Proceedings of the ASME 2022
International Design Engineering Technical Conferences and
Computers and Information in Engineering Conference
IDETC/CIE2022
August 14-17, 2022, St. Louis, Missouri

DETC2022-90163

MULTI-FIDELITY REDUCED-ORDER MODELS FOR MULTISCALE DAMAGE ANALYSES WITH AUTOMATIC CALIBRATION

Shiguang Deng [a], **Carlos Mora** [b], **Diran Apelian** [a], **Ramin Bostanabad** [b, *]
[a] ACRC, Materials Science and Engineering, University of California, Irvine, CA, USA
[b] Mechanical and Aerospace Engineering, University of California, Irvine, CA, USA

ABSTRACT

Predicting the fracture behavior of macroscale components containing microscopic porosity relies on multiscale damage models which typically ignore the manufacturing-induced spatial variabilities in porosity. This simplification is made due to the prohibitive computational costs associated with explicitly modeling spatially varying microstructures in a macroscopic component. To address this challenge, we propose a data-driven framework that integrates a mechanistic reduced-order model (ROM) with a calibration scheme based on latent map Gaussian processes (LMGPs). Our ROM drastically accelerates direct numerical simulations (DNS) by using a stabilized damage algorithm and systematically reducing the degrees of freedom via clustering. Since clustering affects local strain fields and hence the fracture response, we construct a multi-fidelity LMGP to inversely estimate the damage parameters of an ROM as a function of microstructure and clustering level such that the ROM faithfully surrogates DNS. We demonstrate the application of our framework in predicting the damage behavior of a multiscale metallic component with spatially varying porosity.

Keywords: multiscale damage analysis, data-driven calibration, reduced-order model, Gaussian processes, spatially varying microstructures.

1. INTRODUCTION

Predicting the effect of manufacturing-induced microscopic defects on the performance of macroscopic components relies on multiscale simulations where a microstructure or a representative volume element (RVE) is associated with each integration point (IP) of the discretized macrostructure. Traditional multiscale simulations use the finite element method (FEM) at both macroscale and microscale where macroscopic deformation gradients $\mathbf{F}^{\mathrm{M}}$ and RVE effective stress $\boldsymbol{\sigma}^{\mathrm{M}}_{\mathrm{FEM}}$ are exchanged between scales at each iteration, see Figure 1(a). A major challenge associated with such nested simulations is the computational expenses which prohibitively increase in the presence of nonlinear microscale deformations that involve damage. Reducing these costs holds the key to understanding the relation between microscopic defects and components' fracture behavior and, in turn, guiding the "design for fracture" process. To this end, we propose a data-driven framework that has two major components: (1) a mechanistic reduced-order model (ROM) with an adjustable degree of fidelity, and (2) a multi-fidelity modeling and calibration scheme based on latent map Gaussian processes (LMGPs). Integration of these two components enables us to build calibrated multi-fidelity ROMs that can simulate the damage behavior of multiscale materials with spatially varying microstructures.

The rest of the paper is organized as follows. In Section 2, we review existing works on reduced-order modeling and discuss the research gaps that we aim to address. The overview and technical details of our approach are provided in Sections 3 and 4, respectively. We evaluate the performance of our approach in Section 5 and conclude the paper in Section 6.

2. BACKGROUND ON REDUCED-ORDER MODELING

Mechanistic ROMs are increasingly employed to accelerate nonlinear material modeling by using a combination of methods from linear algebra and machine learning that result in reducing the number of unknown variables that characterize, e.g., microstructural strain and stress fields. Transformation field analysis (TFA) and its successor nonuniform transformation

*Corresponding author.
E-mail address: raminb@uci.edu.

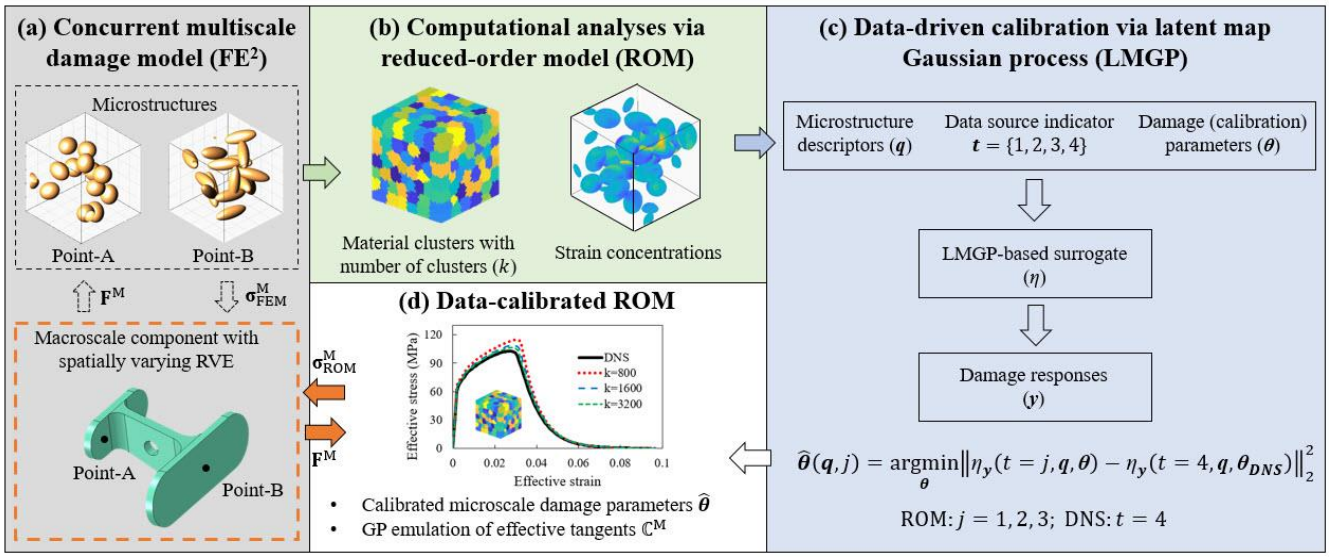

Figure 1 Proposed data-driven framework for multiscale damage modeling: LMGP creates a multi-fidelity emulator for the ROMs and DNS. It is then used in an inverse optimization to determine the damage parameters that must be used in ROMs such that they can approximate DNS as closely as possible. Upon this calibration, a multiscale simulation is run where ROMs are used at the microscale.

field analysis (NTFA) are two of the earliest ROMs [1–3]. These two methods approximate plastic strain as either piecewise constants or spatially varying orthonormal eigenstrains which are pre-selected in an offline stage. These eigenstrains evolve in the online stage based on pre-defined analytical functions that involve thermodynamic forces and potentials.

Clustering-based ROMs are recent techniques that decompose microstructure domains into a set of clusters whose interactions and deformations are modeled. For instance, the clusters in the self-consistent analysis (SCA) [4] method group material points with similar elastic responses and then quantifies cluster-to-cluster interactions by incremental Lippmann-Schwinger equation. Finite element-based cluster analysis [5] approximates the microstructural effective responses by following the cluster minimum complementary energy principle. Deflated clustering analysis (DCA) [6] agglomerates close-by IPs in clusters and the cluster-wise quantities of interests are computed in a multi-grid fashion where unknown variables are projected back and forth between different meshes. In this work, we use cluster-based ROMs as they provide higher efficiency compared to other methods such as TFA.

Successful application of any ROM depends on two primary factors: (i) the coarsening degree (e.g., the chosen number of clusters) which makes a tradeoff between *fidelity* level and computational costs, and (ii) the calibrated material properties. Both of these factors depend on the microstructure as well as the properties of interests. For example, accurate prediction of the damage behavior requires different damage parameters and the number of clusters for the two microstructures in Figure 1(a). In particular, given a desired level of accuracy with respect to high-fidelity direct numerical simulations (DNS), the analysis of the more complex microstructure in Figure 1(a) requires more clusters (i.e., less coarsening or data reduction).

Regarding calibration, we note that clustering material points diffuse stress/strain fields compared to DNS. This diffusion depends on the topology and unrealistically increases the tolerance of the material microstructure to localized phenomena such as damage. Hence, the material properties that characterize damage should be calibrated to counteract the superficial increase in material strength upon clustering. This reduction typically depends on the microstructure topology.

In this paper, we develop a data-driven framework to automate the process of selecting the degree of clustering (i.e., fidelity level) and calibration of ROMs.

3. OVERVIEW OF THE PROPOSED FRAMEWORK

Our framework relies on two primary components for damage modeling in multiscale metals with porosity: a novel cluster-based ROM and LMGP-based calibration which are detailed in Sections 4.4 and 4.3, respectively.

The ROM surrogates DNS and estimates the stress field in a microstructure under arbitrary displacement boundary conditions that may result in plasticity and damage. The fidelity of the ROM is determined by the user-defined parameter k which indicates the number of clusters and balances costs and accuracy.

As argued in Section 2, the material properties that must be used in ROM should be different than the true values that are used in DNS, i.e., the ROM requires calibration. This difference depends on both the microstructure complexity and, more importantly, k. Hence, we use a data-driven approach that relies on emulation via an LMGP to calibrate the material properties for ROMs. In particular, the trained LMGP enables answering the following question: *Given k and one microstructure, what damage parameters should be used in the ROM such that it predicts the same fracture response as DNS which uses the true damage parameters?* As explained below, answering this question relies on solving an inverse optimization problem whose objective function relies on LMGP, see Figure 1.

