



**ARTICLE**

# Reduced Order Machine Learning Finite Element Methods: Concept, Implementation, and Future Applications

Dedicated to Professor Karl Stark Pister for his 95th birthday

Ye Lu<sup>1</sup>, Hengyang Li<sup>1</sup>, Sourav Saha<sup>2</sup>, Satyajit Mojumder<sup>2</sup>, Abdullah Al Amin<sup>1</sup>, Derick Suarez<sup>1</sup>, Yingjian Liu<sup>3</sup>, Dong Qian<sup>3</sup> and Wing Kam Liu<sup>1,\*</sup>

<sup>1</sup>Department of Mechanical Engineering, Northwestern University, Evanston, 60208, USA

<sup>2</sup>Theoretical and Applied Mechanics, Northwestern University, Evanston, 60208, USA

<sup>3</sup>Department of Mechanical Engineering, The University of Texas at Dallas, Richardson, 75080, USA

\*Corresponding Author: Wing Kam Liu. Email: w-liu@northwestern.edu

Received: 01 June 2021 Accepted: 08 July 2021

## ABSTRACT

This paper presents the concept of reduced order machine learning finite element (FE) method. In particular, we propose an example of such method, the proper generalized decomposition (PGD) reduced hierarchical deep-learning neural networks (HiDeNN), called HiDeNN-PGD. We described first the HiDeNN interface seamlessly with the current commercial and open source FE codes. The proposed reduced order method can reduce significantly the degrees of freedom for machine learning and physics based modeling and is able to deal with high dimensional problems. This method is found more accurate than conventional finite element methods with a small portion of degrees of freedom. Different potential applications of the method, including topology optimization, multi-scale and multi-physics material modeling, and additive manufacturing, will be discussed in the paper.

## KEYWORDS

Machine learning; model reduction; HiDeNN-PGD; topology optimization; multi-scale modeling; additive manufacturing

## 1 Introduction and Motivation

The finite element method (FEM) [1,2] has gained an unprecedented success over the last decades and has become an essential tool for simulation based engineering across different fields of the manufacturing industry, including aerospace and automotive industries. An overview of the history and development of FEM can be found in a recent article written by Liu et al. [3]. Despite the rapid evolution of computer hardware, the simulation of real engineering problems can still be prohibitive in terms of computational time. For instance, simulating a car crash problem, which usually requires a large mesh size containing millions or billions elements, may take up to days and even months. This is obviously an obstacle that must be overcome for a more efficient and sustainable engineering design [4].



The development of model reduction methods falls into this perspective. By definition, model reduction refers to a family of methods that reduce the degrees of freedom of numerical systems and thus the underlying computational cost. An instance of these methods is homogenization, such as the coarse-grained model used for molecular dynamic simulations [5]. A similar concept called clustering can also be used to construct reduced order models for efficient multiscale material behavior simulations, as proposed by the so-called self-consistent clustering analysis [6–8]. Another type of model reduction method is the projection-based methods. The proper orthogonal decomposition (POD) is one of the most popular methods. The idea behind this method is finding first a reduced POD basis from a set of offline pre-computed snapshots (data), and then projecting the original model onto that basis so as to reduce the system dimension. The POD reduced model has thus the same dimension of the reduced basis, which is largely inferior to the original size. Consequently, the online computations become extremely fast. The POD based model reduction was originated in the field of computational fluid dynamics and has been applied to solid problems (see e.g., [9–12]). A more mathematically rigorous approach, called reduced basis method, has also been developed within a similar framework [13,14]. Other related techniques, like hyper reduction, can be seen as an enhancement for solving nonlinear problems [15–17]. Contrary to POD, the proper generalized decomposition (PGD) based model reduction does not require a precomputed database. The reduced basis is constructed on-the-fly by solving the partial differential equations of the underlying problem. The PGD was originally proposed in a space-time solution framework [18,19] and has been extended to computing solutions in large parametric domains [20,21]. For overcoming the intrusiveness of the PGD method, the non-intrusive type PGD has also been developed, e.g., high order PGD (HOPGD) and its sparse counterpart [22], which enables a wide application to nonlinear problems, such as welding [22–24], additive manufacturing model calibration [25], crystal plasticity simulations [26], etc. The HOPGD is also a data-driven method and has a direct connection to machine learning techniques.

The machine learning or more general artificial intelligence has gained increasing popularity in recent years for computational science and engineering. The universal approximation theory [27] has provided the foundation for the superior performance of deep neural networks (DNN) for function approximation. Many research works have been done based on a data-driven framework to construct surrogate models that relate the input, like materials, process, or geometry information, to the output mechanical quantities of interest for the purpose of material and process design or fast predictions [28–30]. However, purely data-driven machine learning approaches usually require extensive training data and suffer from the prohibitive cost devoted to the data generation. Experimental or simulation data are not easy to obtain at a low cost in mechanical or physical problems, compared to other fields like image processing. Hence, attempts have been made using physical governing equations to train the DNN instead of using data. The so-called PINN (physics-informed neural networks) [31] is one of the successful approaches. The idea is to approximate the solution functions with DNN and try to minimize a loss function that satisfies the governing partial differential equations. Nevertheless, the training cost and the convergence seem a challenging issue. More recently, a new type of method called hierarchical deep-learning neural networks (HiDeNN) has been proposed [32]. The general HiDeNN framework [33] seems a promising tool to provide a balance between the approximation accuracy and the training cost. A theoretical analysis can be found in a recent paper [34].

The HiDeNN method is based on FEM and provides the capabilities in constructing the FE shape functions. This capability is essential as it enables HiDeNN to build a comprehensive element library that is compatible with the ones used by the major commercial and open source

FE codes. The examples include 2D and 3D isoparametric solid elements, which are known to be the most cost-effective in practical applications. As such, all of the current FEM codes can seamlessly interface with HiDeNN to fully take advantage of its machine learning capabilities. On the other hand, the convergence of the method should be guaranteed with the support of well-developed FEM solvers. Furthermore, the flexibility in the construction of shape functions allows an automatic mesh adaptivity in the HiDeNN. Compared to PINN, this method is based on the weak form of governing equations and should have fewer degrees of freedom, which is expected to enable a higher computational efficiency.

The exploration of machine learning techniques for solving partial differential equations prompt the development of a new generation of FEM. Despite the use of GPU computing, the computational cost seems a remaining issue for this kind of approach. The performance of the method relies strongly on the choice of solution scheme, which seems not unique for general problems. This becomes worse when large numerical systems have to be solved. Hence, we propose to combine model reduction with the machine learning approaches, leading to the concept of reduced order machine learning FEM. In general, the machine learning framework like HiDeNN can be combined with any aforementioned model reduction method, such as POD, PGD, HOPGD, hyper reduction, etc. To illustrate the concept, we introduce an example of this type, called HiDeNN-PGD, which combines the HiDeNN-FEM and the PGD. It has been shown that the HiDeNN-PGD is more accurate than FEM but with much fewer degrees of freedom than both FEM and HiDeNN, thanks to the order reduction by PGD. Furthermore, it seems that the HiDeNN-PGD improves the optimality of the PGD basis and reduces the necessary number of modes for a given accuracy. The observation is more and more significant with increasing degrees of freedom. Thus, we believe that the reduced order machine learning FEM can provide a high performance computing framework for large-size engineering problems with a high level of accuracy.

In this work, we will briefly describe the concept of HiDeNN-PGD and discuss the potential applications of the overall framework to the following challenging problems that require intensive computational efforts.

- Multi-scale modeling of materials: the multi-scale behavior is required for modeling of materials, like composites, for taking into account the microscopic heterogeneity or defects and avoiding making assumptions on the macroscopic constitutive laws. The well-known  $FE^2$  method [35] has proven to be very time-consuming. The proposed HiDeNN-PGD is expected to reduce largely the underlying degrees of freedom and consequently the computational cost.
- Topology optimization: the topological design of materials or structures requires repetitive evaluations of the quantities of interest, like stress field, with respect to the variation of density and geometries. This can be out of reach when the background mesh size increases, especially in 3D cases. Sometimes, super computers are required [36]. Reducing the intrinsic computational complexity and the design variables by HiDeNN-PGD seems to be a promising direction for reducing the design cost to a level accessible for common computer hardware.
- Additive manufacturing: the multi-scale modeling of additive manufacturing systems is challenging in terms of computational time. Therefore, macroscopic models usually neglect some micro-scale detailed physics for a better computational efficiency, which seem to be important in the final manufactured part, such as defects due to the lack of fusion and the surface roughness. In addition, the manufacturing process is controlled by various

parameters, of which the optimal values are unknown and to be determined by parametric studies. The the reduced order machine learning FEM framework can be a powerful tool to reduce the computational cost of such process modeling.

- Multi-physics problems: the multi-physics problems, such as the modeling of soft robotic materials, usually induces a large number of unknown parameters. Simulating this kind of problem requires the consideration of a strong coupling of different phenomena and is consequently prohibitive in time. The HiDeNN-PGD is expected to enable fast predictions and reduce the cost devoted to determining the optimal design parameters.

