Quantifying accuracy of stochastic methods of reconstructing complex materials by deep learning

Serveh Kamrava,[1] Muhammad Sahimi,[1,*] and Pejman Tahmasebi[2]
[1]*Mork Family Department of Chemical Engineering and Materials Science, University of Southern California, Los Angeles, California 90089-1211, USA*
[2]*Department of Petroleum Engineering, University of Wyoming, Laramie, Wyoming 82071, USA*

(Received 17 January 2020; accepted 10 March 2020; published 3 April 2020)

Time and cost are two main hurdles to acquiring a large number of digital image **I** of the microstructure of materials. Thus, use of stochastic methods for producing plausible realizations of materials' morphology based on one or very few images has become an increasingly common practice in their modeling. The accuracy of the realizations is often evaluated using two-point microstructural descriptors or physics-based modeling of certain phenomena in the materials, such as transport processes or fluid flow. In many cases, however, two-point correlation functions do not provide accurate evaluation of the realizations, as they are usually unable to distinguish between high- and low-quality reconstructed models. Calculating flow and transport properties of the realization is an accurate way of checking the quality of the realizations, but it is computationally expensive. In this paper a method based on machine learning is proposed for evaluating stochastic approaches for reconstruction of materials, which is applicable to any of such methods. The method reduces the dimensionality of the realizations using an unsupervised deep-learning algorithm by compressing images and realizations of materials. Two criteria for evaluating the accuracy of a reconstruction algorithm are then introduced. One, referred to as the *internal uncertainty space*, is based on the recognition that for a reconstruction method to be effective, the differences between the realizations that it produces must be reasonably wide, so that they faithfully represent all the possible spatial variations in the materials' microstructure. The second criterion recognizes that the realizations must be close to the original **I** and, thus, it quantifies the similarity based on an *external* uncertainty space. Finally, the ratio of two uncertainty indices associated with the two criteria is considered as the final score of the accuracy of a stochastic algorithm, which provides a quantitative basis for comparing various realizations and the approaches that produce them. The proposed method is tested with images of three types of heterogeneous materials in order to evaluate four stochastic reconstruction algorithms.

DOI: 10.1103/PhysRevE.101.043301

I. INTRODUCTION

Materials with complex morphology, both human made and natural, are ubiquitous [1,2]. Characterizing their morphology—the shape and size of their microscopic elements and the way they are connected together—is a critical first step toward understanding the properties of complex materials and modeling the phenomena that occur there. Thus, over the past several decades a large set of microstructural descriptors has been developed theoretically [1,2] and applied to characterization of a wide variety of complex materials. At the same time, many models have also been proposed for describing the morphology of materials that, due to their complexity, their development has entailed making various simplifications and approximations.

With the tremendous advances in instrumentation, image-based characterization of complex materials, as well as their direct use in computing their properties is gradually becoming the preferred approach. For example, recent advances in imaging techniques have played a fundamental role in gaining deeper understanding of porous materials and their properties [3–15]. Producing high-quality images entails, however, investing a significant amount of time and resources. One fruitful approach to addressing this problem is based on developing computational methods by which one is able to use one or very few images in order to reconstruct a large ensemble of plausible realizations of the same materials, so as to gain a better understanding of the uncertainties associated with computing their properties based on their images. Development of such approaches, which are necessarily stochastic methods [16–21], has recently made considerable progress [22–34], to the point that they may be used to design new materials with novel properties [35].

Stochastic reconstruction approaches may be divided into three main groups: object-based, statistical, and image-based methods, all of which are based on the availability of a limited amount of experimental data. In other words, having access to a dataset is an essential aspect of all such methods. In the object-based methods, the morphological statistics are extracted from the available images in the form of deterministic values or probability distributions. Then, some of the properties are selected from their distributions and, using an optimization technique such as simulated annealing, the initial "objects"—patterns of the morphology—are generated and inserted in the simulation grid until all the predefined

*Corresponding author: moe@usc.edu

constraints on the morpholohy are satisfied. Object-based methods provide satisfactory results if the morphology is relatively simple.

In the statistical methods various spatial correlation functions describing the relationship between the microstructural properties are used to reconstruct realizations of a material [16,17,20–23,25,27,28,31–39] using an optimization technique. In order to capture specific features, various correlations functions, including two- [1,2,14] and three-point [15] correlation functions, have been developed and computed. Provided that the correlation functions include a measure of the connectivity of the clusters of the various phases of a multiphase material, the statistical methods often provide accurate reconstruction [22,26].

The most recent reconstruction algorithms are based on direct use of two- or three-dimensional (3D) images of materials. In these algorithms [25,27,30,40–42] digitized images **I** are directly sampled without extracting any particular statistics or correlations functions. Such methods are able to infer rich information from the images and, hence, are capable of producing high-quality realizations of materials and can be used with both pixel- and pattern-based reconstruction algorithms. The former generates each block in the computational grid separately, whereas the latter reconstructs a group of blocks together, which is computationally more efficient and mimics better extended correlations and connectivity in the microstructures.

Given the variety of the stochastic reconstruction methods, an important issue is a critical evaluation of their accuracy and efficiency. The evaluation may be in terms of the microstructural statistics that are not used in generating the realizations, or based on computing the physical properties of materials, such as their permeability and elastic strength. The latter comparison is, of course, precise but requires intensive computations. Furthermore, comparing the realizations generated by various reconstruction algorithms based on relatively simple statistics, such as the porosity, which do not often shed much light on the complexity of the microstructure, might also not be a valid way of evaluating them.

In this paper we describe an efficient methodology based on machine learning that allows evaluating and ranking of various stochastic algorithms for reconstruction of 2D or 3D discrete or multiphase and continuous images. The methodology evaluates the realizations generated by the reconstruction algorithms in order to identify the optimal method that produces models of the images with maximum similarity with the original **I**, and a reasonable range of variability between the realizations, so that they are not more or less identical. To this end, we develop an autoencoder deep-learning (DL) method in order to reduce the dimensionality of microstructural images. Then a quantitative measure is introduced for quantifying the performance of stochastic algorithms for materials reconstruction.

The reason for using the DL is its ability for capturing the latent complex features in the realizations. Such features cannot be captured or analyzed by the regular statistical methods [43], whereas the DL can represent them highly accurately. Furthermore, as we show below, the standard two-point descriptors, or even a multiple-point connectivity function, cannot accurately differentiate the differences between the realizations generated by various stochastic reconstruction methods, as they produce a slim uncertainty space around the **I**.

The rest of this paper is organized as follows. We first explain two microstructural descriptors that we will use to evaluate the realizations of a complex material. Section III explains the methodology that we propose based on machine learning. We then describe four stochastic reconstruction algorithms that are evaluated in this paper. The methodology is tested in Sec. V with three distinct types of materials and their images. Section VI provides a summary of the paper.

II. MICROSTRUCTURAL DESCRIPTORS

We first describe two microstructural descriptors [1,2] that have been used in some of the stochastic reconstruction of materials. To begin with, we define a phase-indicator function of materials consisting of phases 1 and 2 with volumes Ω_1 and Ω_2 and volume fractions φ_1 and φ_2, where $\Omega_1 \cup \Omega_2 = \Omega$ and $\Omega_1 \cap \Omega_2 = 0$. Then, the indicator function of phase i is defined by

$$I^{(i)}(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in \Omega_i \\ 0, & \mathbf{x} \in \overline{\Omega}_i \end{cases}, \tag{1}$$

with $I^{(1)}(\mathbf{x}) + I^{(2)}(\mathbf{x}) = 1$. The interface between the two phases is defined by the indicator function

$$M(\mathbf{x}) = |\nabla I^{(1)}(\mathbf{x})| = |\nabla I^{(2)}(\mathbf{x})|, \tag{2}$$

which is nonzero when $\mathbf{x}$ is on the interface.