In practice, the above question is answered under two mild assumptions. Firstly, a small set of integer values are considered for k. In this work, we assume $k = 800, 1600,$ or 3200 but more values can be used within our framework. As shown in Section

4.3, all these values are much smaller than the number of elements in a typical mesh and hence result in massive data reduction or coarsening. Secondly, the very high dimensional morphology of microstructures is represented with a reduced set of quantitative descriptors that in our case characterize the geometry and spatial distribution of the pores.

We generate the samples for training the LMGP by design of experiments (DoE) where the inputs are microstructural descriptors and calibration parameters that control the damage behavior. For sample i, we first use a reconstruction algorithm to build the microstructure corresponding to the i^{th} set of descriptors. Then, we calculate the fracture response of the i^{th} microstructure (via DNS or one of the ROMs) while using the i^{th} set of damage parameters. We choose the frequency of using a simulator based on its costs, i.e., we employ an ROM with small k much more than DNS or an ROM with large k.

It is noted that the optimization problem uses LMGP rather than a traditional Gaussian process (GP) since we view the data source indicator as a categorical input rather than a quantitative one, see Figure 1(c). This choice is justified since alternating the data source (e.g., DNS vs. ROM with $k = 800$ vs. ROM with $k = 3200$) encodes the diffusive nature of strain-stress fields which cannot be readily characterized with quantitative inputs. Hence, our treatment of data source motivates the use of LMGP and greatly simplifies the metamodeling task as it eliminates the manual conversion of the source label to a quantitative variable.

Once LMGP is built, we are ready to run a multiscale simulation where ROMs are used at the microscale instead of DNS, see Figure 1(d). We first assign spatially varying microstructures to the IPs of the macro-component. Then, based on the complexity of the microstructures and any prior knowledge (if available) on the macro-locations where excessive deformations can occur (e.g., near sharp corners), we choose the k values for ROM. Next, we use the trained LMGP to assign the damage parameters that must be used at i^{th} macro IP given the k and microstructure assigned to it. Upon this assignment, we conduct the multiscale simulation to find the performance of the macro-component while considering microstructural porosities.

4. TECHNICAL DETAILS

We first provide the details on our ROM and how it can be used for damage modeling in Sections 4.1 through 4.3. Then, we elaborate on the training process of LMGPs in Section 4.4.

4.1. Stabilized micro-damage model for multiscale simulations

Damage includes strain-softening which causes convergence issues in implicit time integration schemes. To address this issue, we use a stabilized damage model [7] to simulate microstructural effective responses during fracture progression. This model decouples damage evolution from elasto-plasticity by introducing three reference RVEs that share state variables with the original damaged RVE. By tracing the elasto-plasticity in one of the referenced RVEs via a classic implicit scheme, the effective fracture stress and states can be mapped to the damaged RVE. Specifically, the homogenized damage stress in an arbitrary RVE can be written as:

$$\mathbf{S_M^d} = \mathbb{C}_{\mathrm{M}}^{\mathrm{d}}\mathbf{E_M^{el}} = \mathbb{C}_{\mathrm{M}}^{\mathrm{d}}(\mathbf{E_M} - \mathbf{E_M^{pl}}) \tag{1}$$

where $\mathbf{S_M^d}$ represents the effective damage stress, $\mathbb{C}_{\mathrm{M}}^{\mathrm{d}}$ is the homogenized macroscale elastic modulus, $\mathbf{E_M}$, $\mathbf{E_M^{el}}$ and $\mathbf{E_M^{pl}}$ are the RVE effective strain, elastic strain, and plastic strain, respectively. The subscript $\mathbf{M}$ indicates that the variable is a macroscopic quantity.

The first reference RVE is introduced to share the same elasto-plastic deformation as the original RVE but without the damage. Its effective stress is therefore computed as:

$$\mathbf{S_M^1} = \mathbb{C}^{\mathrm{el}}\mathbf{E_M^{el}} = \mathbb{C}^{\mathrm{el}}(\mathbf{E_M} - \mathbf{E_M^{pl}}) \tag{2}$$

where $\mathbf{S_M^1}$ and $\mathbb{C}^{\mathrm{el}}$ represent the homogenized stress and (undamaged) elastic modulus, respectively, and the number 1 in the superscript refers to the first referenced RVE. By combining Equations (1) and (2), we can express the referenced stress as:

$$\mathbf{S_M^1} = \mathbb{C}_{\mathrm{M}}^{\mathrm{d}}(\mathbb{C}^{\mathrm{el}-1}\mathbf{E_M^{el}}) \tag{3}$$

The second reference RVE is assumed to share the same effective stress ($\mathbf{S_M^2} = \mathbf{S_M^1}$) and material property as the first RVE but deform elastically. Thus, its effective elastic strain ($\mathbf{E_M^{el}}$) is:

$$\mathbf{E_M^{el}} = \mathbb{C}^{\mathrm{el}-1}\mathbf{S_M^2} = \mathbb{C}^{\mathrm{el}-1}\mathbf{S_M^1} \tag{4}$$

The effective stress and strain of the second reference RVE are equivalently expressed as the volume average of its microscale stress and strain as:

$$\mathbf{S_M^2} = \frac{1}{|\Omega|}\int_{\Omega}\mathbf{S_m^2}d\Omega \tag{5}$$

$$\mathbf{E_M^{el}} = \frac{1}{|\Omega|}\int_{\Omega}\mathbf{E_{m2}^{el}}d\Omega \tag{6}$$

where $|\Omega|$ is the RVE volume, the subscript $\mathbf{m}$ indicates that the variable is a microscopic quantity, and the microscale stress $\mathbf{S_m^2}$ is proportional to the microscale elastic strain $\mathbf{E_{m2}^{el}}$ via:

$$\mathbf{S_m^2} = \mathbb{C}^{\mathrm{el}}\mathbf{E_{m2}^{el}} \tag{7}$$

The third reference RVE has the same elastic strain as the second one ($\mathbf{E_{m3}^{el}} = \mathbf{E_{m2}^{el}}$) but its modulus is assumed to be identical to the original fractured RVE as:

$$\mathbf{S_m^3} = \mathbb{C}_{\mathrm{M}}^{\mathrm{d}}\mathbf{E_{m3}^{el}} \tag{8}$$

$$\mathbb{C}_{\mathrm{M}}^{\mathrm{d}} = (1 - \mathrm{D_m})\mathbb{C}^{\mathrm{el}} \tag{9}$$

where $\mathbb{C}_{\mathrm{M}}^{\mathrm{d}}$ is the damaged tangent moduli, and $\mathrm{D_m}$ is the damage parameter at a microscopic IP. The value of $\mathrm{D_m}$ is determined by the plastic strain states in the first reference RVE:

$$\mathrm{D_m}(\bar{\mathrm{E}}^{\mathrm{pl}}; \alpha, \bar{\mathrm{E}}^{\mathrm{cr}}) = 1 - \frac{\bar{\mathrm{E}}^{\mathrm{cr}}}{\bar{\mathrm{E}}_{\mathrm{m1}}^{\mathrm{pl}}}\exp(-\alpha(\bar{\mathrm{E}}_{\mathrm{m1}}^{\mathrm{pl}} - \bar{\mathrm{E}}^{\mathrm{cr}})) \tag{10}$$

where $\bar{\mathrm{E}}_{\mathrm{m1}}^{\mathrm{pl}}$ is the equivalent plastic strain at a microscale material point, α is the damage evolutionary rate parameter, and $\bar{\mathrm{E}}^{\mathrm{cr}}$ is the critical plastic strain. We note that local damage is initiated ($\mathrm{D_m} = 0$) when effective plastic strain equals the critical strain ($\bar{\mathrm{E}}_{\mathrm{m1}}^{\mathrm{pl}} = \bar{\mathrm{E}}^{\mathrm{cr}}$) and damage reaches total rupture ($\mathrm{D_m} = 1$) when

the effective plastic strain is much larger than the critical plastic strain.

The effective damaged stress of the original RVE is assumed to be equal to the homogenized stress of the third reference RVE and is calculated as:

$$\mathbf{S}_{\mathbf{M}}^{\mathbf{d}} = \mathbf{S}_{\mathbf{M}}^{\mathbf{3}} = \frac{1}{|\Omega|}\int_{\Omega} \mathbf{S}_{\mathbf{m}}^{\mathbf{3}} d\Omega \tag{11}$$

For the multiscale damage analysis in Section 5.4, the macroscale damage parameter is computed as the ratio of the norms of effective stress tensors of the original and the first reference RVE as:

$$\mathrm{D}_{\mathrm{M}} = 1 - \frac{\left\|\mathbf{S}_{\mathbf{M}}^{\mathbf{d}} : \mathbf{S}_{\mathbf{M}}^{\mathbf{1}}\right\|}{\left\|\mathbf{S}_{\mathbf{M}}^{\mathbf{1}} : \mathbf{S}_{\mathbf{M}}^{\mathbf{1}}\right\|} \tag{12}$$

where D_{M} is the homogenized damage parameter representing the fractured status of a macro-material point (and its associated RVE) on a macroscale component.

