Data-Augmented Structure-Property Mapping for Accelerating Computational Design of Advanced Material Systems

by

Ruijin Cang

A Dissertation Presented in Partial Fulfillment
of the Requirement for the Degree
Doctor of Philosophy

Approved July 2018 by the
Graduate Supervisory Committee:

Yi Ren, Chair
Yongming Liu
Yang Jiao
Qiong Nian
Houlong Zhuang

ARIZONA STATE UNIVERSITY

December 2018

ABSTRACT

Advanced material systems refer to materials that are comprised of multiple traditional constituents but complex microstructure morphologies, which lead to their superior properties over conventional materials. This dissertation is motivated by the grand challenge in accelerating the design of advanced material systems through systematic optimization with respect to material microstructures or processing settings. While optimization techniques have mature applications to a large range of engineering systems, their application to material design meets unique challenges due to the high dimensionality of microstructures and the high costs in computing process-structure-property (PSP) mappings. The key to addressing these challenges is the learning of material representations and predictive PSP mappings while managing a small data acquisition budget. This dissertation thus focuses on developing learning mechanisms that leverage context-specific meta-data and physics-based theories. Two research tasks will be conducted: In the first, we develop a statistical generative model that learns to characterize high-dimensional microstructure samples using low-dimensional features. We improve the data efficiency of a variational autoencoder by introducing a morphology loss to the training. We demonstrate that the resultant microstructure generator is morphology-aware when trained on a small set of material samples, and can effectively constrain the microstructure space during material design. In the second task, we investigate an active learning mechanism where new samples are acquired based on their violation to a theory-driven constraint on the physics-based model. We demonstrate using a topology optimization case that while data acquisition through the physics-based model is often expensive (e.g., obtaining microstructures through simulation or optimization processes), the evaluation of the constraint can be far more affordable (e.g., checking whether a solution is optimal or equilibrium). We show that this theory-driven learning algorithm can lead to much

improved learning efficiency and generalization performance when such constraints can be derived. The outcomes of this research is a better understanding of how physics knowledge about material systems can be integrated into machine learning frameworks, in order to achieve more cost-effective and reliable learning of material representations and predictive models, which are essential to accelerate computational material design.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my advisor Dr. Yi Ren. It has been a great honor to be his first Ph.D. student. I appreciate all his contributions of time, idea, patience and funding to make my Ph.D experience productive and stimulating. His guidance and care helped me in all the time of research and writing of this dissertation. I still remembered he taught me to be always curious in study and don't take the things on the paper as they are granted, which is helpful not only in research but also a guidance to discover the mysterious world. I could hardly have imagined having a better advisor and mentor for my Ph.D study.

Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Yang Jiao, Dr. Yongming Liu, Dr. Nian Qiong and Dr. Houlong Zhuang, for their insightful comments and encouragement, but also for the hard question which helped me to widen my research from various perspectives.

I thank my lab-mate, Houpu Yao for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last several years. Also I thank my friends in Complex Material Group, Para Lab and INLAB for problem discussion and the fun.

Last but not the least, I would like to thank my family: my parents and to my wife Xiaoyan for supporting me spiritually throughout writing this thesis and my life in general.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

Advanced materials are being widely studied to address energy, environment, and security challenges. Such materials often comprise of novel structures and exhibit superior properties over conventional materials (McDowell and Olson, 2008a). Examples include artificial photosynthesis materials for solar energy (Wool and Sun, 2011) and catalysts for the production of bio-fuels (Stöcker, 2008), light-weight protection materials and structures for security and defense purposes (Mallick, 2007), and materials with high bio-compatibility for artificial organs (Hollister, 2005). In order to achieve global competitiveness, there is a growing need for accelerating the pace of discovery and deployment of advanced materials.

Traditional ways for material development mainly rely on scientific intuition and trial-and-error experiments, which are time-consuming and computational expensive. Typically this procedure takes about 10 to 20 years from initial research to deployment in industry (Wadia *et al.*, 2009). Computational Material Design (CMD) has been promoted by the Material Genome Initiative to fundamentally change the strategies for developing and manufacturing advanced material systems to meet the urging demands in the mentioned high tech fields. Pioneering work in CMD employs direct gradient-based optimization techniques to explore the material microstructure space and suggests relatively simple microstructures that can be subsequently realized in a composite material system. Conventional definition of material microstructure "features" include manually defined geometric and statistical measures, including grain dimensions, dispersion, and correlation functions. These "features" have been used to explain variance in material properties due to microstructural differences, and thus used for computational material design. For advanced materials with complicated microstructures, however, understanding what material "features" are important for design still remains an unsolved puzzle, and the potentially large number of "features" that can be used to represent the material render the optimization problem difficult to solve.

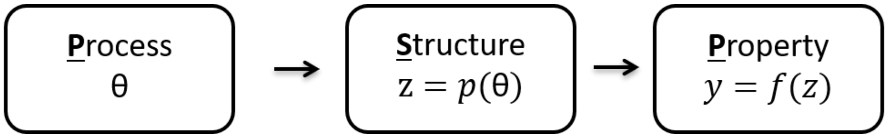


Figure 1: Olson's concept of "Materials by Design": Process-Structure-Property relation of material system

The revealed lack of scalability of CMD tools to advanced materials with multiscale and structured microstructures has led to the proposal of Integrated Computational Material Engineering (ICME) (McDowell and Olson, 2008a; Panchal *et al.*, 2013), with an emphasis on the modeling of material Process-Structure-Property (PSP) mappings through an integration between computation, machine learning, and material sciences (Figure 1). Three core functionalities are needed to realize ICME: (1) a computational tool that is able to efficiently extract structural features of microstructures at multiple length scales and reconstruct physically meaningful material microstructure with the extracted features, (2) a cost-effective predictive model for microstructure property prediction, and (3) an efficient way to search for new material designs within the feature space. Both (2) and (3) rely on the successful learning of a feature representation of the material system, which is ideally low-dimensional (to enable efficient search of new designs) and representative (to offer high reconstruction and prediction quality).

This dissertation thus focuses on developing a design methodology and solution strategies to effectively and efficiently achieve feature learning from and reconstruction of material systems. We start by investigating generative models for learning features from microstructures with hierarchical patterns (Chapter 3, 4). We then address the high data-dependency issue in feature learning (Chapter 5), which is key to material design tasks with high data acquisition costs. Lastly, we look into the lack of physical preservation of reconstructions produced by data-driven learning models, and propose a theory-driven learning framework to significantly improve the performance of microstructure reconstruction in a physical sense (Chapter 6). The remaining of this chapter provides an overview of the challenges of the topic and the corresponding research tasks.

1.1 Challenges in Material Design

1.1.1 Major scientific challenges

Considering the material structure design as an optimization problem, a general problem formulation of material design is as follows: A material microstructure is represented as a 2D or 3D image $z \in Z$, where Z is the image space. Each element of z defines the composition or phase at its location. The image is called feasible when some processing setting $\theta \in \Theta$ exists, such that $z = p(\theta)$, with $p(\cdot)$ being the process-structure mapping, e.g., a physics-based simulation, and Θ the set of all available processing settings. Consider a material property of interest, e.g., an alloy fatigue crack growth property, derived from a structure-property mapping $f(\cdot) : y = f(z)$. Finding the optimal material property y^* through an all-in-one approach, i.e., $\theta^* = argmin_{\theta \in \Theta} f(p(\theta))$ may not be favorable for the following reasons: (1) Finding the optimal θ^* can be computationally intractable if the problem is combinatorial, e.g., a combination of multiple processing techniques is needed. (2) The mapping $p(\theta)$ is often stochastic, i.e, a certain processing setting can lead to a set of microstructure images. One solution is to perform a nested optimization by first finding target microstructure z^* to optimize y, then searching for a processing setting θ^* to match z^*. The latter serves as a feasibility check for the former. Mathematically, we solve the following structure design problem constrained by processing feasibility:

$$\min_{feasible\, z} y = f(z) \tag{1.1}$$

Compared with an all-in-one approach, this approach can speed up the convergence by parallelizing the search for the potentially optimal structures and their feasibility check. The nested problem formulation in Eq. 1.1 allows these models be refined and validated independently. For multiscale material systems, solving Eq. 1.1 will encounter at least two following challenges: 1) the high resolution of material samples required by material property simulations causes the microstructure set Z to be high dimensional and 2) the high cost of evaluating the design $f(\cdot)$, making the overall design process inefficient.

Learning low dimensional representations for high-dimensional multi-scale material systems

The first challenge can be alleviated if the dimensionality of the design space is reduced. This is due to the common existence of structural patterns in the given material system, indicating that there exists a feature representation that concisely encodes the raw material images. Advantages of a feature representations were confirmed by recent studies for homogeneous nanoparticle composite designs tasks (Xu *et al.*, 2014), but this method mainly depends on designer's understanding, which make the robustness limited, Sec 2.1 will explain more about the limitation. Thus, this dissertation will examine unsupervised feature extraction and generation mechanisms. Informally, we use a large material structure-property set to learn a mapping $x = \Phi(z)$ and its inverse $z = \Phi^{-1}(x)$, where x is a learned feature representation from a low-dimensional space X. In addition, developing such feature extraction and generation mechanism for material design usually requires a large number of input samples, which is not always cheap. Existing studies have achieved limited success at this challenge. In this dissertation, we will introduce morphology-aware constraints derived from a small set of material samples to improve the efficiency for network training (Chapter 5).

Improving accuracy and efficiency of model learning

The second challenge is typical to data-driven learning and in particular with deep architecture. As is known that the performance of data-driven method always depends on the data size,

i.e., the larger size the better model performance. However, preparing enough data for model training is not always acceptable due to the limitation of computational budget, especially in industrial field. Thus increase the efficiency of the computation usage is a crucial topic, which involves a good understanding about the design representation, the key structure feature and a goodness measurement method. To this end, the challenges to be tackled before achieving these points include, high-dimensional or ill-defined design spaces such as for topologies (Bendsøe and Kikuchi, 1988; Guo *et al.*, 2018), material microstructures (Cang *et al.*, 2017b, 2018; Bostanabad *et al.*, 2018), or complex geometries (Stiny, 1980; Hsiao and Chen, 1997), low level of data usage, e.g. lack of physical based theory in pure data-driven design mechanisms, expensive evaluations of designs and their sensitivities, e.g., due to model nonlinearity (Allaire *et al.*, 2004; Jung and Gea, 2004), coupled materials or physics (Sigmund, 2001; Wang and Wang, 2004; Zhu *et al.*, 2017), or subjective goodness measures (Ren and Papalambros, 2011; Orbay *et al.*, 2015), or search inefficiency due to the absence of sensitivities Xie and Steven (1993); Querin *et al.* (1998);

Rao *et al.* (2011) or the existence of random variables (Kharmanda *et al.*, 2004).

To this end, this dissertation is motivated to investigate new directions of solutions on material design. Specifically, we will develop machine learning algorithms that integrate data-augmented method with physics-based knowledge to speed-up the design process. As discussed above, two steps are conducted to address the corresponding issues: First, we train generative models that characterize high-dimensional material microstructure into lower-dimensional ones, constraining the microstructure space during material design; second, we train generative models by investigating an active learning mechanism where physics-based theory is applied.

1.2 Research Tasks

1.2.1 Overview of the research activities

This dissertation proposes to accomplish two major tasks derived from the identified challenges: In Task 1 2a, the dissertation will investigate the methods that transform $\mathbf{s}$ (or $\mathbf{z}$ for PS mapping) into and from a lower-dimensional representation $\mathbf{x} \in \mathcal{X}$ through an encoder-decoder pair: $\mathbf{x} = \Phi(\mathbf{s})$ and $\mathbf{s} = \Phi^{-1}(\mathbf{x})$. In general some generalizable statistical learning algorithms will be developed, consisting of some deep networks for extracting material microstructure features and generative models for characterizing structurual pattern distribution. We investigate algorithmic design to achieve a forward network that is able to map the material microstructure images to its low-dimensional feature representation, and realize reconstruction/generation from the learned low dimensional features to physically meaningful material microstructure. In Task 2 2b, we investigate an active learning mechanism where new samples are acquired based on their violation to a theory-driven constraint on the physics-based model. We demonstrate using a topology optimization case that while data acquisition through the physics-based model is often expensive (e.g., obtaining microstructures through simulation or optimization processes), the evaluation of the constraint can be far more affordable (e.g., checking whether a solution is optimal or equilibrium). We show that this theory-driven learning algorithm can lead to much improved learning efficiency and generalization performance when such constraints can be derived.

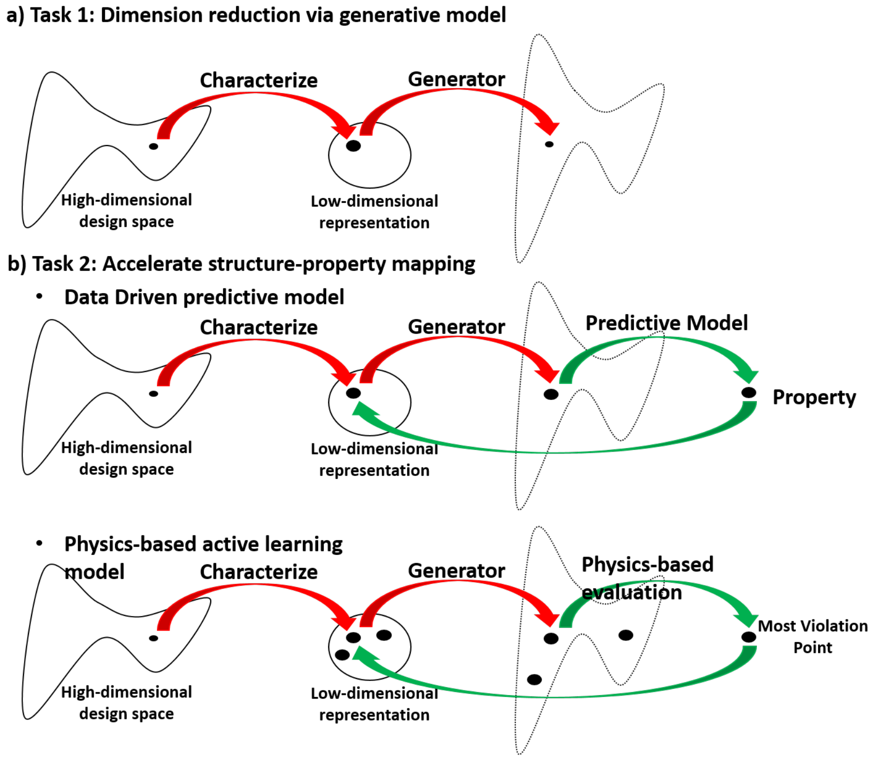

a) Task 1: Dimension reduction via generative model
Characterize
Generator
High-dimensional design space
Low-dimensional representation
b) Task 2: Accelerate structure-property mapping
• Data Driven predictive model
Characterize
Generator
Predictive Model
Property
High-dimensional design space
Low-dimensional representation
• Physics-based active learning model
Characterize
Generator
Physics-based evaluation
Most Violation Point
High-dimensional design space
Low-dimensional representation

Figure 2: (a) task 1, characterize the high dimensionality into a lower dimensional space via a generative model with limited number of data; (b) task 2, accelerate the accuracy and efficiency of representation learning strategy via a physics-based active learning method

1.2.2 Task 1: Dimension reduction and reconstruction via generative model

Given samples $\{\mathbf{s}_i\}_{i=1}^N$ drawn from a distribution $\mathbf{s} \sim p_{\text{data}}(\mathbf{s})$, we aim to learn the transformations $\mathbf{x} = \Phi(\mathbf{s})$ and $\mathbf{s} = \Phi^{-1}(\mathbf{x})$, such that (1) $\mathbf{x}$ has significantly lower dimensions than $\mathbf{s}$; (2) both Φ and Φ^{-1} are inexpensive to compute and differentiable; and most critically, (3) the generated data $\Phi^{-1}(\mathbf{x})$ are "similar" to those from the samples $\{\mathbf{s}_i\}_{i=1}^{N}$, or mathematically, the model distribution $p_{\text{model}}(\mathbf{s})$ matches the data distribution $p_{\text{data}}(\mathbf{s})$. These requirements ensure dimension compression, affordable overhead for the compression and generation, and minimal reconstruction loss to the data. Together with these properties, the compression can be integrated into the equilibrium search loop.

1.2.3 Task 2: Improving network representation learning efficiency and accuracy

Two directions are explored in the task. The first one is trying to find a data-driven predictive model to mimic the structure-property mapping. The challenge for this direction is known to be data demanding. The generator learned from task 1 will be used to address this use, so that the acceleration of the structure-property mapping could be achieved. The assumption for this direction is that the evaluation cost of the design from the generator can be neglected, however, this assumption does not always hold. The second direction, different from pure data-driven model where collects data without considering computational cost, we selectively pick the training data with a very low computational cost by evaluating their violation to a physics-based rule. In this case model is guided to learn the representation from the training data in a much more efficient strategy. In this part, we try to develop a feed forward neural network $\mathbf{x} = g(\mathbf{s}, \boldsymbol{\theta})$ as a generator, which is able to predict a solution $\mathbf{x}$ given a low dimensional problem settings $\mathbf{s}$ and network parameters $\boldsymbol{\theta}$. We call this type of network as a "one-shot" generator, which means that computing $\mathbf{x}$ through g is much less expensive than using an iterative algorithm. We also define the generalization performance of the generator (denoted by $F(\boldsymbol{\theta})$) as the expected performance of its solutions over a distribution of problems specified by a probability density function

$p(\mathbf{s})$:

$$F(\boldsymbol{\theta}) = \mathbb{E}_{p(\mathbf{s})} f(g(\mathbf{s}, \boldsymbol{\theta}), \mathbf{s}), \tag{1.2}$$

where $f(\mathbf{x}, \mathbf{s})$ measures the performance of $\mathbf{x}$ under $\mathbf{s}$. Once the generator get optimized, an optimal solution can be produced given the low dimensional description of the problem.

1.3 Outline of the Dissertation

The outline of this dissertation follows Fig 3. Chapter 2 provides an introduction to background knowledge and existing work. Existing literature on microstructure characterization and reconstruction, and the basic components and techniques in deep learning are reviewed in Chapter 2. In Chapter 3, we describe a Deep Belief Network based generative model, that is able to extract meaningful features, compress the original high dimensional microstructure into a much lower dimension space and also demonstrate the generation ability of the model via the learned low dimensional latent space. In Chapter 4, we modify the structure of the Deep Belief Network based generative model to realize scalable microstructure reconstruction. In Chapter 5, we demonstrate a statistical generative model that learns to characterize high-dimensional microstructure samples using low-dimensional features. We improve the data efficiency of a variational autoencoder by introducing a morphology loss to the training. We demonstrate that the resultant microstructure generator is morphology-aware when trained on a small set of material samples, and can effectively constrain the microstructure space during material design. In Chapter 6, we demonstrate an active learning mechanism where new samples are acquired based on their violation to a theory-driven constraint on the physics-based model. We demonstrate using a topology optimization case that while data acquisition through the physics-based model is often expensive (e.g., obtaining microstructures through simulation or optimization processes), the evaluation of the constraint can be far more affordable (e.g., checking whether a solution is optimal or equilibrium). We show that this theory-driven learning algorithm can lead to much improved learning efficiency and generalization performance when such constraints can be derived. Chapter 7 presents an overall summary of the report and a discussion of future works.

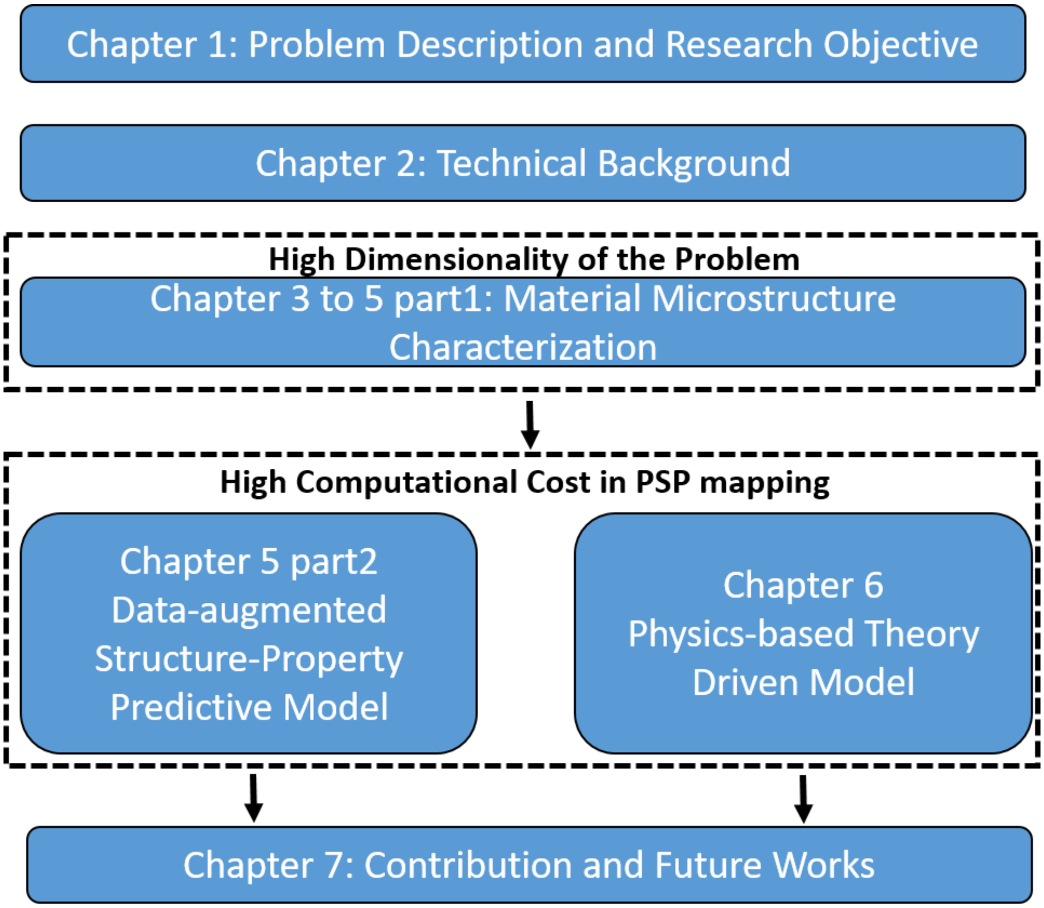

Chapter 1: Problem Description and Research Objective
Chapter 2: Technical Background
High Dimensionality of the Problem
Chapter 3 to 5 part1: Material Microstructure Characterization
High Computational Cost in PSP mapping
Chapter 5 part2 Data-augmented Structure-Property Predictive Model
Chapter 6 Physics-based Theory Driven Model
Chapter 7: Contribution and Future Works

Figure 3: Dissertation outline

Chapter 2

TECHNICAL BACKGROUND

This chapter provides the technical background related to the two research tasks. The first task mainly involves three parts: 1) the characterization method of material microstructure representations, 2) the representations based reconstruction/generation method and 3) the evaluation of material microstructure property. Thus a review of existing stochastic microstructure and reconstruction algorithms is provided in Sec. 2.1; besides, in this dissertation, neural network based models are mainly used for extraction, reconstruction and evaluation purpose, so in Sec. 2.2 and Sec. 2.3, basic concepts of deep learning, its usual components and the idea of generative model are introduced. The second task tries to improve the learning efficiency by introducing an active learning framework, which is also known as meta-learning, so a review of meta-learning is provided in Sec. 2.4.

2.1 Microstructure Characterization and Reconstruction

Conventional material representations can be categorized as physical, statistical, and non-parametric. We briefly introduce the definition of these representations, and their corresponding material generation methods.

Physical representations characterize a material structure using its composition (the percentage of each material constituent) (Broderick *et al.*, 2008a; Ashby, 2005), dispersion (the shortest distance between inclusion pairs and the correlation among different pairs (Steinzig and Harlow, 1999; Tewari and Gokhale, 2004; Rollett *et al.*, 2007; Borbely *et al.*, 2004; Pytz, 2004; Scalon *et al.*, 2003), and geometry (the particles' radius/size distribution, roundness, eccentricity, aspect ratio and others (Rollett *et al.*, 2007; Steinzig and Harlow, 1999; Torquato, 2013; Sundararaghavan and Zabaras, 2005; Basanta *et al.*, 2005; Holotescu and Stoian, 2011; Klaysom *et al.*, 2011; Gruber *et al.*, 2010)). Generating new material structures through physical representations involves minimizing the difference between the target descriptor values and those of the generated structure. While these manually defined representations are physically interpretable, they are not universally applicable, especially to material systems with complex structures (Cang and Ren, 2016).

Statistical representations are based on the N-point correlation functions and are widely used in material characterization and reconstruction (Liu *et al.*, 2013; Borbely *et al.*, 2004; Torquato, 2013; Sundararaghavan and Zabaras, 2005; Basanta *et al.*, 2005). Torquato et al. (Rintoul and Torquato, 1997; Yeong and Torquato, 1998; Torquato, 2013) have shown that the microstructure of heterogeneous materials can be characterized statistically via various types of N-point correlation functions (Okabe and Blunt, 2005; Hajizadeh *et al.*, 2011). Two-point correlation function and lineal path function are widely used as descriptors (Yeong and Torquato, 1998; Øren and Bakke, 2003). Another widely used representation is the two-point statistics calculated based on the frequency domain of the image using fast Fourier transform (FFT) (Fullwood *et al.*, 2008a,b). Similar to physical representations, one can generate structures by matching the statistical descriptors of the generated image with the target values with respect to the pixels of the structure. It has been shown that statistical representations can be used to accurately generation heterogeneous materials (Torquato, 2013; Borbely *et al.*, 2004; Basanta *et al.*, 2005), however, the computational cost for the generation (an optimization through gradient (Fullwood *et al.*, 2008a; Xu *et al.*, 2014) or non-gradient (Yeong and Torquato, 1998; Jiao *et al.*, 2008, 2009; Karsanina *et al.*, 2015) methods) can be high.

Material structures can also be represented as random fields (Roberts, 1997;

Bostanabad *et al.*, 2016; Liu and Shapiro, 2015), where each pixel in the structure is assumed to be drawn from a probability conditioned on other pixels. Of particular interest are Markov random fields where only the neighboring pixels affect the conditional probabilities. The representation can be parametric, when the conditional probability is parameterized, e.g., through a generalized linear model or a neural network, or non-parametric when it is modeled through, e.g., a SVM with nonlinear kernel (Schölkopf and Smola, 2002) or a regression tree (Prasad *et al.*, 2006). The generation of new structures under this representation can be done by jointly searching for a structure that maximizes the joint probability. However, it is shown that sequentially sampling pixel values according to the conditional probability will lead to reasonable generations (Bostanabad *et al.*, 2016).

2.2 Neural Network

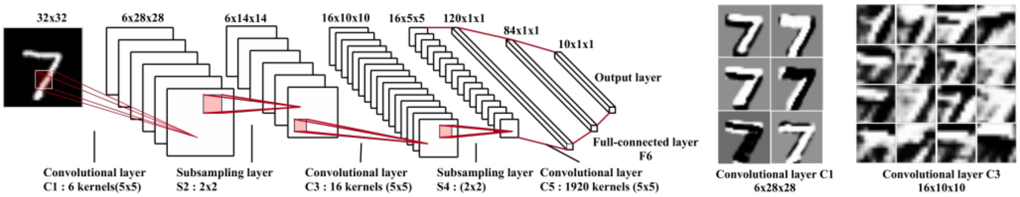
Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction and these methods have dramatically improved lots of field, such as speech recognition, visual object recognition, object detection and many other domains (LeCun *et al.*, 2015).

Representations are the key for various purpose, such as detection or classification. Representation learning is a set of methods that allows us to feed raw data to the machine, and the representations are discovered automatically and deep learning is one of it. The difference between deep learning and "shallow learning" is that, it learns multiple levels of representations via several simple but non-linear modules step by step, from low level representation to high, more abstract level. With the composition of enough such transformations, very complex functions can be learned. The representations at higher layers describe the common feature from the input data. Let's take images as an example. Images can be considered as matrixes with different, due to the value difference at each pixel, the patterns can be observed. The first layer usually learns some basic features that compose the input images, such as edges with different lengths and orientations. The second layer may detect a high level of features composed by the basic edge features that are detected from the fist layer. As the layer gets deeper, more complicated and abstract features could be derived that represent the input image, such as an abstract representation of cars, if the input is different vehicles, or an abstract representation of faces, if the idea is trying to detect human faces. The key advantage of deep learning is that all these features are not predefined by human, and they're all learned in a data-driven way. Thus, such mechanism is very beneficial at discovering some inner structure of complicated, high dimensional, highly-structured data. That's the very reason to take the advantages of deep learning for the research about material design in this dissertation .

2.2.1 Basic convolutional neural network components

Convolutional Neural Network is a well-known deep learning architecture inspired by the natural visual perception mechanism of the living creatures. In 1959, Hubel Wiesel (Hubel and Wiesel, 1968) found that cells in animal visual cortex are responsible for detecting light in receptive fields. Inspired by this discovery, Kunihiko Fukushima proposed the neocognition in 1980 (Fukushima, 1979), which could be regarded as the predecessor of CNN. In 1990, LeCun et al (LeCun *et al.*, 1990) published the seminal paper establishing the modern framework of CNN.

There are numerous variants of CNN architecture for different purpose. However, their basic components are very similar. Taking the famous LeNet-5 (Lee *et al.*, 2009) as an example, it consists of three types of layers, namely convolutional, pooling, and fully-connected layers. The convolutional layer aims to learn feature representations


32x32
6x28x28
6x14x14
16x10x10
16x5x5
120x1x1
84x1x1
10x1x1
Output layer
Full-connected layer F6
Convolutional layer C1 : 6 kernels(5x5)
Subsampling layer S2 : 2x2
Convolutional layer C3 : 16 kernels (5x5)
Subsampling layer S4 : (2x2)
Convolutional layer C5 : 1920 kernels (5x5)
Convolutional layer C1 6x28x28
Convolutional layer C3 16x10x10

(a) LeNet-5 network

(b) Learned features

Figure 4: (a) The architecture of the LeNet-5 netowrk, which works well on digit classification tasks. (b) visualization of features in the LetNet-5 network. Each layer's feature maps are displayed in a different block.

of the inputs. As shown in Figure 4, convolutional layer is composed by a non-linear activation over a convolving process between the kernel and the input data, whose results are often considered as feature map. The feature map describes a connection between the neuron at the current layer and its surrounded neurons in the previous layer. The surrounded area is very similar to the idea of receptive filed as mentioned before. There's one import characters for CNN is that, the kernels at each layer, people also refers as weights, are shared over all spatial locations of the input, which reduces the model complexity and saves the computational cost . Mathematically, the feature value at location k-th feature map of l-th layer, $z^l_{i,j,k}$ is calculated by:

$$z^l_{i,j,k} = (w^l_k)^T x^l_{i,j} + b^l_k \tag{2.1}$$

where w and b are the weight vector and bias term of the k-th filter of the l-th layer respectively, and x is the input patch centered at location (i, j) of the l-th layer. Note that the kernel w that generates the feature map $z^l_{:,:,k}$ is shared. The activation function introduces non-linearity to model, which makes the model able to fit different target with different purpose by increasing its non-linearity. The activation value a of convolutional feature z can be computed as:

$$a^l_{i,j,k} = a(z^l_{i,j,l}) \tag{2.2}$$

Widely used activation functions are sigmoid, tanh and ReLU (Dahl *et al.*, 2013).