An important microstructural descriptor is the lineal-path function $L_2^{(i)}(\mathbf{x}_1, \mathbf{x}_2)$ that provides information on the phase connectedness for short-range connectivities and has been used in many of the past reconstruction works. If we define a function,

$$\lambda^{(i)}(\mathbf{x}_1, \mathbf{x}_2, \alpha) = \begin{cases} 1, & \mathbf{x}_1,\ \mathbf{x}_2 \in \Omega_i(\alpha) \\ 0, & \mathbf{x}_1,\ \mathbf{x}_2 \in \overline{\Omega}_i(\alpha) \end{cases}, \tag{3}$$

for a sample α, then the lineal-path function for phase i is given by

$$L_2^{(i)}(\mathbf{x}_1, \mathbf{x}_2) = \langle \lambda^{(i)}(\mathbf{x}_1, \mathbf{x}_2, \alpha) \rangle, \tag{4}$$

where the averaging is over the samples α. The chord-length probability density function $L_c^{(i)}(z)$ for phase i is very similar to $L_2^{(i)}(z)$ in the sense that it represents a line segment with its interior points in one of the two phases and the end points on the interface between the two phases. More precisely, it represents the probability of finding a chord of length ℓ_c between two points in phase i, which is related to $L_2^i(z)$ via

$$L_2^{(i)}(z) = \frac{\varphi_i \int_0^\infty (y - z) L_c^{(i)}(y) H(y - z) dy}{\int_0^\infty y L_c^{(i)}(y) dy}, \tag{5}$$

where $H(x)$ is the Heaviside step function.

We also define a multiple-point connectivity function $p_m^{(i)}(\mathbf{h}; m)$ that quantifies the long-range connectivity of a material, as it represents the probability of having a sequence of m connected points in phase i of a material in a specific

direction **h** and is defined by

$$p_m^{(i)}(\mathbf{h};m) = \text{Prob}\{I(\mathbf{x})=1, I(\mathbf{x}+\mathbf{h})=1,\ldots, I(\mathbf{x}+m\mathbf{h})=1\}, \quad (6)$$

where $I(\mathbf{x})$ is the indicator function defined earlier. $p_m^{(i)}(\mathbf{h};m)$ accounts for curvilinearity and complexity in a microstructure, as it calculates the probability of finding multiple connected points by considering a tolerance core around a target direction.

III. THE DEEP-LEARNING METHODOLOGY

Unsupervised learning is a type of learning that helps identifying previously unknown patterns in datasets without pre-existing labels and is also known as self-organization that allows modeling of probability densities of given input data. In our work we use an autoencoder DL neural network (NN) to reduce the dimensionality of the realizations of materials, which is an unsupervised DL algorithm that utilizes backpropagation in order to set the target values to be equal to the inputs. \textrmThe algorithm maps the inputs $X = \{x^{(1)}, x^{(2)}, \ldots, x^{(i)}\}$, $x^{(i)} \in \Re^n$ onto the output $y^{(i)}$ by determining a function $h_{w,b}(x) \approx x$, where n is the number of pixels or voxels in the image and w and b denote the weights and biases (see below). Generally speaking, the NNs consist of two parts, namely the encoder Φ and the decoder Ψ, such that

$$\begin{aligned} \Phi &: \chi \rightarrow \mathbf{F}, \\ \Psi &: \mathbf{F} \rightarrow \chi, \\ (\Phi, \Psi) &= \arg\min_{\Psi,\Phi} \|X - (\Psi \circ \Phi)X\|^2, \end{aligned} \quad (7)$$

where $\circ$ denotes the convolution operation. **F** represents the feature map or the latent layer. Recall that the encoder operator Φ compresses each realization to just three numbers (x, y, z) that, in the present work, is the same as **F**. The decoder Ψ starts from the three "coordinates" (x, y, z) and reconstructs the initial images in the second phase. As described in Ref. [44], the autoencoder attempts to learn the function $h_{w,b}(x)$, which is an approximation to the identity function, in order to produce an output similar to $x^{(i)}$. Although the identity function may seem trivial to learn, imposing constraints on the NN by, for example, limiting its number of hidden units, enables one to gain useful insights into the image and discover features in the input data. We shall return to this point shortly.

Before describing further the idea behind the autoencoder DL, we point out that the convolutional NNs (CNNs) consist of a few layers, with the first one representing the input image. Next is the convolutional layer that consists of a set of internal layers that extract important features of the image and reduce the dimensionality of the data through a convolution function. To do so, the CNN needs a kernel function $\mathcal{A}(x)$, for which various forms have been used in the past for generating new features by considering a small shift (stride). One of them is the rectified linear unit (ReLU) that adds a small degree of nonlinearity to the convolved feature map, or the model, which we use in the present study. As the ReLU we used, $\mathcal{A}(x) = \max(0, x)$, which eliminates all the negative values since $\mathcal{A}(x) = 0$ if $x < 0$. ReLU is only mildly nonlinear due to the differences between the shapes of $\mathcal{A}(x)$ for $x < 0$ and $x > 0$.

An important problem with the output feature maps is that they are sensitive to the features' location in the input image. To address the problem one may down-sample (coarsen) the feature maps to reduce their dimensionality, hence making them more robust to changes in the features' positions. The pooling layer (PL), the next layer in the hierarchy of the layers in a CNN, reduces the dimensions of the image further, as it passes through it. It condenses the feature maps by simplifying the information and, hence, reducing the computation, especially when one has a large number of features in the data. In particular, the computation time in the regression stage of the NN is reduced, where overfitting is a major concern. The PL acts on each feature map separately in order to generate a new set of the same number of pooled feature maps, for which a pooling operation, much like a filter, must be selected and applied to the maps. The size of the pooling operation is smaller than that of the feature map. Note that the pooling operation is specified by the user, rather than being learned. Two common functions used in the pooling operation are average pooling, which calculates the average value for each patch on the feature map, and maximum pooling, or max-pooling, which computes the maximum value for each patch of the feature map. In the present paper we use the latter for down-sampling. Afterward, the extracted and summarized features will be connected to a multilayer perceptron network to produce an output image.

If the activation of hidden unit j in the autoencoder, using an input x, is represented by $\mathcal{A}_j^{(\ell_h)}$, then

$$\hat{\rho}_j = \max\left\{\mathcal{A}_j^{(\ell_h)}[x^{(i)}]\right\}, \quad i = 1, 2, \ldots, m, \quad (8)$$

represents the maximum activation of neuron j, where m is the number of training examples and ℓ_h denotes the hidden layer in which the neuron is located. Then the goal is to satisfy the following constraint:

$$\hat{\rho}_j = \rho, \quad (9)$$

where ρ is a parameter to be reproduced as the result of the activation of neuron j. To do so, one adds an extra penalty term to the cost function, the function to be minimized for the difference between the $\hat{\rho}_j$ and ρ, in order to control the difference between the two. One possible penalty term is the Kullback-Leibler (KL) divergence [45], sometimes called the relative entropy, which is a measure of how one probability distribution is different from a second reference probability distribution and is given by

$$D_{\text{KL}}(\rho||\hat{\rho}_j) \sum_{j=1}^{s_2} \rho \log\frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j}, \quad (10)$$

where s_2 represents the number of neurons in the hidden layer. Note that $D_{\text{KL}}(\rho||\hat{\rho}_j)$ increases monotonically as $\hat{\rho}_j$ deviates from ρ and, therefore, $D_{\text{KL}}(\rho) = 0$. With D_{KL} as the penalty term, the overall cost function to be minimized is then given by

$$C_O(W, b) = C(W, b) + \beta_p \sum_{j=1}^{s_2} D_{\text{KL}}(\rho\|\hat{\rho}_j), \quad (11)$$

where $C(W, b)$ is an average sum of the square errors for activation of the hidden layer with respect to the input data, with the error being the difference between the output of the NNs and the actual values, which is calculated by

$$C(W, b) = \frac{1}{2m} \sum_{i=1}^{m} ||O_{w,b}[x^{(i)}] - y^{(i)}||^2. \tag{12}$$

Here $O_{w,b}$ is the output of the NN for the given input data $x^{(i)}$, $y^{(i)}$ are the actual values (or labels), and β_p and b are the weights and biases that are adjusted by the CNN in order to reduce the error produced by it. The weight of the penalty term, which was taken to be $\beta_p = 0.5$, controls how strongly the term influences $C_O(W, b)$.

Using the autoencoder DL, one reduces the dimensionality of the images to some manageable scale. In fact, this is the main role of the DL in this study, i.e., transforming, for example, a 500×500 matrix, to a 1×3 one, which, computationally, is very significant. Although, as described below, we have implemented the algorithm with 2D images, the DL can accomplish the same for 3D realizations just as efficiently. None of the current methods for dimensionality reduction can do the same for large matrices or images **I**, as their output for such severe dimensionality reduction is completely distorted. But, due to its use of an iterative scheme and its ability for discovering latent patterns, the DL is capable of doing so by an advanced approach and producing very accurate results.