4.2. Condensation method

Every macroscopic IP in a multiscale simulation via the stabilized micro-damage model of Section 4.1 requires the tangent (elastic) modulus matrix ($\mathbb{C}^{\mathrm{el}}$), see Equation (2). Since we assign spatially varying RVEs with complex morphologies to macro IPs, $\mathbb{C}^{\mathrm{el}}$ needs to be computed via variational principles for each RVE [8]. This numerical procedure is needed since the constitutive laws of the RVEs are not available in closed form.

As variational calculations are expensive, we employ the condensation method [9] to compute the effective tangent moduli of an RVE. The condensation method starts by partitioning the microstructural system of equations as:

$$\begin{bmatrix} \mathbf{K}_{\mathbf{pp}} & \mathbf{K}_{\mathbf{pf}} \\ \mathbf{K}_{\mathbf{fp}} & \mathbf{K}_{\mathbf{ff}} \end{bmatrix} \begin{bmatrix} \delta\mathbf{u}_{\mathbf{p}} \\ \delta\mathbf{u}_{\mathbf{f}} \end{bmatrix} = \begin{bmatrix} \delta\mathbf{f}_{\mathbf{p}} \\ \mathbf{0} \end{bmatrix} \tag{13}$$

where $\delta\mathbf{u}_{\mathbf{p}}$ and $\delta\mathbf{u}_{\mathbf{f}}$ represent the displacement variations at the prescribed and free nodes in an RVE, and $\delta\mathbf{f}_{\mathbf{p}}$ is the external force on the nodes with prescribed forces. $\mathbf{K}_{\mathbf{pp}}, \mathbf{K}_{\mathbf{pf}}, \mathbf{K}_{\mathbf{fp}}$ and $\mathbf{K}_{\mathbf{ff}}$ are the corresponding partitions of the RVE's stiffness matrix.

Eliminating $\delta\mathbf{u}_{\mathbf{f}}$ from Equation (13) leads to a reduced system, with a reduced stiffness $\mathbf{K}_{\mathbf{r}}$ which directly relates the variations of the prescribed displacements with nodal forces:

$$\mathbf{K}_{\mathbf{r}}\delta\mathbf{u}_{\mathbf{p}} = \delta\mathbf{f}_{\mathbf{p}} \tag{14}$$

$$\mathbf{K}_{\mathbf{r}} = \mathbf{K}_{\mathbf{pp}} - \mathbf{K}_{\mathbf{pf}}(\mathbf{K}_{\mathbf{ff}})^{-1}\mathbf{K}_{\mathbf{fp}} \tag{15}$$

To transform $\mathbf{K}_{\mathbf{r}}$ to the tangent moduli that relate variations of stress and strain, we substitute Equation (14) into the variational form of the macroscopic stress:

$$\mathbf{S}_{\mathbf{M}}(\mathbf{X}) = \frac{1}{|\Omega_{0\mathrm{m}}|}\int_{\Gamma_{0\mathrm{m}}} \bar{\mathbf{t}}_{\mathbf{m}} \otimes (\mathbf{x} - \mathbf{x}_{\mathbf{0}}) d\Gamma \tag{16}$$

where $\boldsymbol{x}$ and $\boldsymbol{x_0}$ are the microscale IPs at the deformed and original configurations, $\mathbf{S}_{\mathbf{M}}$ is the macroscale stress at the macroscopic IP $\mathbf{X}$, $\bar{\mathrm{t}}_{\mathrm{m}}$ is the microscale surface traction, $\Gamma_{0\mathrm{m}}$ is the RVE boundary, and $\otimes$ denotes the tensor product between $\bar{\mathrm{t}}_{\mathrm{m}}$ and the position vector $(\boldsymbol{x} - \boldsymbol{x_0})$. Upon some algebraic modifications, the homogenized tangent (elastic) modulus matrix of an RVE can be obtained as:

$$\mathbb{C}^{\mathrm{el}} = \frac{1}{|\Omega_{0\mathrm{m}}|}\left[(\mathbf{x} - \mathbf{x}_{\mathbf{0}}) \otimes \mathbf{K}_{\mathbf{r}} \otimes (\mathbf{x} - \mathbf{x}_{\mathbf{0}})\right]^{\mathrm{LT}} \tag{17}$$

where'LT' denotes the transposition between the two left indices.

We note that even though the condensation method accelerates the calculation of $\mathbb{C}^{\mathrm{el}}$ for each RVE, parallel computations based on it in a multiscale analysis are memory demanding and still quite expensive. Hence, to avoid the online condensation procedure, we utilize a GP to learn the relation between microstructural morphology and effective elastic tangents for different RVEs which are pre-computed by the condensation method in an offline stage.

4.3. Deflated clustering analysis (DCA)

Computing the elasto-plastic response in the stabilized micro-damage algorithm (see Section 4.1) is needed for every microstructure. This computation is very expensive and so we use the DCA method [6] to dramatically accelerate the computations. The high efficiency of DCA comes from the fact that (1) the degrees of freedom are significantly reduced from a large number of finite elements to a few clusters by employing material clustering techniques, and (2) the algebraic system on the reduced system has better convergence behavior than the classic finite element system with much fewer close-to-zero eigenvalues.

DCA uses clustering to agglomerate neighboring finite elements to a set of interactive irregularly shaped clusters. Clustering is an unsupervised machine learning technique to interpret and group similar data. Among many mature clustering algorithms [10], we adopt k-means clustering [11] in this work due to its simplicity.

We start the k-means clustering by feeding the coordinates of element centers into a feature space where cluster seeds are randomly scattered and serve as initial cluster means. Then, we assign each element to the cluster with the closest mean. Meanwhile, cluster shapes are iteratively updated to minimize the within-cluster variance. Mathematically, the clustering can be stated as the following minimization problem:

$$\mathbf{S} = \underset{\mathbf{S}}{\operatorname{argmin}} \sum_{I=1}^{k} \sum_{n \in S^I} \left\|\varphi_n - \bar{\varphi}_I\right\|^2 \tag{18}$$

where $\mathbf{S}$ represents the k-clusters with $\mathbf{S} = \{S^1, S^2, \dots, S^k\}$. φ_n and $\bar{\varphi}_I$ are the coordinates of the n^{th} element center and the mean of the I^{th} cluster, respectively.

Upon clustering, we construct a reduced mesh by connecting cluster centroids via Delaunay triangularization where topological relations are preserved by checking the connectivity between clusters. We assume the motions of cluster centroids are directly related to the grouped nodes. Specifically, the displacement of the cluster centroid $\mathbf{u}(\mathbf{x})$ is computed by interpolating the nodal displacements via the polynomial augmented radial point interpolation method [12] as:

$$\mathbf{u}(\mathbf{x}) = \sum_{i=1}^{n} R_i(\mathbf{x}) a_i + \sum_{j=1}^{m} Z_j(\mathbf{x}) b_j \tag{19}$$

where a_i is the coefficient of the radial basis function R_i at the i^{th} FE node and b_j is the coefficient of the polynomial basis Z_j.

n and m are the number of cluster nodes and the number of polynomial basis functions, respectively. The coefficients a_i and b_j are determined by enforcing Equation (19) for all nodal displacements in the cluster where polynomial basis and radial coefficients are assumed to satisfy Equation (20) to ensure solution uniqueness [12]:

$$\sum_{i=1}^{n} Z_j(\mathbf{x}) a_i = 0, \qquad j = 1, 2, ..., m \tag{20}$$

We then augment the displacements of cluster centroids with rotational degrees of freedom to represent rigid body motions (three translations and three rotations in 3D) in a deflation space [19–21] where a reduced stiffness matrix is constructed with six degrees of freedom on each node. Performing nonlinear analyses on the reduced mesh and projecting the results back to the finite element nodes at the end of computations reads:

$$\mathbf{u}_i^j = \mathbf{W}_i^j \boldsymbol{\lambda}_i \tag{21}$$

where $\mathbf{u}_i^j$ is the displacement vector at the i^{th} node in the j^{th} cluster, $\boldsymbol{\lambda}_i$ is the rigid-body motion of the centroid of the j^{th} cluster, and $\mathbf{W}_i^j$ is the deflation matrix for the i^{th} node grouped in the j^{th} cluster:

$$\boldsymbol{\lambda}_j = \left[u_{jx}, u_{jy}, u_{jz}, \theta_{jx}, \theta_{jy}, \theta_{jz}\right]^{\mathrm{T}} \tag{22}$$

$$\mathbf{W}_i^j = \begin{bmatrix} 1 & 0 & 0 & 0 & z_i^j & -y_i^j \\ 0 & 1 & 0 & -z_i^j & 0 & x_i^j \\ 0 & 0 & 1 & y_i^j & -x_i^j & 0 \end{bmatrix} \tag{23}$$

where u_{jx} and θ_{jx} are the displacement and rotation of the j^{th} cluster along x axis, and (x_i^j, y_i^j, z_i^j) are the relative 3D coordinates of the i^{th} node with respect to the centroid of the j^{th} cluster.

We note that material points are assumed to share the same stress and strain values in each cluster. Hence, the local plastic strain fields are reproduced in a diffusive manner with lower strain concentrations which, in turn, delay the onset of localized fracture. This diffusive behavior motivates the damage parameter calibration using LMGP in the next section.