Another major component in CNN is pooling, which usually connects between two convolutional layers, whose purpose is to reduce the overall resolution of the feature map at the current layer. Popular pooling methods are mean pooling, max pooling (LeCun *et al.*, 2015), fractional max pooling (Graham, 2014) and etc, but their common advantage is similar, shit-invariance. After several convolutional and pooling layers, there may be one or more fully-connected layers which aim to dimension reduction. They take all neurons in the previous layer and connect them to every single neuron of current layer to generate global semantic information.

The last layer of CNNs is an output layer. For classification tasks, the softmax operator is commonly used. Another commonly used method is SVM, which can be combined with CNN features to solve different classification tasks ((Tang, 2013)). Let θ denote all the parameters of a CNN (e.g., the wiehgt vectors and bias terms). The optimum parameters for a specific task can be obtained by minimizing an appropriate loss function defined on that task. Suppose we have N desired input-ouput relations, where x is the n^{th} input data, $y^{(n)}$ is its corresponding target label and o^n is the output of CNN. The loss of CNN can be calculated as follows:

$$L = \frac{1}{N}\sum_{n=1}^{N} l(\theta; y^n, o^n) \tag{2.3}$$

which will be used for back propagation 2.2.2. Training CNN is a problem of global optimization. By minimizing the loss function, we can find the best fitting set of parameters. Stochastic gradient descent is a common solution for optimizing CNN network ((Abadi *et al.*, 2016; Wijnhoven and de With, 2010; Zinkevich *et al.*, 2010))

2.2.2 Back propagation

The backpropagation procedure to compute the gradient of an objective function (5.5) with respect to the weights of a multilayer stack of modules is nothing more than a practical application of the chain rule for derivatives. They key insight is that the derivative (or gradient) of the objective with respect to the input of a module can be computed by working backwards from the gradient with respect to the output of that module. The backpropagation equation can be applied repeatedly to propagate gradients through all modules, starting from the output at the top (where the network produces its prediction) all the way to the bottom (where the external input is fed). Once these gradients have been computed, it is straightforward to compute the gradients with respect to the weights of each module.

2.3 Deep Generative Models

Different material microstructure has different microstructure features, and within the same material system the microstructure looks similar and we can consider them from the same data distribution. Generative model, put forward in recent years, is designed for such purpose. It is a very powerful tool aiming at learning the data distribution of the input samples, and with known of the data distribution, it will be very easy and efficient for us to generate the new data that follows the similar distribution. On the other word, if it's a material system, we are able to generate new microstructure with various properties.

But learning the exact distribution of the data is hard, no matter implicitly or explicitly, and we are always trying to model a distribution which is as similar as possible to the true data distribution. To this end, we take the advantage of neural networks to learn a function which has the capability to approximate the true data distribution and the model distribution.

Currently there are two most commonly used approaches for generative purpose, Variational Autoencoders (VAE) and Generative Adversarial Networks (GAN). The idea of VAE is aiming at maximizing the lower bound of the data log-likelihood and

GAN is aiming at achieving an equilibrium between Generator and Discriminator. In the following paragraph, I will briefly explain the theory behind of VAE and GAN.

2.3.1 Variational Autoencoder

Variational autoencoders (VAE) (Kingma and Welling, 2013) are extensions of autoencoders (Bengio, 2009). An autoencoder (Fig. 32a) is composed by two parts: an encoder $\mathbf{z} = f(\mathbf{x})$ converts the input $\mathbf{x}$ to a hidden vector $\mathbf{z}$, and a decoder produces a reconstruction $\hat{\mathbf{x}} = g(\mathbf{z})$. An AE is trained to minimize the discrepancy between inputs $\mathbf{x}$ and their corresponding reconstructions $\hat{\mathbf{x}}$.

Variants of autoencoders (e.g., Sparse (Ng, 2011), denoising (Vincent *et al.*, 2008), and contractive (Rifai *et al.*, 2011)) have been proposed to improve the learning of more concise representations from high-dimensional input data, and are widely used for data compression (Le *et al.*, 2015), network pre-training (Bengio *et al.*, 2007), and feature extraction (Xing *et al.*, 2015). However, conventional autoencoders are prone to generating implausible new outputs, see Fig. 36b for examples, as they do not attempt to match the model distribution (the distribution of images generated from the decoder) with the data distribution (the distribution of input images). VAE (Fig. 32b) was introduced to address this issue (Kingma and Welling, 2013), by constraining the distribution of the latent variables $\mathbf{z}$ encoded from the input data to that used for output generation, thus indirectly forcing the match between the distributions of the model outputs and the data.

The VAE model We start with the decoder, which defines the output distribution, $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$, conditioned on the latent vector $\mathbf{z}$, and is parameterized by $\boldsymbol{\theta}$. Let the input data be $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N}$, which defines a data distribution $p(\mathbf{x})$, and let $p(\mathbf{z})$ be the pre-specified sampling distribution in the latent space. The goal is to train a model that matches the marginal distribution $p_{\boldsymbol{\theta}}(\mathbf{x})$ to the data distribution $p(\mathbf{x})$. In the following, we show that this is indirectly achieved by matching the posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ to $p(\mathbf{z})$.

First, the marginal is defined as

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \int_{\mathbf{z}} p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} = E_{\mathbf{z}\sim p(\mathbf{z})}p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}). \tag{2.4}$$

Matching the model and data distributions requires maximizing the likelihood $\Pi_{\mathbf{x}\in\mathcal{X}}p_{\boldsymbol{\theta}}(\mathbf{x})$ with respect to $\boldsymbol{\theta}$. This maximization problem, however, is often intractable due to the numerical integral and the complexity of the decoder network. Nonetheless, it is noted that for a given $\mathbf{x}$, most samples in the latent space will have $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) \approx 0$, as these latent samples do not generate outputs close to $\mathbf{x}$. This leads to the idea of introducing another distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ (the encoder) that takes an $\mathbf{x}$ and outputs $\mathbf{z}$ values that are likely to reproduce $\mathbf{x}$. Ideally, the space of $\mathbf{z}$ that are likely under q will be much smaller than that under $p(\mathbf{z})$, making the computation of $E_{\mathbf{z}\sim q}p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ relatively cheap. However, $E_{\mathbf{z}\sim q}p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ is not the same as $p(\mathbf{x})$, as $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ does not necessarily match to $p(\mathbf{z})$. The relation between the two is derived below.

We start with the Kullback-Leibler divergence between the encoded latent distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ and the posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$:

$$\begin{aligned} D_{KL}\left(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})\right) &= \int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \log \frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}d\mathbf{z} \\ &= -\mathbb{E}_{\mathbf{z}\sim q}[\log p(\mathbf{z},\mathbf{x})] - \mathbb{E}_{\mathbf{z}\sim q}[-\log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})] + \log p(\mathbf{x}). \end{aligned} \tag{2.5}$$

By rearranging Eq. (5.2), we arrive at

$$\log p(\mathbf{x}) - D_{\mathrm{KL}}\left(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})\right) = \mathbb{E}_{\mathbf{z}\sim q}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] - D_{\mathrm{KL}}\left(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\right). \tag{2.6}$$

Note that the right hand side of Eq. (5.3) can be maximized through stochastic gradient descent: The first term $\mathbb{E}_{\mathbf{z}\sim q}[\log p_{\boldsymbol{\theta}(\mathbf{x}|\mathbf{z})}]$ measures the reconstruction error, i.e., the expected difference between an input $\mathbf{x}$ and its reconstructions drawn from the decoder conditioned on the latent variables, which are further drawn from the encoder conditioned on the input. The second term $D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$ measures the difference between the encoded distribution of $\mathbf{z}$ and the modeled latent distribution $p(\mathbf{z})$. Since the encoding and decoding processes involved in these terms are modeled as feedforward networks, the gradients are readily available for backpropagation. Specifically, we model $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}(\mathbf{x};\boldsymbol{\phi}), \boldsymbol{\Sigma}^2(\mathbf{x};\boldsymbol{\phi}))$ as a normal distribution parameterized by the mean $\boldsymbol{\mu}(\mathbf{x};\boldsymbol{\phi})$ and the diagonal variance matrix $\boldsymbol{\Sigma}^2(\mathbf{x};\boldsymbol{\phi})$. $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the outputs of the encoder network. We similarly model $p_{\boldsymbol{\theta}(\mathbf{x}|\mathbf{z})} = \mathcal{N}(\mathbf{x}|\mathbf{f}(\mathbf{z};\boldsymbol{\theta}), \sigma^2\mathbf{I})$ with mean $\mathbf{f}(\mathbf{z};\boldsymbol{\theta})$ and variance σ^2. The function $\mathbf{f}(\mathbf{z};\boldsymbol{\theta})$ is the decoder network; σ determines the importance of the reconstruction of $\mathbf{x}$ during the training of a generative model, and is set to 1 in the proposed model. blueThe inputs to the decoder are drawn from a normal distribution defined by $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. This connection between the encoder and the decoder can be modeled as feedforward through a reparameterization trick (Kingma and Welling, 2013), which enables gradient calculation of the entire network. The prior of the latent distribution, $p(\mathbf{z})$, is often assumed to be standard normal. This is because the transformation from this simple distribution to the potentially highly nonlinear data distribution $p(\mathbf{x})$ can often be achieved by a sufficiently deep decoder.

The left hand side of Eq. (5.3) contains the objective $\log p_{\boldsymbol{\theta}}(\mathbf{x})$ that we want to maximize, and the KL-divergence $D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})) > 0$ that should ideally reach 0. Thus, minimizing

$$L(\theta, \phi, \mathbf{x}) = -\mathbb{E}_{\mathbf{z}\sim q}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] + D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \tag{2.7}$$

will maximize a lower bound of $p(\mathbf{x})$.

2.3.2 Generative Adversarial Networks

Generative adversarial network (GAN) is another way of learning the high-dimensional data distribution via low dimensional representation. Proposed in (Ganin *et al.*, 2016), they can be characterized by training a pair of network in competition with each other, there are two types of sub-networks in GAN and got trained simultaneously in competition. One good analogy is that one network behaves like an paint forger, and the other one play as an paint expert. The forger tries his best to fake some new paints that are as realistic as possible, which is the generator G. The expert, also known as discriminator D, tries to tell the real one and fake one apart. One important note is that the generator has no direct clue about how the real paints looks like, and the only access for measuring its goodness is from the score of the discriminator, which will be used to train the generator to make the fake paint more real. The discriminator will be fed with both fake samples and real samples. Usually both generator and discriminator networks are composed by multi-layer convolutional and fully connected layers. Generative network tries to map the data distribution from a latent space, more formally we can express it as $G : G(z)- > R^{|x|}$, where $z \in R^{|z|}$ is a sample from latent space, $x \in R^{|x|}$ is an image and $| \cdot |$ denotes the number of dimensions. In a basic GAN, the discriminator network, D, may be similarly characterized as a function that maps from input data, such as images or other high dimensional input, to a probability that the input is from the real data distribution, rather than the generator distribution: $D : D(x)- > (0, 1)$. For a fixed generator, G, the discriminator, D, may be trained to classify images as either being from the training data (real, close to 1) or from a fixed generator (fake, close to 0). When the discriminator is optimal, it may be frozen, and the generator G, may continue to be trained so as to lower the accuracy of the discriminator. If the generator distribution is able to match the real data distribution perfectly then the discriminator will be maximally confused, predicting 0.5 for all inputs. In practice, the discriminator might not be trained until it is optimal. Training of GANs is a min-max problem, which involves find the parameters of a generator that maximize the cross-entropy to confuse the discriminator, and meanwhile the discriminator tries to minimize the cross-entropy to increase the classification accuracy for fake and real input. Mathematically:

$$\max_{D} \min_{G} V(G, D) \tag{2.8}$$

where

$$V(G, D) = \mathbb{E}_{p_{data}(x)} logD(x) + \mathbb{E}_{p_g(x)} log(1 - D(x)) \tag{2.9}$$

During the training, the parameters of one model are updated, while the parameters of the other are fixed. (Goodfellow *et al.*, 2014) show that for a fixed generator there is a unique optimal discriminator, $D^*(x) = \frac{p_{data}(x)}{p_{data}(x)+p_g(x)}$. They also show that the generator, G, is optimal when $p_g(x) = p_{data}(x)$, which is equivalent to the optimal discriminator predicting 0.5 for all samples drawn from x. That is to say, the minimax game converges when the discriminator can hardly distinguish between the real one and the fake one. Ideally, network G and D should converge at the same time, so that the generator is able to keep updating, but it is not always this case. GAN has great performance but it is notorious know for the converging issue. Thus several other techniques are applied to alleviate such issue, such as W-GAN (Venter *et al.*, 2001), EB-GAN (Zhao *et al.*, 2016), BE-GAN (Berthelot *et al.*, 2017) and etc.

2.3.3 Conclusion

Deep Learning models are really achieving human level performance in supervised learning but the same is not true for unsupervised learning. Nevertheless, deep learning scientists are working hard to improve the performance of unsupervised models.

In this section we briefly introduce about the two most famous unsupervised learning frameworks of generative models actually work. We got to know the problems in Variational Autoencoders and why Adversarial networks are better at producing realistic images. But there are problems with GANs such as stabilizing their training which is still an active area of research. However GANs are really powerful and currently they are being used in a variety of tasks such as high quality image and video generation, text to image translation, image enhancement, reconstruction of 3D models of objects from images, music generation, cancer drug discovery etc.

2.4 Meta Learning

The idea of meta learning has close connections to transfer learning (Caruana, 1995; Pan and Yang, 2010; Ganin *et al.*, 2016) and life-long learning (Thrun, 1996). While there exists a broad range of problem settings within the literature of meta-learning, the setup commonly involves a student and a learning mechanism that specifies how the student updates its way of learning or solving problems. The student outputs a solution for every input problem, for example, the input can be a labeled dataset and the output a classifier that explains the data, or the input an optimization problem and the output an optimal solution. In the former case, the student updates its way of learning a classifier (Ravi and Larochelle, 2016) or its hypothesis space of classifiers (Vilalta and Drissi, 2002); in the latter, it updates its gradient (Andrychowicz *et al.*, 2016) or non-gradient (Chen *et al.*, 2016) search strategies. These updates are governed by the learning mechanism, the design of which is driven by a goodness measure of the student, e.g., the generalization performance of a classifier or the convergence rate of an optimization solver.

There are three types of approaches to meta-learning, first one is iterative, to train a recurrent model such as a recurrent neural network (RNN) by taking training data as input and output the parameters of a learner model (Schmidhuber, 1987; Bengio *et al.*, 1992; Li and Malik, 2016; Andrychowicz *et al.*, 2016). The second one is to pass the training data and some test input into the model, and then outputs a corresponding prediction for the test example (Santoro *et al.*, 2016; Wang *et al.*, 2010; Mishra *et al.*, 2018). Both of the approaches have the capability to approximately represent any mapping from dataset and test datapoint to label, but it may lack statistical efficiency (Finn and Levine, 2018). The last more recent one is usually called model-agnostic meta-learning (MAML) (Finn and Levine, 2018), which optimizes only for the initial parameters of the learner model by standard gradient descent as the learner's update rule and then the learner will be trained via the gradient descent, which improves the statistical efficiency of meta-learners.

Chapter 3

DEEP BELIEF NETWORK BASED FEATURE EXTRACTION AND RECONSTRUCTION OF COMPLEX MATERIAL MICROSTRUCTURE

3.1 Introduction

Integrated Computational Materials Engineering (ICME) has been promoted by the Material Genome Initiative to fundamentally change the strategies for developing and manufacturing advanced material systems to meet the urging demands in energy, health and national security. Pioneering ICME works based on gradient-based shape and topological optimization and analytical processing-structure-property relations have demonstrated successes in designing a variety of material systems, including functional polymers (Sharma *et al.*, 2014; Baldwin *et al.*, 2015; Ma *et al.*, 2015), alloys and ceramics (Kalidindi and De Graef, 2015; Kaczmarowski *et al.*, 2015; Kirklin *et al.*, 2016), and polymer-matrix composites (Xu *et al.*, 2014, 2015), to name a few. Nonetheless, existing ICME methods lack scalability and have limited applications to multiscale and high-resolution microstructures. To this end, this paper proposes a computational tool for extraction and reconstruction of microstructure features on multiple length scales. The developed methodology from this paper could lead to better definition of the design space and enable computational design of complex material systems. In the remainder of this section, we introduce the mathematical formulation of the material microstructure design problem and review contemporary feature extraction techniques. Specifications of the proposed model and demonstrations of its performance can be found in Section 5.3. The advantages and limitations of the proposed algorithm will be discussed in Section 5.12, followed by conclusions in Section 4.5.

3.1.1 Material design as an optimization problem

We start by providing an overview of ICME to justify the necessity of design at the structure level. Computational material design can be mathematically formulated as follows: A material microstructure is represented as a 2D or 3D image $\mathbf{z} \in \mathcal{Z}$, where $\mathcal{Z}$ is the image space. Each element of $\mathbf{z}$ defines the composition or phase at its location. The image is called *feasible* when some processing setting $\boldsymbol{\theta} \in \Theta$ exists, such that $z = p(\boldsymbol{\theta})$, with $p(\cdot)$ being the process-structure mapping, e.g., a physics-based simulation, and Θ the set of all available processing settings. Consider a material property of interest, e.g., critical fracture force of alloy, derived from a structure-property mapping $f(\cdot)$: $y = f(\mathbf{z})$. Finding the optimal material property y^* through an all-in-one approach, i.e., $\boldsymbol{\theta}^* = arg\min_{\boldsymbol{\theta}\in\Theta} \quad y = f(p(\boldsymbol{\theta}))$, may not be favorable for the following reasons: (1) Finding the optimal $\boldsymbol{\theta}^*$ can be computationally intractable if the problem is combinatorial, e.g., a combination of multiple processing techniques is needed. (2) The mapping $p(\boldsymbol{\theta})$ is often stochastic, i.e., a certain processing setting can lead to a set of microstructure images. One solution is to perform a nested optimization by first finding a target microstructure $\mathbf{z}^*$ that optimizes y, then searching for a processing setting $\boldsymbol{\theta}^*$ to match $\mathbf{z}^*$. This approach can be mathematically formulated as the following structure design problem constrained by processing feasibility:

$$\min_{\text{feasible } \mathbf{z}} \quad y = f(\mathbf{z}). \tag{3.1}$$

3.1.2 Scientific challenges in optimal material structure design

For complex material systems, solving Eq. (3.1) will encounter the following issues: The high sensitivity of material properties on microstructure details requires high-resolution microstructure images leading to high-dimensional $\mathcal{Z}$; and the nonlinear processing-structure mapping makes the feasible domain within $\mathcal{Z}$ costly to be characterized. These issues can be alleviated if the dimensionality of the design space can be reduced. Dimension reduction is possible due to the existence of common structural patterns in the given material system, indicating that there exists a feature representation that concisely encodes the raw material images. Further, since the structural patterns are often related to processing feasibility and material properties, the encoding may lead to easier-to-construct predictive models for the processing-structure-property mappings. Advantages of feature representations were confirmed by recent studies for homogeneous nanoparticle composite design tasks (Xu *et al.*, 2014, 2015). However, current practices rely on the designer's understanding of specific material systems to identify geometric (e.g., particle size and orientation) and statistical descriptors (e.g., correlation functions) that are likely to explain variances in material properties. Generating such quantitative features for complex material systems could be difficult for human designers, or, the manually defined feature set would be insufficient for reconstruction of physically meaningful microstructures. Feature extraction methods through the use of deep convolutional networks could potentially address this issue. Informally, this line of approaches use sampled microstructures to learn a mapping $\mathbf{x} = \Phi(\mathbf{z})$ and its inverse $\mathbf{z} = \Phi^{-1}(\mathbf{x})$ (the unsupervised learning step), where $\mathbf{x}$ is a learned feature representation of $\mathbf{z}$ from a feature space $\mathcal{X}$, before mapping $\mathbf{x}$ to the performance y (the supervised learning step).

A unique challenge in developing a feature extraction mechanism for (material) design is the demanding quality of decoding (or called reconstruction), i.e., the reconstruction $\Phi(\Phi^{-1}(\mathbf{z}))$ shall "match" $\mathbf{z}$, so that any design searched in $\mathcal{X}$ can be correctly validated through a physics-based simulation or an experiment. In Sec. 3.4.5, we examine both the visual appearances, the 2-point correlation functions, and the mean property values of $\Phi(\Phi^{-1}(\mathbf{z}))$ and $\mathbf{z}$. Visually meaningful decoding is yet challenging for existing deep networks, which tends to neglect local details while preserving key features during the reconstruction (Dosovitskiy and Brox, 2015; Mahendran and Vedaldi, 2015; Nguyen *et al.*, 2015). To illustrate, photos can be encoded as text labels with a state-of-the-art deep network (e.g., (Krizhevsky *et al.*, 2012; Simonyan and Zisserman, 2014a)) but image restoration from text labels is hard to be precise (Nguyen *et al.*, 2015; Yan *et al.*, 2015). Existing studies have achieved limited success at this challenge by imposing structures onto the image space, e.g., constraining the smoothness of the reconstructed image (Dosovitskiy and Brox, 2015; Mahendran and Vedaldi, 2015; Yan *et al.*, 2015). This study first investigates a deep network model and postprocessing steps to enable random reconstruction of the complex microstructure of Ti-6Al-4V alloy samples. We then show that the proposed model can be applied to other material systems, including Pb-Sn (lead-tin) alloy, Pore structure of Fontainebleau sandstone, and 2D suspension of spherical colloids.

3.1.3 Contributions and limitations

To the authors knowledge, this is the first study that develops a deep network model for both feature extraction and reconstruction of complex material systems. In addition, our method preserves the statistical and the mechanical properties of interest statistically. Two major limitations still exist: First, while computationally inexpensive, the current reconstruction process is by nature non-scalable. A hybrid approach that combines the proposed feature extraction network and a stochastic material synthesis model will be discussed. Second, the mechanical properties of individual samples are not preserved, demanding higher reconstruction quality and better preservation of property-related microstructure features through the network model.

3.2 Related Work

Design at the microstructure level by changing the composition, phases and morphologies could lead to the discovery of unprecedented material systems with advanced performance (McDowell and Olson, 2008b). The challenges, however, exist in the modeling of interactions among multiple material constituents to predict the properties, and the automation of material design at microstructure level. In the following, we review existing studies that concern the latter challenge, and then introduce background knowledge for the proposed approach based on Convolutional Deep Belief Network (CDBN).

3.2.1 Microstructure parametrization and reconstruction

Conventional material design relies on choosing different material compositions from material databases (Broderick *et al.*, 2008b). For example, carbon-fibre reinforced epoxy are applied to replace the wooden oars for better performance (Ashby and Cebon, 1993). This composition-based approach is often limited and cannot be applied in the design of complex materials systems due to the existence of features other than constituents that govern material properties. Such features include morphology of microstructure, i.e., the spatial arrangement of local microstructural features (Karasek and Sumita, 1996), the heterogeneity in microstructure (Xu *et al.*, 2014) and others. A quantitative representation of material systems is thus needed.

A review from (Xu *et al.*, 2014) summarizes three categories of microstructure representations: (1) physical descriptors (e.g., composition descriptors such as volume fraction, dispersion descriptors (Rollett *et al.*, 2007; Borbely *et al.*, 2004; Tewari and Gokhale, 2004; Pytz, 2004; Steinzig and Harlow, 1999; Scalon *et al.*, 2003) such as average distance between fillers, and geometry descriptors (Rollett *et al.*, 2007;

Steinzig and Harlow, 1999; Torquato, 2013; Sundararaghavan and Zabaras, 2005; Basanta *et al.*, 2005; Holotescu and Stoian, 2011; Klaysom *et al.*, 2011; Gruber *et al.*, 2010) such as the size of fillers), (2) 2-point or N-point correlation functions (Liu *et al.*, 2013; Borbely *et al.*, 2004; Torquato, 2013; Sundararaghavan and Zabaras, 2005; Basanta *et al.*, 2005), and (3) random fields (Torquato, 2013; Quiblier, 1984; Jiang *et al.*, 2013; Grigoriu, 2003). Accordingly, a variety of material reconstruction schemes have been devised for different microstructure representations. Examples include the Gaussian random field method(Roberts, 1997), stochastic optimization method (Yeong and Torquato, 1998; Jiao *et al.*, 2008, 2009; Karsanina *et al.*, 2015), gradient-based method (Fullwood *et al.*, 2008a), phase-retrieval method(Fullwood *et al.*, 2008b), iterative methods based on multi-point statistics(Okabe and Blunt, 2005; Hajizadeh *et al.*, 2011), Bayesian Network Method(Matthews *et al.*, 2016) and cross-correlation functions (Tahmasebi and Sahimi, 2013, 2015), and image synthesis method (Sundararaghavan and Zabaras, 2005; Liu and Shapiro, 2015). Recently, a new reconstruction method is proved to be efficient at synthesizing *Markovian* microstructures (Bostanabad *et al.*, 2016) where the probability distribution of the material composition at each pixel (or voxel) is determined by its *local* surroundings and the conditional probability model can be applied *homogeneously* across a microstructure sample, see Fig. 5a for examples. However, the Markovian assumption may not hold for complex material systems. In fact, the algorithm from (Bostanabad *et al.*, 2016) produces less plausible synthesis results when trained on Ti-6Al-4V alloy samples, see Fig. 5b and c, especially when high variances in grain directions and sizes exist.

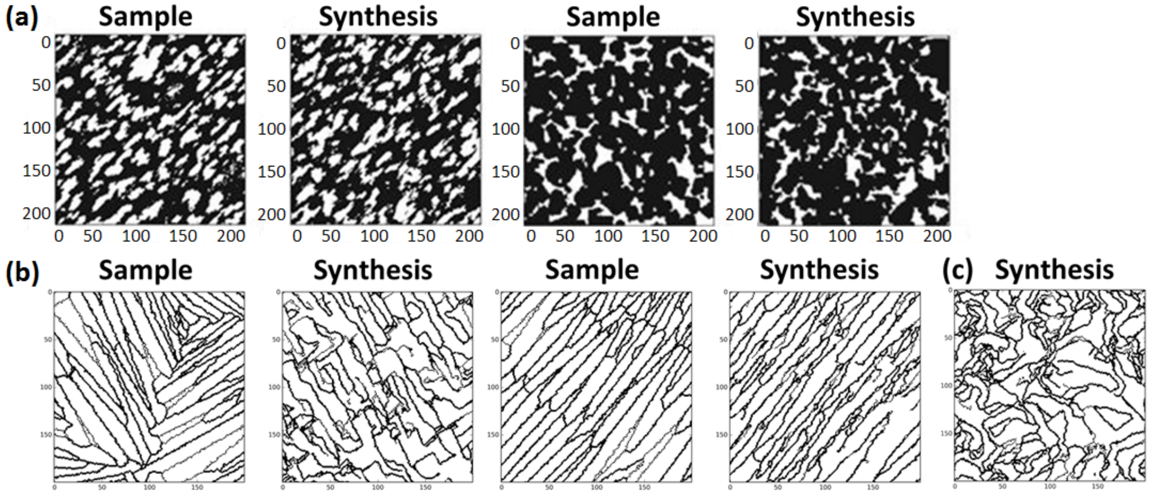


Figure 5: (a) Samples and random synthesis results of material systems that are assumed to be Markovian (Bostanabad *et al.*, 2016). (b-c) Random synthesis of the Ti-6Al-4V alloy microstructure following the synthesis algorithm from (Bostanabad *et al.*, 2016). The synthesis is based on (b) a single sample and (c) 100 samples from Fig. 47. Image courtesy of Dr. Ramin Bostanabad.

3.3 Introduction of Deep Belief Network

3.3.1 Convolutional deep network

Convolutional networks have long been used for feature extraction, with applications to object and voice recognition (Hinton *et al.*, 2012) and detection (Krizhevsky *et al.*, 2012), reinforcement learning (Mnih *et al.*, 2013; Schmidhuber, 2015; Levine *et al.*, 2015), analogy making (Reed *et al.*, 2015) and many others. By stacking multiple layers, a convolutional deep network can learn from input training data multi-scale features that contribute to the explanation of the corresponding outputs, without necessarily being guided by the outputs. For example, using a large set of human face images, a network can extract various face elements, including basic edges from the 1st layer, eyes and noses from the 2nd, and partial faces from the 3rd, see Fig. 6a. A forward pass of an image through the trained network essentially checks the existence of these features in the image, and thus using the output of the network for prediction (e.g., face recognition) is more effective than using raw image pixel values (Lee *et al.*, 2009). While a deep network may involve a large number of model parameters, the training is made tractable by using stochastic gradient descent((Bousquet and Bottou, 2008)) along with back-propagation (for deterministic models) (Rumelhart *et al.*, 1988) and contrastive divergence (for probabilistic models such as Restricted Boltzmann Machines (RBM)) (Hinton, 2002).

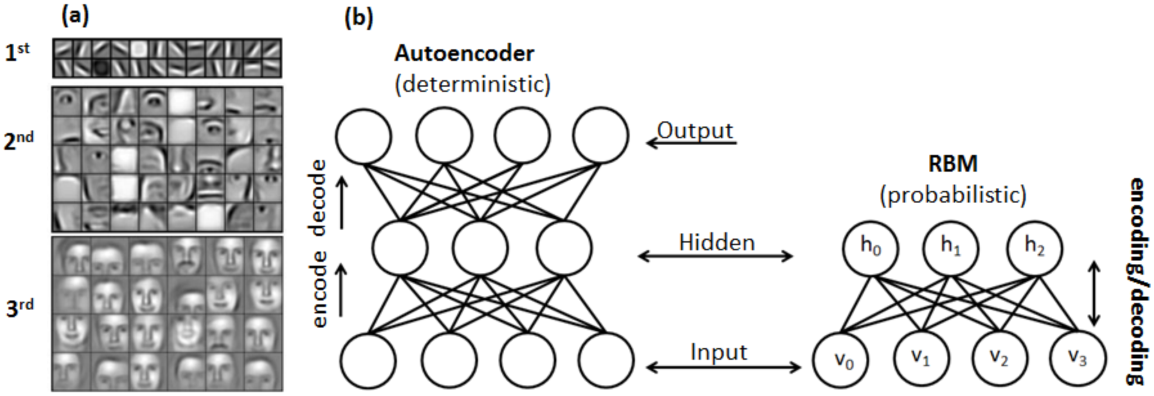


Figure 6: (a) Three layers of human face features extracted at increasing length scales (Lee *et al.*, 2009). (b) A schematic comparison between an RBM and an autoencoder. a_i (b_j) and v_i (h_j) are the bias and state values of visible (hidden) layer, respectively. W_{ij} indicates the weights between the visible and hidden layers.