The paper is organized as follows. [Section 2](#) gives a brief overview of HiDeNN-PGD formulation, in which different solution strategies will be discussed. Some numerical experiments will be presented in [Section 3](#). [Section 4](#) presents the potential application of this method on the challenging problems with current FE analysis. Finally, the paper closes with some concluding remarks.

## 2 HiDeNN-PGD

### 2.1 Formulation

The HiDeNN-PGD relies on the construction of HiDeNN-FEM shape functions. We first introduce the HiDeNN-FEM shape function.

#### 2.1.1 HiDeNN-FEM Shape Function

##### *Shape Function Based on Physical Coordinates*

The idea is to constrain the weights and biases of the DNN to mesh coordinates so as to build FE shape functions, which reads

$$\mathcal{F}_I(x; \mathbf{w}, \mathbf{b}, \mathcal{A}) \rightarrow \mathcal{N}_I(x; \mathbf{x}_I^*, \mathcal{A}) \quad (1)$$

where  $\mathcal{F}_I$  stands for the fully connected DNN structures with weights  $\mathbf{w}$ , biases  $\mathbf{b}$ , and the activation function  $\mathcal{A}$ .  $\mathcal{N}_I$  denotes the FE shape function for the node at position  $\mathbf{x}_I^*$ . [Fig. 1](#) is an illustration of such construction in 3D cases. Assuming a domain  $\Omega$  is discretized by  $np$  points, we can write the HiDeNN-FEM approximation as

$$u^h(\mathbf{x}) = \sum_{I=1}^{np} \mathcal{N}_I(\mathbf{x}; \mathbf{x}_I^*, \mathcal{A}) u_I \quad (2)$$

where  $u_I$  is the discretized nodal solution of the problem,  $u^h$  is the approximated solution function. Considering the vector notation  $\mathcal{N} = [\mathcal{N}_1, \dots, \mathcal{N}_{np}]$  and  $\mathbf{U} = [u_1, \dots, u_{np}]^T$ , the [Eq. \(2\)](#) can be simplified as

$$u^h(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{x}^*, \mathcal{A}) \mathbf{U} \quad (3)$$

In 1D case, this reads

$$u^h(x) = \mathcal{N}(x; x^*, \mathcal{A}) \mathbf{U} \quad (4)$$

Detailed construction of such 1D shape functions using DNN can be found in [\[32\]](#). It should be noticed that the FE shape function is only one of the choices, the HiDeNN structure allows to easily switch from one to another by releasing the constraints on the weights  $\mathbf{w}$  and biases  $\mathbf{b}$ .

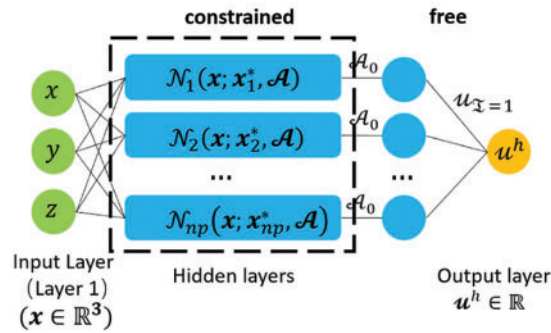


Figure 1: HiDeNN-FEM shape functions [32,34]

Shape Function Based on Element Coordinates

Shape functions employing element coordinates can also be constructed [37] based on the methodology described in the last section and in [34]. These include the ones for general 2D and 3D isoparametric elements that are commonly used by the commercial and open-source codes. This implementation is critical in HiDeNN as it enables major commercial/open-source FE codes to directly integrate with HiDeNN-FEM and fully take advantage of its machine learning capability. On the other hand, there is no need to program separate FE solvers in HiDeNN-FEM as these are already available in the current FE codes. To the best of our knowledge, such a seamless interface does not exist in other machine learning based approach to FEM. Fig. 2 illustrates the steps to construct the shape functions for 2D isoparametric elements and integrate them with nonlinear FE analysis based on total Lagrangian formulation [1]. More details can be found in [37]. Box 1 in Fig. 2 evaluates the shape functions and their derivatives based on given inputs of the isoparametric coordinates  $(\xi, \eta)$  (in most cases these will be for the quadrature points) using HiDeNN [34], expressed as

$$u_e^h(\xi, \eta) = \mathcal{N}(\xi, \eta, \mathcal{A}) U_e \tag{5}$$

where  $(\cdot)_e$  denotes the element-wise displacement.  $\mathcal{N}$  is the shape function in the parametric space.

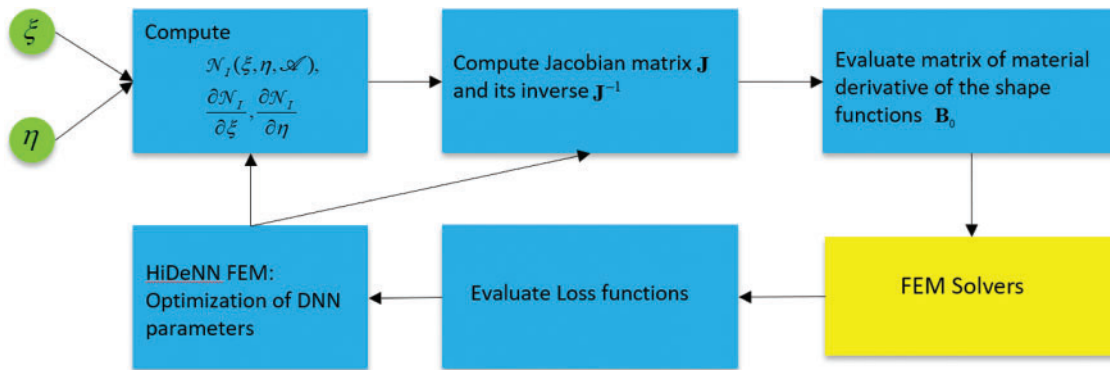
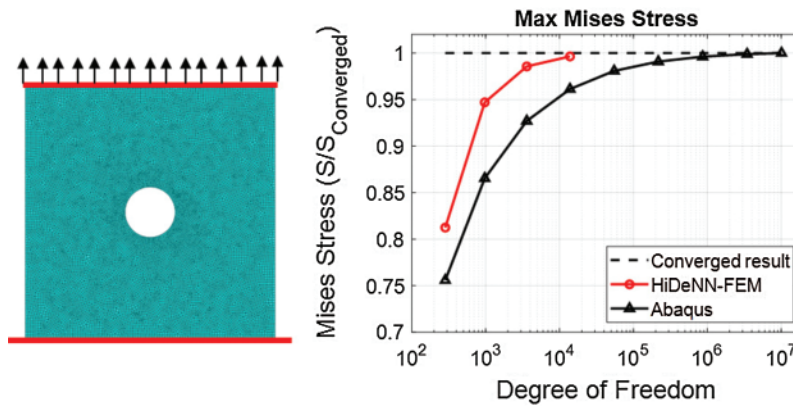


Figure 2: Steps to construct the 2D isoparametric elements and integrate it with nonlinear FE analysis [37]

Boxes 2 and 3 further constructs neural network to compute the Jacobian from the coordinate transformation and its inverse based on the basic building blocks of HiDeNN given in [32,34]. Finally the matrix that contains the shape function derivatives with respect to the material coordinates is evaluated and provided to the FE code through the interface.

Accuracy of the HiDeNN-FEM is illustrated in Fig. 3 and compared with the results obtained from the commercial FE code ABAQUS. Here we consider a plane stress problem of square plate of dimension 1 m by 1 m with center hole of radius of 0.1 m. As shown in Fig. 3a, the plate is modelled as Neo-Hookean material and subjected to traction of 100 kPa on the top surface while the bottom surface is constrained. Fig. 3 shows that HiDeNN-FEM achieves much more rapid convergence than Abaqus. For the same FE model using the standard 4-node quadrilateral element with 13,840 degrees of freedom, error from HiDeNN-FEM is 0.36% in terms of the maximum Mises stress whereas the corresponding is 3.87% from Abaqus.



**Figure 3:** Comparison between HiDeNN-FEM and commercial code Abaqus for a 2D plane stress problem [37]. (a) Problem statement; (b) Normalized maximum Mises stress as a function of degrees of freedom. Dashed line indicates converged result obtained from very refined mesh of 10 M degrees of freedom

### 2.1.2 HiDeNN-PGD for Reducing the Spatial Degrees of Freedom

Before introducing the HiDeNN-PGD, we can do a rough analysis of the degrees of freedom (DoF) for the HiDeNN-FEM method. In conventional DNN, the weight and biases are unknown variables, whereas the HiDeNN-FEM has constrained them to the mesh coordinates and thus the optimal coordinates become the primary unknowns. Therefore, optimizing the weights and biases is equivalent to optimizing the mesh coordinates  $\mathbf{x}^*$  and the nodal solution  $\mathbf{U}$  in HiDeNN-FEM. Now, we can consider a 3D regular domain discretized by a uniform mesh with  $N \times N \times N$  nodes. The total degrees of freedom for the unknowns in HiDeNN-FEM is then

$$DoF = \underbrace{N \times N \times N}_{\text{DoF of } \mathbf{U}} + \underbrace{N \times N \times N}_{\text{DoF of } \mathbf{x}^*} \quad (6)$$

This DoF grows exponentially when increasing the nodes  $N$  in each direction. For example, if  $N$  increases from 10 to 100, the DoF grows from 2000 to  $2 \times 10^6$ . This estimate is based on the assumption that  $u^h$  is a scalar valued function. For a mechanical problem, this  $u^h$  becomes a vector and the DoF grows more quickly.