4.4. Latent map Gaussian Process (LMGP)

GPs are widely used in many applications for emulation [16–19]. The underlying idea of GP modeling is to assume that the data originate from a multivariate normal distribution. With this assumption, GP modeling involves considering a parametric form for the mean and covariance functions of the distribution and, in turn, estimating the parameters of these functions.

Traditional GPs cannot handle categorical inputs because covariance functions rely on the (weighted) distance between inputs while categorical inputs are not typically endowed with a distance measure. To address this limitation of GPs, we have recently developed LMGPs [20] that enable GPs to handle categorical inputs such as the data source indicator in our case. As we show in Section 5.2, the learned latent space of an LMGP provides a nice diagnostic tool that can guide the analysis and design process.

Assume the observations are produced by the single-response function $\eta(\boldsymbol{s})$ which is modeled as:

$$\eta(\mathbf{s}) = f(\mathbf{s})\beta + \xi(\mathbf{s}) + \varepsilon \tag{24}$$

where $\boldsymbol{f}(\boldsymbol{s}) = [f_1(\boldsymbol{s}), \dots, f_h(\boldsymbol{s})]$ is a vector of predefined parametric basis functions depending on the d_s dimensional input vector $\boldsymbol{s} = [s_1, s_1, \dots, s_{d_s}]^T$, $\boldsymbol{\beta} = [\beta_1, \dots, \beta_h]^T$ represent the unknown coefficients of the basis functions, ε is white noise, and $\xi(\boldsymbol{s})$ is a zero-mean GP with covariance function:

$$cov\left(\xi(\mathbf{s}), \xi(\mathbf{s}')\right) = c(\mathbf{s}, \mathbf{s}') = \sigma^2 r(\mathbf{s}, \mathbf{s}') \tag{25}$$

where $c(\cdot,\cdot)$ is the covariance function, σ^2 denotes the amplitude, and $r(\cdot,\cdot)$ is the correlation function. An example $r(\cdot,\cdot)$ is the Gaussian kernel given by:

$$\begin{aligned} r(\mathbf{s}, \mathbf{s}') &= exp\left\{-\sum_{i=1}^{d_s} 10^{w_i} \left(\mathbf{s_i} - \mathbf{s_i}'\right)^2\right\} \\ &= exp\left\{\left(\mathbf{s} - \mathbf{s}'\right)^T \boldsymbol{\Omega}_{\mathbf{s}} \left(\mathbf{s} - \mathbf{s}'\right)\right\} \end{aligned} \tag{26}$$

where $\boldsymbol{w} = [w_1, \dots, w_{d_s}]^T$ is the vector of roughness parameters and $\Omega_{\mathbf{s}} = diag(10^{\boldsymbol{w}})$. As it can be seen, $r(\cdot,\cdot)$ in Equation (26) does not accommodate categorical inputs as the distance between them is not defined.

To handle categorical inputs, LMGP maps them into a quantitative latent space which then makes it possible to use any distance-based correlation function. Specifically, let us denote the categorical inputs via $\boldsymbol{t} = [t_1, \dots, t_{d_t}]^T$ where variable t_i has m_i different levels. Upon mapping, LMGP uses the Gaussian correlation function as:

$$r(\mathbf{u}, \mathbf{u}') = exp\left\{-\left(\mathbf{s} - \mathbf{s}'\right)^T \boldsymbol{\Omega}_{\mathbf{s}} \left(\mathbf{s} - \mathbf{s}'\right) - \left\|\mathbf{z}(\mathbf{t}) - \mathbf{z}(\mathbf{t}')\right\|^2\right\} \tag{27}$$

where $\boldsymbol{u} = [\boldsymbol{s}; \boldsymbol{t}]$ and $\boldsymbol{z}(\boldsymbol{t}) = [z_1(\boldsymbol{t}), \dots, z_{d_z}(\boldsymbol{t})]^T$ is the learned d_z dimensional latent variable representing a particular combination of the categorical variables. $\boldsymbol{z}(\boldsymbol{t})$ is computed by mapping the representation of each combination of the categorical variables $\boldsymbol{\tau}(\boldsymbol{t})$ via:

$$\mathbf{z}(\mathbf{t}) = \boldsymbol{\tau}(\mathbf{t})\mathbf{A} \tag{28}$$

where $\mathbf{A}$ is the projection matrix that is estimated during training. Given a training dataset with n samples, the LMGP parameters (i.e., $\mathbf{A}, \boldsymbol{\beta}, \boldsymbol{w}$, and σ^2) are estimated by maximizing the log-likelihood function:

$$\left[\hat{\mathbf{A}}, \hat{\boldsymbol{\beta}}, \hat{\mathbf{w}}, \hat{\sigma}^2\right] = \underset{\mathbf{A}, \boldsymbol{\beta}, \mathbf{w}, \sigma^2}{\arg\max} \left\{ \begin{matrix} -\frac{n}{2}\log\left(\sigma^2\right) - \frac{1}{2}\log(|\mathbf{R}|) \\ -\frac{1}{2\sigma^2}\left(\mathbf{y} - \mathbf{F}\boldsymbol{\beta}\right)^T \mathbf{R}^{-1}\left(\mathbf{y} - \mathbf{F}\boldsymbol{\beta}\right) \end{matrix} \right\} \tag{29}$$

where $\log(\cdot)$ is the natural logarithm, $|\cdot|$ denotes the determinant operator, $\boldsymbol{y} = [y_{(1)}, \dots, y_{(n)}]^T$ are the n outputs in the training

data, $\boldsymbol{R}$ is the correlation matrix with entries $R_{ij} = r(\boldsymbol{u}_{(i)}, \boldsymbol{u}_{(j)})$, and $\boldsymbol{F}$ is the prior mean basis matrix with entries $F_{ij} = f_j(\boldsymbol{u}_{(i)})$.

Once the parameters are estimated, the predicted response at the query point $\boldsymbol{u}^*$ is obtained via:

$$\hat{y}(\mathbf{u}^*) = f(\mathbf{u}^*)\hat{\boldsymbol{\beta}} + \mathbf{g}^T(\mathbf{u}^*)\mathbf{V}^{-1}(\mathbf{y} - \mathbf{F}\hat{\boldsymbol{\beta}}) \tag{30}$$

where $\boldsymbol{g}(\boldsymbol{u}^*)$ is an $n \times 1$ vector with the i^{th} element $g_i(\boldsymbol{u}^*) = \hat{\sigma}^2 r(\boldsymbol{u}_{(i)}, \boldsymbol{u}^*)$, and $\boldsymbol{V}$ is the covariance matrix with entries $V_{ij} = \hat{\sigma}^2 r(\boldsymbol{u}_{(i)}, \boldsymbol{u}_{(j)})$.

5. NUMERICAL RESULTS

In this section, we apply the proposed data-driven framework to calibrate the ROMs in a multi-fidelity and multiscale model that simulates the damage behavior of a metallic component with spatially varying microstructures. In section 5.1 we train a GP that emulates the condensation method to accelerate the online calculation of $\mathbb{C}^{\text{el}}$ for each macroscopic IP during the multiscale simulation. In Section 5.2, we demonstrate the application of LMGPs in building a multi-fidelity emulator that is used in Section 5.3 to calibrate the damage parameters of the ROMs. Finally, we employ the calibrated ROMs in the multiscale simulation in Section 5.4.

The material studied in this work is cast aluminum alloy A356 whose elastic properties are:

$$E = 5.70\text{E}4 \text{ MPa}, \quad \nu = 0.33 \tag{31}$$

where E and ν are Young's modulus and Poisson's ratio, respectively. The alloy's behavior is modeled following the J2 plasticity theory with the piecewise linear hardening curve shown in Figure 2. We use an associative plastic flow rule with the following yield condition:

$$\bar{\sigma} \le \sigma_Y(\bar{\varepsilon}) \tag{32}$$

where $\bar{\sigma}$, $\bar{\varepsilon}$ and σ_Y are Mises equivalent stress, equivalent plastic strain, and yield stress, respectively.

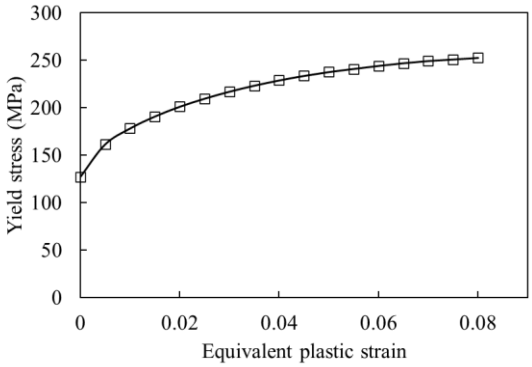

Figure 2 Hardening behavior: Piecewise linear hardening of A356 without damage.

The softening behavior of A356 is modeled by the progressive damage model in Equation (10) with two damage parameters: critical plastic strain ($\bar{\text{E}}^{\text{cr}}$) and damage evolutionary rate parameter (α). The following values are used in DNS while for ROMs they are calibrated based on the microstructural morphology and number of clusters:

$$\bar{\text{E}}^{\text{cr}} = 0.03; \quad \alpha = 100 \tag{33}$$

The proposed method is implemented in Matlab [21] and all simulations are performed on a high-performance cluster paralleled by 40 cores (AMD EPYC processor running at 4.1 GHz) with 120 GB RAM.