The proposed Convolutional Deep Belief Network is one type of deep network that consists of stacked RBM layers. By the nature of RBM, the network is generative, meaning that it can perform both dimension reduction to map an input microstructure to a low-dimensional feature space, and reconstruction of the microstructure for a given sample in the feature space. It should be noted that other models could also be developed for reconstruction purpose, including Autoencoder (Bengio, 2009) (see Fig. 6b for an illustration and (Yumer *et al.*, 2015) for an example), Variational Autoencoder (VAE) (Kingma and Welling, 2013; Yan *et al.*, 2015), and the Generative Adversarial Network (GAN) (Goodfellow *et al.*, 2014).

3.3.2 Restricted Boltzmann Machine

A fully-connected RBM layer consists of visible (input) and hidden (output) nodes, see Fig. 6b. With weighted edges connecting these two sets of nodes, it is a complete bipartite graph. The set of edges connecting from all visible nodes to one hidden node acts as a convolution filter applied to the input image. The output of the convolution is transformed by a sigmoid function, and treated as the *activation* of the corresponding hidden nodes. The *state* of each hidden node is a binary number drawn from a Bernoulli distribution parameterized by the activation. A Convolutional RBM (or CRBM) is a special type of RBM where all image batches in the input layer (of the size of the filters) share K filters, so that the hidden layer forms a binary image with K channels, each produced by convolving the input image with a filter. Thus, a CRBM only has a subset of the visible nodes connecting to each of the hidden nodes (see Fig. 7a).

Learning an RBM (and CRBM) model through an set of input images takes the following procedure. Notations follow Fig. 6. Consider a model with n visible and m hidden nodes, and let v_i and h_j be the states of the ith visible and the jth hidden nodes, respectively. When visible units are binary-valued, the total energy of the layer is defined as:

$$E(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta}) = -\sum_{i=1}^{n} a_i v_i - \sum_{j=1}^{m} b_j h_j - \sum_{i=1}^{n} \sum_{j=1}^{m} v_i W_{ij} h_j. \tag{3.2}$$

In the above equation, $\boldsymbol{\theta} := \{\mathbf{W}, \mathbf{a}, \mathbf{b}\}$ is a set of model parameters to be estimated, within which $\mathbf{W}$ is a matrix of network weights, and $\mathbf{a}$ and $\mathbf{b}$ are the biases in the visible and hidden nodes, respectively. The joint probability of visible and hidden states is defined based on this energy function as:

$$\Pr(\mathbf{v}, \mathbf{h} | \boldsymbol{\theta}) = \frac{\exp(-E(\mathbf{v}, \mathbf{h} | \boldsymbol{\theta}))}{Z(\boldsymbol{\theta})}, \tag{3.3}$$

where $Z(\boldsymbol{\theta}) = \sum\limits_{\mathbf{v},\mathbf{h}} \exp(-E(\mathbf{v},\mathbf{h}|\boldsymbol{\theta}))$ is a partition function to make $\Pr(\mathbf{v},\mathbf{h}|\boldsymbol{\theta})$ a valid probability mass function. The marginal probability of the visible state is:

$$\Pr(\mathbf{v}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \sum e^{-E(\mathbf{v},\mathbf{h}|\boldsymbol{\theta})}. \tag{3.4}$$

Thus, the log-likelihood of the model parameters $\boldsymbol{\theta}$, given training data at the visible layer, can be written as $\phi(\boldsymbol{\theta};\mathbf{v}) = \log(\Pr(\mathbf{v}|\boldsymbol{\theta})) = \phi^+ - \phi^-$, where $\phi^+ = \log\sum_{\mathbf{h}}\exp(-E(\mathbf{v},\mathbf{h}))$ and $\phi^- = \log Z = \log\sum_{\mathbf{v},\mathbf{h}}\exp(-E(\mathbf{v},\mathbf{h}))$ are called the positive and negative parts. Finding the maximum likelihood estimator of $\boldsymbol{\theta}$ requires applying stochastic gradient descend to $\phi(\boldsymbol{\theta};\mathbf{v})$. The gradient of the positive part can be simply calculated as $\frac{\partial\phi^+}{\partial W_{ij}} = v_i\Pr(h_j = 1|v)$, while the negative gradient $\frac{\partial\phi^-}{\partial W_{ij}} = \Pr(v_i = 1, h_j = 1)$ is usually computationally expensive due to the calculation of the partition function. This issue, however, can be resolved by using contrastive divergence (Hinton, 2002) to provide an asymptotic approximation of $\frac{\partial\phi^-}{\partial W_{ij}}$ by iteratively sampling from the conditional probabilities $\Pr(v_i = 1|\mathbf{h}) = \text{sigmoid}\left(\sum_j W_{ij}h_j + a_i\right)$ and $\Pr(h_j = 1|\mathbf{v}) = \text{sigmoid}\left(\sum_i W_{ij}v_i + b_j\right)$.

3.3.3 Probabilistic max-pooling

In a deep network, a max-pooling layer is often inserted after a convolution layer to reduce the number of hidden nodes. This allows fewer network weights to be computed in the follow-up layers and also allow features of larger length scales to be extracted. Specific to CDBN, a probabilistic max-pooling layer segments the output image of a hidden RBM layer into blocks (2×2 in this study) and assigns the associated pooling node a binary value α by drawing from a Bernoulli distribution defined by the activations of the hidden nodes. The activations for the four hidden nodes are denoted by I_i for $i = 1, 2, 3, 4$. The probability for the pooling node to be activated is $\Pr(\alpha = 1) = 1 - (1 + \sum_{i=1}^4 I_i)^{-1}$. For backward pooling during the reconstruction, we assign ones to all four nodes in the block if $\alpha = 1$ or otherwise zeros. Fig. 7b summarizes the pooling procedure.

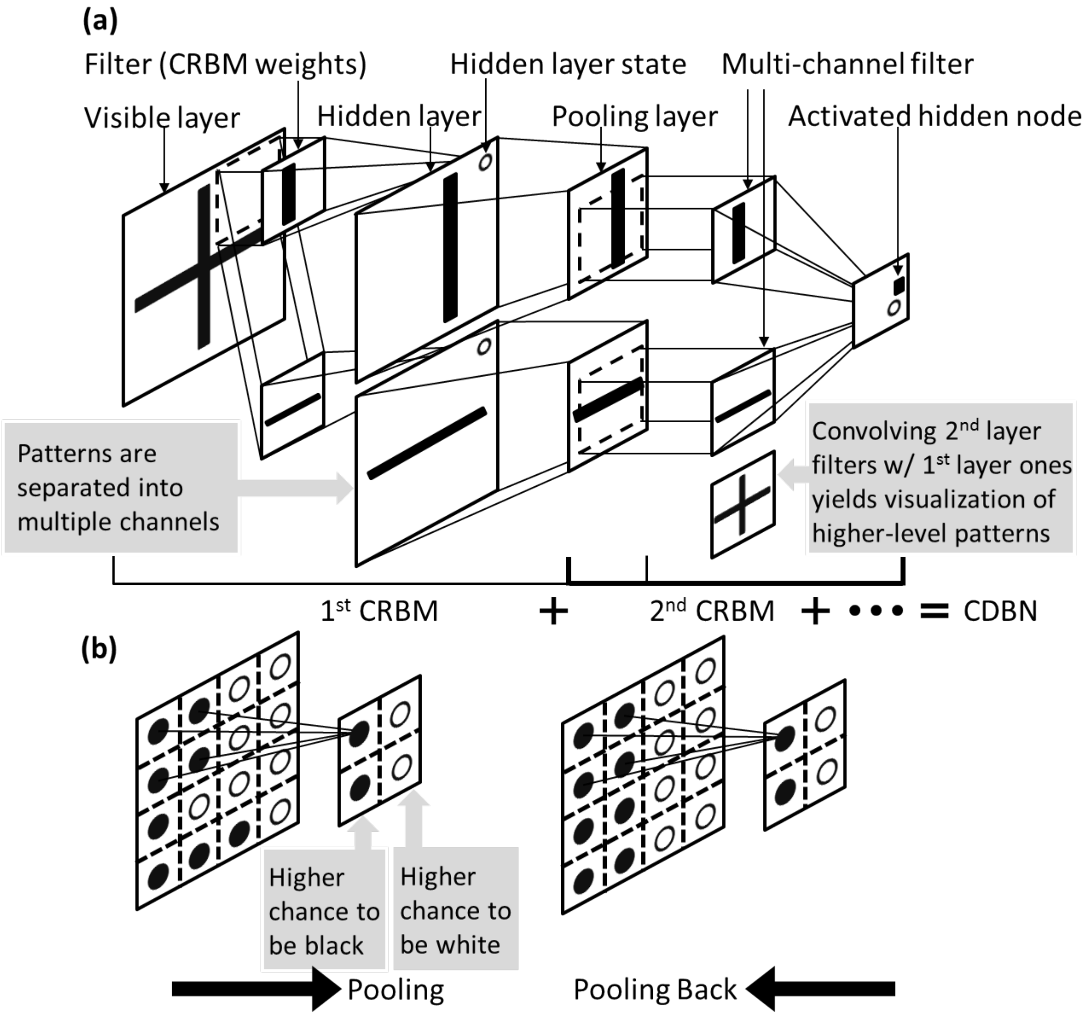

(a)
Filter (CRBM weights)
Hidden layer state
Multi-channel filter
Visible layer
Hidden layer
Pooling layer
Activated hidden node
Patterns are separated into multiple channels
Convolving 2nd layer filters w/ 1st layer ones yields visualization of higher-level patterns
1st CRBM
+
2nd CRBM
+ • • • =
CDBN
(b)
Higher chance to be black
Higher chance to be white
Pooling
Pooling Back

Figure 7: (a) CRBM and pooling layers (b) Forward and backward probabilistic max-pooling procedures with 2×2 blocks

3.4 Proposed CDBN for Feature Extraction and Material Reconstruction

The proposed network has three CRBM layers and two fully-connected RBM layers. Each of the first two CRBM layers is followed by a 2×2 probabilistic max-pooling layer. The convolutional filters for the first three layers are set respectively as follows: 6×6, 9×9, and 9×9 in filter size, 1, 24 and 40 in the number of channels, and 24, 40 and 288 in the number of filters. The size of the five hidden layers are correspondingly $200 \times 200 \times 1$, $97 \times 97 \times 24$, $44 \times 44 \times 40$, $36 \times 36 \times 288$ and 30×1. Notice that the image size shrinks along the depth of the network due to the convolution operation. For example, convolving an image of 200×200 with the filter of 6×6 will lead to an output of $(200 - 6 + 1) \times (200 - 6 + 1)$. This size will further be divided by 4 after the max-pooling process, which makes the dimension 97×97. In addition, the number of filters corresponds to the number of channels of the hidden layer. Table 1 summarizes the model and training parameters. One typical measure to prevent the learning of trivial filters is to set a target sparsity for the hidden nodes, so that on average a filter can only activate a limited number of hidden nodes and thus represents a specific image pattern. For the first four layers, a target sparsity of 0.1 is imposed on the hidden layer activations. The p_λ values specify the weights on the gradient of the sparsity penalty during the training. Both target sparsity and p_λ values are manually tuned so that the training of the network can achieve low reconstruction error. During network training, we set the learning rate to be 0.005, initial momentum to be 0.1, final momentum to be 0.9, and use one iteration of contrastive divergence for gradient approximation. The network weights are initialized by drawing from independent normal distributions with a standard deviation of 0.02. In addition, with $2,000$ epochs training for the first four training layer and $20,000$ for the last layer is applied in our cases. On a workstation withIntel Xeon E5-1620 v3 @3.5GHz CPU, GeForce GTX Titan GPU and 64GB Memory, the training takes 14134 seconds with 100 input images of size 200×200 for each and 1363130 network parameters.

3.4.1 Orientation-invariant filters

One common trait of microstructure images is the existence of low-level features that are similar under linear transformations. For example, grain boundaries of alloy samples in Fig. 5b are only different by orientations (of elongated grains). Therefore filters of the first CRBM layers can be parametrized by linear transformations, so that training of a small set of filters will be sufficient to capture all features, thus significantly accelerate the training process. Such as the material system of Ti64 Fig 8, which is composed by lines with different orientation visually, so the basic features for these kinds of material system should be edges with various orientations. In our proposed model, $K = 2$ rotation-invariant filters are incorporated, and $S = 12$ mannually pre-defined orientations with an interval of 15 degrees are given. This setting will result in 24 filters by training only 2(each one with 12 different orientations). Let the rotation matrices be $\{\mathbf{T}_s\}_{s=1}^{S}$, thus the parameterized filter for the sth orientation and the jth filter be $\mathbf{T}_s^T\mathbf{W}_j$, where $\mathbf{W}_j$ is the jth column of $\mathbf{W}$. Following (Sohn and Lee, 2012), the energy function can be updated as:

$$E(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta}) = -\sum_{j=1}^{K}\sum_{s=1}^{S}(\mathbf{T}_s\mathbf{v})^T\mathbf{W}_j h_{j,s} - \sum_{j=1}^{K}\sum_{s=1}^{S} b_{j,s}h_{j,s} - \mathbf{c}^T\mathbf{v}. \tag{3.5}$$

Note that among all S hidden nodes for the jth filter, at most one can be activated, i.e., $\sum_{s=1}^{S} h_{j,s} \leq 1, \text{for } h_{j,s} \in \{0, 1\}, j = 1, ..., K$. This is achieved by considering $h_{j,s}$ an outcome of a multi-class classifier, with

$$\Pr(h_{j,s} = 1|\mathbf{v}) = \frac{\exp\left((\mathbf{T}_s\mathbf{v})^T\mathbf{W}_j + b_{j,s}\right)}{1 + \sum_{s'=1}^{S}\exp((\mathbf{T}_{s'}\mathbf{v})^T\mathbf{W}_j + b_{j,s'})}. \tag{3.6}$$

The conditional probability of the visible nodes will be kept the same as before. With these updates, training of the rotation-invariant filters follows the discussion in Sec. 3.3.2.

3.4.2 Explanation of the network model

The choice of the network configuration requires some explanation. The first four layers are used to extract features at increasing length scales. The 5th layer is added to further reduce the dimensionality of the data by learning the correlations among the 4th layer activations, i.e., the co-existence of global features in the samples. The number of nodes at layer 4th and 5th are manually configured to achieve a balance between high dimension reduction and low reconstruction error. However, we did not test all possibilities to confirm that our network design yields the best result.

Fig. 8 visualizes filters of the five layers, learned from 100 Ti-6Al-4V alloy samples that are produced from the same laser sintering process. The visualization of each filter is done by iteratively convolving the filter (the network weights) with filters from the previous layers [1] . It is worth noting that the visualizations from the 5th layer include images of almost all black pixels. This is because the output images from these filters have almost uniform pixel values, appearing to be zeros after normalization. Also note that these filters are non-trivial, as the effect of activating multiple nodes at the output layer has nonlinear effect in the input space, i.e., the reconstructed image is not a simple addition of filter visualizations, because of the sigmoid operations throughout the network. To demonstrate, we show in Fig. 9(a)-(c) the reconstructions when the 1st, the 2nd, and both filters are activated.

To further justify the network architecture, we observed through experiments that the random microstructure reconstructions obtained using the 5-layer network have larger variance than those from without the 5th layer, as illustrated in Fig. 10. This result indicates that the 5th layer is effective for avoiding the creation of repetitive microstructure samples.

[1] This procedure is different from microstructure reconstruction (see Sec. 3.4.3) in that the sigmoid function is not applied to the convolution results.

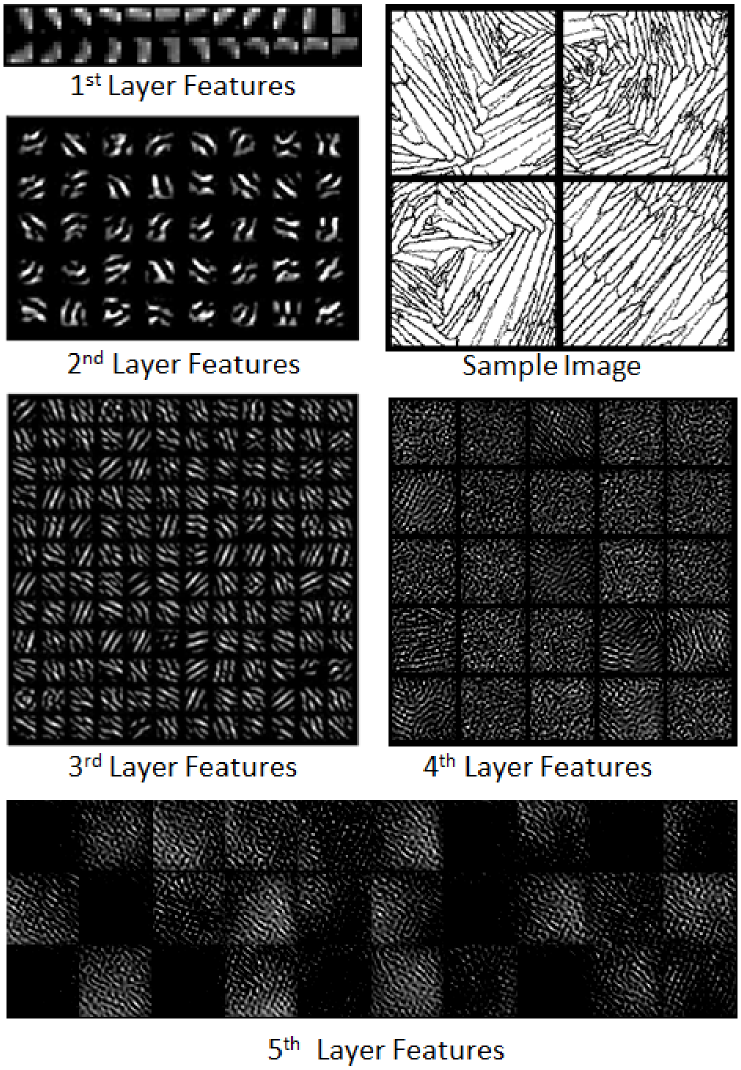
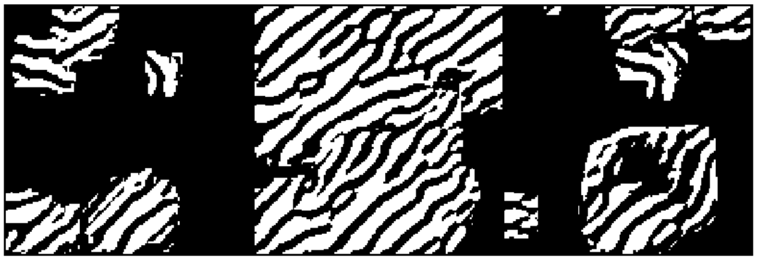


Figure 8: Input Images and Extracted Features. Visualizations of filters for the 5 layers: The 1st layer has two filters, each with 12 orientations (see Sec. 3.4.1). Due to limited space, Only the first 144 and 25 filters from the 3rd and 4th layers, respectively, are shown.

Table 1: Network and algorithmic parameters

Layer	#filter	filter size	sparsity	p_λ	rotation
1^{st}	2	6×6	0.1	10	12
2^{nd}	40	9×9	0.1	10	N/A
3^{rd}	288	9×9	0.1	10	N/A
4^{th}	1000	36×36	0.1	10	N/A
5^{th}	30	1000×1	0	0	N/A

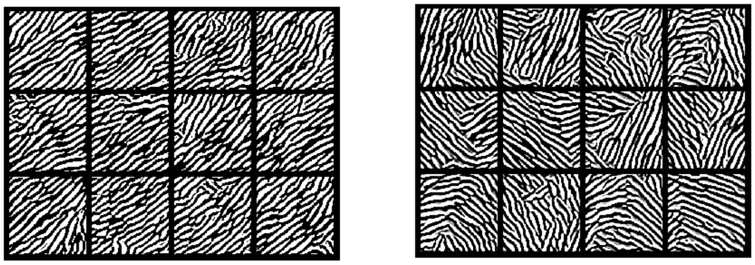


(a) 1^{st} filter activated

(b) 2^{nd} filter activated

(c) 1^{st} + 2^{nd} filter activated

Figure 9: Activation on Different Filters. From left to right, reconstructions when only the 1^{st}, only the 2^{nd}, and both nodes from the 5th layer are activated, respectively



(a). Random reconstruction from 4^{th} Layer

(b). Random reconstruction from 5^{th} Layer

Figure 10: Comparison between random reconstructions from (a) the 4th and (b) the 5th layers, sampled from the corresponding design spaces ($\{0,1\}^{1000}$ in (a) and $\{0,1\}^{30}$ in (b)).

3.4.3 Reconstruction and postprocessing

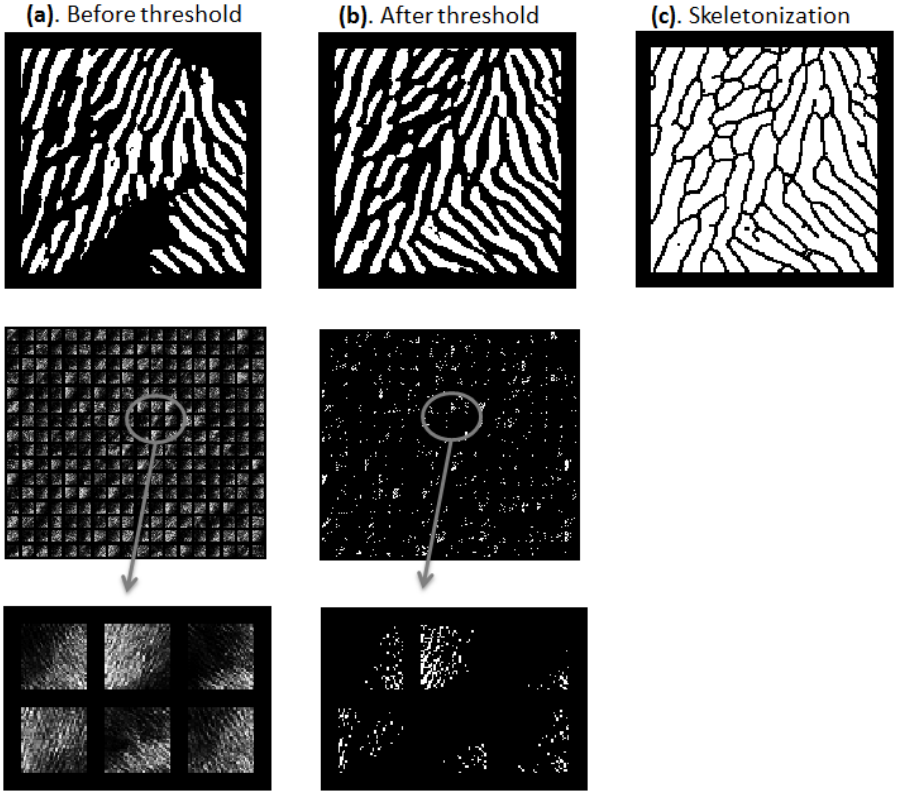


Figure 11: Postprocessing steps: Column (a) shows the original reconstruction and its 3rd layer activations (288 channels) with enlarged sample channels; Column (b) shows the reconstruction after thresholding the 3rd layer activations at 0.5; Column (c) shows the further improved reconstruction after skeletonization.

To reconstruct random microstructures, one starts by randomly assigning binary values to the last layer of the trained CDBN, and inversely sample the previous layers through deconvolution. To speed up the process, we directly treat activations (real values between 0 and 1) as the states of hidden nodes, to avoid drawing from Bernoulli distributions and eliminate randomness during the reconstruction. We discuss below the necessity of post-processing steps to avoid undesirable reconstructions.

First, we observe that with the proposed reconstruction method, the 3rd hidden layer (a $36 \times 36 \times 288$ image) is overly saturated, see Fig. 11a. It happens because the forward path of the proposed network has a Bernoulli sampling process, making the activation hidden layer binaries. But during the reconstruction, the sampling process is avoided in order to eliminate randomness. This causes the issue that the slightly activated nodes in the 3rd layer across its 288 channels, which should have a high chance to be sampled as zeros, to accumulatively affect the input layer, by creating overlapped grain boundaries (as black pixels) and undesirably rendering such regions as voids. Through experiments, we found that thresholding the activations of the 3rd layer at $\tau = 0.5$ achieves the lowest average reconstruction error from the original Ti64 samples, see Fig. 11b. Some random reconstructions using this threshold are shown in Fig. 12a. It is worth noting that this fixed threshold value may result in invalid reconstructions for some binary combinations at the output layer, see Fig. 12a, two rows of the reconstruction microstructures are processed with the same threshold value 0.5, but the performance is very different. The four images in the second row have more unexpected voids than the images in the first row. One potential solution is to fine-tune the threshold for each individual reconstruction based on a heuristic criterion, e.g., one can remove invalid voids in Ti64 reconstructions by thesholding the hidden layer so that the volume fraction of the resultant microstructure falls into a range calculated by valid reconstructions of the original samples, see Fig. 12b for an example. Nonetheless, we are yet to identify thresholding criteria that are universal across all material systems.

Secondly, we notice that the resultant grain boundaries are wider than those from the original samples. This is due to the convolution and the max-pooling operations in the first three layers of the proposed CDBN. Applying an existing skeletonization method to the reconstruction results in the final outcome in Fig. 11c. Lastly, the convolution operation tends to smear the reconstructed image towards its boundary. The reported reconstruction results in this work are the cropped areas (143×143) without the blurred margins, which are resultant from the convolution.

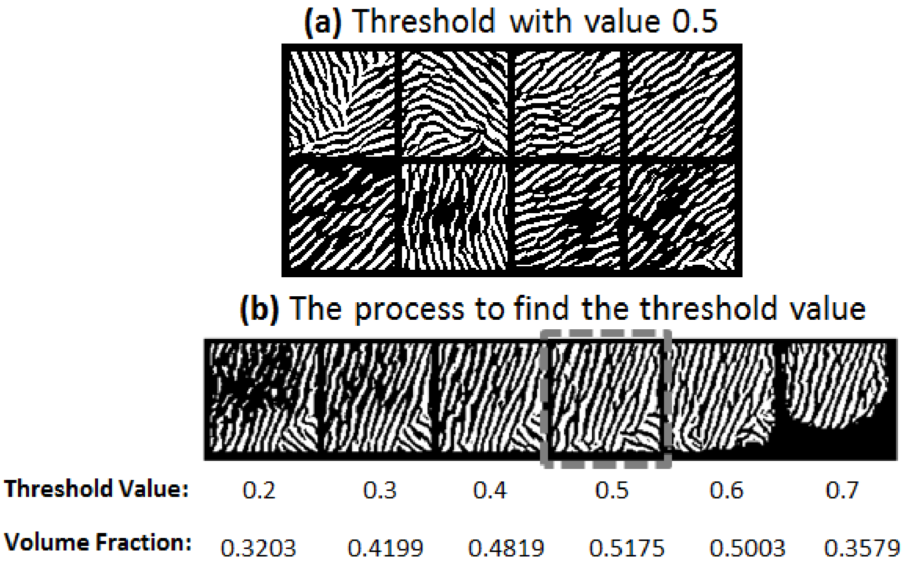


Figure 12: (a) A threshold of 0.5 does not always guarantee valid reconstructions, see the second row (b) Threshold can be fine-tuned based on a heuristic criterion: Here we test a set of threshold values and pick the one that yields a volume fraction closest to the average volume fraction of reconstructions of the original samples ($\rho = 0.51$).

3.4.4 Random reconstruction results

Here we demonstrate the reconstruction performance of the proposed CDBN on four material systems: Ti-6Al-4V (100 images in Fig. 47), Pb-Sn(lead-tin) (60 images in Fig. 48), Pore structure of Fontainebleau sandstone (60 images in Fig. 49) and 2D suspension of spherical colloids (80 images in Fig. 50). Methods used to generate images of these four material systems are as follows: Ti64 are generated from physics-based kinetic Monte Carlo simulations (Y. Jiao and Chawla, 2013) for beta to alpha phase transition starting from beta phase field; Pb-Sn optical images are generated from polished Pb-Sn solder ball surface (Jiao *et al.*, 2013); sandstone images are virtual 2D slices from 3D tomography reconstruction (Li *et al.*, 2016); and random sphere packing are generated using Monte Carlo simulations (Li *et al.*, 2014).

The CDBN specifications follow Sec. 5.3: Its input and output are a 200-by-200 image and a 30-dimensional binary vector, respectively. To generate microstructures, we can assign binary values to the 30 output neurons, and perform convolution backwards using the learned features. Fig. 13 provides a visual comparison between samples and the corresponding random reconstructions. The first three material systems follow almost the same network setting, with the exception for Ti-6Al-4V alloy where the first layer is orientation-invariant. The network for spherical colloids is slightly different than the other three, since its universal local feature, a sphere, can be directly extracted from a single CRBM layer. Thus the rest four layers in this case are all fully-connected RBMs. From the results, it is evident that the proposed network has some general applicability across material systems.

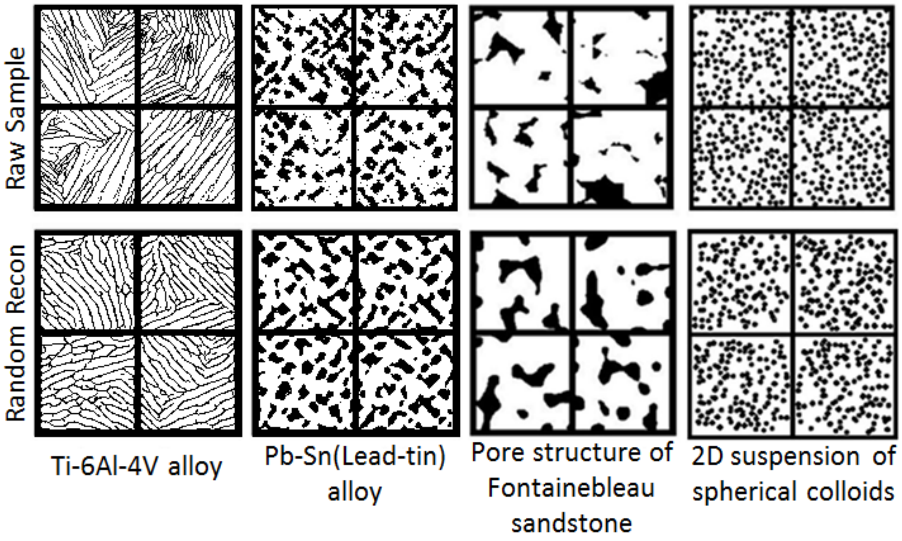

Raw Sample
Random Recon
Ti-6Al-4V alloy
Pb-Sn(Lead-tin) alloy
Pore structure of Fontainebleau sandstone
2D suspension of spherical colloids

Figure 13: A comparison between random reconstructions (bottom) and the original samples (top) for four different material systems.