The HiDeNN-PGD can reduce this growth by considering a separation of variables as follows

$$u^h(\mathbf{x}) = u^h(x, y, z) \approx \sum_{q=1}^Q u_x^{(q)}(x) u_y^{(q)}(y) u_z^{(q)}(z) \quad (7)$$

where  $Q$  is the number of modes, which is expected to be small. The 3D function is separated as the summation of products of 1D functions. This format is known as separation of variables and adopted by PGD based model reduction. We applied then the HiDeNN-FEM shape function, which leads to

$$u^h(\mathbf{x}) = u^h(x, y, z) \approx \sum_{q=1}^Q \underbrace{\mathcal{N}(x; x^*, \mathcal{A}) \boldsymbol{\beta}^{(q)}}_{u_x^{(q)}(x)} \underbrace{\mathcal{N}(y; y^*, \mathcal{A}) \boldsymbol{\gamma}^{(q)}}_{u_y^{(q)}(y)} \underbrace{\mathcal{N}(z; z^*, \mathcal{A}) \boldsymbol{\theta}^{(q)}}_{u_z^{(q)}(z)} \quad (8)$$

where  $\mathcal{N}$  is only the 1D HiDeNN-FEM shape function as in Eq. (4). Even though the problem is 3D, only 1D shape functions are needed, thanks to the separation of variables. With the Eq. (8), we analyze again the DoF for the problem using the uniform mesh of  $N \times N \times N$  nodes. The primary unknowns are nodal values of  $\boldsymbol{\beta}^{(q)}$ ,  $\boldsymbol{\gamma}^{(q)}$ ,  $\boldsymbol{\theta}^{(q)}$ , and the directional mesh coordinates  $x^*$ ,  $y^*$ ,  $z^*$ . The DoF reads then

$$DoF = \underbrace{Q \times (N + N + N)}_{DoF \text{ of } \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\theta}} + \underbrace{N + N + N}_{DoF \text{ of } x^*, y^*, z^*} \quad (9)$$

Since the  $Q$  is a small number compared to the mesh  $N$ , we can expect the DoF of HiDeNN-PGD is much smaller than HiDeNN-FEM. For illustration, if  $Q = 5$  and  $N$  increases from 10 to 100, the DoF grows from 180 to 1800 (which is  $2 \times 10^6$  for HiDeNN-FEM). Hence, the HiDeNN-PGD enables a significant reduction of DoF and breaks the exponential growth of DoF with respect to the discretization.

### 2.1.3 HiDeNN-PGD for Reducing the Degrees of Freedom in a Space-Time-Parameter Domain

The HiDeNN-PGD can also be used to compute space-time-parametric solutions with reduced DoF. This kind of solution is needed when doing parametric studies. The conventional way to do so is to fix the parameters and solve for the corresponding space-time solution, then modify the parameters and solve again the problem. Here, using HiDeNN-PGD, we can compute all the parametric solutions at the same time. This consists in solving the following type of solution with parameters as extra-coordinates. The HiDeNN-PGD in this case reads

$$u^h(\mathbf{x}, t, \boldsymbol{\mu}) = u^h(x, y, z, t, \mu_1, \dots, \mu_k) \approx \sum_{q=1}^Q u_x^{(q)}(x) u_y^{(q)}(y) u_z^{(q)}(z) u_t^{(q)}(t) u_{\mu_1}^{(q)}(\mu_1) \cdots u_{\mu_k}^{(q)}(\mu_k) \quad (10)$$

where  $\boldsymbol{\mu}$  is the control or design parameters that we want to study. These parameters can be related to materials, loading, or geometry. Optimization or calibration problems usually require such parametric solutions to study the influence of parameters on the output quantities.

Similarly to the Eq. (8), we can consider a uniform discretization in the space-time-parameter domain:  $\Omega \times \Omega_t \times \Omega_{\mu_1} \times \dots \times \Omega_{\mu_k}$ , and then apply the 1D HiDeNN-FEM shape functions for

each separated term in the above equation. To make it short, we omit the notation  $\mathcal{A}$  in the shape function.

$$\begin{aligned}
 u^h \approx & \sum_{q=1}^Q \underbrace{\mathcal{N}(x; x^*)}_{u_x^{(q)}(x)} \underbrace{\beta^{(q)}}_{u_y^{(q)}(y)} \underbrace{\mathcal{N}(y; y^*)}_{u_z^{(q)}(z)} \underbrace{\gamma^{(q)}}_{u_t^{(q)}(t)} \underbrace{\mathcal{N}(z; z^*)}_{u_{\mu_1}^{(q)}(\mu_1)} \underbrace{\theta^{(q)}}_{u_{\mu_k}^{(q)}(\mu_k)} \underbrace{\mathcal{N}(t; t^*)}_{f_1^{(q)}} \underbrace{T^{(q)}}_{f_k^{(q)}} \underbrace{\mathcal{N}(\mu_1; \mu_1^*)}_{f_1^{(q)}} \dots \underbrace{\mathcal{N}(\mu_k; \mu_k^*)}_{f_k^{(q)}} \\
 & \dots \underbrace{\mathcal{N}(\mu_k; \mu_k^*)}_{f_k^{(q)}}
 \end{aligned} \tag{11}$$

where  $(\cdot)^*$  are the discrete nodal coordinates of the space-time-parameter mesh. Again, the primary unknown are the nodal solutions of the separated terms and the optimal mesh coordinates. For analyzing the DoF, we assume that each dimension is discretized by  $N$  nodes. The DoF can be computed as

$$\text{DoF} = \underbrace{Q \times (3N + N + kN)}_{\text{DoF of nodal separated terms}} + \underbrace{3N + N + kN}_{\text{DoF of global mesh}} \tag{12}$$

Assuming the  $Q = 5$ ,  $k = 3$ ,  $N = 10$ , the DoF is then 420, whereas the DoF is  $10^7$  and  $2 \times 10^7$  respectively for FEM and HiDeNN-FEM. This reduction allows HiDeNN-PGD compute solutions in a high dimensional space-time-parameter domain.

### 2.1.4 Variational Framework

The HiDeNN-PGD uses a variational framework to compute the primary unknowns: nodal solutions of separated terms and optimal mesh coordinates. Assuming a potential energy exists for the problem, the HiDeNN-PGD solution can be found by solving the following minimization problem.

$$u^h = \underset{\substack{\beta^{(q)}, \gamma^{(q)}, \theta^{(q)}, T^{(q)}, f_1^{(q)}, \dots, f_k^{(q)} \\ x^*, y^*, z^*, t^*, \mu_1^*, \dots, \mu_k^*}}{\text{arg min}} \Pi(u^h) \tag{13}$$

where  $\Pi$  is the potential function. The solution can be found by any adequate optimization algorithm.

In what follows, we restrict us to the case with only space variables (8) and present the detailed implementation of the method.

## 2.2 Numerical Implementation

Let us consider a 2D Poisson problem, which reads

$$\nabla^2 u(x, y) + b(x, y) = 0 \quad \text{in } \Omega \text{ and } u|_{\partial\Omega} = 0 \tag{14}$$

where  $\nabla$  denotes the gradient. The domain  $\Omega$  is assumed to be a rectangular domain.  $b$  is a body source term with the assumption that  $b(x, y) = b_x(x)b_y(y)$ . For arbitrary functions, this kind of separation can be obtained by the HOPGD [22,23].

The solution of this problem can be found by minimizing the following potential function

$$\Pi(u) = \int_{\Omega} \frac{1}{2} (\nabla u)^T \nabla u \, dx \, dy + \int_{\Omega} u b \, dx \, dy \tag{15}$$



Then, the discretized HiDeNN-PGD solution reads

$$u^h(x, y) \approx \sum_{q=1}^Q \mathcal{N}(x; x^*) \boldsymbol{\beta}^{(q)} \mathcal{N}(y; y^*) \boldsymbol{\gamma}^{(q)} \quad (16)$$

where the activation function  $\mathcal{A}$  has been omitted. The number of modes  $Q$  is assumed to be given. Hence, the solution can be computed with

$$u^h = \arg \min_{\substack{\boldsymbol{\beta}^{(q)}, \boldsymbol{\gamma}^{(q)} \\ x^*, y^*}} \Pi(u^h) \quad (17)$$

Solving this minimization problem requires the derivatives of the potential with respect to the unknown variables, i.e.,

$$\frac{\partial \Pi}{\partial \boldsymbol{\beta}^{(q)}}, \frac{\partial \Pi}{\partial \boldsymbol{\gamma}^{(q)}}, \frac{\partial \Pi}{\partial x^*}, \frac{\partial \Pi}{\partial y^*} \quad (18)$$

The updating step reads

$$\boldsymbol{\beta}^{(q)} = \boldsymbol{\beta}^{(q)} - \alpha_1 \frac{\partial \Pi}{\partial \boldsymbol{\beta}^{(q)}}, \quad \boldsymbol{\gamma}^{(q)} = \boldsymbol{\gamma}^{(q)} - \alpha_2 \frac{\partial \Pi}{\partial \boldsymbol{\gamma}^{(q)}}, \quad x^* = x^* - \alpha_3 \frac{\partial \Pi}{\partial x^*}, \quad y^* = y^* - \alpha_4 \frac{\partial \Pi}{\partial y^*} \quad (19)$$

where the learning rate (step size)  $\alpha$  should be determined by the optimization scheme, such as Adam. The iterative procedure should be performed until the  $Q$  modes reach a convergence for  $u^h$  or  $\Pi$ .