5.1. GP modeling for microstructure effective tangents

In damage analysis, the effective elastic tangent matrix plays a fundamental role in relating the effective reference stresses with elastic strains, see Equation (2). However, computing the effective tangents often involves intensive computational efforts even when condensation methods are applied.

To improve efficiency, we develop a GP surrogate to link microstructural morphologies (i.e., pore distribution) with the effective tangent matrix. Specifically, we approximate the complex pores via overlapping ellipsoids whose geometry and spatial distribution in an RVE are characterized by four descriptors including porosity volume fraction V_f, number of pores N_p, aspect ratio between ellipsoidal axes A_r, and average nearest neighbor distance between pore centroids $\bar{r}_d$. In addition, as we work with isotropic microstructural responses, the components of the tangent matrix are reduced to two effective Lame constants (μ and λ). Hence, the GP aims to build a predictive model between $[V_f, N_p, A_r, \bar{r}_d]$ and $[\mu, \lambda]$.

To construct the GP, we first generate a training dataset with 160 RVEs. The inputs in this dataset are generated via DoE where each sample specifies the value of $[V_f, N_p, A_r, \bar{r}_d]$ for an RVE. Then, we use a microstructure reconstruction algorithm [22] to build the RVE corresponding to each sample. Several reconstructed microstructures are shown in Figure 3 where the corresponding $[V_f, N_p, A_r, \bar{r}_d]$ values are enumerated in Table 1 . Next, we use the condensation method to calculate the Lame constants for each RVE. The GP is finally trained as described in Section 4.4. The parameter ranges used in DoE are as follows:

$$\begin{cases} 1\% \le V_f \le 20\% \\ 10 \le N_p \le 100 \\ 1 \le A_r \le 5 \\ 0.1\text{L} \le \bar{r}_d \le 0.5\text{L} \end{cases} \tag{34}$$

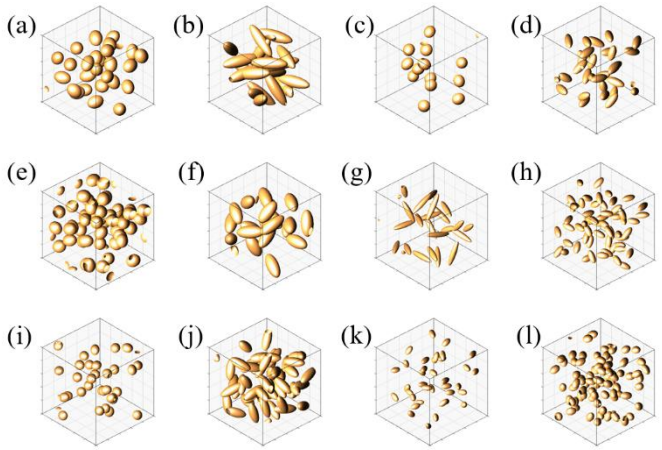

Figure 3 Example reconstructed microstructures: Pore descriptors and effective Lame constants are listed in Table 1.

Table 1 Pore descriptors and effective Lame constants: The numbers correspond to the reconstructed microstructures in Figure 3.

RVE	V_f	N_p	A_r	$\bar{r}_d$	μ (E10)	λ (E10)
(a)	6.56%	26	1.31	23.3	1.94	3.51
(b)	9.21%	20	3.33	19.7	1.82	3.05
(c)	2.06%	13	1.14	28.1	2.08	3.96
(d)	3.29%	29	2.37	20.5	2.03	3.78
(e)	9.97%	48	1.16	20.4	1.85	3.23
(f)	7.80%	20	2.15	25.9	1.89	3.31
(g)	1.92%	22	4.95	22.4	2.08	3.92
(h)	3.12%	60	2.11	16.9	2.04	3.81
(i)	2.61%	31	1.09	21.6	2.07	3.91
(j)	9.70%	51	2.47	18.2	1.82	3.09
(k)	1.15%	36	1.84	21.1	2.11	4.03
(l)	4.48%	77	1.43	14.5	2.01	4.03

To test our GP's accuracy, we split the dataset and use 80% for training and 20% for validation. Comparisons of the predictions against the validation samples are shown in Figure 4.

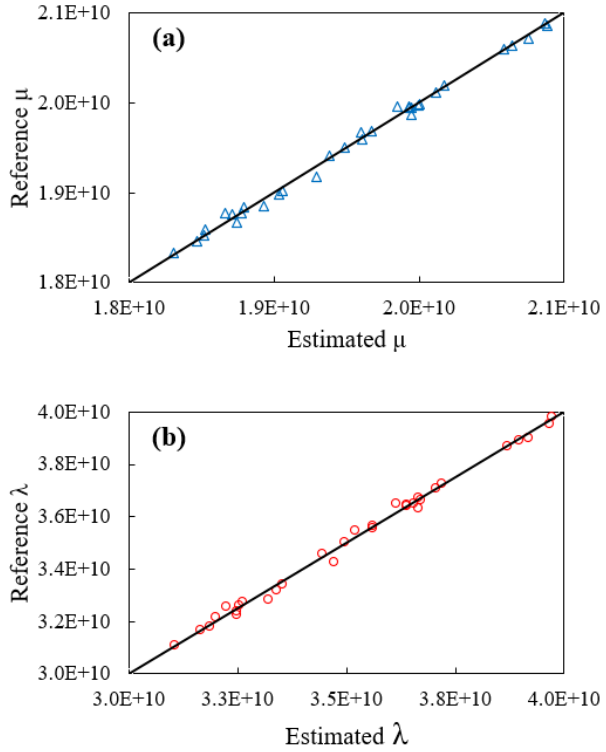

Figure 4 Emulation accuracy: Comparison of actual microstructural effective Lame constants against GP predictions on unseen test samples.

To assess the convergence and whether sufficient training data are used, we then split the dataset to 100 samples for training and 60 samples for testing. We sequentially increase the size of the training data from 10 to 100 and evaluate the accuracy of the corresponding GPs on 60 test samples (all GPs are evaluated on the same set of test samples). The prediction errors are computed by Equation (35) and shown in Figure 5:

$$E_{\mathbf{y}} = \frac{1}{N_v}\sum_{i=1}^{N_v}\frac{\left\|\hat{\mathbf{y}}_i - \bar{\mathbf{y}}_i\right\|}{\left\|\bar{\mathbf{y}}_i\right\|} \tag{35}$$

where N_v is the number of validation samples, $E_{\mathbf{y}}$ is the relative prediction error of responses $\boldsymbol{y} = [\boldsymbol{\mu}, \boldsymbol{\lambda}]$, $\hat{\mathbf{y}}_i$ and $\bar{\mathbf{y}}_i$ are the predicted effective constants for the i^{th} microstructure.

Through investigating Figure 5, we note that the errors monotonically decrease and that with almost 100 samples prediction error has converged. Following these observations, we fit a GP to the entire data and subsequently use it in our microscale damage analyses.

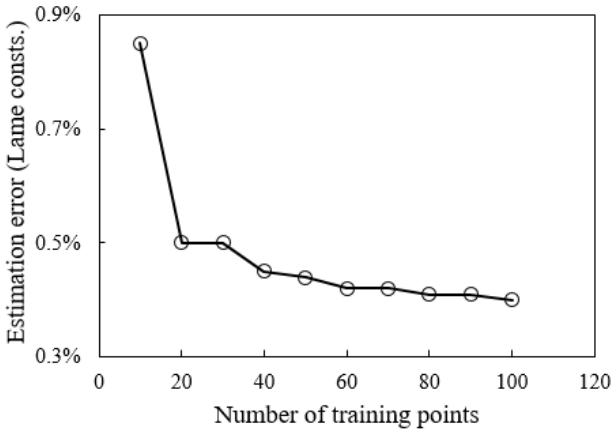

Figure 5 Error convergence: GP estimation errors on predicted Lame constants with respect to the number of training points.

5.2. LMGP modeling of damage model parameters

To showcase the importance of using LMGP for multi-fidelity modeling and calibration, consider the microstructure in Figure 6(a) whose damage parameters are defined in Equation (33). We deform this RVE to the deformation gradient in Equation (36) and obtain its response via DNS with 68675 elements. As shown in Figure 6(b), significant plastic strain concentrations appear in the vicinity of the pores. We then model the same microstructure via an ROM with 3200 clusters and with the same damage parameters as DNS. The results are shown in Figure 6(c) and clearly demonstrate the diffusive nature of clustering. Hence, when using the ROM the magnitude of local plastic strain is lower than DNS which results in delayed fracture initiation, larger material toughness, and higher ultimate tensile strength (UTS), see Figure 7(a).

ROM's accuracy can be improved by calibrating its damage parameters. We illustrate the effects of calibration on local strain concentrations and effective responses in Figure 6(d) and Figure 7(b), respectively. Compared to the ROM with the original damage parameters, the calibrated ROM provides more accurate estimations on both material toughness and ultimate tensile strength (UTS), see the enumerated errors and their norms in Table 2. However, manually calibrating microstructures with various morphologies and different fidelity levels (k) is time-consuming and suboptimal. Hence, we develop an LMGP-based calibration procedure to automatically find the optimal values of damage parameters.