3.4.5 Deeper understanding of the proposed model

To further understand the performance and limitations of the proposed CDBN, we conduct two additional studies: The first concerns the restoration of original samples, and the second validates the reconstruction of the material property of interest, i.e., the critical fracture force computed using a recently developed volume-compensated lattice-particle (VCLP) method (Chen *et al.*, 2014).

For the first study, samples of the original Ti-6Al-4V alloy microstructure images and their corresponding reconstructions are shown in Fig. 15. A close examination reveals discrepancies between the reconstructed and the original images. We believe that such discrepancies are caused by the nature of the network: Similar local patterns are learned as a single filter, and are replaced by this filter during reconstruction. Increasing the number of filters to learn may alleviate this issue, yet will also increase the computational cost for learning the network. Therefore a compromise is necessary. For example, for Ti64, Pb-Sn and Sandstone, we use 2 filters with 12 orientations in the first CRBM layer, and may fail to capture features (e.g., line segments) that do not belong to these pre-set orientations. The use of probabilistic max-pooling may also contribute to the discrepancies as forward and backward pooling do not preserve the activations. Note that preservation of local microstructure patterns can be achieved relatively easily for simple systems such as spherical colloids.

To quantitatively access the accuracy of the reconstructions, we computed the two-point correlation function $S_2(r)$, which gives the probability of two randomly selected points separated by distance r falling into the same phase of interest, for both the samples and the associated reconstructions. Fig. 27 shows the comparison. It can be seen that S_2 for the reconstructed microstructures statistically match the corresponding target structures well, except for the Ti64 system. We note that for the Ti64 system, both S_2 for the target and reconstruction exhibit clear oscillations for small r values, indicating significant short range correlations due to the mutual exclusion volume between the grains. The wavelengths associated with the oscillations indicate the average grain width, and are the same in both functions. The oscillation in the reconstructed S_2 is apparently stronger, which is due to more uniform grain size and shape distribution in the reconstructed microstructure, and the missing of local details from the samples.

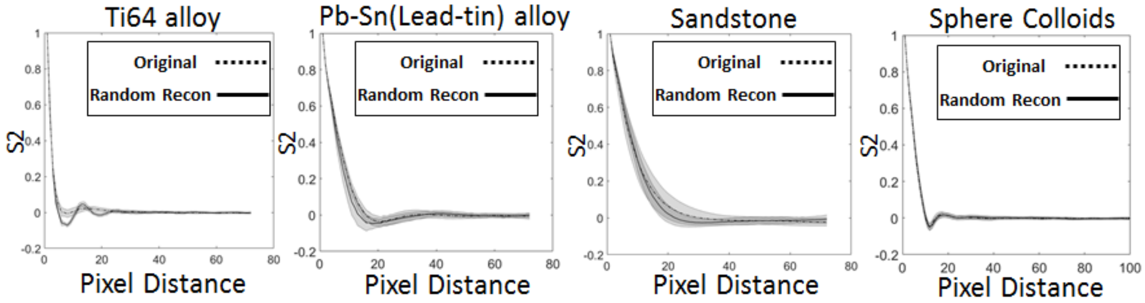


Figure 14: The 2-point correlation functions for the four different materials.

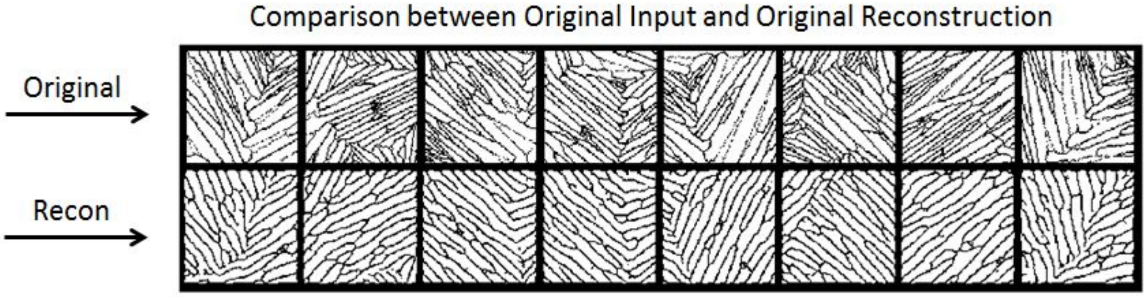

Comparison between Original Input and Original Reconstruction
Original
Recon

Figure 15: Comparison between the samples (top) and their reconstructions (bottom). Details of the two fail to match.

In addition, we investigate the performance-wise discrepancies introduced by the reconstructions. Specifically, we calculate the critical fracture force of three sets of samples: the original samples, their reconstructions, and 100 random reconstructions, using VCLP. The simulation results from four material systems are summarized in Fig. 16, with two findings: (1) the average fracture forces of the three groups are statistically similar within each material system; and (2) discrepancies in the fracture force are visible between the original samples and their reconstructions, due to the aforementioned reconstruction error. We argue that finding (1) is a desirable result, while individual-wise property matching may not be necessary. This is because each processing setting may produce various structures with small variance in their property values. Thus, during the structural design, it is required that the set of

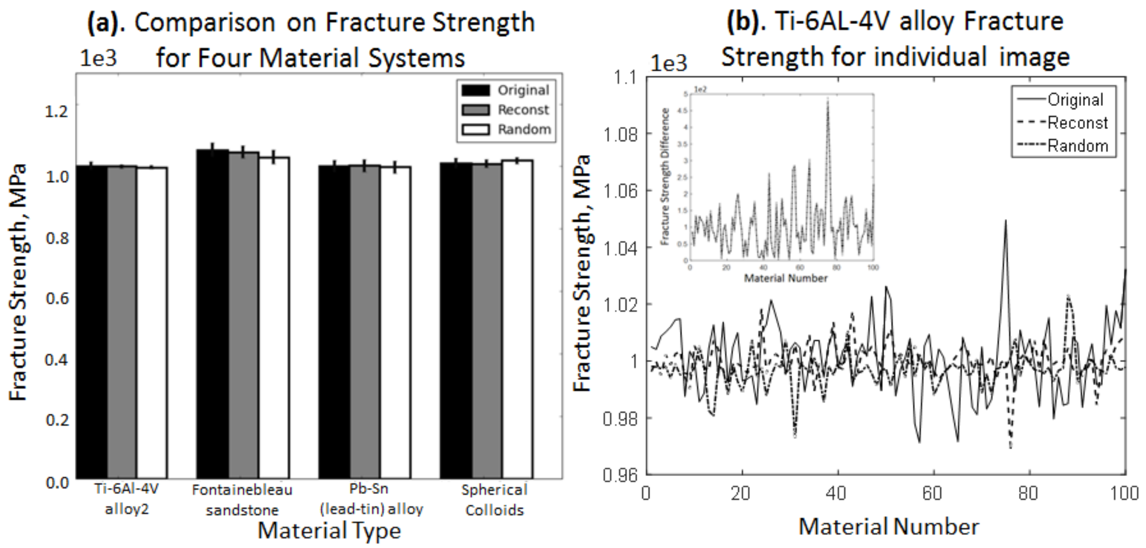


Figure 16: a) Comparison on critical fracture strength among the four material systems between the original samples (black), their corresponding reconstructions(gray) and random reconstructions (white) (b) Comparison on critical fracture strength among individual original images and the related reconstruction images.

reconstructions of all samples derived from the same processing parameters (as is the case in this study) have statistically the same property as the original samples.

3.5 Discussion

Here we provide more discussions on the limitations and future directions of the presented work.

3.5.1 Limitations

First, we note that the network is designed to only reconstruct images of a fixed size, which is not desirable when scalable synthesis of material systems is needed. One potential solution is to apply a conditional probability model (e.g., (Bostanabad *et al.*, 2016)) to the outputs of a hidden layer of the CDBN that represent the activations of key microstructure patterns. The intuition is that while the Markovian assumption may not directly hold at the image level (i.e., image pixels cannot be homogeneously inferred from their surroundings), the activations of local patterns may still be captured by this assumption. Such a hybrid model could be used to produce new microstructure patches given its surrounding patches, thus achieving synthesis of complex material samples.

Secondly, identification of proper network specifications (e.g., the number of layers, the number of filters, and filter sizes) is non-trivial. In fact, an arbitrary choice of these parameters could lead to abysmal feature extraction and reconstruction performance. Due to the unknown sensitivity of the performance with respect to the modeling parameters, the authors cannot claim that the presented model is universally applicable to all material systems, despite the fact that the model worked well on the four demonstrated systems.

Thirdly, the size of training samples can significantly affect the reconstruction performance. To show this, we learned three network models based on 10 and 50 training samples randomly drawn from the 100 Ti64 images. We then reconstruct 100 random images using each of the trained models and image-wise variances for each model are calculated as 0.2644, 0.2863, and 0.4205 respectively. As expected, the result shows that larger training data lead to more distinct filters, which leads to the reconstructions with high variance Fig. 17.

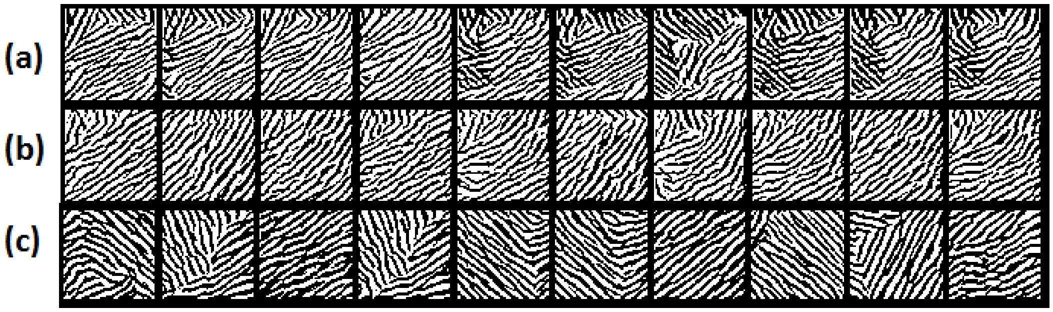


Figure 17: (a-c) Sample reconstructions (without skeletonization) based on models trained from 10, 50, and 100 samples, respectively. The variances among random reconstructions for these three cases are 0.2644, 0.2863, and 0.4205, respectively. The variance is calculated from 100 reconstructions in each case.

Lastly, it is also worth noting that the proposed network essentially learns a lower-dimensional manifold from the image space using limited samples. However, it does not provide a feasible region within this manifold. As a result, random reconstructions are not guaranteed to be physically meaningful, and a validation through processing-structure mapping is needed. It is therefore desirable that certain known physics-based rules (i.e., constraints on features) can be incorporated into the feature extraction process to eliminate or reduce the chance of creating infeasible reconstructions.

3.5.2 From material reconstruction to material design

As mentioned in the introduction, the feature learning method proposed in this work has two potential contributions to computationally efficient material design: The reduced design space will enable more tractable search for complex microsturcture designs, and allows statistical models with better prediction performance to be built for process-structure-property mappings. Achieving the latter, however, requires the learned features to be able to explain variance in material properties (in the structure-property mapping), and in processing settings (in the process-structure mapping). While this study does not directly demonstrate the connection between features and properties or processing parameters, successful reconstructions through the developed model showed its potential at learning microstructure patterns, which could be fine-tuned through a supervised model to statistically explain properties or processing parameters.

3.6 Conclusions

This work presented a novel methodology for feature extraction and reconstruction of complex microstructures through a convolutional deep belief network. Using four different material systems, we showed that the proposed 5-layer CDBN model, along with postprocessing techniques, can achieve significant dimension reduction, visually and statistically plausible random reconstruction, and statistically preserve the critical fracture strength values after reconstruction. Key limitations include (1) the lack of scalability in reconstruction, (2) the individual discrepancies in fracture strength between original and reconstructed images, and (3) the lack of guarantee of the validity of the reconstructions, i.e., the threshold for the reconstruction is only empirically chosen. Nonetheless, the demonstrated advantages over existing material reconstruction approaches showed that our method could lead to more efficient representations of complex microstructures, which is critical to accelerating ICME.

Chapter 4

SCALABLE MICROSTRUCTURE RECONSTRUCTION WITH MULTI-SCALE PATTERN PRESERVATION

4.1 Introduction

Last chapter we propose an automated feature learning techniques that is to identify an encoder that maps microstructure to a lower-dimensional feature space and a decoder that creates a microstructure based on given features (Cang and Ren, 2016; Coates *et al.*, 2010; Boureau *et al.*, 2008; Vincent *et al.*, 2010). The two-way mapping allows the optimization to be performed in the feature space, solutions from where can be evaluated through the decoded microstructures. However, the proposed approach requires the generated images to have the same size as the training samples, and thus are not scalable for designing microstructures of an arbitrary size. Alternatively, Markovian Random Field (MRF) models have been applied to directly learn the pixel value distribution of images (Bostanabad *et al.*, 2016). MRF is complementary to feature learning, in that it allows scalable reconstruction yet may produce undesired microstructures when the Markovian assumption fails to hold (e.g., when complicated local patterns exist, see Fig. 1 in (Cang and Ren, 2016) for example).

It is thus natural to propose a hybrid model that combines the advantages of feature learning and MRF. Specifically, we use a convolutional deep belief network (CDBN) (Lee *et al.*, 2009) to extract local microstructure patterns, and then use a MRF model to learn the distributions of the *activation* of these patterns. We show that the proposed model can generate random microstructure samples of arbitrary sizes, that are statistically and visually similar to the training samples of complex material systems. Fig 18 presents the random generation of Ti-6Al-4V, Pb-Sn alloy, Sandstone, and Spherical Colloids, along with their training samples.

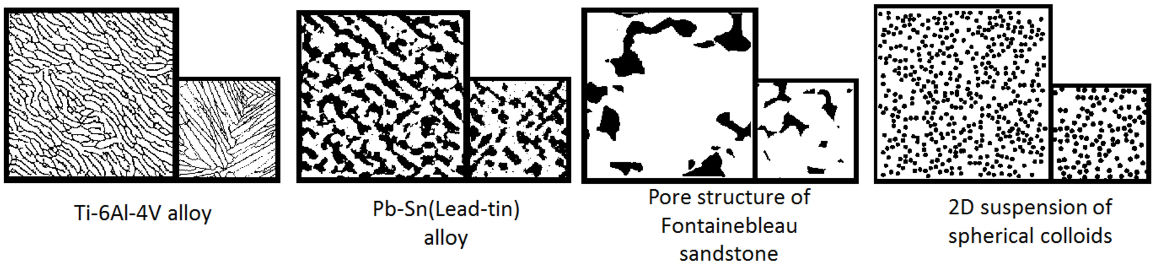

Ti-6Al-4V alloy
Pb-Sn(Lead-tin) alloy
Pore structure of Fontainebleau sandstone
2D suspension of spherical colloids

Figure 18: Our hybrid method combines advantages from feature learning and Markovian random field to enable scalable random reconstruction with local-pattern preservation. Four material systems are tested: the larger images are random reconstructions by learning from their corresponding samples shown on the right (multiple samples are used for learning, only one is shown).

4.2 Learning Distributions of Local Microstructure Patterns

Following ((Cang *et al.*, 2016)), the proposed CDBN is composed by three stacked CRBM layers, each followed by a probabilistic max-pooling layer that reduces the height and the width of the output by a given factor. The outputs from the last pooling layer is modeled to follow a random field, so that new microstructures can be generated by first sampling an output layer of the CDBN using the random field, and then performing convolutions backward through the CDBN. See Fig. 19.

4.2.1 Learning local microstructure patterns

We briefly introduce key elements of the CDBN and provide details of the proposed CDBN settings.

Restricted Boltzmann Machine (RBM) A RBM is a bipartite graphical model with visible ($\mathbf{v}$) and hidden ($\mathbf{h}$) nodes. The model is parameterized by the weight matrix $\mathbf{W}$ for its edges and biases $\mathbf{a}$ and $\mathbf{b}$ for visible and hidden nodes, respectively.

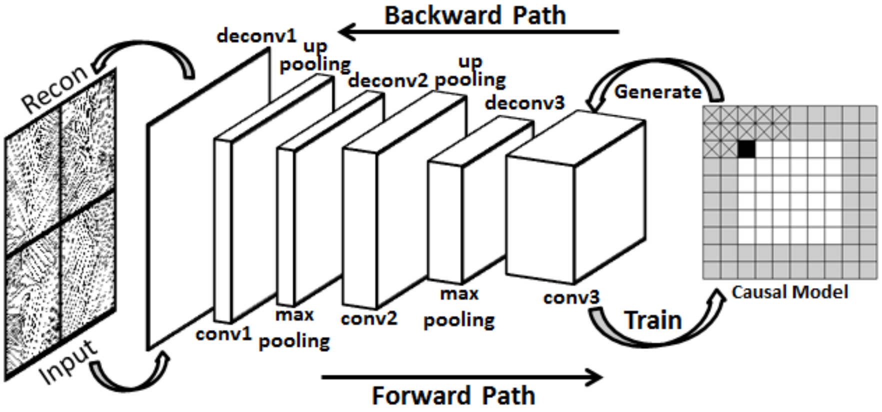


Figure 19: The proposed network with three convolutional RBM layers followed by a random field model(causal model). Image generation is done by sampling from the random field and performing backward convolution through the CDBN.

The joint probability of $\mathbf{v}$ and $\mathbf{h}$ is modeled as

$$p(\mathbf{v},\mathbf{h}) = \frac{\exp\left(-E(\mathbf{v},\mathbf{h}|\boldsymbol{\theta})\right)}{\sum\limits_{\mathbf{v},\mathbf{h}} \exp\left(-E(\mathbf{v},\mathbf{h}|\boldsymbol{\theta})\right)}, \tag{4.1}$$

where

$$E(\mathbf{v},\mathbf{h}|\boldsymbol{\theta}) = -\sum_{i=1}^{n} a_i v_i - \sum_{j=1}^{m} b_j h_j - \sum_{i=1}^{n}\sum_{j=1}^{m} v_i W_{ij} h_j \tag{4.2}$$

is the Boltzmann energy, and $\boldsymbol{\theta} := \{\mathbf{W},\mathbf{a},\mathbf{b}\}$ is the parameter set. Given a set of inputs to the visible nodes (e.g., microstructure images), the network weights and biases can be learned by maximizing the marginal probability

$$p(\mathbf{v}) = \frac{\sum\limits_{\mathbf{h}} \exp\left(-E(\mathbf{v},\mathbf{h}|\boldsymbol{\theta})\right)}{\sum\limits_{\mathbf{v},\mathbf{h}} \exp\left(-E(\mathbf{v},\mathbf{h}|\boldsymbol{\theta})\right)}. \tag{4.3}$$

This optimization problem can be solved by stochastic gradient descent (Bottou, 2010) and contrastive divergence (Hinton, 2002). In a *convolutional* RBM, edges share weights in such a way that the encoding ($\mathbf{Wv}+\mathbf{b}$) and the decoding ($\mathbf{Wh}+\mathbf{a}$) are convolution operations.

Probabilistic max-pooling A probabilistic max-pooling layer((Lee *et al.*, 2009)) is attached to each CRBM to reduce its size. This allows faster training and extraction of features of larger scales for the follow-up CRBMs, and builds into these follow-up filters a degree of invariance to translations and distortions. In forward pooling, hidden channels are divided into smaller blocks, each corresponding to a node in the pooling layer. For max-pooling with factor α, the block is of size $\alpha \times \alpha$. The states of the pooling nodes are drawn from Bernoulli distributions with parameters defined by the states of the corresponding block of hidden nodes. For backward pooling, all nodes in the block are assigned the same state as the corresponding pooling node.

4.2.2 A Markov Random Field model for pattern activations

We introduce an existing MRF model((Liu and Vadim, 2015; Sundararaghavan, 2014)), and its application to the output layer of the CDBN, instead of direct application to the microstructure image. While we focus on 2D images, the extension of this model to 3D images can be achieve with minor modification((Bostanabad *et al.*, 2016)).

Model overview The model assumes that each pixel in a 2D image follows a stationary MRF((Bostanabad *et al.*, 2015)), i.e., a pixel value Y_{ij} at location (i, j) follows a distribution $p(Y_{ij}|X_{ij}; \phi)$ that is conditioned on its neighboring pixels X_{ij} and parameterized by ϕ. When Y_{ij} is binary, the distribution is Bernoulli. The parameters ϕ can be learned through a set of (Y_{ij}, X_{ij}) pairs sampled from given images through supervised learning. Therefore, this model offers both an encoder and a decoder: An image (or a set of images) can be mapped to a single point ϕ in the feature space; inversely, one can generate images by maximizing the log likelihood $\sum_{ij} \log\left(p(Y_{ij}|X_{ij}; \phi)\right)$ with respect to the image, for a given ϕ. A greedy solution to this inverse problem is stated in (Bostanabad *et al.*, 2015): Starting from an empty image, we draw Y_{ij} sequentially from the top-left to the bottom-right corner,using only those neighbouring pixels that have been drawn. See Fig. 20 for an illustration.

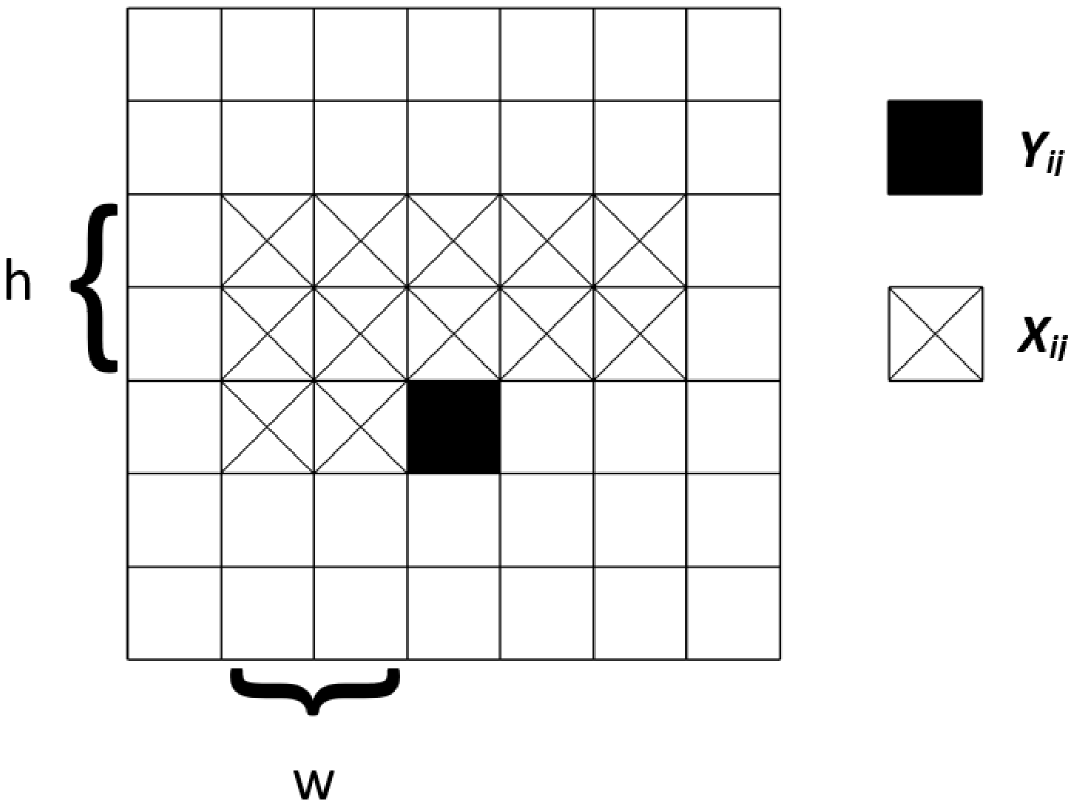


Figure 20: The pixel Y_{ij} is conditioned on its $(h(2w+1)+w)$ neighboring pixels X_{ij}

It is important to note that unlike CDBN, the encoder-decoder pair in this case does not preserve specific images, i.e., the similarity between images is measured only in the feature space. From a material design perspective, this model is useful for material systems where the statistics (such as ϕ) rather than the image representations is sufficient and effective at explaining the variances in properties and processing parameters. We will elaborate on this point in Sec. 5.12.

Setting boundary padding One may notice that there is a need to provide boundary padding to images for both encoding and decoding. This is because the MRF model requires each pixel to be dependent on its neighbors, including the ones on the boundary. Boundary padding is thus created around the image to provides sufficient preceding pixels to draw Y_{ij}s on the boundary. See Fig. 21 for an illustration. The padding is set to black or white, whichever is dominating in the sample images.

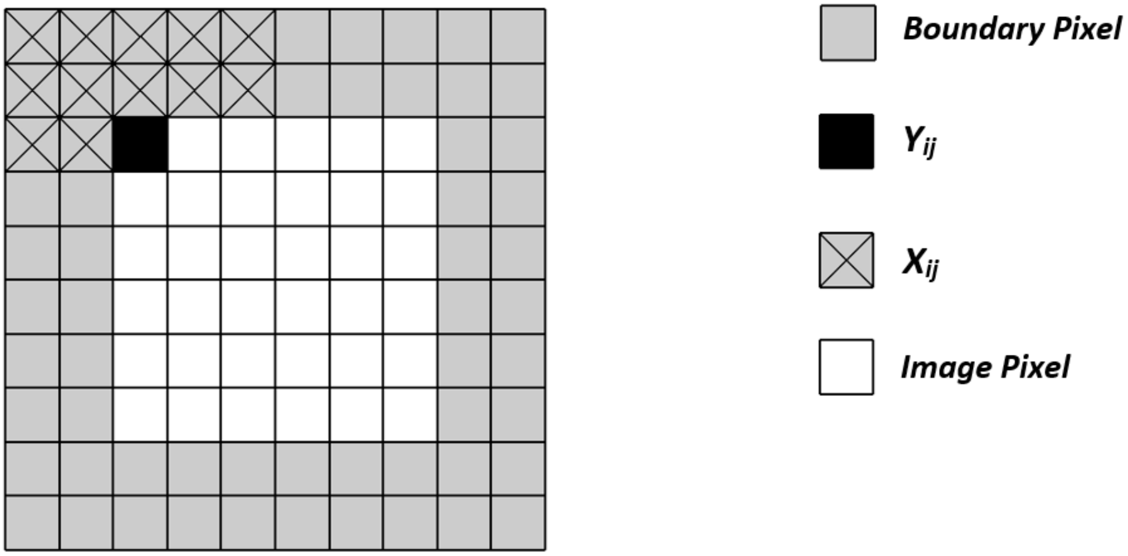


Figure 21: The padding is shown in grey. Y_{ij} on the boundary depends on X_{ij} in the padding. Padding thickness is $(2h + 1)$.

Learning the conditional probability model We learn K conditional probability models, where K is the number of channels of the CDBN output layer of a particular material system. To do so, we first randomly sample $(Y_{ij}^{(k)}, X_{ij}^{(k)})$ pairs from each channel across all sample images, where $X_{ij}^{(k)}$ are preceding pixels in the neighbour of $Y_{ij}^{(k)}$. As will be explained in Sec. 4.3.1, the best neighbouring window size $(h + w + 1)$ varies with material systems. Note that in addition to the neighboring pixels, we also include in $X_{ij}^{(k)}$ an independent variable $\gamma_{ij}^{(k)}$ that represents the percentage of activated nodes from the preceding channels at location

(i, j): $\gamma_{ij}^{(k)} = \sum_{k'=1}^{k-1} Y_{ij}^{(k')}/(k-1)$. For the first channel, this variable is omitted. This treatment prevents overly activating one location across channels by accident. The prepared data is then used to train K models using a classification tree algorithm (*fitctree*) in MATLAB. The tree depth is $n-1$ where n is the training sample size. The minimal leaf size is 1, and the minimal parent size is 10.

Reconstruction with a consistent volume fraction Once trained, the classifiers output the probabilities $p_{ij}^{(k)} := p(Y_{ij}^{(k)} = 1 | X_{ij}^{(k)}; \phi)$ for given $X_{ij}^{(k)}$. Each channel is generated using the corresponding classifier through the method described above. To ensure volume fraction conservation in each channel, we use the iterative procedure described in (Bostanabad *et al.*, 2015): In each iteration, we calculate the volume fraction gap (δ) between the current reconstruction and the training images. If $\delta > 0.0025$, we adjust the conditional probability as ${p'}_{ij}^{(k)} = p_{ij}^{(k)} + c\sqrt{p_{ij}^{(k)}(1 - p_{ij}^{(k)})}$, where the algorithmic parameter c is updated as $c = c + 0.005\text{sign}(\delta)$, initialized at $c = 0$. This procedure reduces the probability of activating a node when the reconstructed volume fraction is higher than that of the samples, and vice versa.

Addressing correlations among channels We noticed that the activations across channels in the CDBN output layer are correlated. And the same correlation occurs across images. See Fig. 22 for example. The proposed random field model, however, does not capture this correlation, as the classification trees are trained almost independently except for the consideration of γ. To enable correlations, we propose to identify the clustering of channels based on their activation patterns from the training samples. Specifically, we first calculate the channel-wise mean activation across all samples, although the activation pattern is very different for every image, the same correlation occurs across all the training samples, in this case K-means((MacQueen

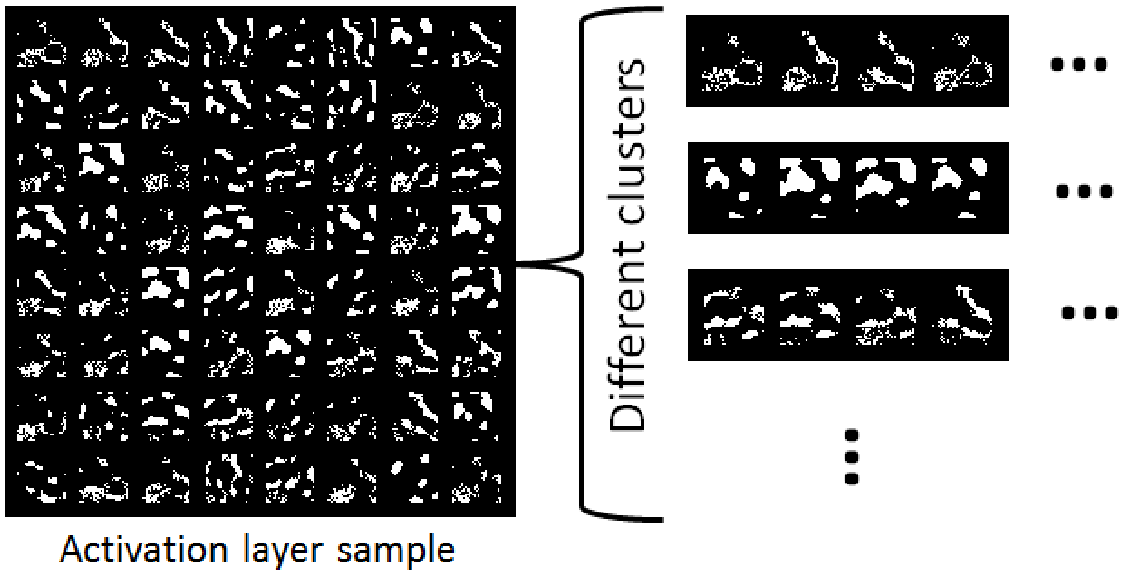


Figure 22: Sample activation layer from sandstone. Clusters are also observed in other material systems.

et al., 1967)) is applied to cluster these mean activation. The resultant clusters will be applied during the reconstruction, where we use the same patterns for channels within a cluster. The appropriate number of clusters is determined by calculating the reconstruction error, i.e., image-wise difference, between the original samples and the reconstructed ones with shared patterns. When calculating this difference, the shared pattern for each cluster is chosen randomly from one of its channels.