### 2.2.1 Example of one Mode Case

For a better illustration, we consider the one-mode case

$$u^h(x, y) \approx \mathcal{N}(x; x^*) \boldsymbol{\beta}^{(1)} \mathcal{N}(y; y^*) \boldsymbol{\gamma}^{(1)} \quad \text{or} \quad u^h(x, y) \approx \mathcal{N}(x; x^*) \boldsymbol{\beta} \mathcal{N}(y; y^*) \boldsymbol{\gamma} \quad (20)$$

The gradient of  $u^h$  becomes

$$\nabla u^h = \begin{bmatrix} \mathbf{B}(x; x^*) \boldsymbol{\beta} \mathcal{N}(y; y^*) \boldsymbol{\gamma} \\ \mathcal{N}(x; x^*) \boldsymbol{\beta} \mathbf{B}(y; y^*) \boldsymbol{\gamma} \end{bmatrix} \quad (21)$$

where  $\mathbf{B}(x; x^*) = \frac{\partial \mathcal{N}(x; x^*)}{\partial x}$  and  $\mathbf{B}(y; y^*) = \frac{\partial \mathcal{N}(y; y^*)}{\partial y}$ .

Substituting (20) and (21) into (15), then

$$\Pi(u^h) = \frac{1}{2} \boldsymbol{\beta}^T \mathbf{K}_{xx} \boldsymbol{\beta} \boldsymbol{\gamma}^T \mathbf{M}_{yy} \boldsymbol{\gamma} + \frac{1}{2} \boldsymbol{\beta}^T \mathbf{M}_{xx} \boldsymbol{\beta} \boldsymbol{\gamma}^T \mathbf{K}_{yy} \boldsymbol{\gamma} - \boldsymbol{\beta}^T \mathbf{F}_x \boldsymbol{\gamma}^T \mathbf{F}_y \quad (22)$$

where

$$\mathbf{K}_{xx} = \int_{\Omega_x} \mathbf{B}^T \mathbf{B} dx, \quad \mathbf{M}_{xx} = \int_{\Omega_x} \mathcal{N}^T \mathcal{N} dx, \quad \mathbf{K}_{yy} = \int_{\Omega_y} \mathbf{B}^T \mathbf{B} dy, \quad \mathbf{M}_{yy} = \int_{\Omega_y} \mathcal{N}^T \mathcal{N} dy \quad (23)$$

and

$$\mathbf{F}_x = \int_{\Omega_x} \mathcal{N} b_x dx, \quad \mathbf{F}_y = \int_{\Omega_y} \mathcal{N} b_y dy, \quad (24)$$

Therefore, we have the derivatives as

$$\frac{\partial \Pi}{\partial \boldsymbol{\beta}} = \mathbf{K}_{xx} \boldsymbol{\beta} \boldsymbol{\gamma}^T \mathbf{M}_{yy} \boldsymbol{\gamma} + \mathbf{M}_{xx} \boldsymbol{\beta} \boldsymbol{\gamma}^T \mathbf{K}_{yy} \boldsymbol{\gamma} - \mathbf{F}_x \boldsymbol{\gamma}^T \mathbf{F}_y \quad (25)$$

$$\frac{\partial \Pi}{\partial \boldsymbol{\gamma}} = \boldsymbol{\beta}^T \mathbf{K}_{xx} \boldsymbol{\beta} \mathbf{M}_{yy} \boldsymbol{\gamma} + \boldsymbol{\beta}^T \mathbf{M}_{xx} \boldsymbol{\beta} \mathbf{K}_{yy} \boldsymbol{\gamma} - \boldsymbol{\beta}^T \mathbf{F}_x \mathbf{F}_y \quad (26)$$

$$\frac{\partial \Pi}{\partial x^*} = \frac{1}{2} \boldsymbol{\beta}^T \frac{\partial \mathbf{K}_{xx}}{\partial x^*} \boldsymbol{\beta} \boldsymbol{\gamma}^T \mathbf{M}_{yy} \boldsymbol{\gamma} + \frac{1}{2} \boldsymbol{\beta}^T \frac{\partial \mathbf{M}_{xx}}{\partial x^*} \boldsymbol{\beta} \boldsymbol{\gamma}^T \mathbf{K}_{yy} \boldsymbol{\gamma} - \boldsymbol{\beta}^T \frac{\partial \mathbf{F}_x}{\partial x^*} \boldsymbol{\gamma}^T \mathbf{F}_y \quad (27)$$

$$\frac{\partial \Pi}{\partial y^*} = \frac{1}{2} \boldsymbol{\beta}^T \mathbf{K}_{xx} \boldsymbol{\beta} \boldsymbol{\gamma}^T \frac{\partial \mathbf{M}_{yy}}{\partial y^*} \boldsymbol{\gamma} + \frac{1}{2} \boldsymbol{\beta}^T \mathbf{M}_{xx} \boldsymbol{\beta} \boldsymbol{\gamma}^T \frac{\partial \mathbf{K}_{yy}}{\partial y^*} \boldsymbol{\gamma} - \boldsymbol{\beta}^T \mathbf{F}_x \boldsymbol{\gamma}^T \frac{\partial \mathbf{F}_y}{\partial y^*} \quad (28)$$

Finally, the nodal solution and mesh coordinates can be updated using (19) until the variation of the potential energy  $\Pi$  becomes small enough. It should be noticed that the HiDeNN-PGD can be degenerated to PGD by fixing the mesh coordinates.

### 3 Numerical Experiment

This section presents a numerical example using the proposed HiDeNN-PGD framework. The numerical results will be compared with FEM, HiDeNN (HiDeNN-FEM) and a conventional PGD method.

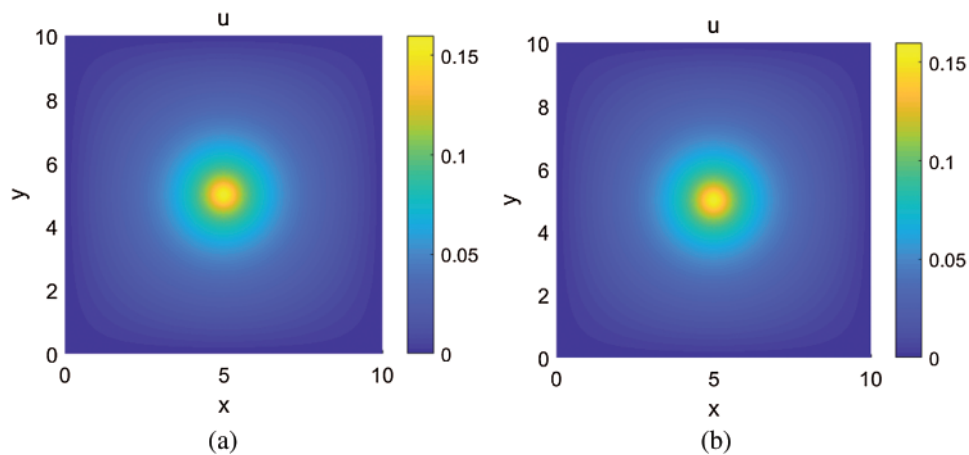
The test example is a Poisson problem described by the Eq. (14) with the following body source term

$$b(x, y) = \exp\{-10(x-5)^2 - 10(y-5)^2\} \quad (29)$$

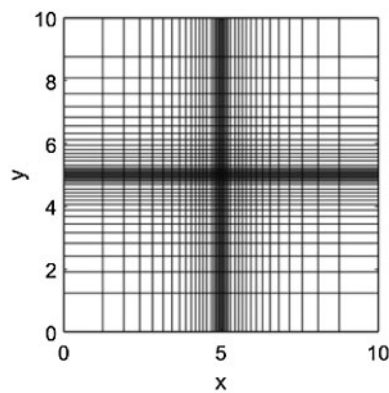
The geometry is a square domain, as shown in Fig. 4a. A very coarse uniform mesh ( $40 \times 40$  elements) is used for the HiDeNN-PGD method. We prescribed  $Q = 4$  as the number of modes. The HiDeNN-PGD consists in finding out the modes of  $\boldsymbol{\beta}^{(q)}$ ,  $\boldsymbol{\gamma}^{(q)}$ , and the optimized mesh coordinates  $x^*$ ,  $y^*$  by minimizing the corresponding potential energy. The final solution  $u^h$  obtained HiDeNN-PGD is depicted in Fig. 4. Fig. 4a shows the HiDeNN-PGD solution reconstructed by the 4 modes of  $\boldsymbol{\beta}^{(q)}$ ,  $\boldsymbol{\gamma}^{(q)}$ . It is shown that the solution agrees well with a reference FEM solution which is obtained on a very fine mesh. In terms of mesh coordinate optimization, the HiDeNN-PGD only allows separated directional movements due to the separation of variables. Therefore, the final optimized mesh is well structured, as shown in Fig. 5. However, if HiDeNN mesh is used, the distortion becomes a real issue, which is undesired. In this sense, the HiDeNN-PGD presents a big advantage on the mesh distortion controlling when optimizing the coordinates.