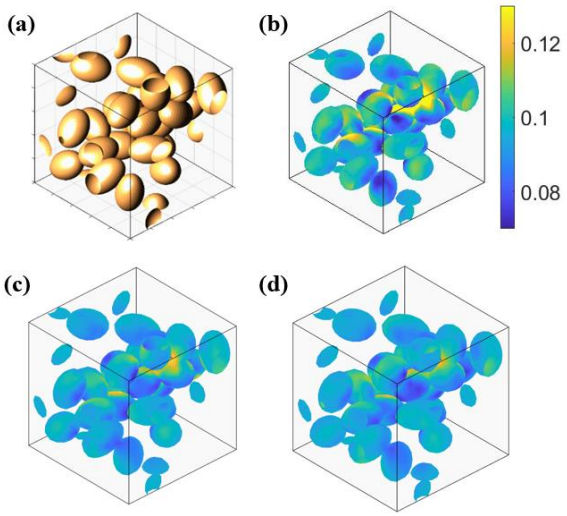

Figure 6 Equivalent plastic strain fields: (a) the porosity morphology of a microstructure with 25 pores, **(b)** plastic strain simulated via DNS, **(c)** plastic strain approximated by ROM (k=3200) without calibration, and **(d)** plastic strain approximated by ROM (k=3200) with calibration.

$$\mathbf{F}^{\mathrm{M}} = \begin{bmatrix} 1.1 & 0 & 0 \\ 0 & 0.95 & 0 \\ 0 & 0 & 0.95 \end{bmatrix} \tag{36}$$

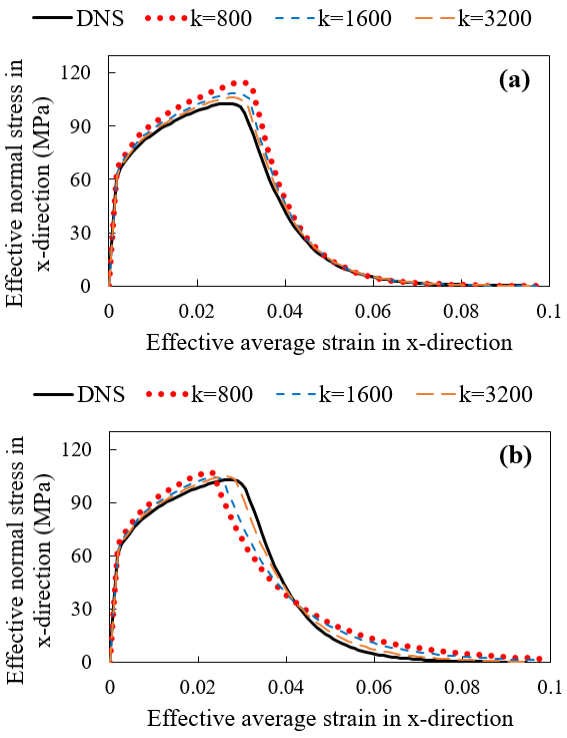

Figure 7 Importance of calibration: (a) effective stress-strain curves without damage parameters calibration, and **(b)** the effective response with calibration.

Table 2 ROM prediction errors: the errors in UTS and toughness are obtained by comparing the predictions to DNS.

ROM clusters (k)	Error w/o LMGP calibration (%)			Error with LMGP calibration (%)		
	UTS	Toughness	Error norm	UTS	Toughness	Error norm
800	11.5	10.6	15.61	3.74	3.64	5.22
1600	5.5	5.8	7.99	1.28	1.96	2.34
3200	3.2	3.7	4.86	1.55	1.73	2.33

To use LMGP for calibrating damage parameters, we generate a dataset that consists of six inputs $\boldsymbol{x} = [x_1, \dots, x_6]^T$ and two outputs $\boldsymbol{y} = [y_1, y_2]^T$, as shown in Table 3. The first four inputs represent the pore descriptors (i.e., $[V_f, N_p, A_r, \bar{r}_d]$) and the last two inputs represent damage parameters (i.e., evolutionary rate parameter α and critical effective plastic strain $\bar{\mathrm{E}}^{\mathrm{cr}}$). The outputs are the UTS and material toughness.

We append each sample with a categorical variable encoding data source, denoted by $t_1 = \{1,2,3,4\}$ where label 4 corresponds to DNS while labels 3, 2, and 1 correspond to ROM with $k = 3200$, $k = 1600$, and $k = 800$ respectively. To enable LMGP to simultaneously surrogate multiple responses, we also appended the samples with a second categorical variable encoding the type of outputs by $t_2 = \{1, 2\}$ where label 1 corresponds to UTS and label 2 indicates material toughness. The resulting training dataset is shown in Table 3.

We note that our dataset is highly unbalanced since we have fewer samples from high-fidelity sources which require intensive computational efforts. In particular, we have a total of 300 data points where only $n_h = 15$ samples are obtained via DNS while $n_{l1} = 45$, $n_{l2} = 90$, and $n_{l3} = 150$ samples are built via the ROM with 3200, 1600, and 800 clusters, respectively.

Once LMGP is trained, we can visualize the learned latent space where each combination of the two categorical variables is mapped to a point. The learned positions are demonstrated in Figure 8 and are consistent with our expectations. Specifically, the eight latent positions correspond to all possible combinations of the two categorical variables. The first digit of the label encodes the data source while the second label encodes the damage response. The latent points with the same responses are grouped by two vertical lines: while the points on the left correspond to the UTS, the points on the right are for the toughness. Therefore, we note that the four fidelity levels are described by vertical coordinates while the two responses are represented by horizontal coordinates.

From Figure 8, we also observe that the relative distances are directly related to the data sources' fidelity levels. For instance, the positions of $k = 3200$ (labels 31 and 32) are further from $k = 800$ (labels 11 and 12) than $k = 1600$ (labels 21 and 22), but the closet to DNS (labels 41 and 42). In addition, we observe that the scale of the horizontal axis is one order of magnitude larger than the vertical axis, indicating a higher correlation between fidelity levels than the types of responses.

The observation that the distance between DNS and $k = 800$ for both responses is around 0.02, suggests a large correlation ($\exp\{-0.02^2\} = 0.9996$) between the two data sources, see Equation (27). LMGP can therefore use any useful knowledge from low-fidelity data to improve its accuracy in

emulating the high-fidelity source (i.e., DNS). Additionally, we notice that the distance between two responses is about 0.6, which results in the correlation value of $\exp\{-0.6^2\} = 0.6977$. In other words, the two responses are positively correlated which coincides with our expectation, since the delayed fracture prediction of ROM not only increases UTS but also enlarges material toughness.

Table 3 LMGP's training dataset: There are four microstructure descriptors and two damage parameters (x_5 and x_6). The two categorical inputs distinguish data source and response type. The data are color-coded based on t_2 (green is UTS and blue toughness).

x_1	x_2	x_3	x_4	x_5	x_6	t_1	t_2	y
0.021	13	1.14	28.1	54.7	0.015	4	1	$1.12\cdot10^8$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0.066	26	1.31	23.3	71.2	0.017	4	1	$1.15\cdot10^8$
0.098	87	1.89	12.4	75.6	0.020	3	1	$1.13\cdot10^8$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0.045	77	1.43	14.5	80.7	0.023	3	1	$1.26\cdot10^8$
0.030	70	3.93	12.6	73.4	0.066	2	1	$1.21\cdot10^8$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0.026	31	1.10	21.6	98.3	0.029	2	1	$1.33\cdot10^8$
0.078	34	2.77	17.4	21.3	0.012	1	1	$1.08\cdot10^8$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0.016	88	3.13	14.4	61.7	0.027	1	1	$1.36\cdot10^8$
0.021	13	1.14	28.1	54.7	0.015	4	2	$3.14\cdot10^6$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0.067	26	1.31	23.3	71.2	0.017	4	2	$3.00\cdot10^6$
0.098	87	1.89	12.4	75.6	0.020	3	2	$3.26\cdot10^6$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0.045	77	1.43	14.5	80.7	0.023	3	2	$3.93\cdot10^6$
0.030	70	3.93	12.6	73.4	0.066	2	2	$3.07\cdot10^6$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0.026	31	1.10	21.6	98.3	0.029	2	2	$4.72\cdot10^6$
0.078	34	2.77	17.4	21.3	0.012	1	2	$3.17\cdot10^6$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0.016	88	3.13	14.4	61.7	0.027	1	2	$5.05\cdot10^6$

To assess LMGP's accuracy, we use 300 samples for training and 100 samples ($n_h = 5$, $n_{l1} = 15$, $n_{l2} = 30$, $n_{l3} = 50$) for validation. LMGP's prediction accuracy is quantified by mean squared error (MSE) in Table 4 where we observe that the surrogate's prediction error decreases as we use higher fidelity level sources. We compare LMGP's predictions with validation values in Figure 9 where we note the predictions for both UTS and toughness are quite accurate. Based on this figure, the predictions of UTS present larger errors than toughness. One plausible reason is that UTS, a point measurement of the maximum stress that an RVE can tolerate, is sensitive to some factors that are not captured in this model, e.g., crack propagation direction. However, RVE toughness which is a measurement of the amount of released energy during damage evolution can be characterized by our model's variables sufficiently well.