As a simple example, consider [44] a 10×10 image with the input data being the pixel values. Thus, $n = 100$, and we assume that there are $s_2 = 50$ neurons in the hidden layer of the NN. With only 50 neuron in the hidden layer, the NN must learn a "compressed" representation of the input image with $n = 100$. That is, given only the vector of the hidden unit activation in $\Re^{50}$, it is forced to reconstruct the input image with 100 pixel values. If the pixel values are completely random with no correlations between them, then the compression and reconstruction would be very difficult, if not impossible. In practice, however, any real material, and thus its image, contains correlations in its morphology and, therefore, the algorithm can discover at least some of them. Thus, the encoder compresses the input image into a latent-space representation, while the decoder reconstructs the initial image from the representation, which is why the two NNs with suitable constraints are excellent tools for dimensionality reduction (compression), as well as learning data projection. Schematic representation of the networks is shown in Fig. 1.

The outcome is a set of points that represents an image with a lower dimension. Thus, after acquiring the point data, similarly to analysis of variance in statistics, we propose to quantify two types of variabilities or uncertainty spaces: (i) internal, which represents the variability or the uncertainty space *between the realizations*, R_e, and (ii) external, which is the uncertainty space for the differences between R_e and *the input*—the **I**. The most accurate reconstruction algorithm is then one that produces very different realizations that share the basic features with **I**. In other words, producing a set of diverse realizations that do not have any common features with the **I** is not a useful exercise; rather the aim is to maximize the internal variability *between* the realizations, so as to generate as many plausible realizations of a complex material and, at the same time, minimize their differences with the **I**. The reconstruction algorithms are then ranked based on such concepts.

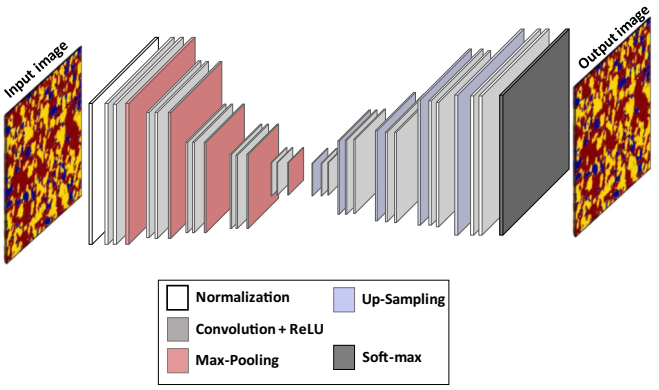

FIG. 1. Schematic representation of an autoencoder deep-learning algorithm.

We define the internal variability as the average distance between all pairs of the realizations,

$$\zeta_I = \frac{\sum_{i=1}^{N_r} \sum_{j=1}^{N_r} \mathcal{D}(R_{ei}, R_{ej})}{N_r(N_r - 1)}, \tag{13}$$

where $\mathcal{D}$ is the Euclidean distance between two points in a pair of realizations and N_r is the number of realizations. Similarly, the external variability is quantified by

$$\zeta_E = \frac{\sum_{i=1}^{N_r} \mathcal{D}(R_{ei}, \mathbf{I})}{N_r}. \tag{14}$$

The final score s for each reconstruction algorithm is then defined by

$$s = \frac{\zeta_I}{\zeta_E}. \tag{15}$$

An accurate reconstruction method generates realizations that have high internal variability ζ_I and small external variability ζ_E. Thus, in principle, the higher s, the more accurate are the reconstruction method and the realizations that it produces. Note that computation of ζ_I and ζ_E are done after the DL algorithm compresses the realizations and extracts their most important features.

IV. THE RECONSTRUCTION METHODS

Using the algorithm described in the previous section, we carried out computations with four distinct reconstruction methods. They are the cross correlation-based simulation (CCSIM) method, the single-normal equation simulation (SNESIM) algorithm, the sequential indicator simulation (SISIM) approach, and what is referred to as the filter simulation (FILTERSIM). The four methods were originally developed for reconstructing geomaterials but, similarly to all the reconstruction methods, they can be used for generating plausible realizations of any type of complex material and media, given a limited amount of data or one or a few of their images. The same type of computations can, obviously, be carried out with the reconstruction methods that use, for

example, simulated annealing (see, e.g., Yeong and Torquato [17,18]; Hamzehpour *et al.* [21]; Jiao *et al.* [22,23,26]). One motivation for carrying out the computations for the aforementioned algorithms is that they were originally developed for reconstructing realizations of large-scale porous media that are typically highly heterogeneous and, thus, complex, and for which the authors have considerable experience. In addition, the aim of this paper is not to evaluate and rank any specific reconstruction algorithm but only to compare various stochastic algorithms based on the method proposed in order to demonstrate how the proposed concepts numerically rank reconstruction methods when they produce very different or similar realizations. In what follows, we describe briefly each of the four methods.

A. The CCSIM algorithm

We describe the CCSIM algorithm, developed recently [25,27,40,41], for 2D images of materials; its extension to 3D will then be clear. In the CCSIM approach the realizations are represented by computational grids **G** that are divided into overlapping blocks of sizes $T_x \times T_y$. The **I** is also partitioned into blocks whose number and sizes are the same as those of **G**. The neighboring blocks share overlap regions **OL** with sizes $\ell_x \times \ell_y$. Beginning from any block of **G**, the algorithm visits each grid block along a one-dimensional raster path, selects at random a pattern of heterogeneity from the **I** for each grid block, and inserts it in the visited block. The inserted pattern is referred to as the *data event* $\mathbf{D}_T$, with the word "event" implying that the inserted pattern in the block may change again during reconstruction. Then, the next pattern is selected based on the similarity between the neighborhood blocks and the **I**, meaning that, instead of considering all the previously reconstructed blocks, only those in the neighborhood of the current blocks are used for the calculations. Next, the similarity between the neighboring blocks and the **I** is quantified based on a cross-correlation function $\psi(i, j; x, y)$ that represents a convolution between the **I** and $\mathbf{D}_T(x, y)$:

$$\psi(i, j; x, y) = \sum_{x=0}^{\ell_x - 1} \sum_{y=0}^{\ell_y - 1} \mathbf{I}(x + i, y + j)\mathbf{D}_T(x, y), \qquad (16)$$

with $i \in [0, T_x + \ell_x - 1)$ and $j \in [0, T_y + \ell_y - 1)$,

Thus, one uses an overlap region of size $\ell_x \times \ell_y$ between two neighboring blocks and a data event $\mathbf{D}_T$ to match the patterns in the **I**. The overlap region contains a set of pixels or voxels that one picks from the previously constructed blocks and utilizes them in Eq. (15) for identifying the next pattern of heterogeneity. As was shown in the original papers that developed the method, in order to minimize the Euclidean distance (difference) between the constructed blocks and the data, $\psi(i, j; x, y)$ must be maximum or, in practice, larger than a preset threshold. After calculating $\psi(i, j; x, y)$ for various patterns and selecting those for which it exceeds the threshold, one of the acceptable ones is selected randomly and inserted in the block currently being visited in **G**. The process is repeated until all blocks of **G** are reconstructed. Typically, the size of the **OL** regions is about 1/5–1/6 of the blocks' size.

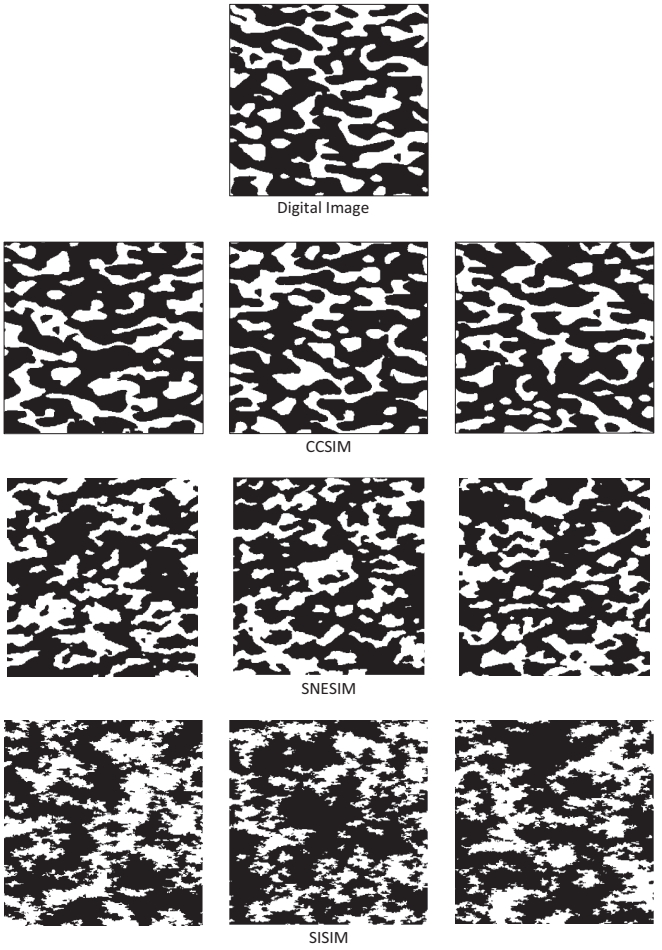

FIG. 2. Comparison between the original digital image **I** of a material with a binary microstructure and the realizations generated by three stochastic reconstuction algorithms.