The HiDeNN-PGD also makes the modes different to the conventional PGD. As shown in Fig. 6, the HiDeNN-PGD modes seem more concentrated on the region of interest. This difference comes from the mesh adaptivity and the solution strategy we proposed. The HiDeNN-PGD is expected to obtain optimized modes, compared to conventional PGD.

For further comparing the performance of each method, we illustrate the accuracy of PGD, FEM, HiDeNN and HiDeNN-PGD on the same coarse  $40 \times 40$  mesh in Table 1. The error is computed by taking the difference to the reference fine mesh FEM solution and using an energy norm relying on the gradient of the solution. As expected, the HiDeNN-PGD and HiDeNN are both more accurate than FEM, thanks to the mesh adaptivity. By increasing the number of mode, the PGD only approaches the FEM. In terms of DoF, the HiDeNN-PGD has a reduction of factor 20 over the HiDeNN method. The DoF is also smaller than PGD when compared at the same level of accuracy. The HiDeNN-PGD only requires two modes for a very good accuracy, whereas PGD needs more than 6 modes. This comparison confirms that HiDeNN-PGD has optimized separated modes.

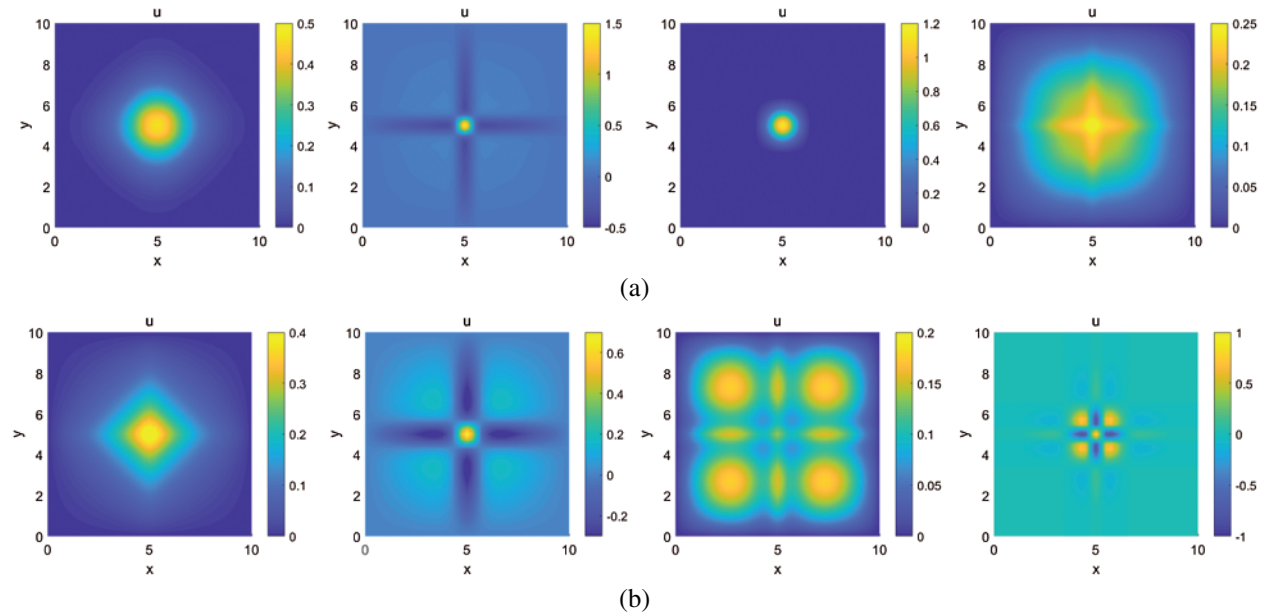


**Figure 4:** Comparison of HiDeNN-PGD and FEM solutions [34]. (a) HiDeNN-PGD, mesh:  $40 \times 40$  (b) Reference FEM, mesh:  $4000 \times 4000$



**Figure 5:** HiDeNN-PGD optimized mesh from an initial uniform mesh

This example has demonstrated the capability and potential of HiDeNN-PGD for reducing the DoF while keeping a good accuracy of solutions. The computational solver and the choice of the number of modes for a better numerical performance will be the next topics we will investigate, but not in the scope of this paper. Indeed, these aspects have been mentioned in our recent paper [34], which studied theoretically the proposed HiDeNN-PGD method.



**Figure 6:** Comparison of HiDeNN-PGD and PGD modes [34]. (a) HiDeNN-PGD modes (b) PGD modes

**Table 1:** Accuracy comparison for different methods on the coarse 40 by 40 mesh

Mode number	PGD		FEM		HiDeNN-PGD		HiDeNN	
	DoF	Err (%)	DoF	Err	DoF	Err %	DoF	Err
1	78	38.167	<b>1521</b>	<b>11.659%</b>	156	37.357	<b>4719</b>	<b>2.102%</b>
2	156	16.500	—	—	<b>234</b>	<b>9.293</b>	—	—
⋮	⋮	⋮	—	—	⋮	⋮	—	—
6	468	11.666	—	—	546	3.661	—	—
8	<b>624</b>	<b>11.659</b>	—	—	—	—	—	—
20	1560	11.659	—	—	—	—	—	—

## 4 Perspectives and Applications

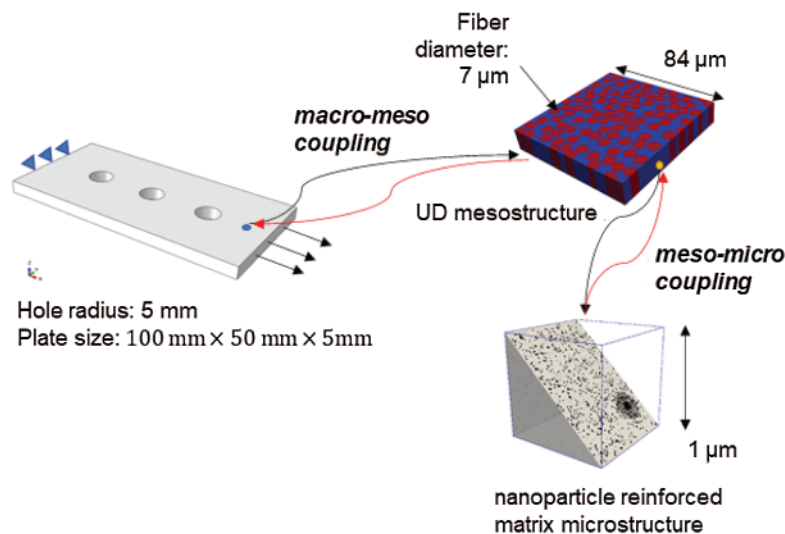
This section will discuss the potential applications of the HiDeNN-PGD and the more general reduced order machine learning FEM on the real problems in computational science and engineering. In particular, we use HiDeNN-ROM (reduced order model) for the general reduced order machine learning FEM. We will consider two types of applications: 1) accelerating computations, 2) efficient parametric studies.

### 4.1 Multi-Scale Modeling of Composites

Composite materials are hierarchical in nature and need to be modeled in multiple length scale to capture multiscale physics. As shown in Fig. 7, a composite plate with holes is demonstrated as a three lengthscale problem. The macro or part scale is the plate with the hole whereas a material point in the macroscale can be modeled as a representative volume

element (RVE) of unidirectional fiber composites at the mesoscale. Often the polymer matrix are reinforced with nanofillers to impart toughness in the matrix which leads the problem to be a three-scale problem. Numerical modeling of a three-scale composite is computationally intractable with the traditional approach such as FE for each length scale. Previously, two scale problems are solved using  $FE \times FE$  [35] formulation, but with a fine mesh in the lower scale the problem computational effort increases significantly. Recently, the data-driven reduced order method such as Self-consistent Clustering Analysis (SCA) [6], Multiresolution Clustering Analysis [7], FEM-cluster based analysis method (FCA), Virtual clustering analysis (VCA), NTFA, POD, etc., Li et al. [28] has been proposed which is quite successful for two-scale concurrent composite modeling. However, the offline database computation for SCA type methods can be quite expensive and not feasible for the parametric studies of the lower scale effect on the part scale solution. The computational bottleneck can be resolved by combining SCA-PGD (or SCA-HOPGD) approach for the microscale and mesoscale problems, whereas the HiDeNN-PGD or other HiDeNN-ROM methods can be adopted for the part scale problem. We envision this approach will help us study the parametric space at microstructure level and solve the part scale details with the adaptive feature of HiDeNN methods. To illustrate our vision, we divide the problem into following three steps:

- Prediction of reinforced matrix properties using SCA-PGD at microscale
- SCA-PGD based reduce order surrogate model of UD composite at mesoscale
- HiDeNN-PGD (or HiDeNN-ROM) based solution for composite part at macroscale.