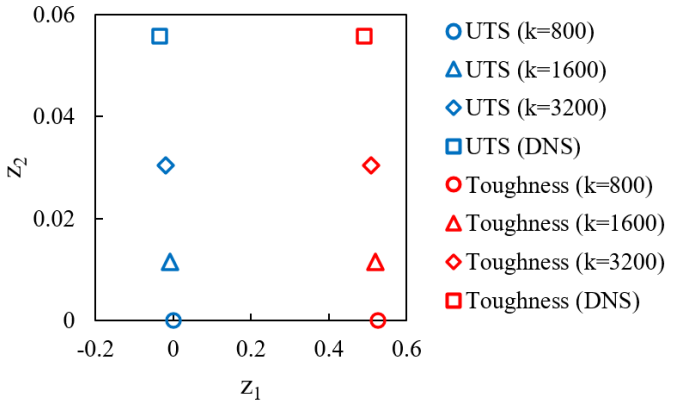

Figure 8 Learnt latent space of LMGP: Each latent position encodes simulation fidelity level and damage response.

Table 4 Error analysis: LMGP's prediction MSE for the two damage responses and four data sources.

Source	MSE	
	y_1 (UTS)	y_2 (fracture energy)
DNS	$6.3966\cdot10^{11}$	$7.4737\cdot10^8$
ROM with k = 3200	$7.1468\cdot10^{11}$	$4.3607\cdot10^8$
ROM with k = 1600	$3.2451\cdot10^{12}$	$3.5422\cdot10^9$
ROM with k = 800	$7.2695\cdot10^{12}$	$9.9943\cdot10^9$

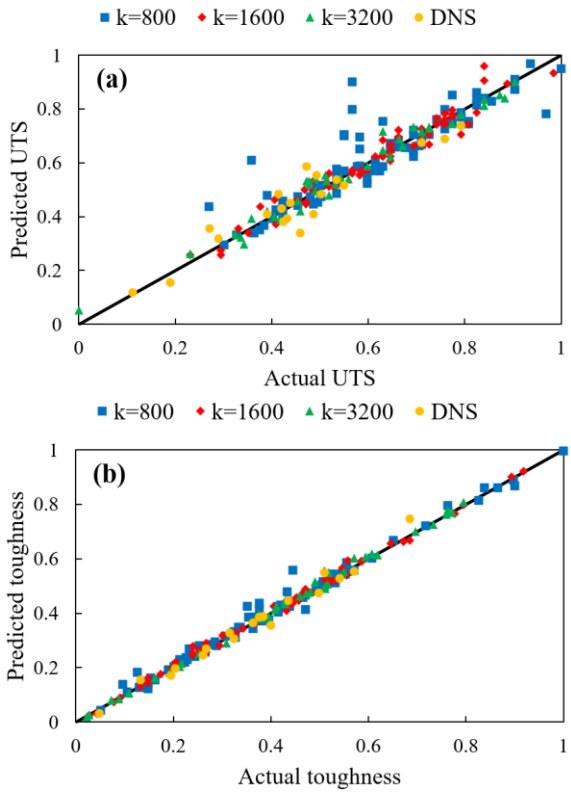

Figure 9 Performance on unseen test data: Comparison of the true responses against the LMGP's predictions for UTS and toughness.

5.3. Calibration of damage parameters

To calibrate the damage parameters for each ROM, we need to solve an inverse optimization problem whose objective function is evaluated via LMGP. We estimate the calibration parameters for the i^{th} microstructure and the j^{th} source-level such that the estimated damage responses from ROM match the ones from DNS that uses $\alpha_{DNS} = 100$ and $\bar{E}^{cr}_{DNS} = 0.03$. The optimization problem is hence formulated as:

$$\left[\hat{\alpha}, \bar{E}^{cr}\right] = \arg\min_{\alpha, \bar{E}^{cr}} \left\| \boldsymbol{y_p}\left(\boldsymbol{x}^i_{DNS}\right) - \boldsymbol{y_p}\left(\boldsymbol{x}^i_j\right) \right\|^2 \qquad (37)$$

where $\boldsymbol{y_p}(\cdot)$ are the predicted damage responses by LMGP and $\boldsymbol{x}^i_{DNS} = [V^i_f, N^i_p, A^i_r, \bar{r}^i_d, \alpha_{DNS}, \bar{E}^{cr}_{DNS}, t_1 = 4, \boldsymbol{t}_2]$ is the input vector of the i^{th} microstructure for predicting the responses of DNS. Analogously, $\boldsymbol{x}^i_j = [V^i_f, N^i_p, A^i_r, \bar{r}^i_d, \alpha, \bar{E}^{cr}, t_1 = j, \boldsymbol{t}_2]$ is the input vector of the i^{th} microstructure for predicting the damage responses for ROM at the j^{th} level (note that we pass $\boldsymbol{t}_2$ as a vector to get both responses).

We use a gradient-based optimization method to solve Equation (37). In Figure 10, we show the calibrated damage parameters for 20 randomly selected microstructures.

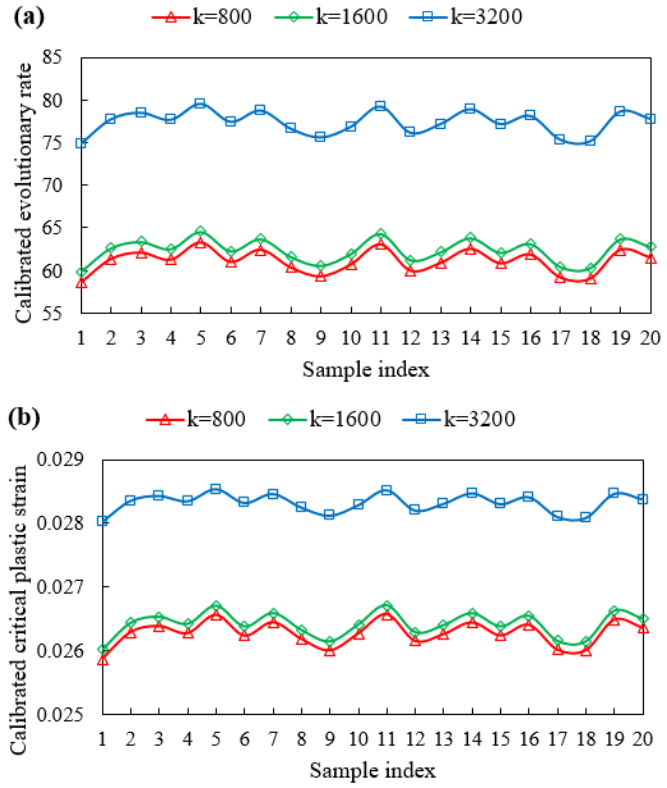

Figure 10 Calibrated ROM damage parameters: (a) evolutionary rate parameter, and **(b)** the critical effective plastic strain.

We observe the same trend across all samples: (i) the ROM's calibrated damage parameters are smaller than those of DNS, and (ii) the values of calibrated damage parameters get closer to DNS as we increase the number of clusters (k). To understand the underlying reason, we refer to Figure 7(a): as k decreases, the localized plastic strain is more diffusive than its DNS counterpart, resulting in a delay of damage initiation in the stress-strain curve. Therefore, to counteract this diffusive behavior, the calibrated damage parameters reduce the strength of the materials such that the ROM can faithfully approximate DNS.

5.4. Concurrent multiscale damage analyses

We apply the proposed multiscale damage model to a 3D L-shape bracket in this section to simulate the impact of micro-porosity on macroscopic fracture behavior. The dimensions of the L-bracket are shown in Figure 11 which is fixed on the top surface and is subject to a Dirichlet boundary condition on the right surface ($d = 20$mm). The bracket model is discretized with 2113 linear tetrahedron elements with reduced integrations.

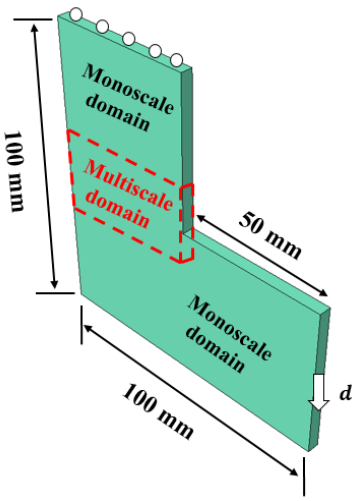

Figure 11 multiscale model: The dimensions and boundary conditions of the 3D L-shape bracket with a thickness of 5 mm.

For multiscale analysis, we divide the bracket into two subdomains: a monoscale region and a multiscale region with spatially varying porosity distribution. This choice is motivated by the observation that under large deformations the fracture happens in the multiscale domain (where high accuracy and microstructural effects are needed) and hence the other regions of the bracket can be modeled as a single scale.

For each of the 147 IPs in the multiscale region, we assume it is randomly associated with a microstructure from the database generated in 5.2. The effective damage behavior in each microstructure are simulated by ROM with three options for the number of clusters: 800, 1600, or 3200. For each ROM with a selected cluster number, its optimal damage parameters are readily available from the LMGP-based calibration process described in Section 5.3.