B. The SNESIM algorithm

The algorithm was proposed by Strebelle [46]. Briefly, one scans the **I** once, computes all the conditional probabilities for a given pixels' (voxels') configuration (the template), and stores them in a dynamic search tree. In this way, there is no need to rescan the entire **I** in order to reconstruct a block, as was done in the methods that had been developed prior to Strebelle's, because one can directly use the tree structure to retrieve the conditional probabilities. At each step of the reconstruction the method utilizes the original data and the previously reconstructed grid blocks in order to advance. The algorithms can be used only with discrete multiphase **I** [47] (see below) and was improved significantly by Straubhaar *et al.* [48], who used a structure list for storing the data, instead of a search tree. Since, compared with a tree, a list is much less computationally demanding, the method improved on the SNESIM.

C. The SISIM algorithm

This method classifies each grid block into a category of phases of a disordered multiphase material with specific characteristics. The category is built up based on a variety of the

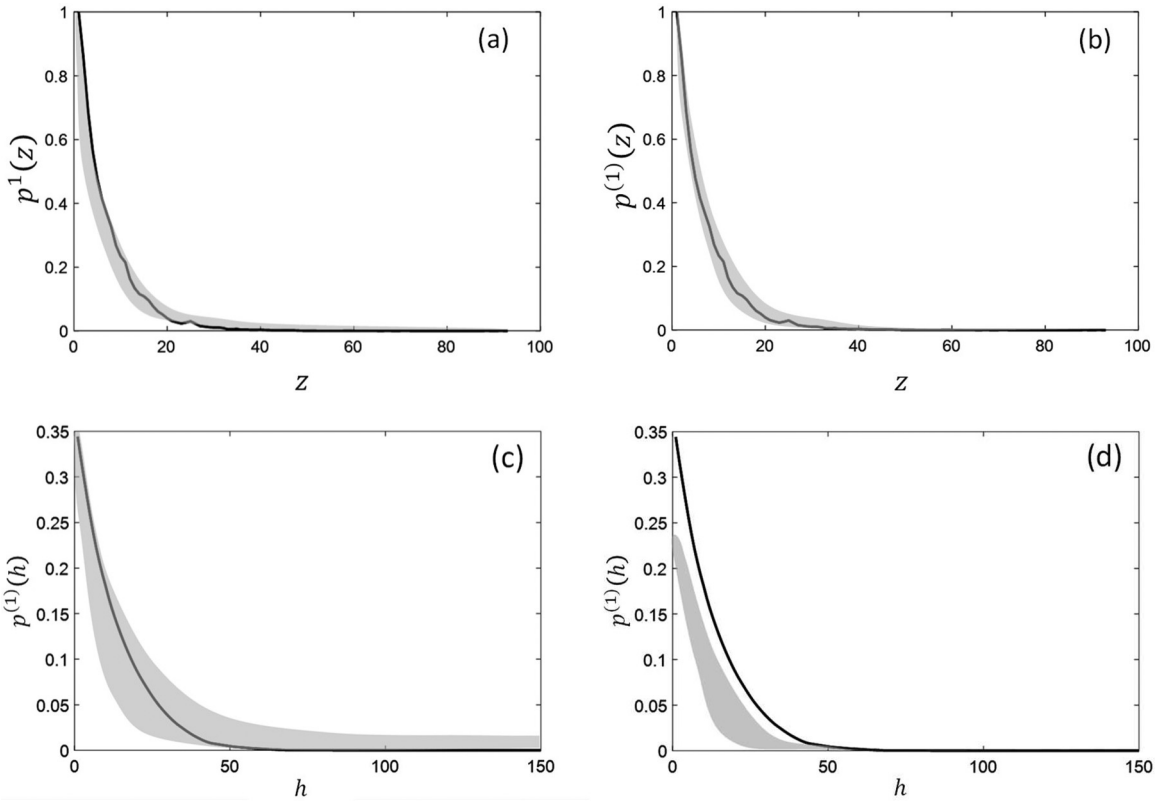

FIG. 3. Comparison between (top) the computed chord-length density function and (bottom) multiple-point connectivity function $p(h)$ for the digital image shown in Fig. 2 and its realizations generated by two reconstruction methods: [(a) and (c)] the CCSIM and [(b) and (d)] the SISIM. The black curves are the computed results for the **I**, while the colored areas indicate the uncertainty space for the realizations generated by the CCSIM and SISIM.

available data. It is assumed that no two identical phases coexist in the same grid block. After each grid block is assigned to its phase category, its property value is attributed to it from the probability distribution function (PDF) of the corresponding phase. Thus, the method is also called *sequential indicator simulation-probability distribution function*. The overall PDF of the phases represents the pattern of their occurrence over the length scale of the material to be reconstructed. The overall procedure for the algorithm is as follows.

(i) A random walk is taken through the computational grid that represents the material, such that the unconditioned blocks, i.e., those that do not contain any hard data that must be honored, are visited once and only once.

(ii) For each visited unconditioned grid block the prespecified number of conditioning phase data from the already reconstructed blocks, as well as any other available data, are identified.

(iii) A process called *indicator kriging* is carried out in order to estimate the conditional probability for each phase category. Originally developed for geostatistical applications, kriging [49] is a method for interpolating properties for which the interpolated values are modeled by a Gaussian process and governed by the prior covariances. With suitable assumptions on the prior covariances, kriging provides the best linear unbiased prediction of the interpolated values. Indicator kriging uses indicator functions in order to estimate the transition probabilities from one block to the next. It proceeds by first constructing an *indicator semivariogram*γ_I. Suppose that one has N_h pairs of data points $x(y_i)$ and $x(y_i + h)$ with $i = 1, 2, \ldots, N_h$ that are separated by a distance h. Then, the semivariogram $\gamma(h)$, which is a measure of the spatial correlations between the data, is defined by

$$\gamma(h) = \frac{1}{N_h} \sum_{i=1}^{N_h} [x(y_i) - x(y_i + h)]^2 \, . \tag{17}$$

The indicator semivariogram is contructed the same way by first introducing a critical threshold x_c that varies between a minimum and maximum value. Then, an indicator function $I(x_i)$, a generalization of the indicator function defined earlier, is defined by

$$I(x_i) = \begin{cases} 1 & x(y_i) \leqslant x_c, \\ 0 & x(y_i) > x_c, \end{cases} \tag{18}$$

where $x_i = x(y_i)$. The cumulative probability distribution function (CPDF) $\hat{F}(x_c)$ is then constructed by

$$\hat{F}(x_c) = \frac{1}{M} \sum_{i=1}^{M} I(x_i), \tag{19}$$

where $M < N_h$. The indicator semivariogram γ_I is then constructed based on $I(x_i)$, and the CPDF is used for estimating the conditional probabilities.

(iv) Each phase's probability is then normalized by the sum of the probabilities of all the phases. The result is then used to construct a local CPDF.