**Figure 7:** A schematic of three scale composite materials. The part scale is a plate with three holes, mesostructure is unidirectional (UD) fiber reinforced and microstructure is the nanoparticle reinforced matrix

#### 4.1.1 Prediction of Reinforced Matrix Properties Using SCA-PGD at Microscale

As shown in the Fig. 7, the particle reinforced matrix is considered as the microscale of the composite. The materials properties are needed to be calibrated and passed to the mesoscale to correctly account for the matrix properties at the mesoscale. For the microscale, the composite can be designed by varying the volume fraction of the particles, as well as the temperatures

that can change the matrix properties quite significantly. Therefore, in this design problem, the volume fraction and the temperature are the two design parameters that we can consider for the microscale. However, varying the volume fraction and the temperatures, we need to perform new simulations every time for new values. The PGD-based methods (e.g., [22]) that provide parametric materials property prediction can be very useful in this case in the mesoscale. However, it should be noted that PGD may need an offline database that requires reasonable amount of time. Here, we can combine the prediction process with the SCA method that can significantly speed up the offline data generation process. For two given parameters, volume fraction ( $\phi$ ), and temperature ( $T$ ), the stress-strain can be approximated in a PGD form as

$$\sigma(\varepsilon, \phi, T) \approx \sum_{q=1}^Q u_{\varepsilon}^{(q)}(\varepsilon) u_{\phi}^{(q)}(\phi) u_T^{(q)}(T) \quad (30)$$

where  $u_{\varepsilon}^{(m)}(\varepsilon)$ ,  $u_{\phi}^{(m)}(\phi)$ ,  $u_T^{(m)}(T)$  are the separated modes associated with different parameters. The equivalent reinforced matrix hardening properties (yield stress, hardening exponent) can be then extracted from this stress-strain prediction and used for the mesoscale modeling of the matrix phase.

#### 4.1.2 SCA-PGD Based Reduced Order Surrogate Model of UD Composite at Mesoscale

The matrix properties have been calibrated from the microscale as discussed above. For the mesoscale, we can then construct a reduced order surrogate model based on, again, a combined SCA-PGD method. This reduced order surrogate model takes the macrostrain as input and predict the homogenized stress response and pass it to the macroscale HiDeNN-PGD model. Previously, SCA alone is used for the homogenization purpose, which restricts to explore the parametric space at the mesoscale since the computational effort becomes infeasible with a number of varying parameters [38]. Similar to the microscale, we can study the mesoscale volume fraction and temperature and consider them as input parameters for the PGD reduced order surrogate model. Again, SCA is used to generate the training data. The general PGD model can be written as

$$\sigma^M(\varepsilon^M, \Phi, T) \approx \sum_{q=1}^Q \mathbf{u}_{xx}^{(q)}(\varepsilon_{xx}^M) u_{yy}^{(q)}(\varepsilon_{yy}^M) u_{zz}^{(q)}(\varepsilon_{zz}^M) u_{xy}^{(q)}(\varepsilon_{xy}^M) u_{yz}^{(q)}(\varepsilon_{yz}^M) u_{zx}^{(q)}(\varepsilon_{zx}^M) u_{\Phi}^{(q)}(\Phi) u_T^{(q)}(T) \quad (31)$$

where  $\sigma^M$ ,  $\varepsilon^M$ ,  $\phi$ ,  $T$  are homogenized stress, macro strain, volume fraction of the UD fiber in the mesostructure and temperature, respectively.

#### 4.1.3 HiDeNN-PGD Based Solution for Composite Part at Macroscale

The HiDeNN-PGD or HiDeNN-ROM method is a good choice for part scale modeling as geometrical features such as the holes, bends can be resolved with adaptive mesh feature of HiDeNN for a better accuracy. Also, the part can be studied at different temperatures. The overall reduced order machine learning framework enables the rapid design by varying the microstructure descriptors and monitoring their response in real time. The final solution at the part scale level can be expressed similarly to the Eq. (11).

## 4.2 Topology Optimization for Materials Design

Topology optimization (TO) has been developed in last few decades with the purpose to find the optimal material layout to reach the best performance under design constraints [39,40]. Due



to the computation limitation and curse of dimensionality, the computational cost is burdensome in multiscale, multiphysics, and high resolution topology optimization [36,41]. Since most topology design are based on the voxel mesh, this gives the possibility to use HiDeNN-PGD to reduce the number of DoF in the design algorithm. Considering the Solid Isotropic Material with Penalization (SIMP) method [39], the density is assigned to each element of the background domain as

$$E(\rho) = E_{void} + \rho^p (E_{solid} - E_{void}), \tag{32}$$

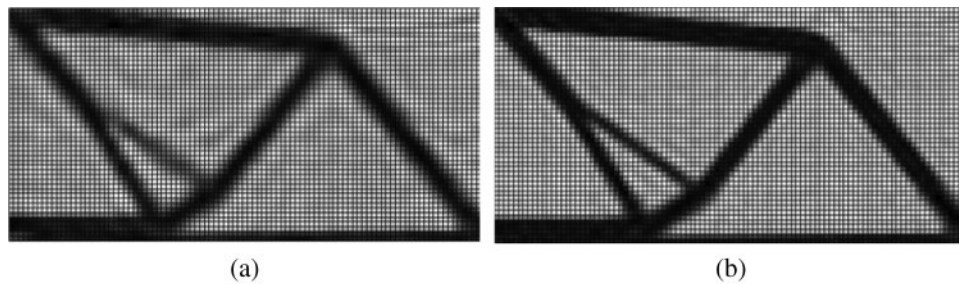
where  $E_{solid}$  and  $E_{void}$  are respectively reference solid and very weak materials,  $\rho \in [\rho_{min}, 1]$ . The penalization parameter  $p \geq 1$  is used to enforce final designs being either 0 or 1. To maximize the structural stiffness, the optimization problem reads:

$$\min c = \sum_i^L \mathbf{F}_i^T \mathbf{u}_i \tag{33}$$

subject to

$$\begin{cases} \mathbf{K}(\rho)\mathbf{u}_i = \mathbf{F}_i, & i = 1, \dots, L \\ \frac{V(\rho)}{V^*} - 1 \leq 0 \\ \rho_{min} \leq \rho_j \leq 1 & j = 1 \dots N \end{cases} \tag{34}$$

where  $L$  is the total DoF for displacement and  $N$  is the total number of design variables. The three constraints are: mechanical equilibrium equation, volume constraint, and design parameter constraints. The optimization requires repetitively solving the mechanical equation, which is the most computational costly part and can be accelerated by the HiDeNN-PGD or PGD. To do so, the density function needs to be decomposed, like the body force in Eq. (14). This reduces also the number of design variables. Results using PGD for TO are shown in Fig. 8. It can be seen that the design problem is reducible without sacrificing the final design accuracy. Compared to FEM-TO, the final design seems very similar, as shown in Fig. 9. The current PGD-TO is faster (with a factor 2) than conventional FEM-TO. The algorithm can be further enhanced using HiDeNN-PGD. In addition, it is found that the speedup is higher when dealing with high dimensional problems. The results demonstrate that HiDeNN-PGD has good potentials to reduce the computational cost in high resolution TO problems.



**Figure 8:** PGD based topology design. (a) Optimized design with 11 density modes (b) Optimized design with 19 density modes