4.2.3 Post-processing

Post-processing is needed to improve the quality of reconstructions for Ti64. As discussed in ((Cang and Ren, 2016)), the third hidden layer of the CDBN can be overly saturated, which leads to regions in the input layer with overlapped grain boundary or undesirable voids. The same issue is observed in the hybrid model. Our current solution is to set an upper bound on the activation rate of all channels. Fig. 23 shows the activations of the CDBN output layer above and their corresponding reconstructions below. One can see that directly applying the generated activation will lead to less desirable outcomes. Experiments show that reducing the activation rate to 15% will sufficiently improve the reconstruction quality, because by reducing the activation rate cross all the channels avoids the over-activation issue Fig 26. Finally, skeletonization is used to further match the style of the generated images to the original ones.

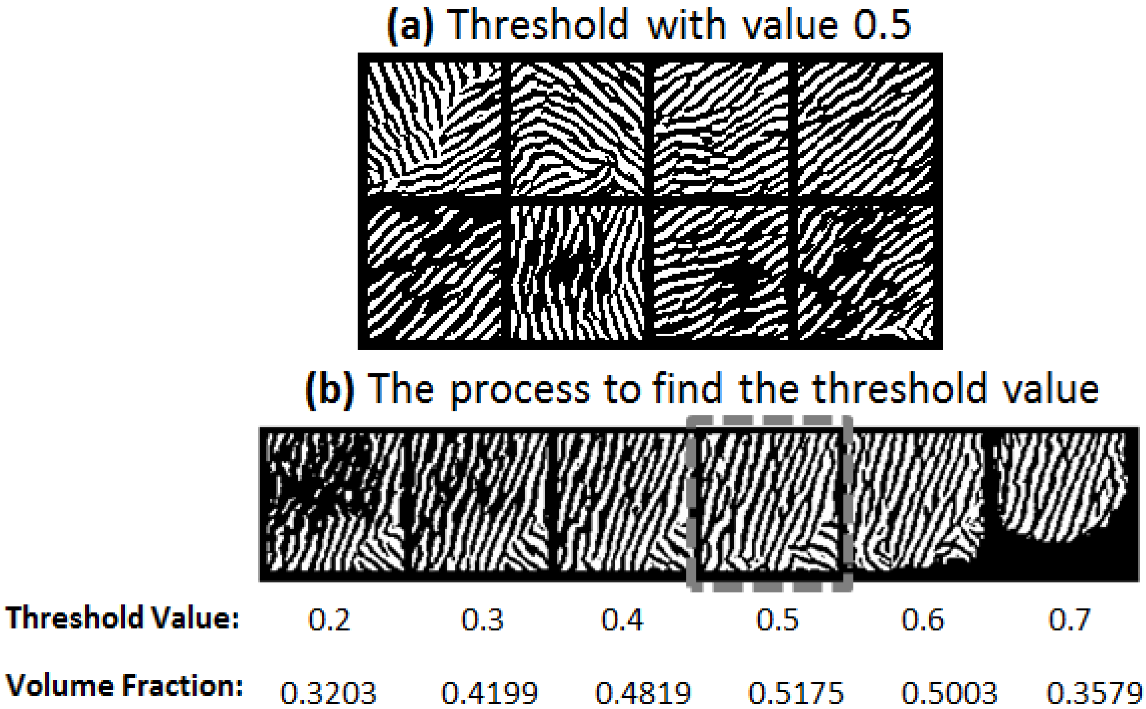

(a) Threshold with value 0.5
(b) The process to find the threshold value
Threshold Value: 0.2 0.3 0.4 0.5 0.6 0.7
Volume Fraction: 0.3203 0.4199 0.4819 0.5175 0.5003 0.3579

Figure 23: (Top) CDBN output activations generated by MRF for Ti64; (Bottom) The corresponding generated images; (Left) No activation reduction after sampling; (Middle) 85% reduction on the total activation; (Right) Result after skeletonization

4.3 Verification of the Proposed Method

This section elaborates on the algorithmic settings of the proposed hybrid model for four material systems: Ti-6Al-4V alloy (Ti64), Pb-Sn alloy (Pb-Sn), sandstone (SS), and spherical colloids (SC). Due to differences in their local patterns and distributions of these patterns, the settings vary across these systems. We then compare the generation quality between the hybrid and the MRF models. Our data consists of 100 samples for Ti64, 60 samples for Pb-Sn and SS, and 80 samples for SC. All sample images used for model training are listed in the Appendix. We also show generations of different image sizes using the proposed method.

4.3.1 Algorithmic settings

Settings of the MRF model We apply the MRF model directly to microstructure images to produce benchmark images. Due to the sensitivity of the MRF approach to the window size (as illustrated in Fig 24), we determine the optimal size for each material system via cross-validation (CV) by starting with a relatively large window size and reducing the size till the CV error of the model does not further decrease. The CV errors are measured by the differences between the two-point correlation functions of the generated images and those of the original samples. The optimal sizes are thus derived as 7×7, 17×17, 19×19, 15×15, for Ti64, Pb-Sn, SS and SC, respectively. The minimum CV errors are 7.7% for Ti64, 4% for Pb-Sn, 0.95% for SS, and 2.2% for SC. This data indicates that Ti64 follows the Markov assumption of MRF less strictly than the other material systems.

Settings of the hybrid model The configurations of the CRBM layers are tuned according to each material in order to achieve good generation quality, as different material systems may have patterns of different length scales. All models have three CRBM layers followed by probabilistic max-pooling layers, except for SC due to its simple spherical feature, in which case a single CRBM layer is used. For Ti64, Pb-Sn, and SS, the convolutional filter sizes are 6×6, 9×9, and 9×9 for the first, second, and third CRBM layers, respectively; the number of filters for the first and second layers are 24 and 40 respectively; The number of filters for the third layer varies

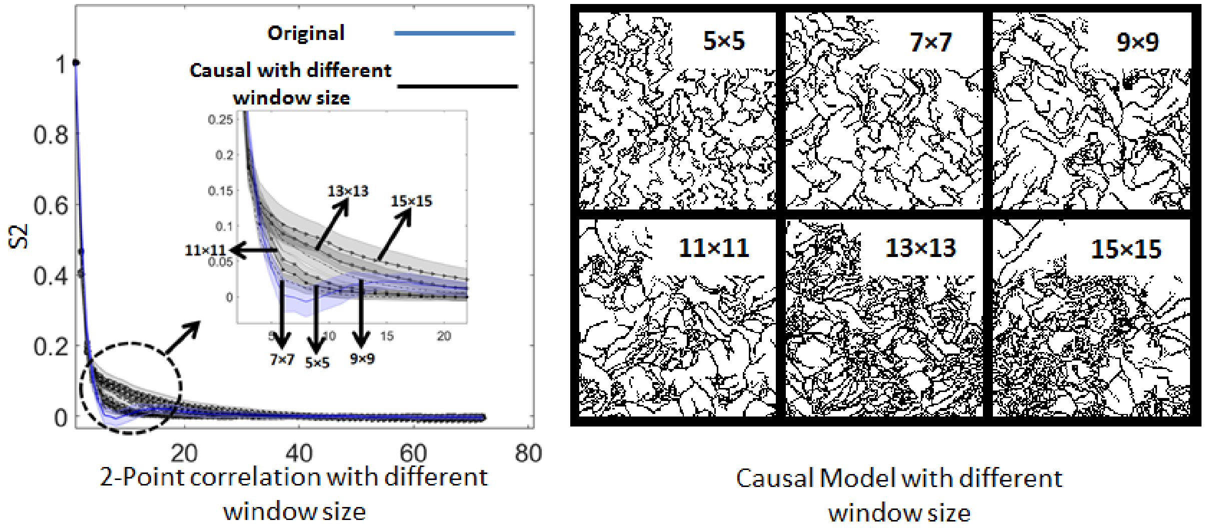


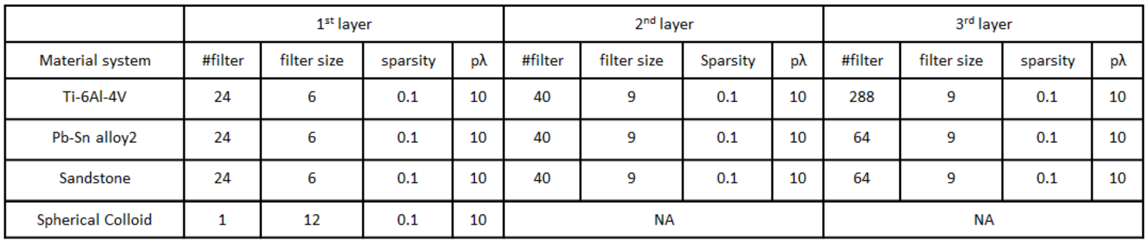
Figure 24: (Left) 2-point correlation functions with window sizes: 5 × 5, 7 × 7, 9 × 9, 11 × 11, 13 × 13 and 15 × 15 (Right) Corresponding generations

according to the microstructure complexity. Experiments showed that 288, 64, and 64 filters will be sufficient for Ti64, Pb-Sn and SS, respectively. We notice that using these reduced amount of filters for Pb-Sn and SS leads to 6% of average pixel-wise reconstruction error, compared with those with 288 filters. Max-pooling layers with a factor of $\alpha = 2$ is applied to Ti64, Pb-Sn, and SS, and $\alpha = 3$ for SC.

The MRF model is applied to the last activation layers. Similar to the direct MRF models, the window size for training the models is chosen through CV. For all materials, the optimal window size is found to be 5 × 5.

Target sparsity for the hidden nodes are set during the training of the CRBM network in order to prevent learning trivial features. This ensures that, during the training, a filter can only activate a limited number of hidden nodes and thus represent a specific image pattern. Table 2 summarizes the model and training parameters.

The number of clusters for the MRF applied to the CDBN output is tuned for each material system. It is identified that 8 clusters work well for Pb-Sn and SS. Fig. 25 compares the reconstructions from 4 to 64 clusters. Pb-Sn is shown as an



	1st layer				2nd layer				3rd layer			
Material system	#filter	filter size	sparsity	pλ	#filter	filter size	Sparsity	pλ	#filter	filter size	sparsity	pλ
Ti-6Al-4V	24	6	0.1	10	40	9	0.1	10	288	9	0.1	10
Pb-Sn alloy2	24	6	0.1	10	40	9	0.1	10	64	9	0.1	10
Sandstone	24	6	0.1	10	40	9	0.1	10	64	9	0.1	10
Spherical Colloid	1	12	0.1	10	NA				NA			

Table 2: CDBN settings

example. Note that the reconstruction error decreases with more clusters. No cluster is needed for SC since only one channel exists. Ti64 material system is not suitable for clustering as doing so leads to undesirable reconstructions, as is demonstrated in the top row of Fig. 26. By checking the summation of all the activation channels (the bottom row of Fig. 26), we notice that the regions with high activations, due to shared patterns within clusters, lead to voids in the reconstructions.

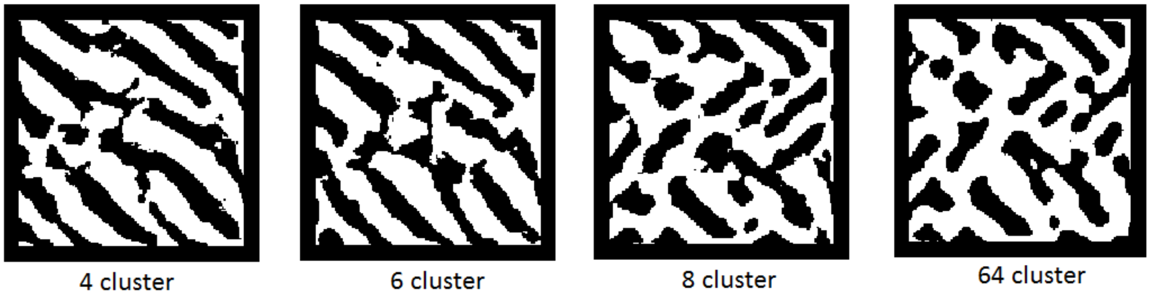


Figure 25: Different cluster number on the 64 channels' activation layer based reconstruction for Pb-Sn alloy is shown, as the cluster number increases the final reconstruction gets close.

Settings of the similarity measure A similarity measure is needed for evaluating the quality of reconstructions and random generations. In this work, we measure the similarities between sample images and reconstructions both visually and through 2-point correlation functions (S_2), which is a very effective choice of measurement for homogeneous and isotropic materials (Liu *et al.*, 2013; Jiao *et al.*, 2008), such as Pb-Sn, SS and SC. While commonly agreed similarity measure for complicated material systems such as Ti64 is currently missing, the same measure is applied for

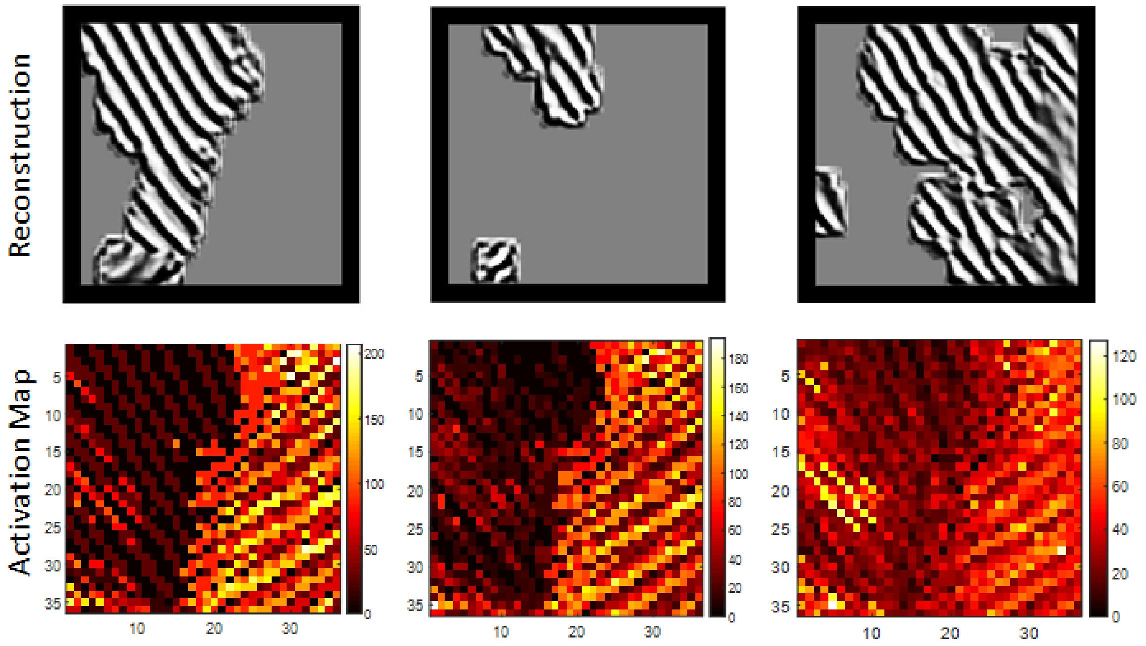


Figure 26: Clustering the the activation channels for Ti64 into three different groups, 8, 16, 36, from left to right. None of them produce a good reconstruction as shown in the upper row. The lower row presents the summation of their corresponding activation channels.

consistency. To calculate 2-point correlation functions for bi-phase microstructures, we sample all the possible pairs of points (x_1, x_2) for distances from 1 to half of the length of the image. At each distance, the percentage of pairs with matched pixel values is calculated.

4.3.2 Comparisons on random generations

Fig. 27 compares the performance between MRF, CDBN((Cang and Ren, 2016)) and the proposed hybrid method, using the four material systems. For each system, we present the original microstructures, and random generations by MRF (with the best window size), hybrid and CDBN, from top to bottom. 2-point correlation functions are calculated for 30 random generations from each method, in addition to the original samples. [1] Mean and standard deviations (as the shades) are reported.

Major findings are as follows: (1) The proposed method produces reconstructions that are visually similar to the samples from CDBN, while MRF has less favorable reconstructions particularly on Ti64 due to the existence of local patterns which breaks the Markovian assumption. MRF also produces more white areas in Pb-Sn in comparison with the hybrid and the CDBN methods. In addition, a closer investigation shows that MRF fails to constrain the filler particles in SC to be spherical. Nonetheless, MRF produces local patterns closer to the samples in SS. We believe that this is due to the drawback of the current feature learning implementation that is less sensitive to capturing sharp edges of local patterns. (2) None of the reconstructions for Pb-Sn, SS, and SC are statistically different from the original samples with respect to S_2. For Ti64 however, the findings are intriguing: While MRF produces closer S_2 to the samples, these functions do not capture the critical characteristics of this material system, as they monotonically decreases along the distance. To elaborate, S_2 for Ti64 should exhibit clear oscillations for small distance values, to reflect significant short range correlations due to the mutual exclusion volume between the grains. The wavelengths associated with the oscillations also indicate the average grain width. The characteristics are preserved in reconstructions from the hybrid and the CDBN methods, while their S_2s have higher amplitudes in the oscillation than those from the samples. This is due to the more uniform grain sizes and shape distributions in the reconstructions and the missing of local details exist in the samples.

[1] The number of random generations is limited by the high cost of sampling the large number (288) of channels for the hybrid method for Ti64.

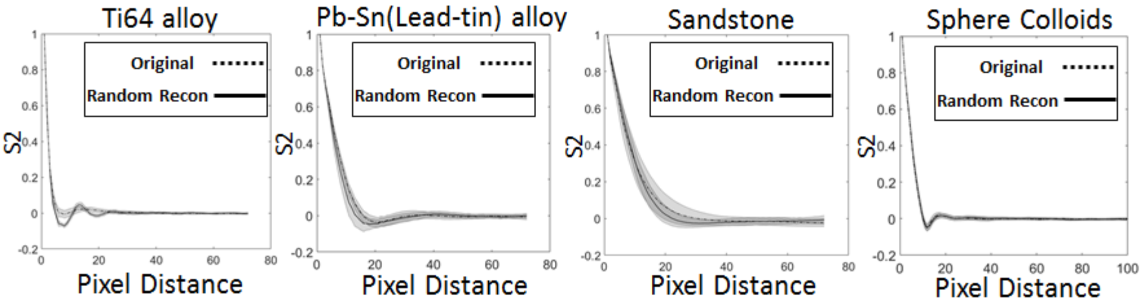


Figure 27: (Top) Comparison of the sample means and sample standard deviations of 2-point correlation functions of: the original samples, reconstructions from MRF with the best window sizes, the hybrid method, and CDBN (Bottom) The corresponding reconstructions (from top to bottom: original, MRF, hybrid, CDBN)

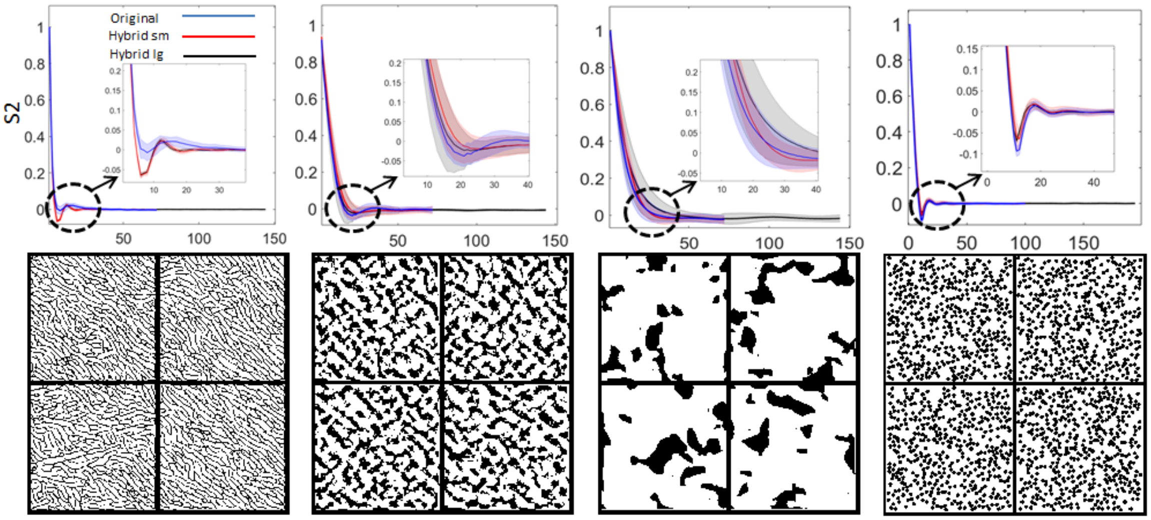
4.3.3 Generation at an arbitrary size

With the help of the MRF model on the CDBN output, we can produce reconstructions with an arbitrary size. Fig. 28 shows samples for larger reconstructions: 288×288 for Ti64, Pb-Sn and SS, 378×378 for SC. Their 2-point correlation functions are also calculated and compared with those of the original samples, and generations of the original sizes (144×144 for Ti64, Pb-Sn and SS, and 189×189 for SC). The result shows that characteristics of the 2-point correlation functions are preserved across generation sizes.

4.4 Discussion

Limitations and future directions are summarized as follows.

Reconstruction quality We note that the proposed model is yet to achieve high accuracy in reconstruction for complex material systems, especially when detailed features (e.g., the connection of grain boundaries in Ti64) are critical for material properties, even when post-processing techniques are manually applied. The application of post-processing also introduces additional human knowledge into the reconstruction procedure, making the method not fully automated for all material systems. The lack



Ti-6Al-4V alloy(288×288)

Pb-Sn(Lead-tin) alloy(288×288)

Pore structure of Fontainebleau sandstone(288×288)

2D suspension of spherical colloids(378×378)

Figure 28: Generations of size 288×288 for Ti64, Pb-Sn, and SS, 378×378 for SC, and their 2-point correlation functions.

of reconstruction accuracy is largely due to the blurring effect of the convolutions. We also note that neither VAE nor GAN in their original formulations((Kingma and Welling, 2013; Goodfellow *et al.*, 2014)) produces high quality random generations for complex material systems such as Ti64, as shown in Fig. 29, although the networks can reconstruct the training samples well. Note that these generations are too blurred to skeletonize. VAE suffers from its limitation of only allowing continuous features, i.e., a real-valued weighting on various discrete local patterns does not produce meaningful patterns. The GAN model may suffer from its intrinsic difficulty in balancing the convergence of the generator and the descriminator.

The Markovian assumption It is also important to note that MRF relies on the strong assumption that the image to be generated follows a stationary random field, and each pixel value is only conditioned on its limited neighbors. In this work, we used CDBN to remove local patterns, as an attempt to acquire the resulting activation

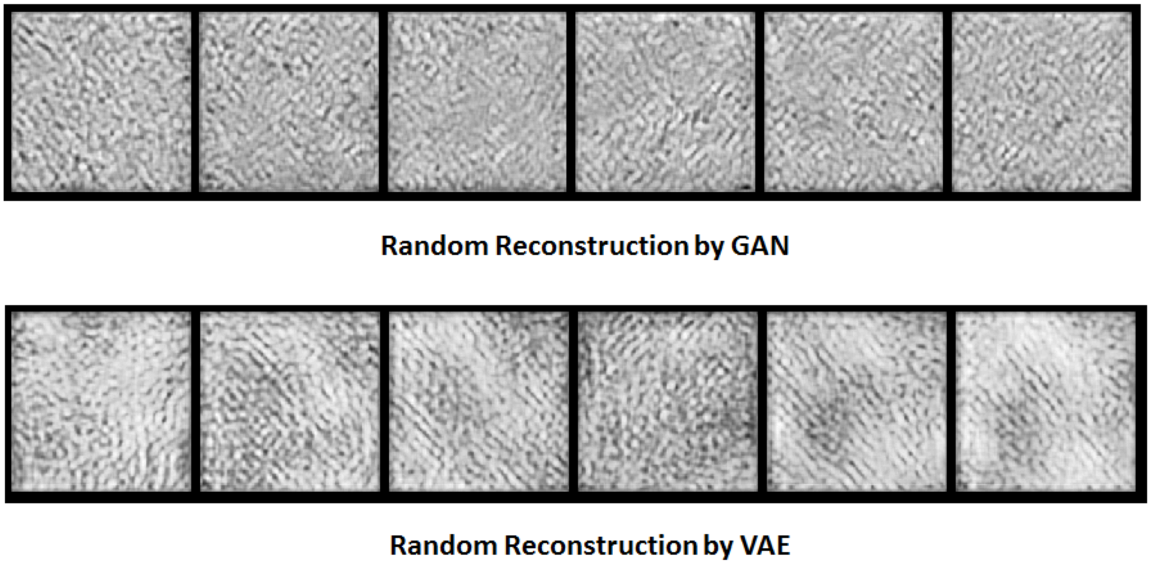


Figure 29: Random generations from VAE and GAN.

images these MRF properties. This attempt will fail if a material system, after abstracted by CDBN, cannot be considered as a MRF. CDBNs used in this work also requires careful tuning, based on the designer's understanding of the representative length scales of local patterns. Compared with conventional procedures for hand-picking features, our approach avoids asking for specifying details of human knowledge during the construction of the decoder, yet still requires heuristic inputs from the human designer.

Optimal material design in a statistical sense As we mentioned earlier, the application of MRF allows a distribution of microstructures to be represented by the same feature vector, i.e., the MRF parameters. This model is useful when the variances in the properties of interest can be effectively explained by these parameters. This is, however, not always the case. For example, as is shown in ((Cang *et al.*, 2016)), the fracture strength of Ti64 samples from the same processing setting is sensitive to their microstructure details. Thus it is not meaningful to represent a distribution of designs with the same feature when the property is to be optimized with respect to a particular microstructure. Nonetheless, like other material properties, the fracture strength of Ti64 exhibits correlation with laser sintering parameters, as can be seen in Fig. 30. Therefore, using MRF parameters (with the help of CDBN) to represent microstructures is meaningful when the goal is to identify a distribution of microstructures from the same processing setting that has the optimal mean property. Our future work will include microstructure samples from multiple processing settings in training the hybrid model. We will examine if the resulting MRF parameters are effective at explaining the properties and the processing parameters.

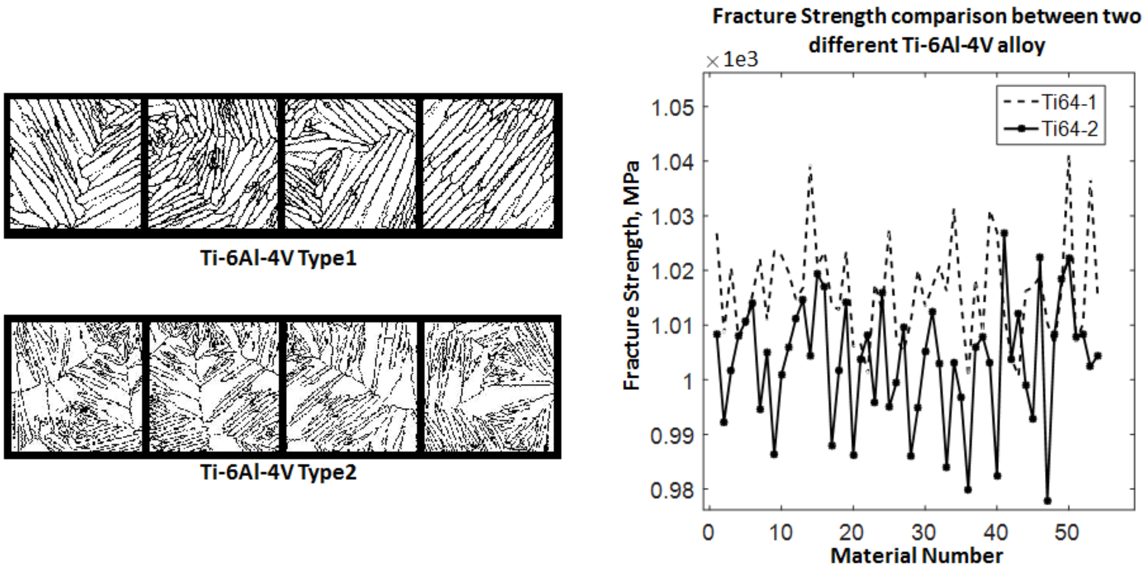


Figure 30: (Right) Two types for Ti64 samples from different processing parameters (Left) Their corresponding fracture strengths

4.5 Conclusions

In this work we developed a method to enable random generation of 2D material microstructure images of an arbitrary size that are visually and statistically similar to the authentic ones, based on a given set of samples. This is achieved by learning multiscale patterns through a convolutional deep belief network, and modeling the distributions of these patterns through a Markov Random Field. Technical challenges in integration these two models are discussed in detail. Model parameters are tuned by matching 2-point correlation functions to those of the training samples. We show that the proposed hybrid method performs particularly well on Ti-6Al-4V alloy both visually and statistically, in comparison with an existing MRF model.

Chapter 5

IMPROVING DIRECT PHYSICAL PROPERTIES PREDICTION OF HETEROGENEOUS MATERIALS FROM IMAGING DATA VIA CONVOLUTIONAL NEURAL NETWORK AND A MORPHOLOGY-AWARE GENERATIVE MODEL

5.1 Introduction

Direct prediction of material properties through predictive models has attracted interests from both material and data science communities. Predictive models have the potential to mimic highly nonlinear physics-based mappings, thus reducing dependencies on numerical simulations or experiments during material design, and enabling tractable discovery of novel yet complex material systems (Jain *et al.*, 2013; Hautier *et al.*, 2010; Kondo *et al.*, 2017). Nonetheless, the construction of predictive models for nonlinear functions, such as material structure-property mappings, is known to be data-demanding, especially when the inputs, e.g., material microstructures represented as 2D or 3D images, are high-dimensional (Huan *et al.*, 2016). Thus, the added value of predictive models quickly diminishes as the acquisition cost increases for material samples. We investigate in this paper a computational approach to generate artificial material samples with negligible cost, by exploiting the fact that all samples within one material system share similar morphology. More concretely, we define morphology as a style vector quantified from a microstructure sample, and propose a generative model that learns from a small set of authentic samples, and creates an arbitrary amount of artificial samples that share the same distribution of morphologies as the authentic ones.