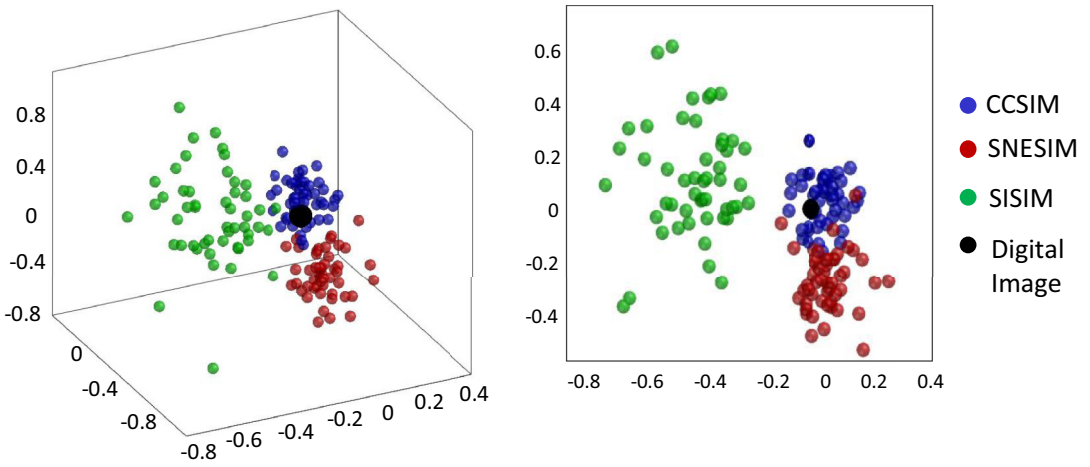

FIG. 4. Uncertainty space representation of the realizations of the digital image **I** of Fig. 2, after they were processed by the deep-learning algorithm. The **I** is shown as a black circle at the center. The others are for the realizations generated by the CCSIM (blue, those located around **I**), SNESIM (red, those located at the bottom of the right image), and SISIM (green, those located on the left side of the right image) algorithms.

(v) A random number r, distributed uniformly in [0,1], is generated and used together with the local CPDF in order to determine the reconstructed phase in the visited unconditioned grid block.

(vi) For each unconditioned block, in the sense defined earlier, the random path steps (ii)–(v) are repeated. The final result is a computational grid as the model of the material with the distribution of the phases.

D. The FILTERSIM algorithm

To address the difficulties of the SNESIM approach for 3D media, as well as to make it applicable to continuous images, several refinements were proposed [50,51]. Though such methods are accurate in 2D, their high computational cost makes them impractical for 3D applications. One of the main reasons for their high computational cost is that for each grid block one must compare a data event with all the patterns of heterogeneity in the database or image. To address this problem Zhang *et al.* [51] introduced the FLITERSIM algorithm. They used a set of six and nine filters for 2D and 3D reconstruction, respectively, in order to coarsen ("summarize") the basic spatial properties of the heterogeneity patterns contained in the **I**, which reduce significantly the dimensionality of the patterns' space and, hence, the computation time. The patterns are first filtered using linear filters and, according to some similarity criteria, are grouped in distinct clusters. Then, for each cluster, a prototype pattern is computed that represents some sort of the average of all the patterns in that cluster. The rest of the algorithm proceeds, in each grid block, by selecting the most similar prototype and randomly drawing a pattern from that cluster and is repeated for all the blocks. Clearly, the use of a limited number of filters reduces the computation cost of the algorithm. The method does, however, have its shortcomings, the most important of which is that, it uses a limited set of linear filters that may not always convey all the important information and variability of the heterogeneity of the **I**.

V. RESULTS AND DISCUSSION

The proposed DL method was implemented with three types of materials with different number of phases. They include a wide range of microstructures, from a relatively simple binary material to a continuous image of the microstructure of a highly heterogeneous one. All the computations were carried out on a GPU, an NVIDIA GeForce GTX 1660 Ti with a total memory of 38 661 MB and shared memory of 32 670 MB, with 1536 CUDA cores. The total computation time was about 2 h. Since the main idea of the proposed method is to capture the features at small and large scales, one can speed up the computations further by upscaling the realizations.

A. A two-phase microstructure

As the first example we consider a two-phase functionally graded material, with one phase being macrocopically connected. Each phase is uniform, but its clusters are spatially distributed. The **I**, taken from Srividhya *et al.* [52] and shown in Fig. 2, contains a combination of long- and short-range features. To demonstrate the performance of the proposed DL algorithm, three realizations of the **I** were generated by the CCSIM, SNESIM, and SISIM algorithms and are shown in Fig. 2. Visual inspection indicates that the realizations generated by the CCSIM and SNESIM are far more similar to the **I** than those produced by the SISIM. Looks can, however, be deceiving and, thus, we need to quantify the differences.

Let 1 and 2 denote, respectively, the white and black phases of the material whose image is shown in Fig. 2. To make a

TABLE I. Comparison of the internal similarities ζ_I of the three algorithms.

Algorithm	SNESIM	CCSIM	SISIM
SNESIM	1	1.24	0.41
CCSIM	–	1	0.85
SISIM	–	–	1

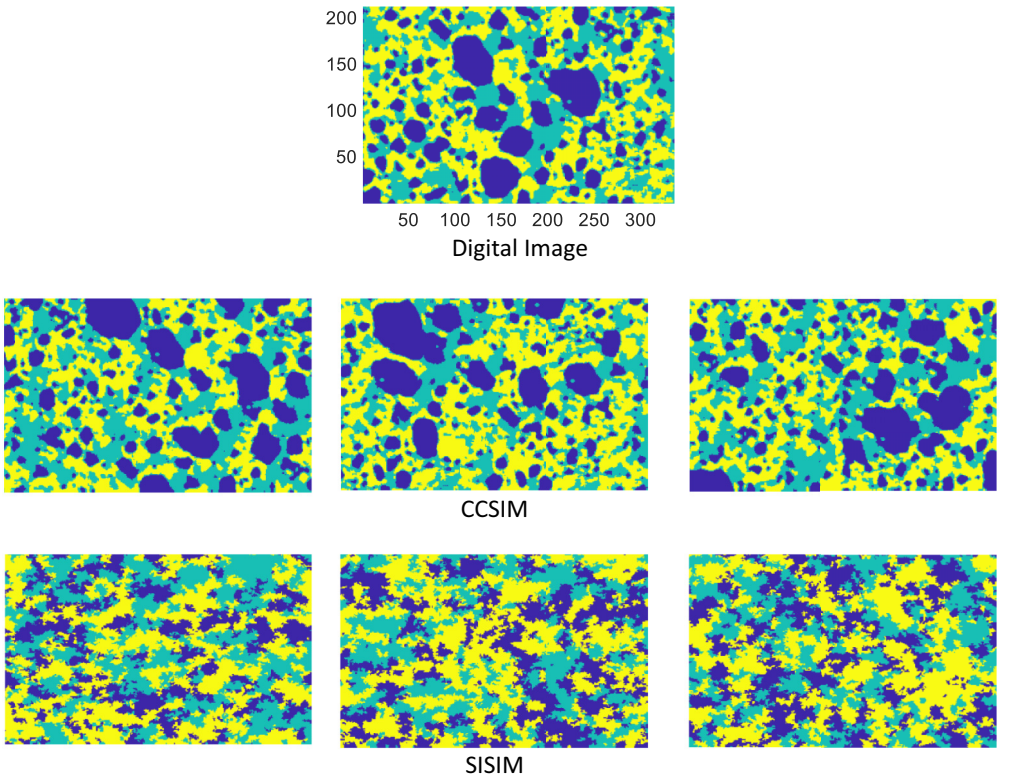

FIG. 5. Comparison between the original digital image **I** of a multiphase material and the realizations generated by the two stochastic reconstruction algorithms.

quantitative comparison between the realizations and the **I**, we computed the two microsructural descriptors described earlier, namely the chord-length density function $p^{(1)}(z)$ and the multiple-point connectivity function $p^{(1)}(h)$ (for $m = 150$), both for phase 1. The results, shown in Fig. 3, indicate that the chord-length density function does not differentiate the differences between the realizations generated by the CCSIM, SISIM, and the **I**. Although the multiple-point connectivity function produces more realistic variations, it is still inadequate for distinguishing the various patterns generated by the three reconstruction algorithms. Furthermore, the range of uncertainty indicated by the chord-length density function is very narrow, which is not the case in reality.

Using the proposed DL method, the "distances" between the various realizations were compared. Recall that each realization is compressed by the DL to just three numbers, which we take them to be its "coordinates" (x, y, z). Figure 4 presents the results, in which the original **I** has been given the coordinates (0,0,0) at the center of the plot. The results indicate a large uncertainty space for the SISIM method (shown by green) far from the **I**, hence showing that the apparent similarity between the realizations generated by the SISIM algorithm and the **I** is not significant. Likewise, the results for the SNESIM algorithm indicate reasonable similarity with the **I**, though they are not well distributed around the **I**. On the other hand, the results for the CCSIM algorithm indicate an acceptable uncertainty space consistent with Figs. 2 and 3, as they are well scattered all around the **I**.

