi.org/10.1088/2632-2153/acb19c>.
- Shannon, C. E. 1948. "A mathematical theory of communication." *Bell Syst. Tech. J.* 27 (3): 379–423. <https://doi.org/10.1002/j.1538-7305.1948.tb01338.x>.
- Shin, Y. 2020. "On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs." *Commun. Comput. Phys.* 28 (5): 2042–2074. <https://doi.org/10.4208/cicp.OA-2020-0193>.
- Soize, C. 2015. "Polynomial chaos expansion of a multimodal random vector." *SIAM/ASA J. Uncertainty Quantif.* 3 (1): 34–60. <https://doi.org/10.1137/140968495>.
- Soize, C. 2021. "PLoM softwares." Accessed February 17, 2023. <https://pagespro.univ-gustave-eiffel.fr/christian-soize>.
- Soize, C. 2022. "Probabilistic learning constrained by realizations using a weak formulation of fourier transform of probability measures." *Comput. Stat.* 38 (4): 1879–1925. <https://doi.org/10.1007/s00180-022-01300-w>.
- Soize, C., and R. Ghanem. 2016. "Data-driven probability concentration and sampling on manifold." *J. Comput. Phys.* 321 (Sep): 242–258. <https://doi.org/10.1016/j.jcp.2016.05.044>.
- Soize, C., and R. Ghanem. 2020a. "Probabilistic learning on manifolds." *Found. Data Sci.* 2 (3): 279–307. <https://doi.org/10.3934/fods.2020013>.
- Soize, C., and R. Ghanem. 2022. "Probabilistic learning on manifolds (PLoM) with partition." *Int. J. Numer. Methods Eng.* 123 (1): 268–290. <https://doi.org/10.1002/nme.6856>.
- Soize, C., and R. G. Ghanem. 2020b. "Probabilistic learning on manifolds constrained by nonlinear partial differential equations for small datasets." Preprint, submitted October 27, 2020. <http://arxiv.org/abs/2010.14324>.
- Wang, Q., J. S. Hesthaven, and D. Ray. 2019. "Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem." *J. Comput. Phys.* 384 (Jun): 289–307. <https://doi.org/10.1016/j.jcp.2019.01.031>.
- Weinan, E., and B. Yu. 2017. "The deep ritz method: A deep learning-based numerical algorithm for solving variational problems." Preprint, submitted September 30, 2017. <https://arxiv.org/abs/1710.00211>.
- You, H., Y. Yu, M. D'Elia, T. Gao, and S. Silling. 2022. "Nonlocal kernel network (NKN): A stable and resolution-independent deep neural network." *J. Comput. Phys.* 469 (Jun): 111536. <https://doi.org/10.1016/j.jcp.2022.111536>.
- Zhong, K., J. Gual, and S. Govindjee. 2021. "PLoM python package." Accessed February 22, 2023. <https://github.com/sanjayg0/PLoM>.
- Zhu, Y., and N. Zabaras. 2018. "Bayesian deep convolutional encoder-decoder networks for surrogate modeling and uncertainty quantification." *J. Comput. Phys.* 366 (Jun): 415–447. <https://doi.org/10.1016/j.jcp.2018.04.018>.
- Zongyi, L., H. Zhengyu, and V. Vansh. 2022. "Geometry-aware Fourier neural operator (Geo-FNO)." Accessed February 2, 2023. <https://github.com/neuraloperator/Geo-FNO>.