In our multiscale simulations, we ensure the released fracture energy is consistent between the scales by equating microstructure volumes to macroscopic mesh sizes. Additionally, we apply a nonlocal damage function with a feature size of 15mm on the bracket model to prevent pathological mesh dependency and convergence difficulty.

We demonstrate the simulated fracture pattern and load-displacement response (with and without multiscale treatment) in Figure 12. In Figure 12(a), fractures are represented by the effective damage values D_M from Equation (12) $D_M = 1$ represents complete rupture. From Figure 12(b), we observe that

porous microstructures significantly deteriorate the bracket's load-carrying capacity which drops by 10.22% from 70.86N to 63.62N, and the whole structure breaks at a much lower displacement boundary condition. Therefore, compared to the single-scale model that only considers dense materials and neglects pores, the multiscale model provides us with a more realistic prediction considering fractures across scales.

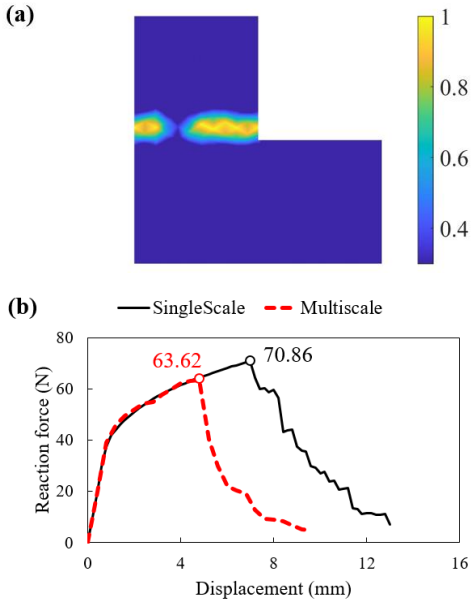

Figure 12 Results of the multiscale damage analysis: (a) top view of the fracture patterns on the L-bracket model, and **(b)** the force-displacement responses.

6. CONCLUSION

We propose a multi-fidelity reduced-order model for multiscale damage analysis that considers manufacturing-induced spatially-varying porosity. Our model is not only significantly faster than classic multiscale simulations but also has lower memory requirements. Our approach relies on a mechanics-based ROM that accelerates the microscale elasto-plastic deformations by clustering the degrees of freedom. Since this clustering increases the microstructure tolerance to damage initiation and evolution, we develop a calibration scheme to estimate the damage parameters that must be used in ROM such that it can faithfully approximate high-fidelity simulations.

We use LMGPs to build a multi-fidelity emulator and then use it in our calibration scheme. In addition to providing high accuracy and versatility, we show that the learned latent space of LMGP provides insights into the problem. In particular, we demonstrate the relative accuracy between the four simulators that model microstructural damage behavior.

Upon calibration of our ROMs, we use them in a multiscale simulation to study the effect of porosity on the macroscopic response of an L-bracket. Our results indicate that porosity noticeably decreases the strength of the material and hence must be considered in "design for fracture".

ACKNOWLEDGEMENTS

We acknowledge the support from National Science Foundation (award number OAC-2103708) as well as Early Career Faculty grant from NASA's Space Technology Research Grants Program (award number 80NSSC21K1809). The authors also thank the ACRC consortium members for their support.

REFERENCES

[1] G. J. Dvorak, "Transformation field analysis of inelastic composite materials," *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, vol. 437, no. 1900, pp. 311–327, May 1992, doi: 10.1098/rspa.1992.0063.

[2] J. C. Michel and P. Suquet, "Nonuniform transformation field analysis," *International Journal of Solids and Structures*, vol. 40, no. 25, pp. 6937–6955, Dec. 2003, doi: 10.1016/S0020-7683(03)00346-9.

[3] S. Roussette, J. C. Michel, and P. Suquet, "Nonuniform transformation field analysis of elastic–viscoplastic composites," *Composites Science and Technology*, vol. 69, no. 1, pp. 22–27, Jan. 2009, doi: 10.1016/j.compscitech.2007.10.032.

[4] Z. Liu, M. A. Bessa, and W. K. Liu, "Self-consistent clustering analysis: An efficient multi-scale scheme for inelastic heterogeneous materials," *Computer Methods in Applied Mechanics and Engineering*, vol. 306, pp. 319–341, Jul. 2016, doi: 10.1016/j.cma.2016.04.004.

[5] G. Cheng, X. Li, Y. Nie, and H. Li, "FEM-Cluster based reduction method for efficient numerical prediction of effective properties of heterogeneous material in nonlinear range," *Computer Methods in Applied Mechanics and Engineering*, vol. 348, pp. 157–184, May 2019, doi: 10.1016/j.cma.2019.01.019.

[6] S. Deng, C. Soderhjelm, D. Apelian, and R. Bostanabad, "Reduced-Order Multiscale Modeling of Plastic Deformations in 3D Alloys with Spatially Varying Porosity by Deflated Clustering Analysis," *Computational Mechanics, accepted*, doi: 10.1007/s00466-022-02177-8.

[7] Z. Liu, M. Fleming, and W. K. Liu, "Microstructural material database for self-consistent clustering analysis of elastoplastic strain softening materials," *Computer Methods in Applied Mechanics and Engineering*, vol. 330, pp. 547–577, Mar. 2018, doi: 10.1016/j.cma.2017.11.005.

[8] C. Miehe, "Numerical computation of algorithmic (consistent) tangent moduli in large-strain computational inelasticity," *Computer Methods in Applied Mechanics and Engineering*, vol. 134, no. 3, pp. 223–240, Aug. 1996, doi: 10.1016/0045-7825(96)01019-5.

[9] V. Kouznetsova, W. A. M. Brekelmans, and F. P. T. Baaijens, "An approach to micro-macro modeling of heterogeneous materials," *Computational Mechanics*, vol. 27, no. 1, pp. 37–48, Jan. 2001, doi: 10.1007/s004660000212.

[10] M. R. Ackermann, J. Blömer, D. Kuntze, and C. Sohler, "Analysis of Agglomerative Clustering," *Algorithmica*, vol. 69, no. 1, pp. 184–215, May 2014, doi: 10.1007/s00453-012-9717-4.

[11] A. Likas, N. Vlassis, and J. J. Verbeek, “The global k-means clustering algorithm,” *Pattern Recognition*, vol. 36, no. 2, pp. 451–461, Feb. 2003, doi: 10.1016/S0031-3203(02)00060-2.

[12] G. R. Liu, *Mesh Free Methods: Moving Beyond the Finite Element Method*, 1st edition. CRC Press, 2002.

[13] T. Jönsthövel, M. B. Gijzen, C. Vuik, C. Kasbergen, and A. Skarpas, “Preconditioned Conjugate Gradient Method Enhanced by Deflation of Rigid Body Modes Applied to Composite Materials,” *CMES - Computer Modeling in Engineering and Sciences*, vol. 47, pp. 97–118, Jul. 2009.

[14] R. Aubry, F. Mut, S. Dey, and R. Löhner, “Deflated preconditioned conjugate gradient solvers for linear elasticity,” *International Journal for Numerical Methods in Engineering*, vol. 88, no. 11, pp. 1112–1127, 2011, doi: 10.1002/nme.3209.

[15] P. Yadav and K. Suresh, “Large Scale Finite Element Analysis Via Assembly-Free Deflated Conjugate Gradient,” *Journal of Computing and Information Science in Engineering*, vol. 14, no. 4, Oct. 2014, doi: 10.1115/1.4028591.

[16] C. E. Rasmussen, “Gaussian Processes in Machine Learning,” in *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures*, O. Bousquet, U. von Luxburg, and G. Rätsch, Eds. Berlin, Heidelberg: Springer, 2004, pp. 63–71. doi: 10.1007/978-3-540-28650-9_4.

[17] R. Bostanabad, Y.-C. Chan, L. Wang, P. Zhu, and W. Chen, “Globally Approximate Gaussian Processes for Big Data With Application to Data-Driven Metamaterials Design,” *Journal of Mechanical Design*, vol. 141, no. 11, Sep. 2019, doi: 10.1115/1.4044257.

[18] R. Planas, N. Oune, and R. Bostanabad, “Extrapolation With Gaussian Random Processes and Evolutionary Programming,” presented at the ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Nov. 2020. doi: 10.1115/DETC2020-22381.

[19] R. Planas, N. Oune, and R. Bostanabad, “Evolutionary Gaussian Processes,” *Journal of Mechanical Design*, vol. 143, no. 11, May 2021, doi: 10.1115/1.4050746.

[20] N. Oune and R. Bostanabad, “Latent map Gaussian processes for mixed variable metamodeling,” *Computer Methods in Applied Mechanics and Engineering*, vol. 387, p. 114128, Dec. 2021, doi: 10.1016/j.cma.2021.114128.

[21] *“MATLAB. (2010). version 7.10.0 (R2010a). Natick, Massachusetts: The MathWorks Inc.”*

[22] R. Bostanabad *et al.*, “Computational microstructure characterization and reconstruction: Review of the state-of-the-art techniques,” *Progress in Materials Science*, vol. 95, pp. 1–41, Jun. 2018, doi: 10.1016/j.pmatsci.2018.01.005.