The computed results for the internal similarity between the realizations, without considering the **I**, are shown in Table I. They are normalized based on the results for the SNESIM approach. The computed results for the external similarity—between the **I** and the realizations—are shown in Table II. They indicate, for example, that the realizations generated by the SISIM algorithm are very dissimilar to the **I**, which is consistent with Figs. 2 and 4. The algorithms are then ranked based on the final score s, defined by Eq. (15), for which the results are presented in Table III. Thus, accuracy of the algorithms is ranked as follows: CCSIM > SNESIM > SISIM.

TABLE II. Comparison between the external similarities ζ_E of the three algorithms and the **I**.

Algorithm	SNESM	CCSIM	SISIM
SNESIM	1	1.52	0.34
CCSIM	–	1	0.21
SISIM	–	–	1

TABLE III. Comparison between the final scores s of the three algorithms.

Algorithm	SNESIM	CCSIM	SISIM
SNESIM	1	0.47	1.24
CCSIM	–	1	4.05
SISIM	–	–	1

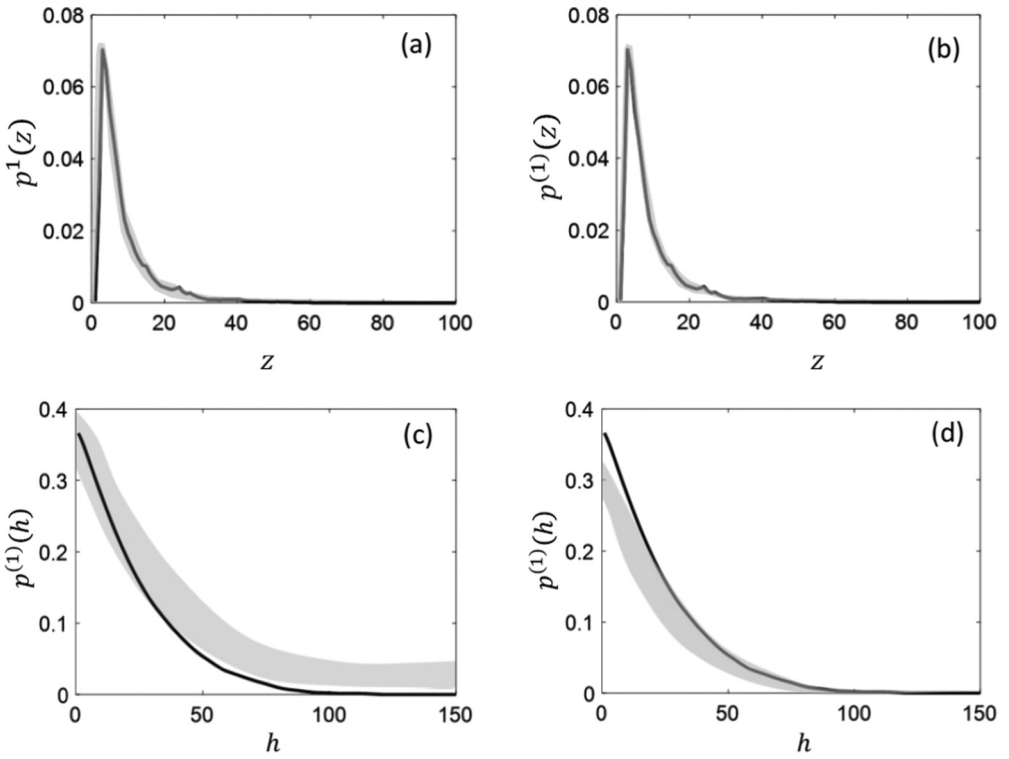

FIG. 6. Comparison between (top) the computed chord-length density function, and (bottom) multiple-point connectivity function $p(h)$ for the image shown in Fig. 5 and its realizations generated by two reconstruction methods: [(a) and (c)] the CCSIM and [(b) and (d)] the SNESIM. The black curves are the computed results for the **I**, while the colored areas indicate the uncertainty space for the realizations generated by the CCSIM and SISIM.

B. A multiphase material

As the second example we used the image of a yttria-dispersed ferritic stainless steel sample [53]. The material represents a multiphase medium with stationary properties. We used the CCSIM and SISIM algorithms to generate multiple realizations for the **I**, examples of which are shown

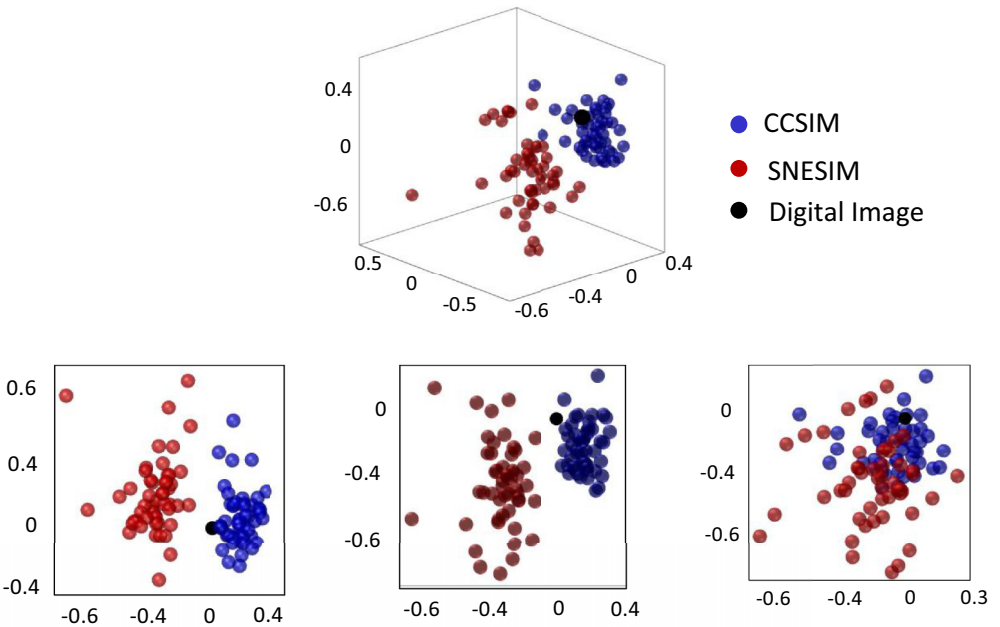

FIG. 7. Uncertainty space representation of the realizations of the digital image of Fig. 5 after they were processed by the deep-learning algorithm. The **I** is shown as a black circle at the center. The others are for the realizations generated by the CCSIM (blue, those located in the right of the middle image) and SNESIM (red, those located in the left of the middle image) algorithms.

TABLE IV. Comparison between the final scores s for two algorithms.

Algorithm	CCSIM	SISIM
CCSIM	1	2.68
SISIM	–	1

TABLE V. The final scores for the two algorithms.

Algorithm	CCSIM	FILTERSIM
CCSIM	1	2.54
FILTERSIM	–	1

in Fig. 5. Visual inspection of the results indicates that the CCSIM method produces higher-quality realizations. To make the comparison quantitative, we computed the two microstructural descriptors, the chord-length density, and the multiple-point connectivity functions for the **I** and its realizations. The results are presented in Fig. 6 and indicate, for example, that the chord-length density function generates a very small uncertainty space, whereas the realizations produced by the SISIM algorithm indicate wide variations and appear to be very different from the **I**. Similarly, the computed multiple-point connectivity function does not manifest the actual properties of the realizations. For example, the long-range connectivity that we computed for the realizations generated by the SISIM method is not exhibited by Fig. 6(d). Furthermore, the short-range connectivity of the realizations produced by the CCSIM algorithm is underestimated. As such, both functions are unable to differentiate the variability among the realizations, as well as between them and the **I**.

Using the proposed DL method, the conclusion is quantified and confirmed. Figure 7 presents the relative distances between the realizations and the **I**. Although the realizations generated by the SISIM algorithm manifest significant variability, they all are very far from the **I**. On the other hand, the CCSIM algorithm produces realizations that not only exhibit reasonable variability but are also close to the **I**, as the spatial distribution of the patterns indicates. This is also confirmed by the computed final scores s, compiled in Table IV, where the results were normalized by that of the CCSIM.