The key contribution of the paper is the introduction of a morphology constraint on the generative model that significantly improves the morphological consistency between the artificial and authentic samples from benchmark generative models. To demonstrate the utility of the proposed model, we run a case study on the prediction of the Young's modulus, the diffusion coefficient and the permeability coefficient of sandstone microstructures. We show that the generated artificial samples from the proposed model can improve the prediction performance more effectively than those from a state-of-the-art Markov Random Field (MRF) model.

As an overview, the proposed model follows the architecture of a variational autoencoder (Kingma and Welling, 2013) that learns to encode material microstructures into a lower-dimensional latent space and to decode samples from the latent space back into microstructures. Both the encoder and the decoder are composed of feed-forward convolutional neural networks for extracting and generating local morphological patterns, and are jointly trained to minimize the discrepancy between the artificial and authentic samples. The target morphology is quantified from the authentic samples by an auxiliary network. The idea of quantifying material morphology through a deep network is inspired by the style transfer technique originally developed for image synthesis (Gatys *et al.*, 2016).

The rest of the paper is structured as follows: In Sec. 5.2 we review related work on material representations and reconstruction, based on which we delineate the novelty of this paper. We then introduce background knowledge on variational autoencoder and style transfer. Sec. 5.3 elaborates on the details of the proposed model. Sec. 5.11.4 presents a case study on the prediction of sandstone properties, where we demonstrate the superior performance of the proposed model against the benchmarks, in both microstructure generation, and the resultant property prediction accuracy. In Sec. 5.12 we summarize findings from the case study and propose potential future directions.

Sec. 5.6 concludes the paper.

5.2 Background

5.2.1 Data science challenges in computational materials science

Incorporating data science into material discovery (Jain *et al.*, 2013) and design (Curtarolo *et al.*, 2013) faces unique challenges with high dimensionality of material representations and the lack of material data due to high acquisition costs. We review existing work that address these challenges to some extent.

Challenge 1: Mechanisms for understanding material representations A common approach to addressing the issue of high dimensionality is to seek for a representation, i.e., an encoder-decoder pair, for a material system: The encoder transforms microstructures to their reduced representations, and the decoder generates (i.e., reconstructs) them back from their representations. A good encoder-decoder pair should both achieve significant dimension reduction, and good matching between the data distribution (i.e., the distribution of authentic samples) and the model distribution (i.e., the distribution defined by the decoder). This is often feasible for material systems with consistent and quantifiable morphologies among their samples, as reviewed below.

Existing encoders for material representations can be categorized as physical and statistical, some of which have led to accelerated design of various material systems (Torquato, 2013; Jiang *et al.*, 2013; Grigoriu, 2003; Xu *et al.*, 2014; Yu *et al.*, 2017). Among all, **physical encoders** characterize microstructures using composition (e.g., the percentage of each material constituent) (Broderick *et al.*, 2008a; Ashby, 2005), dispersion (e.g., inclusions' spatial relation, pair correlation,the ranked neighbor distance (Steinzig and Harlow, 1999; Tewari and Gokhale, 2004; Rollett *et al.*, 2007; Borbely *et al.*, 2004; Pytz, 2004; Scalon *et al.*, 2003)), and geometry features (e.g., the radius/size distribution, roundness, eccentricity, and aspect ratio of elements of the microstructure (Rollett *et al.*, 2007; Steinzig and Harlow, 1999; Torquato, 2013; Sundararaghavan and Zabaras, 2005; Basanta *et al.*, 2005; Holotescu and Stoian, 2011; Klaysom *et al.*, 2011; Gruber *et al.*, 2010)). Among **statistical encoders** are the N-point correlation functions (Liu *et al.*, 2013; Borbely *et al.*, 2004; Torquato, 2013; Sundararaghavan and Zabaras, 2005; Basanta *et al.*, 2005). Torquato et al. (Rintoul and Torquato, 1997; Yeong and Torquato, 1998; Torquato, 2013) show that the microstructure of heterogeneous materials can be characterized statistically via various types of N-point correlation functions (Okabe and Blunt, 2005; Hajizadeh *et al.*, 2011). Similar descriptors include lineal path function (Øren and Bakke, 2003) and statistics calculated based on the frequency domain using fast Fourier transformation (Fullwood *et al.*, 2008a,b). Another type of statistical encoders are **random fields** (Roberts, 1997; Bostanabad *et al.*, 2016; Liu and Shapiro, 2015), which define joint probability functions on the space of microstructures. Typical probability models include Gaussian random fields (Quiblier, 1984; Ostoja-Starzewski, 1998; Torquato, 2013) which treats binary microstructure images as level sets, and Markov random fields, where each pixel of the microstructure is assumed to be drawn from a probability function conditioned on its neighbouring pixels (Bostanabad *et al.*, 2016).

Decoding of representations, i.e., generation of microstructure through existing physical and statistical representations, involves optimization in the microstructure space: For physical representations and N-point correlation functions, a microstructure is searched to minimize its difference from the target descriptors. For random fields, the generation can be done by maximizing the joint probability through Markov chain Monte Carlo simulations (Bostanabad *et al.*, 2016; Liu and Shapiro, 2015). While it is shown that material generation through these representations is feasible (Borbely *et al.*, 2004; Basanta *et al.*, 2005; Torquato, 2013; Xu *et al.*, 2014), the computational costs for the optimization through gradient (Fullwood *et al.*, 2008a; Xu *et al.*, 2014) and non-gradient (Yeong and Torquato, 1998; Jiao *et al.*, 2008, 2009; Karsanina *et al.*, 2015) methods are often high.

In addition to the difficulties in decoding, the existing encoders are not universally applicable, especially to material systems with complex morphology. More specifically, matching in the representation space does not guarantee the match in the microstructure space. An example can be found in Fig. 31, where we compare two-point correlation functions of Ti64 alloy samples and three sets of artificial images (see details from (Bostanabad *et al.*, 2016; Cang *et al.*, 2017b,a)). The visually more plausible set has worse match to the target with respect to the Euclidean distance in the discretized 2-point correlation space.

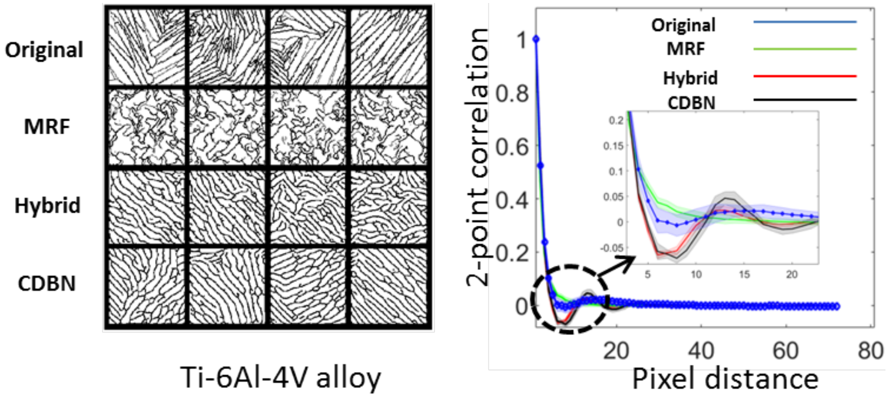


Figure 31: Comparison of two-point correlation functions among four sets of images. From left to right: Authentic microstructure samples, samples generated by a Markov random field model (Bostanabad *et al.*, 2016), samples generated by a hybrid model with deep belief network and Markov random field (Cang *et al.*, 2017a), and samples generated by a deep belief network (Cang *et al.*, 2017b). Better matching in the discretized 2-point correlation space does not indicate better microstructure generations.

These existing difficulties lead to the need for new mechanisms to define material representations. We propose four metrics for evaluating the utility of a material representation: interpretability, dimensionality, expressiveness, and generation cost:

Physical descriptors and correlation functions are designed to be interpretable and relatively low dimensional, yet may not be expressive enough to capture complex morphologies and requires optimization during generation; random fields are relatively expressive (and some permit fast generation (Bostanabad *et al.*, 2016)), but are often high-dimensional and less interpretable. Both categories of representations are material specific, i.e., new representations need to be manually identified for new material systems. Cang et al. (Cang *et al.*, 2017b) proposed to learn statistical generative models from microstructures to automatically derive expressive, low dimensional representations that enables fast microstructure generation. They showed that a particular type of generative model, called Convolutional Deep Belief Network (CDBN) (Lee *et al.*, 2009), can produce reasonable microstructures for material systems with complex morphologies, by extracting morphology patterns at different length scales from samples, and decode an arrangement of these patterns (the hidden activation of the network) back into a microstructure. Nonetheless, CDBNs are trained layer-by-layer, and thus require additional material-dependent parameter tuning to achieve plausible generations.

Different from (Cang *et al.*, 2017b), this paper proposes a model that directly enforces the matching of the artificial microstructures to the authentic ones, thus avoiding additional parameter tuning. Our model is also fully differentiable, making it much easier to train than CDBNs and scalable to deep architectures. Lastly, the introduction of the morphology constraint further refines the artificial microstructures under a small amount of training data. With all these improvements, the proposed model is generally applicable to the extraction of low dimensional material representations, and is expressive for generating microstructures with morphologies of decent complexity.

Challenge 2: Effective material data acquisition methods Unlike scenarios in contemporary machine learning where a large quantity of data is available, in computational material design tasks we may only have a limited amount of microstructure and property samples, and the acquisition cost for additional samples is usually high. This challenge calls for the incorporation of active learning methods into the design. The key idea of active learning is to minimize the data acquisition cost for learning a model (e.g., generative models for capturing microstructure distributions, or discriminative models for predicting process-structure-property mappings), by optimally controlling the balance between data exploitation and exploration (Tong and Chang, 2001; Settles, 2010). In computational material science, such techniques have been adopted to accelerate the process of material discovery for desired properties (Ling *et al.*, 2017; Lookman *et al.*, 2016). While this paper does not focus directly on active learning, we demonstrate that the quality of the microstructure data, along with its quantity, can significantly influence the prediction performance of a statistical structure-property mapping, and thus justify the value of the proposed generative model.

5.2.2 Preliminaries on predictive and generative models

This paper will involve three networks: The generative network for creating artificial material microstructures, the auxiliary convolutional network for quantifying material morphologies, and another convolutional residual network (ResNet) for structure-property prediction. We provide technical backgrounds for these models below.

Generation through Variational Autoencoder Variational autoencoders (VAE) (Kingma and Welling, 2013) are extensions of autoencoders (Bengio, 2009). An autoencoder

(Fig. 32a) is composed by two parts: an encoder $\mathbf{z} = f(\mathbf{x})$ converts the input $\mathbf{x}$ to a hidden vector $\mathbf{z}$, and a decoder produces a reconstruction $\hat{\mathbf{x}} = g(\mathbf{z})$. An AE is trained to minimize the discrepancy between inputs $\mathbf{x}$ and their corresponding reconstructions $\hat{\mathbf{x}}$. Variants of autoencoders (e.g., Sparse (Ng, 2011), denoising (Vincent *et al.*, 2008), and contractive (Rifai *et al.*, 2011)) have been proposed to improve the learning of more concise representations from high-dimensional input data, and are widely used for data compression (Le *et al.*, 2015), network pre-training (Bengio *et al.*, 2007), and feature extraction (Xing *et al.*, 2015). However, conventional autoencoders are prone to generating implausible new outputs, see Fig. 36b for examples, as they do not attempt to match the model distribution (the distribution of images generated from the decoder) with the data distribution (the distribution of input images). VAE (Fig. 32b) was introduced to address this issue (Kingma and Welling, 2013), by constraining the distribution of the latent variables $\mathbf{z}$ encoded from the input data

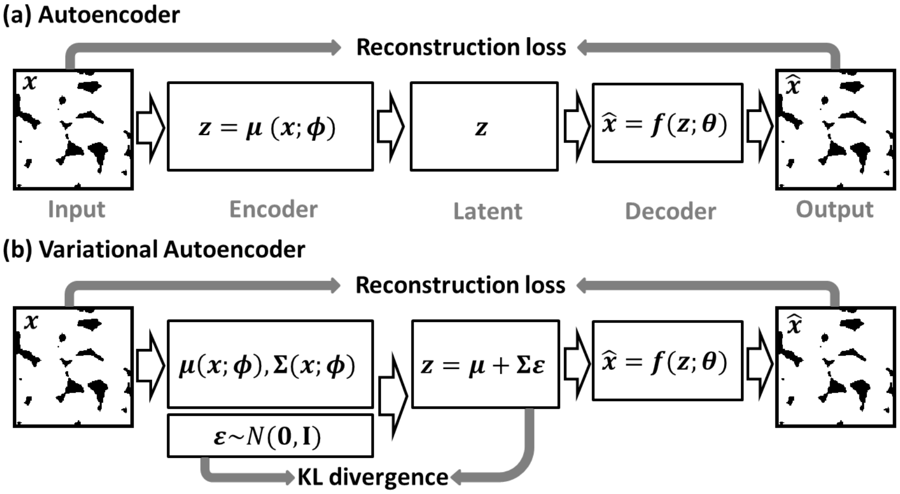


Figure 32: (a) An autoencoder converts inputs to a latent space and reconstructs outputs from samples in the latent space. It is trained to minimize the reconstruction error. (b) A variational autoencoder converts inputs to a distribution in the latent space. It is trained to minimize both the reconstruction error and the Kullback-Leibler (KL) divergence between the latent distribution and a sample distribution (often standard normal).

to that used for output generation, thus indirectly forcing the match between the distributions of the model outputs and the data.

The VAE model We start with the decoder, which defines the output distribution, $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$, conditioned on the latent vector $\mathbf{z}$, and is parameterized by $\boldsymbol{\theta}$. Let the input data be $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N}$, which defines a data distribution $p(\mathbf{x})$, and let $p(\mathbf{z})$ be the pre-specified sampling distribution in the latent space. The goal is to train a model that matches the marginal distribution $p_{\boldsymbol{\theta}}(\mathbf{x})$ to the data distribution $p(\mathbf{x})$. In the following, we show that this is indirectly achieved by matching the posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ to $p(\mathbf{z})$.

First, the marginal is defined as

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \int_{\mathbf{z}} p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} = E_{\mathbf{z}\sim p(\mathbf{z})}p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}). \tag{5.1}$$

Matching the model and data distributions requires maximizing the likelihood $\Pi_{\mathbf{x}\in\mathcal{X}}p_{\boldsymbol{\theta}}(\mathbf{x})$ with respect to $\boldsymbol{\theta}$. This maximization problem, however, is often intractable due to the numerical integral and the complexity of the decoder network. Nonetheless, it is noted that for a given $\mathbf{x}$, most samples in the latent space will have $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) \approx 0$, as these latent samples do not generate outputs close to $\mathbf{x}$. This leads to the idea of introducing another distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ (the encoder) that takes an $\mathbf{x}$ and outputs $\mathbf{z}$ values that are likely to reproduce $\mathbf{x}$. Ideally, the space of $\mathbf{z}$ that are likely under q will be much smaller than that under $p(\mathbf{z})$, making the computation of $E_{\mathbf{z}\sim q}p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ relatively cheap. However, $E_{\mathbf{z}\sim q}p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ is not the same as $p(\mathbf{x})$, as $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ does not necessarily match to $p(\mathbf{z})$. The relation between the two is derived below.

We start with the Kullback-Leibler divergence between the encoded latent distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ and the posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$:

$$\begin{aligned} D_{KL}\left(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})\right) &= \int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \log \frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})} d\mathbf{z} \\ &= -\mathbb{E}_{\mathbf{z}\sim q}[\log p(\mathbf{z},\mathbf{x})] - \mathbb{E}_{\mathbf{z}\sim q}[-\log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})] + \log p(\mathbf{x}). \end{aligned} \tag{5.2}$$

By rearranging Eq. (5.2), we arrive at

$$\log p(\mathbf{x}) - D_{\mathrm{KL}}\left(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})\right) = \mathbb{E}_{\mathbf{z}\sim q}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] - D_{\mathrm{KL}}\left(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\right). \tag{5.3}$$

Note that the right hand side of Eq. (5.3) can be maximized through stochastic gradient descent: The first term $\mathbb{E}_{\mathbf{z}\sim q}[\log p_{\boldsymbol{\theta}(\mathbf{x}|\mathbf{z})}]$ measures the reconstruction error, i.e., the expected difference between an input $\mathbf{x}$ and its reconstructions drawn from the decoder conditioned on the latent variables, which are further drawn from the encoder conditioned on the input. The second term $D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$ measures the difference between the encoded distribution of $\mathbf{z}$ and the modeled latent distribution $p(\mathbf{z})$. Since the encoding and decoding processes involved in these terms are modeled as feedforward networks, the gradients are readily available for backpropagation. Specifically, we model $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}(\mathbf{x};\boldsymbol{\phi}), \boldsymbol{\Sigma}^2(\mathbf{x};\boldsymbol{\phi}))$ as a normal distribution parameterized by the mean $\boldsymbol{\mu}(\mathbf{x};\boldsymbol{\phi})$ and the diagonal variance matrix $\boldsymbol{\Sigma}^2(\mathbf{x};\boldsymbol{\phi})$. $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the outputs of the encoder network. We similarly model $p_{\boldsymbol{\theta}(\mathbf{x}|\mathbf{z})} = \mathcal{N}(\mathbf{x}|\mathbf{f}(\mathbf{z};\boldsymbol{\theta}), \sigma^2\mathbf{I})$ with mean $\mathbf{f}(\mathbf{z};\boldsymbol{\theta})$ and variance σ^2. The function $\mathbf{f}(\mathbf{z};\boldsymbol{\theta})$ is the decoder network; σ determines the importance of the reconstruction of $\mathbf{x}$ during the training of a generative model, and is set to 1 in the proposed model. The inputs to the decoder are drawn from a normal distribution defined by $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. This connection between the encoder and the decoder can be modeled as feedforward through a reparameterization trick (Kingma and Welling, 2013), which enables gradient calculation of the entire network.

The prior of the latent distribution, $p(\mathbf{z})$, is often assumed to be standard normal. This is because the transformation from this simple distribution to the potentially highly nonlinear data distribution $p(\mathbf{x})$ can often be achieved by a sufficiently deep decoder.

The left hand side of Eq. (5.3) contains the objective $\log p_{\boldsymbol{\theta}}(\mathbf{x})$ that we want to maximize, and the KL-divergence $D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})) > 0$ that should ideally reach 0. Thus, minimizing

$$L(\theta, \phi, \mathbf{x}) = -\mathbb{E}_{\mathbf{z}\sim q}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] + D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \tag{5.4}$$

will maximize a lower bound of $p(\mathbf{x})$.

Prediction through Residual Network (ResNet) We build a predictive structure-property model using the frame work of a ResNet (He *et al.*, 2016). A conventional feed-forward neural network models the function $f(x) = y$ directly, while a ResNet models the residual $f(x) = y - x$ between inputs x and outputs y. The operation of $y = f(x) + x$ is performed by a shortcut connection and element-wise addition as shown in Fig. 33c. ResNet has shown to resolve the degradation problem effectively, i.e., the problem that deeper networks often introduce challenges in convergence and lead to worse prediction performance (He *et al.*, 2016). In this study, we applied an 18-layer ResNet to learn the structure-property mappings of interest.

Style transfer (ST) ST was first introduced to integrate the content of one image with the style of another (Gatys *et al.*, 2016). To perform ST, a "style vector" is first computed for an image, that consists of the variance-covariance matrices of the hidden states of a pre-trained CNN (we use a VGG Net (Simonyan and Zisserman, 2014a) in this paper) activated by the input image. A new image can then be created by minimizing the difference between its style vector and the target one. One can also preserve the image content, i.e., the hidden states of the deepest layers, by adding an additional loss on the content difference.

As will be discussed in Sec. 5.3, the proposed method extends ST to a generative setting, by directly adding a style (morphology) penalty to the training loss of a VAE. There are two major difference in comparison: First, the standard ST generates images by minimizing the style loss with respect to a high-dimensional image space, while our model is learned to directly generate style-consistent images, thus is significantly faster than standard ST; second, the standard ST uses a single style vector as a target, while our model can encode a distribution of styles in its latent space, and thus is more expressive as a generator.

5.3 Proposed Models

5.3.1 Network specifications

The generative model The proposed generative model is composed of two networks, a VAE for image encoding and decoding, and an auxiliary network for computing the style vector (**G**). See model summary in Fig. 33a. For the VAE, its encoder has convolutional layers of a fixed filter size (4×4), max-pooling layers after each other convolutional layers, and two fully-connected layers of sizes 256 and 16, respectively. Each pooling layer is of stride 2, i.e., it is composed with filters of size 2×2 applied to the hidden layers, down-sampling these layers by a factor of 2 along both width and height. Together, the encoder reduces the dimension of the input from 128×128 to 16×1. The architecture of the decoder is almost symmetric to that of the encoder, except that the representation (after sampling) is expanded to 32 dimensions through a fully connected layer.

The predictive model An 18-layer ResNet is built to predict material properties from microstructures. A schematic of this predictive model is shown in Fig. 33b. The network is composed by one convolutional block, two max pooling layers, four

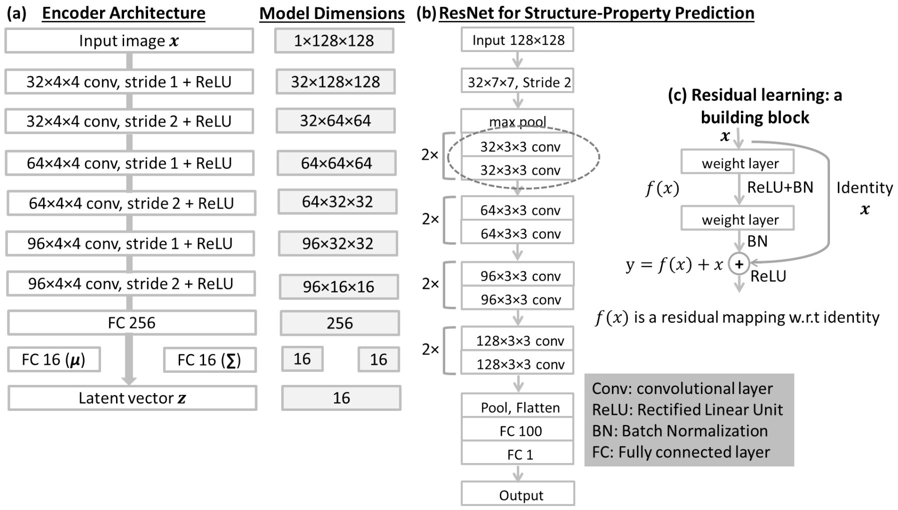


Figure 33: (a) The VAE encoder architecture. (b) The 18-layer ResNet architecture. The circled part represent a residual block and is explained in (c).

residual block and two fully connected layers. The number and the size of the weights for every individual layer are listed in the figure. Max pooling and convolutional layers are used to progressively reduce the spatial dimensions of the hidden layers.

The morphology model To acquire a rich set of morphology patterns and make this acquisition process general, we propose to use a VGG network (Simonyan and Zisserman, 2014b) to quantify morphology. The VGG net is pre-trained on over a million images (Deng *et al.*, 2009) to learn a rich set of basic feature representations, and is then integrated into the generative model. The schematic for the resultant morphology-aware VAE is shown as Fig. 34.

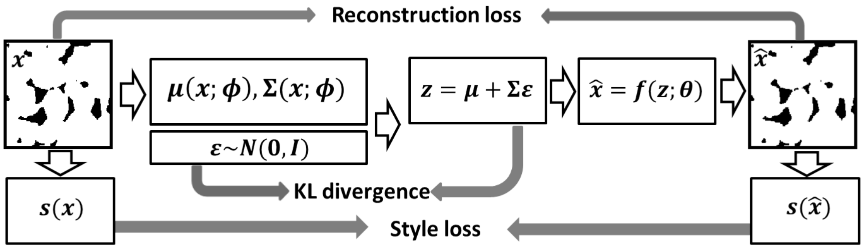


Figure 34: Schematic of the proposed morphology-aware VAE

5.3.2 Model training

The loss function of the generative model Training the generative model involves minimizing a loss function with four components:

$$L(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathcal{X}) = L_{\text{RECON}} + L_{\text{KL}} + L_{\text{ST}} + L_{\text{MC}}, \tag{5.5}$$

where $L_{\text{RECON}} + L_{\text{KL}}$ are the standard reconstruction and KL divergence losses for VAE (see Sec. 5.2.2), L_{ST} is the style loss, and L_{MC} is an additional loss to prevent mode collapse (introduced below).

The reconstruction loss

$$L_{\text{RECON}} = -\sum_{\mathbf{x}_i \in \mathcal{X}} \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)}[\log p_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z})] \tag{5.6}$$

measures the reconstruction error of all pairs of authentic microstructures ($\mathbf{x}_i \in \mathcal{X}$) and their reconstructions. The KL divergence loss

$$L_{\text{KL}} = \sum_{\mathbf{x}_i \in \mathcal{X}} D_{KL}(q_{\phi}(\mathbf{z}|\mathbf{x}_i)||p(\mathbf{z})) \tag{5.7}$$

measures the difference between the encoder distribution $q_{\phi}(\mathbf{z}|\mathbf{x}_i)$ and the prior $p(\mathbf{z})$. The style loss

$$L_{\text{ST}} = \sum_{s=1}^{N} \sum_{p=1}^{N} \sum_{l} ||\mathbf{G}^{s,l} - \mathbf{G}^{p,l}||_F^2 \tag{5.8}$$

measures the total style difference between two sets of N images with indices s and p, representing the generated and the authentic ones, respectively. Given a morphology style model, the Gram matrix $\mathbf{G}^{s,l}$ of image s at the lth hidden layer is defined through elements $G_{i,j}^{s,l} = \sum_k f_{i,k}^{s,l} f_{j,k}^{s,l}$, where feature element $f_{i,k}^{s,l}$ represents the hidden activation corresponding to the ith convolutional filter at spatial location k. $|| \cdot ||_F$ is the Frobenius normal. We note that the definition of feature maps will affect the outcome of style transfer. In general, deeper layers capture styles at larger length scales. We use the first four layers of a standard VGG net. This definition of features is empirical.

Lastly, the mode collapse loss is introduced to prevent the model from producing clustered samples (Zhao *et al.*, 2016), by forcing the generated samples to be different in terms of their activations in the style network. The loss is defined as:

$$L_{\mathrm{MC}} = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} (\frac{\mathbf{s}_i^T \mathbf{s}_j}{\|\mathbf{s}_i\| \|\mathbf{s}_j\|})^2, \tag{5.9}$$

where the style vector $\mathbf{s}$ contains the concatenated and vectorized Gram matrices as defined above.

The training of the generative model The generative model is trained to minimize the loss defined in Eq. (5.5) with respect to model parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ through stochastic gradient descent. To approximate the gradient of the loss, $\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}}(L_{\mathrm{RECON}} + L_{\mathrm{KL}} + L_{\mathrm{ST}} + L_{\mathrm{MC}})$, we randomly sample 20 authentic material microstructures from $\mathcal{X}$ and generate another 20 artificial microstructures in each training iteration. The artificial microstructures are needed to calculate $\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}}(L_{\mathrm{ST}} + L_{\mathrm{MC}})$. To prevent neighbouring latent samples to be matched to drastically different style (morphology) targets, we propose to create artificial microstructures by the following procedure: We take each authentic sample $\mathbf{x}$ and pass it through the encoder to get $\boldsymbol{\mu}(\mathbf{x};\boldsymbol{\phi})$ and $\boldsymbol{\Sigma}(\mathbf{x};\boldsymbol{\phi})$. We then draw a random sample $\mathbf{z}$ from the latent space following the normal distribution $\mathcal{N}(\mathbf{z}|\boldsymbol{\mu}(\mathbf{x};\boldsymbol{\phi}), \boldsymbol{\Sigma}(\mathbf{x};\boldsymbol{\phi}))$, and derive its decoded microstructure $\mathbf{x}' = f(\mathbf{z};\boldsymbol{\theta})$ to pair with $\mathbf{x}$. By matching the style of $\mathbf{x}$ and $\mathbf{x}'$, we ensure that samples close to each other in the latent space will have similar morphologies. We demonstrate the necessity of this treatment in Fig. 35, where we show the generation results from an alternative method where microstructures from a standard normal distribution are randomly paired with the authentic ones for style loss calculation, leading to averaged morphology across all random generations (Fig. 35b).

We use Adam optimizer (Kingma and Ba, 2014) for training, with 200K iterations, a learning rate of 0.001, and a batch size of 20. The Adam parameters β_1 and β_2 are set as 0.9 and 0.999, respectively.

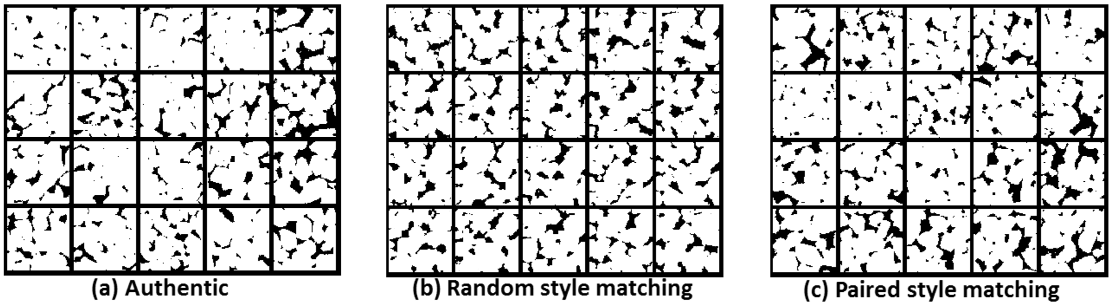


Figure 35: (a) Samples of authentic microstructures (b) Artificial microstructures when random samples are drawn from a standard normal distribution during the training (c) Artificial microstructures when random samples are drawn to pair with the authentic ones

The training of the ResNet We divide the raw data into training, validation, and test sets. The loss function for training the predictive structure-property model is the mean square error of the predicted properties of the training data. Prediction performance of the model on the validation set is monitored to prevent the model from overfitting. The performance of the model is then measured by R-square value of the test data. We use Adam for training with a learning rate of 0.001, and a batch size of 20. The Adam parameters are the same as in training the generative model. Batch normalization (Ioffe and Szegedy, 2015) is used.

5.4 Case Study and Results

In this section, we demonstrate key contributions of the proposed model through a case study: (1) By incorporating the style loss, the proposed generative model creates material microstructures with better visual and statistical similarity to the authentic ones than a state-of-the-art Markov random field (MRF) approach; and (2) we can improve the prediction of material properties more effectively by using the artificial data generated from the proposed model than by those generated from a MRF. The case study uses a set of sandstone microstructures and their properties, including Young's modulus, diffusivity, and fluid permeability. Details of this material system are introduced as follows.