**Figure 9:** Comparison of FEM-TO and PGD-TO results for a half MBB beam. (a) FEM-TO (b) PGD-TO

### 4.3 Parametric Learning and Uncertainty Quantification for Additive Manufacturing

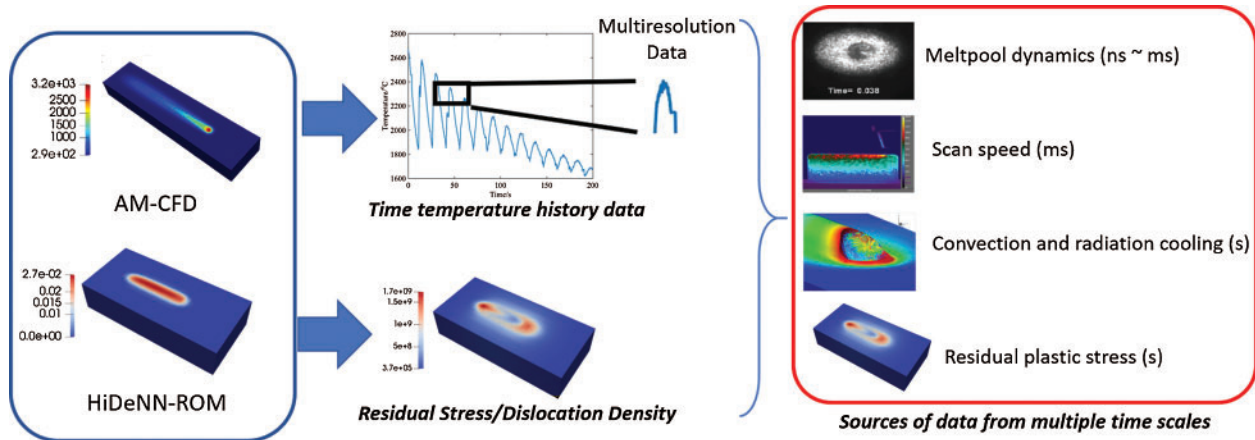
Metal additive manufacturing is considered as the manufacturing technology of the future. However, design and development of additively manufactured metallic part is thwarted by the highly non-equilibrium nature of the process and resulting microstructure. The variable nature of the microstructure at different locations in the part gives rise to uncertainty in the resulting mechanical properties. The process conditions also result in residual stress inside the part, which is controlled by the microstructure as well. In turn, the residual stress will affect the microstructural feature such as the grain formation and dislocation distribution. The porosity formed during the manufacturing process is another concern in additive manufacturing. These microstructural features are the determining factors or parameters for the final mechanical properties such as the ultimate tensile strength or fatigue crack incubation life. The microstructural features are dependent on various process parameters such as scan speed, laser power, hatch spacing, powder composition, etc. The process-structure-property chain development by computational aid requires a method that can efficiently compute the field (thermal and mechanical) quantities for a large number of parameters. The proposed reduced order machine learning FEM framework has an immense prospect in this regard.

The outline of the application in the context of additive manufacturing is shown in Fig. 10. The AM-CFD [25] is the computational fluid dynamics code that can faithfully simulate the manufacturing process as a function of different process input parameters. This code can be extended with HiDeNN-FEM concept. The thermal history coming from the AM-CFD can be combined with HiDeNN-ROM to compute the residual stress field. Fig. 11 illustrates an example of residual stress obtained by an advanced ROM [42], which agrees well with the FEM solution. The final computed residual stress in terms of, e.g., two input parameters and locations (different material points) can be written in the following form:

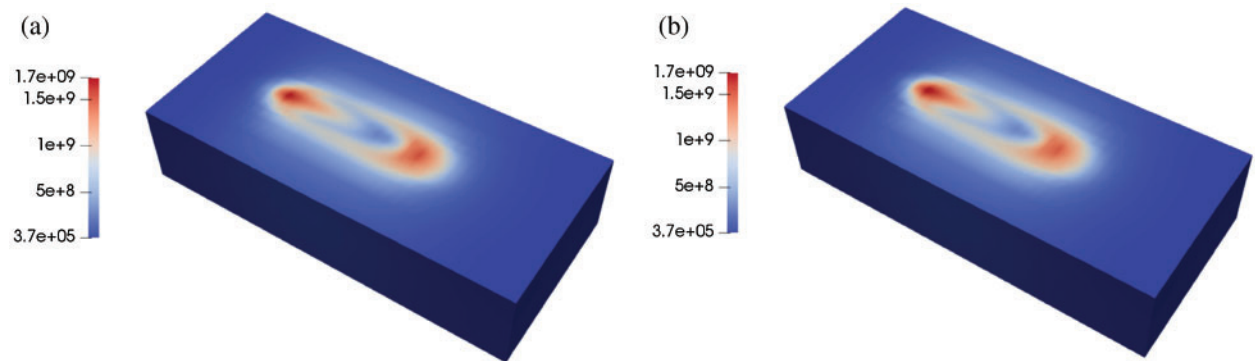
$$\sigma_{res}(x, \mu_1, \mu_2) \approx \sum_{q=1}^Q u_x^{(q)}(x) u_{\mu_1}^{(q)}(\mu_1) u_{\mu_2}^{(q)}(\mu_2) \quad (35)$$

Here,  $\sigma_{res}$  is the residual stress,  $x$  is the material point, and  $\mu_1, \mu_2$  are two manufacturing parameters. The resulting outcome (time-temperature history and residual stress) can be related to the microstructural features and different underlying physical phenomena. For example, the time temperature data holds the information coming from different time scales: nano to millisecond (melt pool dynamics), millisecond (scan speed), second (convection and radiation cooling and residual stress evolution). One way of relating this time-temperature history with mechanical

properties is to apply feature engineering such as wavelet transformation and convolutional neural network [43]. In order to use these data-science tools, we need a large high fidelity database that can be generated quickly with the proposed HiDeNN-ROM.



**Figure 10:** A schematic diagram showing the potential application of HiDeNN-ROM to metal additive manufacturing. The time-temperature history contains multi-resolution information embedded. This multi-resolution time-temperature history and residual stress data can be obtained from the HiDeNN-ROM



**Figure 11:** Additive manufacturing residual stress obtained by FEM and advanced ROM [42]. (a) FEM (b) Advanced ROM

**4.4 Multi-Physics Problems: Field Interaction Modeling of Robotic Materials**

The multiphysics problems involves a large number of parameters that need to be considered for optimization, uncertainty quantification, and design. These challenging problems can be addressed by the proposed HiDeNN-PGD and HiDeNN-ROM.

Robotic materials are a class of multifunctional materials that combines responses from different physical field interactions to achieve desired functions. Earlier robotic materials were fabricated using soft matter that exhibited highly nonlinear behavior under the influence of environmental factors such as temperature, pressure, magnetic, electric, and optical fields [44]. Later research with soft robotic materials pushed their limits by considering inclusions of 2D materials or

nanofillers, expanding the problem in multiple length scales. By design, these materials need to interact with external stimuli to attain necessary functionality. Therefore, the analysis needs to consider numerous design iterations involving solutions of multiple physical field interactions. In such scenarios, the use of conventional FEA methods is costly and a hindrance to effectively iterate through multiple length scales and multiple field interactions. The proposed HiDeNN-ROM framework can potentially handle the challenge of dimensional reduction by—first optimizing the nodal positions smartly and thus promoting the essence of  $r$  and  $h$  adaptivity [32], and secondly decoupling the field solvers in an attempt to accelerate the solution techniques using separation of variations as introduced in this work. After solving the problem, the field responses can assist in engineering the configuration for required functionality.

## 5 Conclusion

This paper proposes a new type of numerical method for analyzing the partial differential equations in computational science and engineering. The so-called reduced order machine learning FEM is a combination of scientific machine learning methods and model reduction techniques. We introduce a particular type of this method, called HiDeNN-PGD, which is actually a combination of HiDeNN-FEM and PGD based model reduction. Advantages of this synergy in accuracy and computational efficiency have been shown with an example of Poisson problems. It is found the HiDeNN-PGD is more accurate with a few modes than FEM and PGD. The potential applications of such method to real life problems have been discussed. We propose a well designed structure of the general HiDeNN-ROM for 1) the multi-scale analysis of composite materials, 2) topology optimization, 3) additive manufacturing. In particular, preliminary numerical results based on PGD-topology optimization have been reported for the first time. It is found that the proposed framework has a good potential to enable high resolution topology optimization at a significantly reduced cost. Finally, the application of the proposed method to multiphysics problems: the field interaction modeling of robotic materials, has also been discussed. We believe that the proposed reduced order machine learning FEM framework is a general and powerful tool for analyzing a large class problems with a high dimensional nature.

**Acknowledgement:** The authors would like to acknowledge the support of the National Science Foundation under Grant Nos. CMMI-1762035 and CMMI-1934367 and AFOSR under Grant No. FA9550-18-1-0381.

**Funding Statement:** WKL, YL, HL, SS, SM, AAA are supported by NSF Grants CMMI-1934367 and 1762035. In addition, WKL and SM are supported by AFOSR, USA Grant FA9550-18-1-0381.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

1. Belytschko, T., Liu, W. K., Moran, B., Elkhodary, K. (2013). *Nonlinear finite elements for continua and structures*. John Wiley & Sons.
2. Hughes, T. J. (2012). *The finite element method: Linear static and dynamic finite element analysis*. Courier Corporation.
3. Liu, W. K., Li, S., Park, H. (2021). Eighty years of the finite element method: Birth, evolution, and future. arXiv preprint arXiv:2107.04960.