C. A continuous microstructure

The third image, shown in Fig. 8, represents the image of a low-carbon steel whose microstructure consists mostly of ferrite with the darker pearlite regions around the ferrite grains [54]. Because the image is on a gray scale, its reconstruction in its current form is difficult. Despite this, we used the CCSIM and FILTERSIM algorithms to generate realizations of the image. The SNESIM and SISIM approaches are not applicable to such images. The results are also shown in Fig. 8. The realizations generated by the FILTERSIM algorithm are not similar to the **I**, as they contain a significant number of artifacts. For example, the spatial connectivity of the black phase is not reproduced, and the spatial distribution of the gray phase does not match the given features in the **I**.

In addition to the distinct visual differences between the generated realizations, we also quantified the similarity and the uncertainty space using the proposed DL method. The

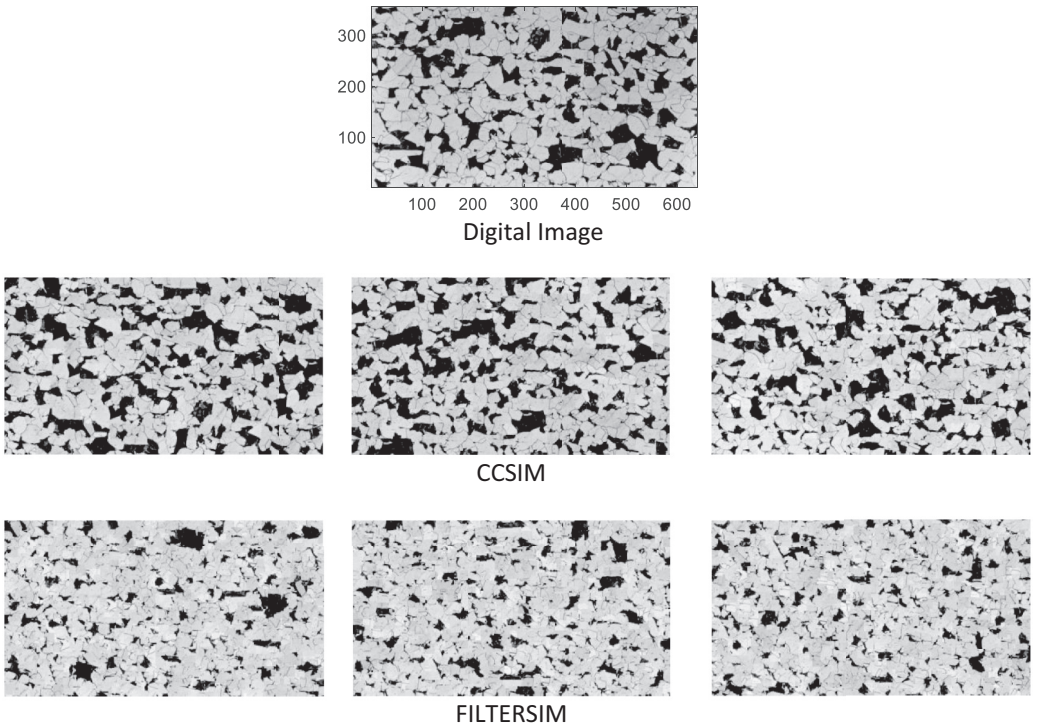

FIG. 8. Comparison between the original digital image **I** of a material with a continuous microstructure and the realizations generated by two stochastic reconstruction algorithms.

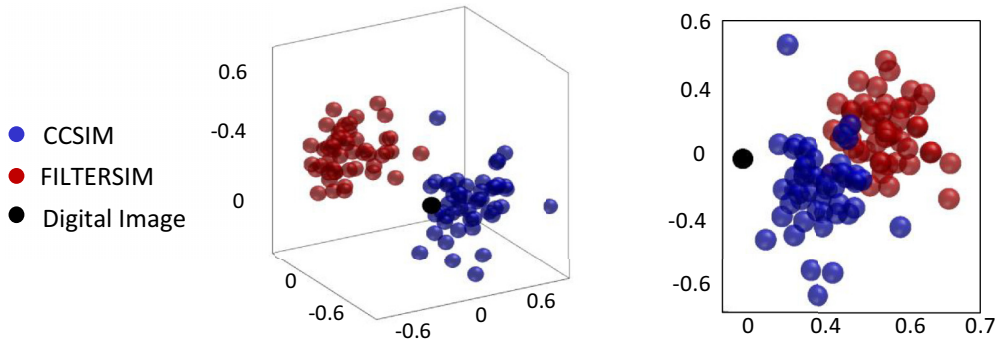

FIG. 9. Uncertainty space representation of the realizations of the digital image **I** of Fig. 8 after they were processed by the deep-learning algorithm. The **I** is shown as a black circle at the center. The others are for the realizations generated by the CCSIM (blue, those located in lower-left side of the right image) and FILTERSIM (red, those located in upper-right side of the right image) algorithms.

final scores s for the two algorithms were computed and are listed in Table V in which the results were normalized by that of the CCSIM, while the results for the uncertainty space are shown in Fig. 9. Consistent with Fig. 8, both sets of results indicate that drastic differences exist between the realizations generated by the FILTERSIM algorithm and the **I**. Both CCSIM and FILTERSIM algorithms reproduce similar uncertainty space between the realizations, but their external uncertainty spaces are not comparable.

VI. SUMMARY

Despite the development of several reconstruction methods for modeling of heterogeneous materials over the past two decades, the question of their validity for accurate representation of materials has remained open. At the same time, with advances in instrumentation, accurate 2D and even 3D images of heterogeneous materials are becoming increasingly available. This paper presented a method based on machine learning for evaluating stochastic approaches for reconstruction of materials and making a comprehensive quantitative comparison between the realizations generated by them. The method reduces the dimensionality of the realizations using a deep-learning algorithm by coarsening the given image(s). Two criteria for evaluating the similarities between the realizations, as well as between them and the original digitized image, were introduced. First, for a reconstruction method to be effective, the differences between the realizations that it produces must be reasonably broad, so that they faithfully represent the possible spatial variations in the microstructure of the materials and their plausible representation. We refer to this as the internal uncertainty space. At the same time, the realizations and the **I** of materials must be completely similar. Thus, in a similar fashion, a second criterion, the external uncertainty space, is defined by which the similarity between each realization and the **I** is quantified. Finally, the ratio of two uncertainty indices associated with them is considered as the final score of a stochastic algorithm, which provides a quantitative basis for comparing various approaches. The proposed method was tested with images of various heterogeneous materials in order to evaluate four stochastic reconstruction algorithms.

ACKNOWLEDGMENTS

Work at the University of Southern California was supported in part by the Petroleum Research Fund, administered by the American Chemical Society Grant No. 56112-ND9.

[1] S. Torquato, *Random Heterogeneous Materials* (Springer, New York, 2002).
[2] M. Sahimi, *Heterogeneous Materials* (Springer, New York, 2003).
[3] J. H. Kinney and N. C. Nichols, X-Ray tomographic microscopy (XTM) using synchrotron radiation, Annu. Rev. Mater. Sci. **22**, 121 (1992).
[4] A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging* (SIAM, Philadephia, PA, 2001).
[5] L. Babout, E. Maire, J. Y. Buffiére, and R. Fougéres, Characterization by X-ray computed tomography of decohesion, porosity growth and coalescence in model metal matrix composites, Acta Mater. **49**, 2055 (2001).
[6] A. Borbély, F. Csikor, S. Zabler, P. Cloetens, and H. Biermann, Three-dimensional characterization of the microstructure of a metal-matrix composite by holotomography, Mater. Sci. Eng. A **367**, 40 (2004).
[7] J. Baruchel, P. Bleuet, A. Bravin, P. Coan, E. Lima, A. Madsen, W. Ludwig, O. Pernot, and J. Susini, Advances in synchrotron hard X-ray based imaging, Compt. Rend. Phys. **9**, 624 (2008).
[8] H. Toda, S. Yamamoto, M. Kobayashi, K. Uesugi, and H. Zhang, Direct measurement procedure for three-dimensional local crack driving force using synchrotron X-ray microtomography, Acta Mater. **56**, 6027 (2008).
[9] A. Weck, D. S. Wilkinson, E. Maire, and H. Toda, Visualization by X-ray tomography of void growth and coalescence leading to fracture in model materials, Acta Mater. **56**, 2919 (2008).
[10] G. T. Herman, *Fundamentals of Computerized Tomography* (Springer, Berlin, 2009).