5.4.1 A case study on the sandstone system

Sandstones are a class of important geological porous materials in petroleum engineering. In the simplest form, a mono-mineral sandstone is composed of a solid (rock) phase and a pore (void) phase, both of which are typically percolating. Extensive research work has been carried out to model the microstructure and physical properties of such porous materials (Sahimi, 2011; Radlinski *et al.*, 2004; Milliken and Laubach, 2000; Blair *et al.*, 1996; Coker *et al.*, 1996; Antonellini *et al.*, 1994; Al-Raoush and Willson, 2005; Appoloni *et al.*, 2002; Li *et al.*, 2017; Hajizadeh *et al.*, 2011). For example, a fast and independent architecture of artificial neural network has been developed for accurately predicting fluid permeability (Tahmasebi and Hezarkhani, 2012).

We will focus on three different material properties: Young's modulus (stiffness) E, diffusivity D and fluid permeability k. These properties are respectively sensitive to distinct microstructural features of the materials. In particular, E is mainly determined by the morphology and volume fraction of the rock phase. The diffusivity D is most sensitive to the local pore size distribution. The fluid permeability k depends on the degree of connectivity of the pore phase as well as the pore/rock interface morphology. Thus, successfully improving the prediction accuracy for all three material properties poses a stringent test for our morphology-aware generative model for effectively populating the microstructure sample space.

Physics-based material property calculations The material properties used to train the network model are first computed using the effective medium theory (Torquato, 2013). The Young's modulus can be written as

$$E = \frac{9K_e G_e}{3K_e + G_e}, \tag{5.10}$$

where K_e and G_e are respectively the bulk and shear modulus of the materials. We use analytical approximations of K_e and G_e obtained by truncating the associated strong-contrast expansions after the third order term (Torquato, 1997), which are respectively given by

$$\phi_2 \frac{\kappa_{21}}{\kappa_{e1}} = 1 - \frac{(d+2)(d-1)G_1\kappa_{21}\mu_{21}}{d(K_1+2G_1)}\phi_1\chi \tag{5.11}$$

and

$$\phi_2 \frac{\mu_{21}}{\mu_{e1}} = 1 - \frac{2G_1\kappa_{21}\mu_{21}}{d(K_1+G_1)}\phi_1\chi - \frac{2G_1\kappa_{21}\mu_{21}}{d(K_1+2G_1)}\phi_1\chi - \frac{1}{2d}[\frac{dK_1+(d-2)G_1}{K_1+2G_1}]^2\mu_{21}^2\phi_1\eta_2, \tag{5.12}$$

where ϕ_1 and ϕ_2 are respectively the volume fraction of the rock phase and void phase (i.e., porosity); d is the spatial dimension of the material system; K_p and G_p are respectively the bulk and shear modulus of phase p; and the scalar parameters κ_{pq} and μ_{pq} $(p, q = 1, 2, e)$ are respectively the bulk and shear modulus polarizability, i.e,

$$\kappa_{pq} = \frac{K_p - K_q}{K_q + \frac{2(d-1)}{d}G_q} \tag{5.13}$$

and

$$\mu_{pq} = \frac{G_p - G_q}{G_q}\frac{1}{1 + \frac{\frac{d}{2}K_q + \frac{(d+1)(d-2)}{d}G_1}{K_q+2G_q}}. \tag{5.14}$$

The quantities χ and η_2 are the microstructural parameters associated with the void (pore) phase involving the three-point correlation functions S_3 and two-point correlation functions S_2 of the pore phase, i.e.,

$$\chi = \frac{9}{2\phi_1\phi_2}\int_0^\infty \frac{dr}{r}\int_0^\infty \frac{ds}{s}\int_{-1}^1 d(\cos\theta)P_2(\cos\theta)[S_3(r,s,t) - \frac{S_2(r)S_2(t)}{\phi_2}] \tag{5.15}$$

and

$$\eta_2 = \frac{5}{21}\chi + \frac{150}{7\phi_1\phi_2}\int_0^\infty \frac{dr}{r}\int_0^\infty \frac{ds}{s}\int_{-1}^1 d(\cos\theta)P_4(\cos\theta)[S_3(r,s,t) - \frac{S_2(r)S_2(t)}{\phi_2}], \tag{5.16}$$

where $t = (r^2 + s^2 - 2rs\cos\theta)^{1/2}$, and P_2 and P_4 are respectively the Legendre polynomials of order two and four, i.e.,

$$P_2(x) = \frac{1}{2}(3x^2 - 1), \quad P_4(x) = \frac{1}{8}(35x^4 - 30x^2 + 3). \tag{5.17}$$

In our system, the void phase possesses zero elastic moduli, i.e., $K_2 = G_2 = 0$, and the moduli of the rock phase are respectively $K_1 = 4.3$ GPa and $G_2 = 3.8$ GPa. The three-point parameters χ and η_2 are computed by first evaluating the correlation functions S_3 and S_2 of the void phase from the microstructure data and then evaluating the integrals. In the reported results, the overall Young's modulus of the sandstone is re-scaled with respect to that of the rock phase.

Similar to the elastic moduli, we use an analytical approximation (Torquato, 1985; Jiao and Torquato, 2012) obtained by truncating the strong-contrast expansion of the effective diffusion coefficient D after the third order term:

$$\frac{D}{D_2} = \frac{1 + 2\phi_1\beta_{12} - 2\phi_2\chi\beta_{12}^2}{1 - \phi_1\beta_{12} - 2\phi_2\chi\beta_{12}^2}, \tag{5.18}$$

where ϕ_1 and ϕ_2 are respectively the volume fraction of the rock and pore phases. The polarization parameter β_{12} is given by

$$\beta_{12} = \frac{D_2 - D_1}{D_2 + 2D_1}, \tag{5.19}$$

where D_1 and D_2 are respectively the diffusivity of the rock and pore phases. In our sandstone system, the rock diffusivity is typically negligibly small compared with the pore phase diffusivity, and thus $D_1/D_2 \approx 0$. The three-point parameter χ for the rock phase is given by Eq. (5.15), in which the correlation function S_3 and S_2

are computed from the microstructural data for the rock phase (instead of the pore phase as in the case of elastic moduli). The computed D values are normalized with respect to D_2 in the reported data.

Finally, the fluid permeability is computed using the approximation (Torquato and Lu, 1990)

$$k = \frac{2}{3(1-\phi_2)^2} \int_0^\infty [S_2(r) - \phi_2^2] r dr, \tag{5.20}$$

where ϕ_2 is the volume fraction of the pore phase and S_2 is the associated two-point correlation function.

5.4.2 Results

The effect of style loss on material generation In Fig. 36, we show that a standard VAE has undesirable generation performance with a small training set (200 samples), and that the incorporation of the style loss can effectively improve the generation quality under the same sample size. Further, the proposed approach also produces more plausible microstructures than the MRF method (Bostanabad *et al.*, 2016), as one can observe the scattering of unrealistically small particles of the "stone" phase from the latter. We also tested the scalability of the proposed method by increasing the input image size. Fig. 36g shows that our method can robustly learn high-quality generations when the image size is increased from 128-by-128 to 256-by-256, while keeping the same sample size.

Structure-property predictions Here we investigate how the prediction accuracy of a structure-property model improves with the additional training data, acquired through the proposed method and the MRF. To do so, we start by deriving baseline models (for all three structure-property mappings) using 100 data points for training and another 100 for validation. The baseline prediction performance is evaluated

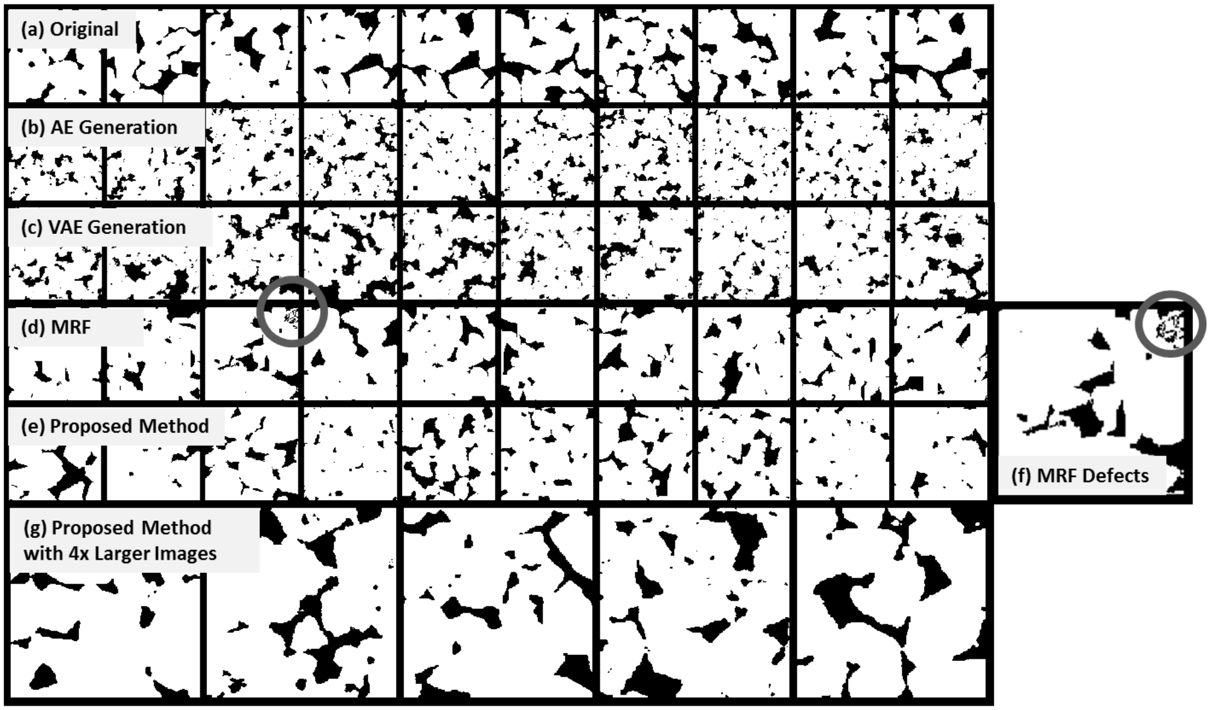


Figure 36: (a) The authentic sandstone microstructure samples (b) Generations from a standard autoencoder (c) Generations from a standard VAE (d) Generations from the MRF method in (Bostanabad *et al.*, 2016) (e) Generations from the proposed method (f) An enlarged example of MRF generation with artificial defects (g) Generations from the proposed method based on authentic samples four times larger than in (e)

using a third set of 100 test data points. The true property values are calculated following Sec. 5.4.1, and normalized for training.

Two sets of artificial microstructures, 1000 samples each, are created using the proposed method and the MRF, respectively. The true properties of these samples are computed. We then use these additional samples to improve the performance of the predictive models. Specifically, an increasing number of additional data points (from 50 to 1000) from the two sets are randomly added to the original training set to update the predictive model. We perform 10 independent random draws for each size of the additional data, and report the means and variances of the resultant test R-squares. The exception is with data size 1000, where all data points are used all together.

Results are presented in Fig. 37 and summarized as follows. (1) Microstructures generated from the proposed method have property distributions similar to those of the authentic samples, while those generated from MRF have significantly smaller variances (Fig. 37a). While the MRF method ensures variance in the microstructure space, as is evident from Fig. 36, the lack of variance in the volume fraction of its generations may cause the observed small variances in the properties. It is worth noting that while a modified MRF approach that considers variance in volume fraction may improve its generation quality, our approach does not require human identification or design of such features, as variances in morphology are directly learned through the generative model. (2) The statistical difference in properties observed in (1) may explain the significant gap in the prediction performance of the resultant structure-property models, as seen in Fig. 37b.

The narrow distribution of properties of the MRF method also explains the drop in model performance from 800 to 1000 samples: By feeding the network data from one distribution while asking it to predict on those drawn from another, we may encounter deteriorated performance even with an increased training data size.

5.5 Discussion

Applications of the proposed method The proposed generative model captures low-dimensional representations of data points distributed in high-dimensional spaces, and uniquely refines the manifolds through morphology constraints. Such capability of learning hidden data structures through limited samples is largely desired, and we demonstrated in the paper its application to improving material structure-property prediction. The value of this application is particularly significant when the bottleneck of the material design task is in the high acquisition cost of microstructure samples.

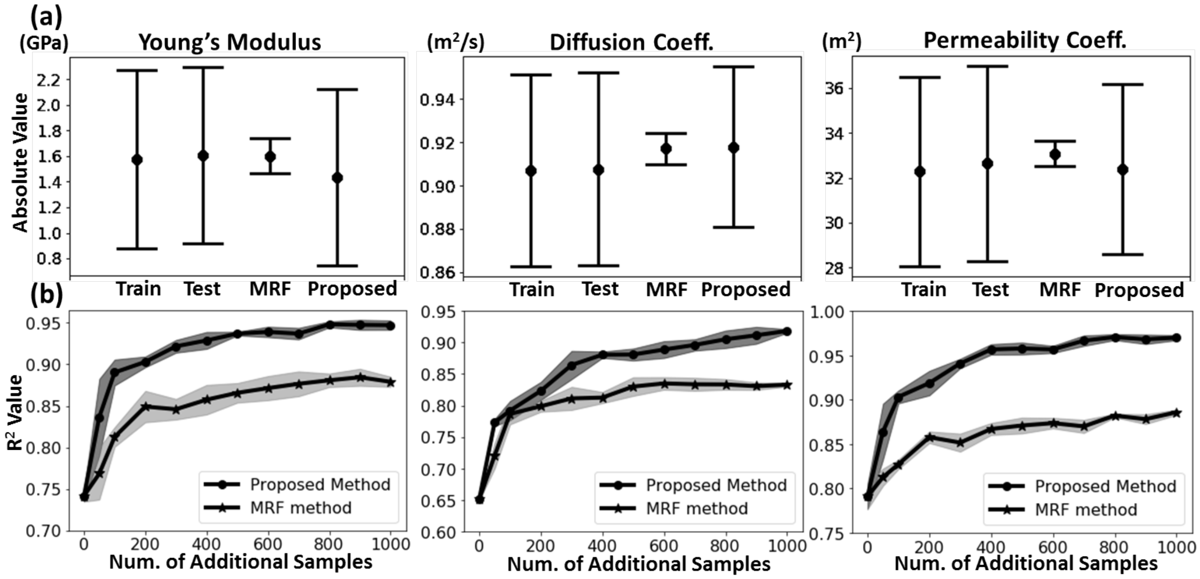


Figure 37: (a) Property distributions of microstructures generated from the proposed and the MRF (Bostanabad *et al.*, 2016) models. (b) Test R-square values at an increasing number of additional data points

Since the generative model essentially approximates the feasible set of microstructures (e.g., for a given set of processing parameters), it is also be used to effectively constrain the design space during microstructure optimization. To show this, we use a pseudo design case where the Young's modulus of a sandstone sample is to be optimized with respect to its microstructure, as is shown in Fig. 38: Gradient descent is applied to both the representation and the image space, with gradients computed from the predictive model for Young's modulus derived in Sec. 5.11.4 (with all artificial samples). This preliminary case shows that the generative model offers a computationally tractable way (due to its differentiable nature and its induced low-dimensional representation) to constrain the design space during microstructure optimization, leading to more morphologically reasonable solutions.

The connection between material representations and property prediction

In this study we created separate models for microstructure generation and structure-

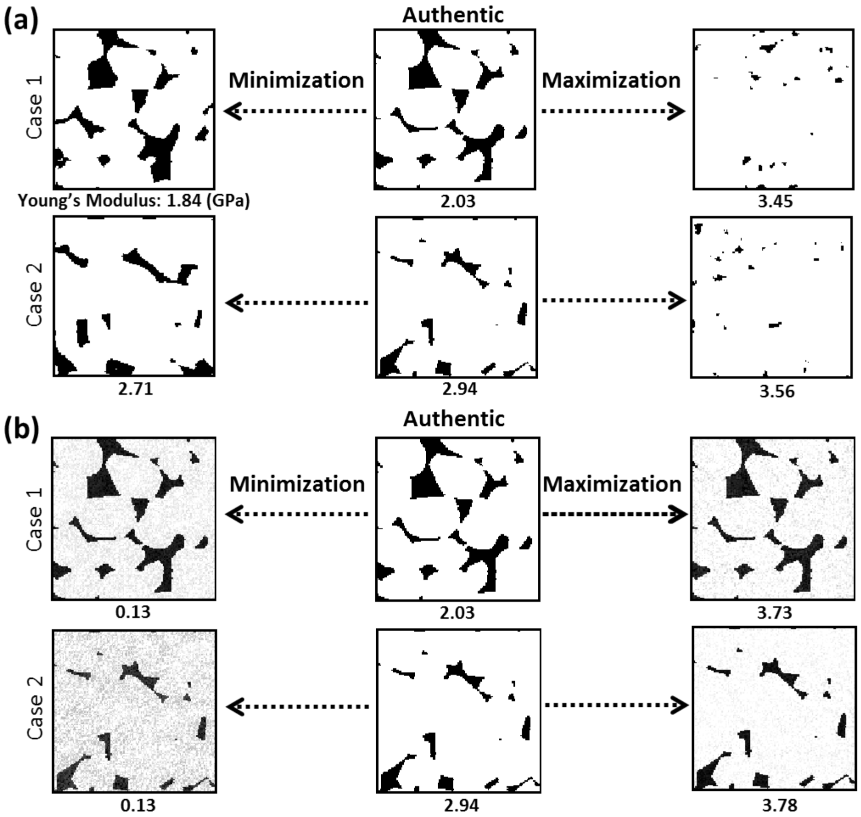


Figure 38: Applying gradient descent on the structure-property mapping derived in Sec. 5.4.2 (with all artificial samples included) to (a) the representation space learned through the proposed generative model and (b) the image space

property mapping. While it has been speculated that microstructure representation learned through a generative model may help improve the prediction performance of a structure-property mapping by treating the representation as inputs to the predictive model, we decided not to test this hypothesis in this paper. The rationale is as follows: From a theoretical perspective, the training of a representation that encodes microstructure data is not designed to improve the accuracy of structure-property predictions. In fact, it is possible that a low-dimensional microstructure representation that offers low reconstruction error may encode two microstructures with drastically different properties similarly in the representation space, making it difficult to create high-accuracy predictive models directly from the representations.

Nonetheless, we note that it is possible to learn representations that are both useful for reconstruction and for property prediction. This requires formulating the learning problem to include both the losses from a generative model and those for a discriminative (predictive) model, see (Sabour *et al.*, 2017) as an example. Our findings along this direction will be reported through other venues.

Physics-based generative models The successful application of morphology loss in this paper inspires an important question: If simple morphology constraints can be used to improve the generation quality of microstructures, can we leverage other knowledge about material systems to further improve the generation quality of their structures? For example, would it be possible to learn to generate microstructures for given processing parameters directly based on a physics-driven process-structure mapping, rather than through the data created by the mapping? Such theory-driven, as opposed to data-driven, generative models can potentially reach higher generation qualities, as their outputs are calibrated to be coherent with physical laws instead of through numerical interpolation among data points. Again, our findings in this vein will be reported through follow-up papers.

5.6 Conclusions

This paper proposed a method for generating an arbitrary amount of artificial microstructure samples with low computation cost and a small amount of training samples. The key contribution is the incorporation of a style loss into the training to significantly improve the quality of the artificial microstructures. For verification, we applied the proposed method to generating sandstone samples. Results showed that our method is more data-efficient at improving the prediction performance of a structure-property mapping than a state-of-the-art Markov Random Field method.

The findings from this paper suggests that integration of material specific knowledge into generative models can be a promising direction towards data-efficient generation of complex microstructures and accelerated microstructure-mediated material design.

5.7 Introduction

This paper is motivated by the observation that experienced human engineers can quickly generate solutions based on accumulated knowledge, while algorithms are only programmed to solve individual problems from scratch, even when the problems are structured similarly. The lack of ability to generalize from experience makes algorithms often too slow to respond to real-world challenges, especially when a stringent time or cost budget is in place. For example, the design of vehicle body-in-white is often done by experienced structure engineers, since topology optimization (TO) on full-scale crash simulation is not yet fast enough to respond to requests from higher-level design tasks, e.g., geometry design with style and aerodynamic considerations, and thus may slow down the entire design process [1] .

Research exists in developing deep neural network models that *learn* to create structured solutions in a *one-shot* fashion, circumventing the need of iterations (e.g., in solving systems of equations (Tompson *et al.*, 2016), simulating dynamical systems (Chu and Thuerey, 2017), or searching for optimal solutions (Giryes *et al.*, 2018; Sosnovik and Oseledets, 2017; Ulu *et al.*, 2016)). Learning of such models through data, however, is often criticized to have limited generalization capability, especially when highly nonlinear input-output relations or high-dimensional output spaces exist (Szegedy *et al.*, 2013; Kurakin *et al.*, 2016; Huang *et al.*, 2017). In the context of TO, this means that the network may create structures with unreasonably poor physical properties when it responds to new problem settings. More concretely, consider a topology with a tiny crack in one of its trusses. This design would be far from optimal if the goal is to lower compliance, yet standard data-driven learning mechanisms do not prevent this from happening, i.e., they don't know that they don't

[1] The authors learned about this challenge through personal communication with an engineer at a major car manufacturer.

know (physics).

Our goal is to create a learning mechanism that knows what it does not know, and thus can self-improve in an effective way. Specifically, we are curious about how physics-based knowledge, e.g., in the forms of dynamical models, theoretical bounds, and optimality conditions, can be directly injected into the learning of networks that perform one-shot solution generation. We call this type of learning mechanisms "*theory-driven*". This idea is particularly enchanting as such knowledge widely exists in engineering disciplines and is often differentiable, which is important for gradient-based learning algorithms.

As an initial attempt in this direction, this paper will focus on TO, and uses a compliance minimization problem (Wu *et al.*, 2018) as a running example. To overview, we introduce an interaction between a student (the neural network) and a teacher (the optimality conditions underlying the TO): The student learns from existing data and is tested on unseen problems. Deviation of the student's solutions from the optimality conditions is quantified, and used to choose new data points for the student to learn from. We use this mechanism to learn a solution generator from a distribution of compliance minimization problems with different load settings. We show that the proposed method has significantly better performance in predicting the optimal topologies for unseen loads, than using a static data collection, under the same computational cost for data acquisition.

The proposed method will enable learning of solution generators for engineering problems that need to be solved repetitively with variations in their settings, e.g., optimal design (structures, networks, geometries, etc.) as an inner loop of a larger-scale system design, or real-time optimal control with context-dependent parameters. The generated solutions kick-start iterative solvers with good initial guesses and thus shorten the design (solution searching) processes.

The rest of the paper is structured as follows: In Sec. 5.8 we review related work at the intersection of generative design and machine learning, and highlight the new contributions from this paper. We then revisit in Sec. 5.9 the compliance minimization problem, which will be used as a running example throughout the paper, and explain the need for learning one-shot solution generators. Sec. 5.10 introduces the proposed learning mechanism, which will then be validated in Sec. 5.11 against two benchmark methods on the running example. Sec. 5.12 discusses the connection between the proposed method and recent developments in machine learning, and suggests future directions. Sec. 5.13 concludes the paper.

5.8 Related Work

Existing research on generative design answer three types of questions: (1) What is the design *representation*? (2) What is the *goodness* measure of a design? And based on these two, (3) how do we *search* for a good design? Challenges in answering these questions include *high-dimensional or ill-defined design spaces* such as for topologies (Bendsøe and Kikuchi, 1988; Guo *et al.*, 2018), material microstructures (Cang *et al.*, 2017b, 2018; Bostanabad *et al.*, 2018), or complex geometries (Stiny, 1980; Hsiao and Chen, 1997), *expensive evaluations* of designs and their sensitivities, e.g., due to model nonlinearity (Allaire *et al.*, 2004; Jung and Gea, 2004), coupled materials or physics (Sigmund, 2001; Wang and Wang, 2004; Zhu *et al.*, 2017), or subjective goodness measures (Ren and Papalambros, 2011; Orbay *et al.*, 2015), or *search inefficiency* due to the absence of sensitivities (Xie and Steven, 1993; Querin *et al.*, 1998; Rao *et al.*, 2011) or the existence of random variables (Kharmanda *et al.*, 2004).

This paper takes a different angle by focusing on design tasks for which answers to the above three questions exist, yet applying them to real-world design tasks is computationally unaffordable, thus the need for algorithms that learn to *improve* their efficiency through problem solving. Below we review related work, and explain how this paper is similar and different from them.

To start with, the task of improving learning efficiency through experience is known as *meta-learning* (Vilalta and Drissi, 2002; Hochreiter *et al.*, 2001) (with close connections to transfer learning (Caruana, 1995; Pan and Yang, 2010; Ganin *et al.*, 2016) and life-long learning (Thrun, 1996)). While there exists a broad range of problem settings within the literature of meta-learning, the setup commonly involves a student and a learning mechanism that specifies how the student updates its way of learning or solving problems. The student outputs a solution for every input problem, for example, the input can be a labeled dataset and the output a classifier that explains the data, or the input an optimization problem and the output an optimal solution. In the former case, the student updates its way of learning a classifier (Ravi and Larochelle, 2016) or its hypothesis space of classifiers (Vilalta and Drissi, 2002); in the latter, it updates its gradient (Andrychowicz *et al.*, 2016) or non-gradient (Chen *et al.*, 2016) search strategies. These updates are governed by the learning mechanism, the design of which is driven by a goodness measure of the student, e.g., the generalization performance of a classifier or the convergence rate of an optimization solver.

The problem of learning to generate designs can be cast as a meta-learning problem, where the student is the generator that takes in settings of the design problem (e.g., the distribution and magnitude of loads, the material properties, or the boundary conditions for a topology optimization problem) and outputs a design solution, whereas the learning mechanism updates the architecture or parameters of the generator.

It is worth noting two differences between the proposed method and contemporary meta-learning. First, typical mechanisms proposed in meta-learning literature are iterative, e.g., in forms of recurrent neural networks (Ravi and Larochelle, 2016; Andrychowicz *et al.*, 2016; Wang *et al.*, 2016). This choice of model is due to the iterative nature of problem solving and the need of memory in decision making during the iteration. In contrary, we model the transition from problem settings to solutions using a feedforward neural network, which is one-shot in nature. We made this choice based on the finding that optimal solutions to a distribution of TO problems often form a continuous manifold (see Sec. 5.9 for details), which suggests that directly learning the manifold through a feedforward network might be achievable, in which case we circumvent the challenges from modeling iterative solvers.

Secondly, we note that existing meta-learning tasks are often set in contexts where large data acquisition is affordable. This does not hold in our case, since finite element analysis and design sensitivity analysis are costly yet necessary for training the neural network. This requires us to focus on adaptively choosing data points to improve the generator, thus rendering our approach somewhat more similar to active learning (Tong and Koller, 2001; Settles, 2012), where the goal is to improve data efficiency of learning by querying data based on the learned model. Nonetheless, active learning strategies are usually statistics-based. For example, in the context of classification, new data points are chosen based on uncertainty of their predicted labels (Lewis and Catlett, 1994) or their predicted contribution to the prediction error of the learned model (Roy and McCallum, 2001). These methods, however, are not suitable for our case since we care about the physical optimality of the generated topologies, rather than the pixel-wise matching between the generated topologies and the corresponding true optima (e.g., l2-norm or cross-entropy defined on image differences often used as metrics of prediction error). The method we introduce is thus distinctively different from active learning, as we choose data based on the optimality conditions of TO, which are problem-dependent and theory-driven.

Lastly, the proposed learning method is aligned with the recent surge of machine learning techniques with integrated physics knowledge. Among this body of work, (Wu *et al.*, 2016) proposes to learn a neural network for predicting intrinsic physical properties of objects with the assistance of a physical simulator that computes object interactions based on the predicted properties. (Jonschkowski *et al.*, 2017) developed an encoder that computes object positions and velocities from images of objects, by enforcing these properties to be compliant with common sense. Similarly, (Stewart and Ermon, 2017) proposed a physics-based regularization to learn object trajectories and human movements from videos. Instead of being taught physics, (Denil *et al.*, 2016) demonstrated a grounded way for machines to acquire an intuition of the physical world through reinforcement learning. Our paper is similar to those learning mechanisms with injected physics constraints, while employing adaptive sampling rather than batch-mode training to avoid expensive simulations.

5.9 Problem Statement

5.9.1 One-shot solution generator

We define a one-shot solution generator as a feedforword neural network $\mathbf{x} = g(\mathbf{s}, \boldsymbol{\theta})$ that computes a solution $\mathbf{x}$ (e.g., a topology) given problem settings $\mathbf{s}$ (e.g., loads) and network parameters $\boldsymbol{\theta}$. It is one-shot in the sense that computing $\mathbf{x}$ through g is much less expensive than using an iterative algorithm. We also define the generalization performance of the generator (denoted by $F(\boldsymbol{\theta})$) as the expected performance of its solutions over a distribution of problems specified by a probability density function $p(\mathbf{s})$:

$$F(\boldsymbol{\theta}) = \mathbb{E}_{p(\mathbf{s})} f(g(\mathbf{s}, \boldsymbol{\theta}), \mathbf{s}), \tag{5.21}$$

where $f(\mathbf{x}, \mathbf{s})$ measures the performance of $\mathbf{x}$ under $\mathbf{s}$.

5.9.2 The compliance minimization problem

We now review the mechanical compliance minimization problem introduced in (Wu *et al.*, 2018) to substantiate $f(\mathbf{x}, \mathbf{s})$. In this context, a solution $\mathbf{x}$ is a N-by-1 vector with values between 0 and 1, elements of which control a density vector, denoted as $\boldsymbol{\rho}$, of the corresponding physical elements of a meshed structure through the following relation:

$$\rho_e = \frac{\tanh(\beta/2) + \tanh(\beta(\tilde{x}_e - 0.5))}{2\tanh(\beta/2)}, \tag{5.22}$$

where x_e and ρ_e are the elements of $\mathbf{x}$ and $\boldsymbol{\rho}$, respectively, for $e = 1, \cdots, N$. The shape parameter β controls the sharpness of the transition to the density ρ_e from a filtered variable $\tilde{x_e}$, which is a weighted average of neighbours of x_e:

$$\tilde{x}_e = \frac{\sum_{i \in \mathcal{M}_e} \omega_{i,e} x_i}{\sum_{i \in \mathcal{M}_e} \omega_{i,e}}, \tag{5.23}$$

where $\mathcal{M}_e$ is the set of neighbours of element e, and weights $\omega_{i,e}$ are defined as

$$\omega_{i,e} = 1 - ||z_i - z_e||_2 / r_e, \tag{5.24}$$

with z_i the coordinates of meshed element i, and r_e the filter radius. In TO, this filter (Eq. (5.23)) is used to prevent convergence to impractical checkerboard topologies (Wu *et al.*, 2018).