4. Chen, Q., Xie, Y., Ao, Y., Li, T., Chen, G. et al. (2021). A deep neural network inverse solution to recover pre-crash impact data of car collisions. *Transportation Research Part C: Emerging Technologies*, 126, 103009. DOI 10.1016/j.trc.2021.103009.
5. van der Giessen, E., Schultz, P. A., Bertin, N., Bulatov, V. V., Cai, W. et al. (2020). Roadmap on multiscale materials modeling. *Modelling and Simulation in Materials Science and Engineering*, 28(4), 043001. DOI 10.1088/1361-651X/ab7150.
6. Liu, Z., Bessa, M., Liu, W. K. (2016). Self-consistent clustering analysis: An efficient multi-scale scheme for inelastic heterogeneous materials. *Computer Methods in Applied Mechanics and Engineering*, 306, 319–341. DOI 10.1016/j.cma.2016.04.004.
7. Yu, C., Kafka, O. L., Liu, W. K. (2021). Multiresolution clustering analysis for efficient modeling of hierarchical material systems. *Computational Mechanics*, 67(5), 1293–1306. DOI 10.1007/s00466-021-01982-x.
8. Shakoor, M., Kafka, O. L., Yu, C., Liu, W. K. (2019). Data science for finite strain mechanical science of ductile materials. *Computational Mechanics*, 64(1), 33–45. DOI 10.1007/s00466-018-1655-9.
9. Berkooz, G., Holmes, P., Lumley, J. L. (1993). The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics*, 25(1), 539–575. DOI 10.1146/annurev.fl.25.010193.002543.
10. Willcox, K., Peraire, J. (2002). Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal*, 40(11), 2323–2330. DOI 10.2514/2.1570.
11. Goury, O., Amsallem, D., Bordas, S. P. A., Liu, W. K., Kerfriden, P. (2016). Automatised selection of load paths to construct reduced-order models in computational damage micromechanics: From dissipation-driven random selection to Bayesian optimization. *Computational Mechanics*, 58(2), 213–234. DOI 10.1007/s00466-016-1290-2.
12. Lu, Y., Blal, N., Gravouil, A. (2018). Space–time pod based computational vademecums for parametric studies: Application to thermo-mechanical problems. *Advanced Modeling and Simulation in Engineering Sciences*, 5(1), 1–27. DOI 10.1186/s40323-018-0095-6.
13. Maday, Y., Rønquist, E. M. (2002). A reduced-basis element method. *Journal of Scientific Computing*, 17(1), 447–459. DOI 10.1023/A:1015197908587.
14. Rozza, G., Huynh, D. B. P., Patera, A. T. (2007). Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15(3), 1. DOI 10.1007/BF03024948.
15. Lu, Y., Jones, K. K., Gan, Z., Liu, W. K. (2020). Adaptive hyper reduction for additive manufacturing thermal fluid analysis. *Computer Methods in Applied Mechanics and Engineering*, 372, 113312. DOI 10.1016/j.cma.2020.113312.
16. Carlberg, K., Farhat, C., Cortial, J., Amsallem, D. (2013). The gnat method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows. *Journal of Computational Physics*, 242, 623–647. DOI 10.1016/j.jcp.2013.02.028.
17. Ryckelynck, D. (2009). Hyper-reduction of mechanical models involving internal variables. *International Journal for Numerical Methods in Engineering*, 77(1), 75–89. DOI 10.1002/nme.2406.
18. Ladevèze, P., Passieux, J. C., Néron, D. (2010). The latin multiscale computational method and the proper generalized decomposition. *Computer Methods in Applied Mechanics and Engineering*, 199(21–22), 1287–1296. DOI 10.1016/j.cma.2009.06.023.
19. Dureisseix, D., Ladevèze, P., Schrefler, B. A. (2003). A latin computational strategy for multiphysics problems: Application to poroelasticity. *International Journal for Numerical Methods in Engineering*, 56(10), 1489–1510. DOI 10.1002/nme.622.
20. Chinesta, F., Ladeveze, P., Cueto, E. (2011). A short review on model order reduction based on proper generalized decomposition. *Archives of Computational Methods in Engineering*, 18(4), 395–404. DOI 10.1007/s11831-011-9064-7.
21. Paillet, C., Néron, D., Ladevèze, P. (2018). A door to model reduction in high-dimensional parameter space. *Comptes Rendus Mécanique*, 346(7), 524–531. DOI 10.1016/j.crme.2018.04.009.
22. Lu, Y., Blal, N., Gravouil, A. (2018). Adaptive sparse grid based hopgd: Toward a nonintrusive strategy for constructing space-time welding computational vademecum. *International Journal for Numerical Methods in Engineering*, 114(13), 1438–1461. DOI 10.1002/nme.5793.



23. Lu, Y., Blal, N., Gravouil, A. (2018). Multi-parametric space-time computational vademecum for parametric studies: Application to real time welding simulations. *Finite Elements in Analysis and Design*, 139, 62–72. DOI 10.1016/j.finel.2017.10.008.
24. Lu, Y., Blal, N., Gravouil, A. (2019). Datadriven hopgd based computational vademecum for welding parameter identification. *Computational Mechanics*, 64(1), 47–62. DOI 10.1007/s00466-018-1656-8.
25. Gan, Z., Jones, K. K., Lu, Y., Liu, W. K. (2021). Benchmark study of melted track geometries in laser powder bed fusion of inconel 625. *Integrating Materials and Manufacturing Innovation*, 10, 177–195. DOI 10.1007/s40192-021-00209-4.
26. Saha, S., Kafka, O. L., Yu, C., Liu, W. K. (2021). Microscale structure to property prediction for additively manufactured IN625 through advanced material model parameter identification. *Integrating Materials and Manufacturing Innovation*, 10, 142–156. DOI 10.1007/s40192-021-00208-5.
27. Hornik, K., Stinchcombe, M., White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), 359–366. DOI 10.1016/0893-6080(89)90020-8.
28. Li, H., Kafka, O. L., Gao, J., Yu, C., Nie, Y. et al. (2019). Clustering discretization methods for generation of material performance databases in machine learning and design optimization. *Computational Mechanics*, 64(2), 281–305. DOI 10.1007/s00466-019-01716-0.
29. Mozaffar, M., Bostanabad, R., Chen, W., Ehmann, K., Cao, J. et al. (2019). Deep learning predicts path-dependent plasticity. *Proceedings of the National Academy of Sciences*, 116(52), 26414–26420. DOI 10.1073/pnas.1911815116.
30. Lu, X., Giovanis, D. G., Yvonnet, J., Papadopoulos, V., Detrez, F. et al. (2019). A data-driven computational homogenization method based on neural networks for the nonlinear anisotropic electrical response of graphene/polymer nanocomposites. *Computational Mechanics*, 64(2), 307–321. DOI 10.1007/s00466-018-1643-0.
31. Raissi, M., Perdikaris, P., Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378, 686–707. DOI 10.1016/j.jcp.2018.10.045.
32. Zhang, L., Cheng, L., Li, H., Gao, J., Yu, C. et al. (2021). Hierarchical deep-learning neural networks: Finite elements and beyond. *Computational Mechanics*, 67(1), 207–230. DOI 10.1007/s00466-020-01928-9.
33. Saha, S., Gan, Z., Cheng, L., Gao, J., Kafka, O. L. et al. (2021). Hierarchical deep learning neural network (HiDeNN): An artificial intelligence (AI) framework for computational science and engineering. *Computer Methods in Applied Mechanics and Engineering*, 373, 113452. DOI 10.1016/j.cma.2020.113452.
34. Zhang, L., Lu, Y., Tang, S., Liu, W. K. (2021). HiDeNN-PGD: Reduced-order hierarchical deep learning neural networks. arXiv preprint arXiv:2105.06363.
35. Feyel, F., Chaboche, J. L. (2000). Fe<sub>2</sub> multiscale approach for modelling the elastoviscoplastic behaviour of long fibre SiC/Ti composite materials. *Computer Methods in Applied Mechanics and Engineering*, 183(3–4), 309–330. DOI 10.1016/S0045-7825(99)00224-8.
36. Aage, N., Andreassen, E., Lazarov, B. S., Sigmund, O. (2017). Giga-voxel computational morphogenesis for structural design. *Nature*, 550(7674), 84–86. DOI 10.1038/nature23911.
37. Liu, Y., Saha, S., Suarez, D., Liu, W. K., Qian, D. (2021). HiDeNN-FEM: A seamless machine learning approach to nonlinear finite element analysis (in Preparation).
38. Gao, J., Mojumder, S., Zhang, W., Li, H., Suarez, D. et al. (2021). Concurrent n-scale modeling for non-orthogonal woven composite. arXiv preprint arXiv:2105.10411.
39. Bendsoe, M. P., Sigmund, O. (2013). *Topology optimization: Theory, methods, and applications*. Springer Science & Business Media.
40. Wang, M. Y., Wang, X., Guo, D. (2003). A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 192(1–2), 227–246. DOI 10.1016/S0045-7825(02)00559-5.
41. Aage, N., Giele, R., Andreasen, C. S. (2021). Length scale control for high-resolution three-dimensional level set-based topology optimization. *Structural and Multidisciplinary Optimization*. DOI 10.1007/s00158-021-02904-4.



42. Lu, Y., Mojumder, S., Saha, S., Liu, W. K. (2021). Extended tensor decomposition model reduction methods: Training, prediction, and design under uncertainty (in Preparation).
43. Xie, X., Bennett, J., Saha, S., Lu, Y., Cao, J. et al. (2021). Mechanistic data-driven prediction of as-built mechanical properties in metal additive manufacturing. *npj Computational Materials*, 7. DOI 10.1038/s41524-021-00555-z.
44. Jing, L., Li, K., Yang, H., Chen, P. Y. (2020). Recent advances in integration of 2D materials with soft matter for multifunctional robotic materials. *Materials Horizons*, 7(1), 54–70. DOI 10.1039/C9MH01139K.