[11] M. Y. Wang, J. J. Williams, L. Jiang, F. De Carlo, T. Jing, and N. Chawla, Dendritic morphology of α-Mg during the solidification of Mg-based alloys: 3D experimental characterization by X-ray synchrotron tomography and phase-field simulations, Scr. Mater. **65**, 855 (2011).

[12] D. Brandon and W. D. Kaplan, *Microstructural Characterization of Materials* (Wiley, New York, 2013).

[13] E. Padilla, V. Jakkali, L. Jiang, and N. Chawla, Quantifying the effect of porosity on the evolution of deformation and damage in Sn-based solder joints by X-ray microtomography and microstructure-based finite element modeling, Acta Mater. **60**, 4017 (2012).

[14] H. Malmir, M. Sahimi, and M. R. Rahimi Tabar, Statistical characterization of microstructure of packings of polydisperse hard cubes, Phys. Rev. E **95**, 052902 (2017).

[15] H. Malmir, M. Sahimi, and Y. Jiao, Higher-order correlation functions in disordered media: Computational algorithms and application to two-phase heterogeneous materials, Phys. Rev. E **98**, 063317 (2018).

[16] P. M. Adler, C. G. Jacquin, and J. A. Quiblier, Flow in simulated porous media, Int. J. Multiphase Flow **16**, 691 (1990).

[17] C. L. Y. Yeong and S. Torquato, Reconstructing random media, Phys. Rev. E **57**, 495 (1998).

[18] C. L. Y. Yeong and S. Torquato, Reconstructing random media. II. Three-dimensional media from two-dimensional cuts, Phys. Rev. E **58**, 224 (1998).

[19] P. Levitz, Off-lattice reconstruction of porous media: Critical evaluation, geometrical confinement and molecular transport, Adv. Colloid Interface Sci. **76–77**, 71 (1998).

[20] S. Bekri, K. Xu, F. Yousefian, P. M. Adler, J.-F. Thovert, J. Muller, K. Iden, A. Psyllos, A. K. Stubos, and M. A. Ioannidis, Pore geometry and transport properties in North Sea chalk, J. Pet. Sci. Eng. **25**, 107 (2000).

[21] H. Hamzehpour, M. R. Rasaei, and M. Sahimi, Development of optimal models of porous media by combining static and dynamic data: The permeability and porosity distributions, Phys. Rev. E **75**, 056311 (2007).

[22] Y. Jiao, F. H. Stillinger, and S. Torquato, Modeling heterogeneous materials via two-point correlation functions. II. Algorithmic details and applications, Phys. Rev. E **77**, 031135 (2008).

[23] Y. Jiao, F. H. Stillinger, and S. Torquato, A superior descriptor of random textures and its predictive capacity, Proc. Natl. Acad. Sci. USA **106**, 17634 (2009).

[24] Y. Jiao and S. Torquato, Quantitative characterization of the microstructure and transport properties of biopolymer networks, Phys. Biol. **9**, 036009 (2012).

[25] P. Tahmasebi and M. Sahimi, Reconstruction of three-dimensional porous media using a single thin section, Phys. Rev. E **85**, 066709 (2012).

[26] Y. Jiao, E. Padilla, and N. Chawla, Modeling and predicting microstructure evolution in lead/tin alloy via correlation functions and stochastic material reconstruction, Acta Mater. **61**, 3370 (2013).

[27] P. Tahmasebi and M. Sahimi, Cross-Correlation Function for Accurate Reconstruction of Heterogeneous Media, Phys. Rev. Lett. **110**, 078002 (2013).

[28] S. Chen, H. Li, and Y. Jiao, Dynamic reconstruction of heterogeneous materials and microstructure evolution, Phys. Rev. E **92**, 023301 (2015).

[29] S. Chen, A. Kirubanandham, N. Chawla, and Y. Jiao, Stochastic multi-Scale reconstruction of 3D microstructure consisting of polycrystalline grains and second-phase particles from 2D micrographs, Metall. Mater. Trans. A **47**, 1440 (2016).

[30] P. Tahmasebi, Accurate modeling and evaluation of microstructures in complex materials, Phys. Rev. E **97**, 023307 (2018).

[31] P.-E. Øren and S. Bakke, Reconstruction of Berea sandstone and pore-scale modeling of wettability effects, J. Pet. Sci. Eng. **39**, 177 (2003).

[32] B. Biswal and R. Hilfer, Microstructure analysis of reconstructed porous media, Physica A **266**, 307 (1999).

[33] B. Biswal, C. Manwart, R. Hilfer, S. Bakke, and P.-E. Øren, Quantitative analysis of experimental and synthetic microstructures for sedimentary rock, Physica A **273**, 452 (1999).

[34] B. Biswal, P.-E. Øren, R. J. Held, S. Bakke, and R. Hilfer, Stochastic multiscale model for carbonate rocks, Phys. Rev. E **75**, 061303 (2007).

[35] D. Chen and S. Torquato, Designing disordered hyperuniform two-phase materials with novel physical properties, Acta Mater. **142**, 152 (2018).

[36] M. E. Kainourgiakis, E. S. Kikkinides, A. Galani, G. C. Charalambopoulou, and A. K. Stubos, Digitally reconstructed porous Media: Transport and sorption properties, Transp. Porous Media **58**, 43 (2005).

[37] Z. Ma and S. Torquato, Precise algorithms to compute surface correlation functions of two-phase heterogeneous media and their applications, Phys. Rev. E **98**, 013307 (2018).

[38] H. Okabe and M. J. Blunt, Prediction of permeability for porous media reconstructed using multiple-point statistics, Phys. Rev. E **70**, 066135 (2004).

[39] C. E. Zachary and S. Torquato, Improved reconstructions of random media using dilation and erosion processes, Phys. Rev. E **84**, 056102 (2011).

[40] P. Tahmasebi and M. Sahimi, Enhancing multiple-point geostatistical modeling. I: Graph theory and pattern adjustment, Water Resour. Res. **52**, 2074 (2016).

[41] P. Tahmasebi and M. Sahimi, Enhancing multiple-point geostatistical modeling. II: Iterative simulation and multiple distance functions, Water Resour. Res. **52**, 2099 (2016).

[42] A. Hajizadeh, A. Safekordi, and F. A. Farhadpour, A multiple-point statistics algorithm for 3D pore space reconstruction from 2D images, Adv. Water Resour. **34**, 1256 (2011).

[43] X. Tan, P. Tahmasebi, and J. Caers, Comparing training-image based algorithms using an analysis of distance, Math. Geosci. **46**, 149 (2014).

[44] http://ufldl.stanford.edu/tutorial/unsupervised/Autoencoders/.

[45] S. Kullback and R. A. Leibler, On information and sufficiency, Ann. Math. Stat. **22**, 79 (1951).

[46] S. Strebelle, Conditional simulation of complex geological structures using multiple-point geostatistics, Math. Geol. **34**, 1 (2002).

[47] S. Strebelle and T. Zhang, Non-Stationary Multiple-point Geostatistical Models, in *Geostatistics Banff 2004*, edited by O. Leuangthong and C. V. Deutsch, Quantitative Geology and Geostatistics Vol. 14 (Springer, Dordrecht, 2005).

[48] M. Sahimi, *Flow and Transport in Porous Media and Fractured Rock*, 2nd ed. (Wiley-VCH, Weinheim, 2011), Chap. 5.

[49] J. Straubhaar, P. Renard, G. Mariethoz, R. Froidevaux, and O. Besson, An improved parallel multiple-point algorithm using a list approach, Math. Geosci. **43**, 305 (2011).

[50] B. Arpat and J. Caers, Stochastic simulation with patterns, Math. Geol. **39**, 177 (2007).

[51] T. Zhang, P. Switzer, and A. G. Journel, Filter-based classification of training image patterns for spatial simulation, Math. Geol. **38**, 63 (2006).

[52] S. Srividhya, K. Basant, R. K. Gupta, A. Rajagopal, and J. N. Reddy, Influence of the homogenization scheme on the bending response of functionally graded plates, Acta Mech. **229**, 4071 (2018).

[53] R. Shashanka and D. Chaira, Effects of Nano-Y2O3 and sintering parameters on the fabrication of PM duplex and ferritic stainless steels, Acta Metall. Sin. (Engl. Lett.) **29**, 58 (2016).

[54] https://www.doitpoms.ac.uk/miclib/micrograph_record.php?id=202.