The connection from $\mathbf{x}$ to the global stiffness matrix of the topology (denoted as $\mathbf{K}$) can be established through the density vector $\boldsymbol{\rho}$. Given loads $\mathbf{s}$ and boundary conditions, the displacement $\mathbf{u}$ of the structure can be found by solving $\mathbf{Ku} = \mathbf{s}$, under the assumption that $\mathbf{K}$ is independent of $\mathbf{u}$ (e.g., linear elastic material and small displacement). The compliance minimization problem can be formulated as

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}, \mathbf{s}) = \frac{1}{2}\mathbf{u}^T\mathbf{K}\mathbf{u} \\ \text{subject to} \quad & \mathbf{K}\mathbf{u} = \mathbf{s}, \\ & x_e \in [0,1],\ \forall e \in \{1, \cdots, N\}, \\ & g_0(\mathbf{x}) = \frac{1}{N}\sum_e \rho_e - \alpha \leq 0, \\ & g_1(\mathbf{x}) = ||\bar{\rho}||_p - \alpha \leq 0. \end{aligned} \tag{TO}$$

Here the constraint g_0 (g_1) limits the global (local) density of the structure to be lower than a threshold α. $||\mathbf{x}||_p = (\frac{1}{N}\sum_e x_e^p)^{1/p}$ is the p-norm defined on $\mathbb{R}^N$. p is set to 16 following (Wu *et al.*, 2018) so that $||\mathbf{x}||_p$ approximately computes the maximum of $|\mathbf{x}|$ while being differentiable. The averaged local density $\bar{\rho}_e = (\sum_{i\in\mathcal{N}_e} \rho_i)/(\sum_{i\in\mathcal{N}_e} 1)$ is defined on the neighborhood $\mathcal{N}_e = \{i|\ ||x_i - x_e||_2 \leq R_e\}$ with radius R_e. Note that R_e for the local density constraint is different from the filter radius r_e. The optimality conditions of (TO) are listed as follows

$$\begin{aligned} & \nabla_{\mathbf{x}} L := -\mathbf{u}^T\frac{\partial \mathbf{K}}{\partial \mathbf{x}}\mathbf{u} + \mu_0\nabla_{\mathbf{x}} g_0 + \mu_1\nabla_{\mathbf{x}} g_1 + \boldsymbol{\mu}_u^T - \boldsymbol{\mu}_l^T = \mathbf{0}^T \\ & x_e \in [0,1],\ \forall e \in \{1, \cdots, N\}, \\ & g_0 \leq 0,\ g_1 \leq 0,\ \mu_0 g_0 = 0,\ \mu_1 g_1 = 0, \\ & \boldsymbol{\mu}_l^T\mathbf{x} = 0, \boldsymbol{\mu}_u^T(\mathbf{x} - \mathbf{1}) = 0 \\ & \mu_0 \geq 0,\ \mu_1 \geq 0,\ \boldsymbol{\mu}_u \geq \mathbf{0},\ \boldsymbol{\mu}_l \geq \mathbf{0}, \end{aligned} \tag{5.25}$$

where $\nabla_{\mathbf{x}} y(\mathbf{x}, \cdot)$ is the partial derivative of function y with respect to variables $\mathbf{x}$, and is defined as a row vector. Finding a solution to comply with Eq. (5.25) can be done through a gradient-based solver, e.g., an augmented Lagrangian algorithm (see Sec. 6.3.3). However, we need to note that the computational cost for converging to an optimal solution usually does not scale well. In particular, solving Eq. (5.25) with a problem size $N = 4800$ requires on average around 5000 finite element analyses (i.e., computing $\mathbf{u}$ from $\mathbf{K}\mathbf{u} = \mathbf{s}$).

5.9.3 Learning a solution generator

With the above setup, the problem of learning a one-shot solution generator can be formulated as follows:

$$\begin{aligned}
\min_{\boldsymbol{\theta}} \quad & F(\boldsymbol{\theta}) = \mathbb{E}_{p(\mathbf{s})} f(\mathbf{x}, \mathbf{s}) \\
\text{subject to} \quad & \mathbf{x} = g(\mathbf{s}, \boldsymbol{\theta}), \\
& \mathbf{K}\mathbf{u} = \mathbf{s}, \\
& f(\mathbf{x}, \mathbf{s}) = \frac{1}{2}\mathbf{u}^T \mathbf{K}\mathbf{u} \\
& g_0(\mathbf{x}) = \frac{1}{N}\sum_e \rho_e - \alpha \leq 0, \\
& g_1(\mathbf{x}) = ||\bar{\rho}||_p - \alpha \leq 0.
\end{aligned} \tag{P}$$

We will force network outputs to be within $(0,1)^N$ by attaching sigmoid activations to its output layer.

5.10 Learning with a Physics-based Criterion

(P) can be solved by matching the input-output pairs of the generator to a dataset $\mathcal{D} = \{\mathbf{x}_i^D, \mathbf{s}_i^D\}_{i=1}^n$ where $\mathbf{x}_i^D$ are optimal for $\mathbf{s}_i^D$. This leads to the data-driven learning formulated as follows:

$$\min_{\boldsymbol{\theta}} \sum_{\mathcal{D}} ||g(\mathbf{s}_i^D, \boldsymbol{\theta}) - \mathbf{x}_i^D||_2^2. \tag{P1}$$

As we reviewed in the last section, collecting $\mathcal{D}$ can be costly due to the iterative nature of solving the topology optimization problem (TO). On the other hand, checking the compliance of an arbitrary topology $\mathbf{x}$ to the optimality conditions (Eq. (5.25)) only requires solving $\mathbf{K}\mathbf{u} = \mathbf{s}$ once. This finding indicates that the optimality conditions may offer affordable means to identify new data point that will most effectively improve the generalization performance of g. Specifically, we define the deviation of solution $\mathbf{x}$ from the optimality conditions as

$$d(\mathbf{x}, \boldsymbol{\mu}) = ||\nabla_{\mathbf{x}} L||_2^2 + w_0 g_0^2 + w_1 g_1^2 \tag{5.26}$$

where the algorithmic parameters w_0 and w_1 weight the penalties on constraints g_0 and g_1, respectively. One issue in evaluating d is that we do not know the values of the Lagrangian multipliers (which are denoted by $\boldsymbol{\mu}^T = [\mu_0, \mu_1, \boldsymbol{\mu}_l^T, \boldsymbol{\mu}_u^T]$) before solving the problem. To this end, we propose to find $\boldsymbol{\mu}^*$ that minimizes the deviation of $\nabla_{\mathbf{x}} L$ from $\mathbf{0}$ subject to their constraints from (P):

$$\begin{aligned} \min_{\boldsymbol{\mu}} \quad & ||\nabla_{\mathbf{x}} L||_2^2 \\ \text{subject to} \quad & \boldsymbol{\mu} \geq 0,\ \boldsymbol{\mu}^T \mathbf{g} = 0. \end{aligned} \tag{P2}$$

By solving (P2) for all $\mathbf{s}$ in a validation set $\mathcal{S}_v$, we can then choose a new training data point $(\mathbf{s}^*, \mathbf{x}^*)$ for which the minimal deviation $d(g(\mathbf{s}^*, \boldsymbol{\theta}), \boldsymbol{\mu}^*)$ is the largest among $\mathcal{S}_v$. It is worth noting that (P2) is a $(2N+2)$-dimensional quadratic programming problem and can be solved efficiently using standard solvers (e.g., sequential quadratic programming).

The learning algorithm can now be summarized in Alg. 1. Details on setting the initial training set $\mathcal{S}_0$, the validation set $\mathcal{S}_v$, the computational budget B, and the budget lower bound b will be introduced along the case studies in Sec. 5.11.

5.11 Case Studies

This section presents two case studies where we demonstrate the superior learning efficiency of the proposed algorithm in comparison with two benchmark mechanisms. The first benchmark uses a static dataset ($\mathcal{D}_{\text{static}}$) for training, and is denoted as *Benchmark I*. The second benchmark, denoted as *Benchmark II*, is similar to the proposed approach, but chooses data points using a different heuristic. For all three

Algorithm 1: Theory-driven learning

```
input  : Problem distribution $p(\mathbf{s})$
output: Learned model parameters $\boldsymbol{\theta}^*$
Draw initial problem set $\mathcal{S}_0$ from $p(\mathbf{s})$;
Find optimal $\mathbf{x}_i$ for each $\mathbf{s}_i$ in $\mathcal{S}_0$;
Initialize dataset $\mathcal{D} = \mathcal{D}_0 = \{(\mathbf{x}_i, \mathbf{s}_i)\}_{i=1}^{|\mathcal{S}_0|}$;
Draw validation problems $\mathcal{S}_v$ from $p(\mathbf{s})$;
Set computation budget $B = B_0$ and budget lower bound $b$;
while $B > b$ do
    Update $\boldsymbol{\theta}^*$ by solving (P1) based on $\mathcal{D}$;
    Calculate $\boldsymbol{\mu}_i^*$ and $d(g(\mathbf{s}_i, \boldsymbol{\theta}^*), \boldsymbol{\mu}_i^*)$ for all $\mathbf{s}_i$ in $\mathcal{S}_v$ based on Eq. (5.26);
    Find $\mathbf{s}^* \in \mathcal{S}_v$ with the highest $d$ value;
    Derive $\mathbf{x}^*$ for $\mathbf{s}^*$ by solving (TO);
    Record $\delta B$ as the number of $\mathbf{Ku} = \mathbf{s}$ solved in solving (TO) and computing
      Eq. (5.26);
    Update the budget $B = B - \delta B$;
    $\mathcal{D} = \mathcal{D} + (\mathbf{x}^*, \mathbf{s}^*)$, $\mathcal{S}_v = \mathcal{S}_v - \mathbf{s}^*$;
end
```

learning mechanisms (Benchmarks I, II, and ours), the topology optimization problem is solved by an Augmented Lagrangian algorithm, details of which is deferred to the appendix.

5.11.1 The heuristic of Benchmark II

In Benchmark II, we evaluate the performance of the generator g by measuring the difference between the compliance produced by g and the predicted compliance based on the training data $\mathcal{D}$:

$$d_h(g(\mathbf{s}_i, \boldsymbol{\theta}^*)) = |f(g(\mathbf{s}_i, \boldsymbol{\theta}^*), \mathbf{s}_i) - \hat{f}(\mathbf{s}_i)|, \tag{5.27}$$

where $\mathbf{s}_i$ comes from the validation set $\mathcal{S}_v$, and $\hat{f}$ is an ordinary least square model that fits to $\{(\mathbf{s}_i^*, f(\mathbf{x}_i^*, \mathbf{s}_i^*) \mid (\mathbf{x}_i^*, \mathbf{s}_i^*) \in \mathcal{D}\}$. In this study, polynomial models are used for curve fitting. The validation data point with the highest value of d_h is chosen, and its true optimal topology is then computed and used to improve the generator.

5.11.2 Study setups

The topology optimization problems to be solved follow (TO). Two cases are created to demonstrate the scalability of the proposed method. In Case 1, the input $\mathbf{s}$ represents a single load applied to a 2D structure represented by a 40-by-120 mesh, see Fig. 39a. In this case, the input to the generator is the angle of the load, uniformly distributed in between 0 and π, i.e., $p(\mathbf{s}) = 1/\pi$. In Case 2, $\mathbf{s}$ encodes (1) the x- and y-coordinates of the loading, which is drawn uniformly from all nodes in the highlighted area in Fig. 39b, and (2) the direction of the point load uniformly drawn from 0 to 2π.

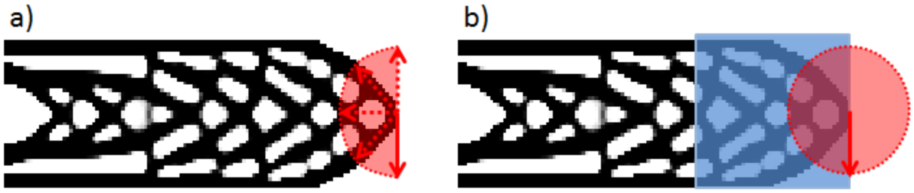


Figure 39: Sample solutions for (a) Case 1, where a point load is applied to the middle node at the tip of the cantilever beam, with directions uniformly distributed in $[0, \pi]$, and (b) Case 2, where the point load is applied to a node in the highlighted area which occupies one-third of the design space, with its direction drawn from $[0, 2\pi]$

To compare the three learning mechanisms, each is executed 10 times to account for the randomness in the sampling of the initial dataset $\mathcal{D}_0$ (and $\mathcal{D}_{\text{static}}$ in the case of Benchmark I) and the validation set $\mathcal{S}_v$ at each iteration. The generalization performance is measured by

$$F(\boldsymbol{\theta}^*) = \sum_{\mathbf{s}_i \in \mathcal{S}_t} f(g(\mathbf{s}_i, \boldsymbol{\theta}^*), \mathbf{s}_i), \tag{5.28}$$

where $\mathcal{S}_t$ is a separate test set drawn from $p(\mathbf{s})$. For Case 1, the sample sizes are $|\mathcal{S}_0| = 5$, $|\mathcal{S}_v| = 100$, $|\mathcal{S}_t| = 100$, and $|\mathcal{D}_{\text{static}}| = 16$. For Case 2, since the problem space is much larger, the sample sizes are set to $|\mathcal{S}_0| = 1000$, $|\mathcal{S}_v| = 100$, $|\mathcal{S}_t| = 1000$, and $|\mathcal{D}_{\text{static}}| = 7000$. In addition, since validating a large number of inputs becomes expensive, we uniformly sample 100 validation points from $|\mathcal{S}_v|$ to perform active learning in each iteration. For both cases, the computational budget B_0 is set as $b_{\min}|\mathcal{D}_{\text{static}}|$, where $b_{\min}$ is the minimal solution cost among all problems sampled for Benchmark I. The budget lower bound b is set to the maximal cost among the same set of problems. This setting ensures that the adaptive methods (Benchmark II and the proposed method) will always use less computational resource than the static method for topology optimization, thus creating a comparison in favor of the latter. For the following results, we set $w_0 = w_1 = 1$ for the proposed method. A full parametric study on these hyper-parameters has not been conducted, yet the effectiveness of the current setting is validated (Sec. 5.11.4).

5.11.3 Architectures of the solution generators

The architectures of the solution generators are summarized in Fig. 40. For Case 1, we use a two-dimensional input to represent the x- and y-components of the point load. For Case 2, we use a three-dimensional input to represent the x, y location and the orientation of the point load. The choice of these input representations are based on empirical tests of the generalization performance of the learned models.

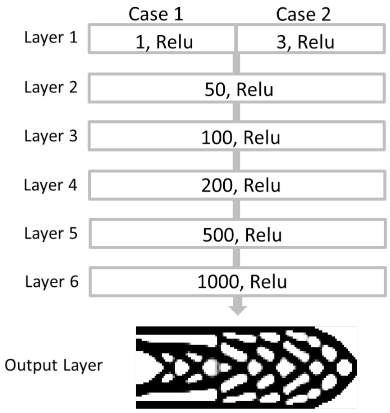


Figure 40: Architectures of the solution generators

5.11.4 Results

Case 1 results For Case 1, Fig. 41a compares the compliance of the topologies generated by all three learning mechanisms with the ground truth for all test inputs; Fig. 41b reports the corresponding compliance gaps produced by these mechanisms. The generalization performance of a learning mechanism is measured by the mean and the standard deviation of the average compliance gaps across all test inputs. The result shows that the proposed mechanism outperforms the benchmarks at predicting optimal topologies for unseen loading conditions in a low-dimensional case. To further demonstrate the difference between the three mechanisms, we visualize and compare generations for four test loads in Fig. 39. The loading directions are marked in the last row of the figure, which are 0.3π, 0.4π, 0.45π,0.55π respectively. The resultant compliance values are shown at the bottom of each topology.

Challenges in Case 2 Case 2 examines the performance of the proposed mechanism under a higher-dimensional and larger input space. We notice that the learning becomes significantly more challenging in this case. see Sec 5.12). Specifically, for all

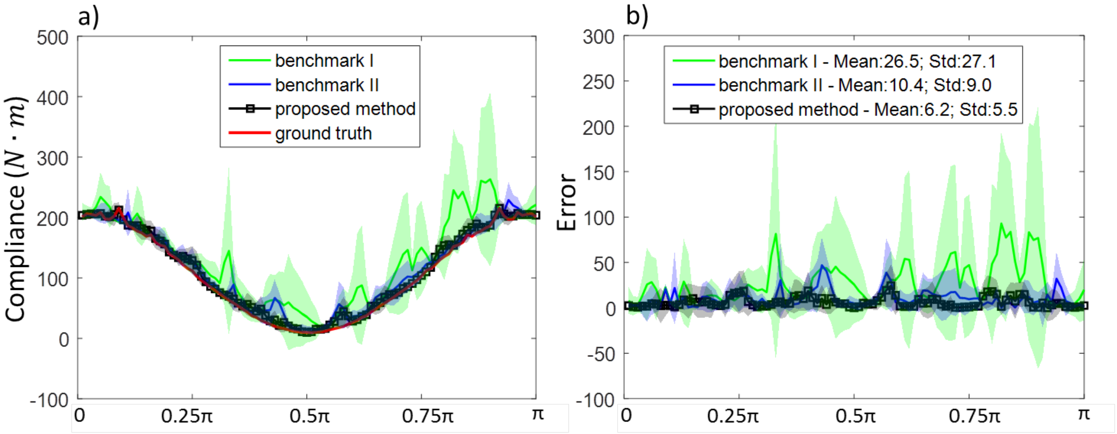


Figure 41: (a) Comparison between the compliance of the ground truth topologies (red curve) and those from the generated solutions from all three learning mechanisms. 10 independent experiments are conducted for each learning mechanism. The means and the standard deviations are reported. (b) The absolute compliance gaps.

three algorithms under the same budget (which is equivalent to solving 7000 TO problems), there exist inputs for which the generated topologies have significantly larger compliance than the ground truth. We mark test data points with a compliance gap of over 1000 as failed designs. For Benchmark I, II and the proposed method, the mean failure rates over the entire test are 14%, 0.8%, and 0.6% respectively. Some failed generations are shown in Fig. 43 along side the corresponding ground truth.

An augmented learning objective To alleviate this issue, we introduce design sensitivity $\nabla_{\mathbf{x}} f$ at the optimal solution as a weighting factor of the learning loss in (P1), based on the insight that structural elements with higher sensitivity contribute more to the compliance and thus should have higher priority during model training. Concretely, the training problem at each iteration of the active learning process is now formulated as

$$\min_{\boldsymbol{\theta}} \sum_{(\mathbf{s}_i,\mathbf{x}_i)\in\mathcal{D}} ||g(\mathbf{s}_i,\boldsymbol{\theta})-\mathbf{x}_i||^2_{\mathbf{\Lambda}_i}, \tag{5.29}$$

where $\mathbf{\Lambda}_i = diag([\lambda_{i,1},\cdots,\lambda_{i,N}])$ is a diagonal weighting matrix, $\lambda_{i,e} = \frac{\nabla_{\mathbf{x}} f - \min(\nabla_{\mathbf{x}} f)}{\max(\nabla_{\mathbf{x}} f)-\min(\nabla_{\mathbf{x}} f)}$, and $||\mathbf{x}||^2_{\mathbf{\Lambda}} = \mathbf{x}^T\mathbf{\Lambda}\mathbf{x}$. It is important to note that $\mathbf{\Lambda}_i$ is a byproduct of computing $\mathbf{x}_i$

images/top_com.png

Figure 42: Case 1 topologies predicted by (a) Benchmark I, (b) Benchmark II, (c) the proposed method. (d) Ground truth computed using the Augmented Lagrangian solver. Compliance values are shown at the bottom of each topology. Loading directions are marked as red arrows.

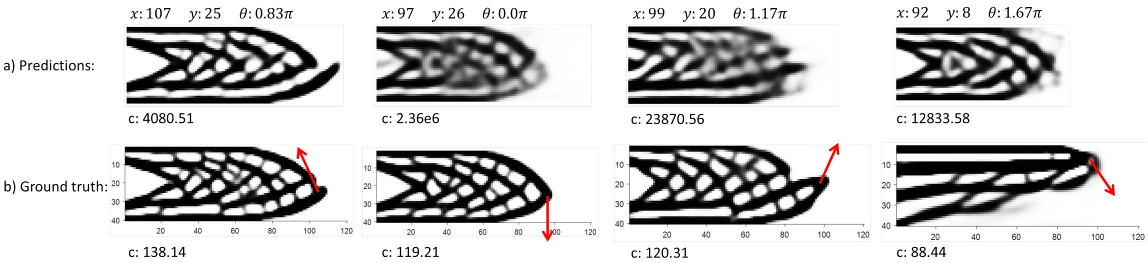


Figure 43: Example of failed generations with significant larger compliance than the corresponding ground truth. Loading conditions are marked at the top and compliance at the bottom. Some generations are close to the ground truth in the pixel space but miss critical elements, such as the first three. Some of the generations are simply off, such as the last one.

Table 3: Comparisons on the mean count of failures from 1000 test data points and the median compliance gaps for Case 2

	Failure rate: mean (std)	Median compliance gap: mean (std)
Benchmark I	5.12% (1.2%)	22.12 (3.34)
Benchmark II	0.64% (0.42%)	6.64 (0.87)
Propose method	**0.16%** (0.09%)	**5.2** (0.6)

and does not cost extra budget.

Case 2 results By introducing this augmentation, the mean rate of failures drops to 5.12%, 0.64%, and 0.16% for Benchmark I, II, and the proposed method, respectively. Since the compliance distributions are far from normal due to the failed designs, we report the means and standard deviations of the median compliance gaps from all experiments and learning mechanisms instead. These results are summarized in Table 3. As a demonstration, we compare in Fig. 44 the generations from all mechanisms under four test settings. The proposed method has the closest compliance to the ground truth.

Validity of hyper-parameter settings For both case studies we set the weights for the local and global constraints (w_0 and w_1 in Eq. 5.26) to be $w_0 = 1$ and $w_1 = 1$.

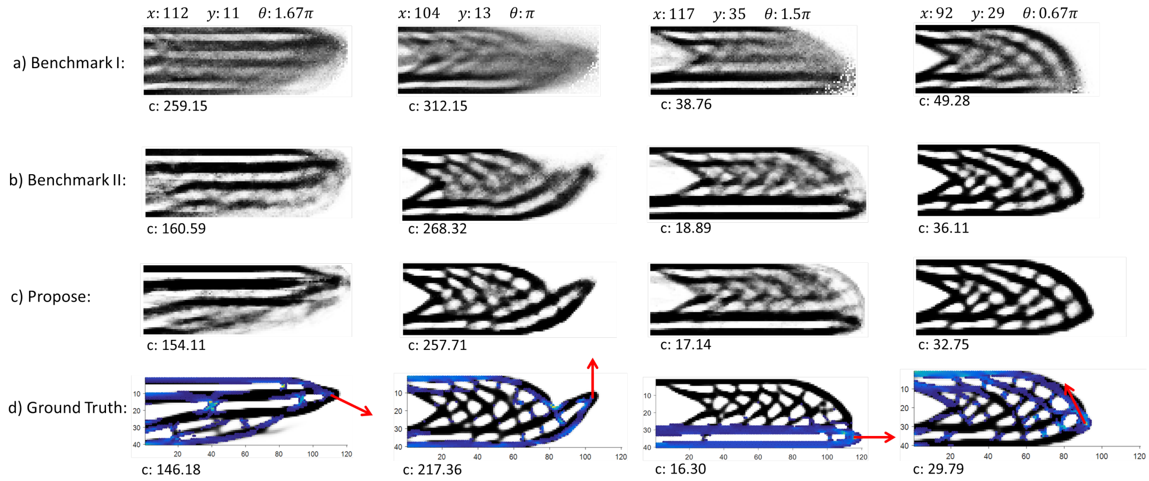


Figure 44: Case 2 optimal topologies predicted by (a) Benchmark I, (b) Benchmark II, and (c) the proposed method. (d) The ground truth topologies with design sensitivity information: cold (warm) color indicates lower (higher) design sensitivity. The point loads are marked as arrows with detailed input values reported at the top of the figure. The compliance values are shown at the bottom of each topology.

To validate this setting, we monitor the violation to the constraints by solutions generated for all test inputs. For a proper measure of constraint satisfaction, test cases where constraints are satisfied will be ignored in the calculation of the mean violation. It is noted that ground truth topologies have none-zero violation due to the non-zero error thresholds set in the augmented Lagrangian algorithm. In Case 1 (Case 2), the mean violation to the global volume fraction constraint is $g_0 = 4.89\%$ (3.0%) for the ground truth and $g_0 = 8.76\%$ (1.6%) for the generated solutions. In both cases, no violation to the local constraints are observed for either the ground truth or the generated solutions. This result indicates that the setting of the hyper-parameters leads to the learning of a generator with reasonable compliance to the global and local volume fraction constraints. A parameteric study is yet needed to fully characterize the tradeoff between the learning of effective topologies and that of constraint compliance by tuning the hyper-parameters.

Comparison on solution generation speed Lastly, we shall note that the time required for generating solutions through the learned model is negligible (in the order of 10^{-2} seconds per topology) compared with that through a TO solver (in the order of 10^2 to 10^3 seconds per topology).

5.12 Discussion

We now discuss the remaining issues related to learning manifolds of optimal solutions using neural networks.

5.12.1 Curse of dimensionality and potential solutions

So far we assumed that the pre-defined network architecture is able to capture the underlying manifold in high-dimensional spaces. As we see from the two cases, increasing the latent dimension of the manifold from one to three already raised the data demand significantly, indicating that the manifold of optimal topologies underlying Case 2 is much more "complex" than that of Case 1. Indeed, from Narayanan and Mitter (Narayanan and Mitter, 2010), the sampling complexity of learning a manifold (to within a specified tolerance) is exponential on the intrinsic dimension (the dimension of the input), linear on the intrinsic volume (the size of the input space), and polynomial on the curvature of the manifold.

This leads to two legitimate concerns. The first regards model sufficiency: predefining the network architecture of the generator could be a stab in the dark when a new manifold is to be learned, as we do not know whether the network is sufficiently flexible to fit to the manifold. The second regards data sufficiency: real-world design problems may have solution manifolds that are too data-demanding for active learning alone to handle.

Lei et al. (Lei *et al.*, 2018) recently proposed an approach to the first concern in the context of piecewise linear networks (with ReLU activations). It is shown that both the manifold and the network complexity can be measured by the number of polyhedral cells induced respectively by the geometry of the manifold and the architecture of the network. Further studies following these complexity measures may lead to protocols for determining the network architecture of the generator before learning of generator weights takes place, essentially by computing an upper bound on the number of planes needed to locally and linearly approximate the underlying manifold. An efficient algorithm for doing so, however, is yet to be developed.

The second concern, however, is more critical to the application of the proposed method. One potential solution is based on the insight that the governing equations for training can be derived at arbitrary spatial resolution of the structure. While the intrinsic dimension and volume do not change across resolutions, we hypothesize that lowering the resolution will reduce the manifold curvature, and thus reduce the sampling complexity. In the context of TO, with lower resolution of the structure (i.e., less elements), we expect an easier learning problem. Based on this hypothesis, it is possible that a hierarchical network architecture can be learned progressively to alleviate the curse of dimensionality: At each spatial resolution level, we learn a generator that predicts the transition from coarse solutions to the ones that satisfy the optimality conditions at this level. The coarse solutions are proposed by the generator learned from the lower-level resolution. The investigation of this approach will be reported in a separate paper. It should be reiterated that despite the inevitable scalability challenge, the value of learning models through domain-specific theories is clearly demonstrated in this study.

5.12.2 From theory-driven data selection to full theory-driven learning

Another direction to explore is based on the note that the proposed method does not solve the learning problem (P) directly. Rather, we collect true solutions and fit a generator (i.e., a neural network) to it, with the hope that by intelligently collecting true solutions on fly, the fit model will effectively converge to the true solution manifold governed by the optimality conditions. An interesting question is whether solving (P) directly through a gradient-based method can be achieved and will be more effective than the presented method. If we consider the presented learning mechanism as theory-driven data selection, then solving (P) directly will be full theory-driven learning. More concretely, in each iteration of the learning, we would need to solve a batch of TO problems *partially*, i.e., finding feasible topologies that reduce the violation to the optimality conditions, and use the resultant changes in topologies to update the solution generator. The key difference between the presented method and a full theory-driven learning mechanism is that while the former guarantees optimality of known solutions as long as the network is flexible enough to fit through these solutions, the latter does not have such a guarantee at any time during the training; instead, it requires less cost per iteration (since it does not completely solve TO problems), and may afford more iterations (batches of TO problems). Fig. 45 visualizes the difference between the two using a simple 2D illustration, where the circle is the unknown solution manifold, the curve represents a solution generator, and the dots are sampled solutions.

5.12.3 Interpretability of the generators

One additional challenge we face is the interpretation of the learned generator. Given the fact that the generators perform reasonably well in the two cases, one

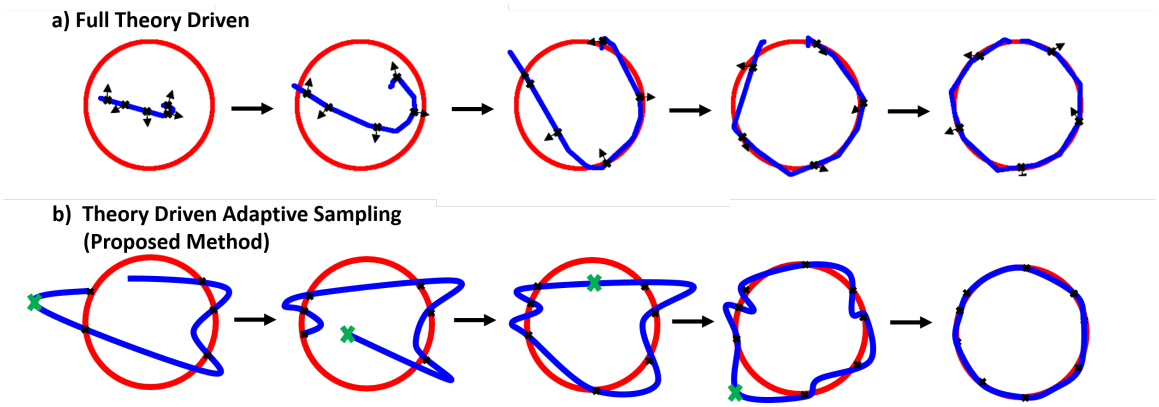


Figure 45: (a) Full theory-driven learning: In each iteration, random inputs are picked to compute the improvement directions of the model towards the ground truth. (b) Theory-driven adaptive sampling (this paper): Sample inputs are selected based on their violation to the optimality conditions. The corresponding ground truth (the crosses) are revealed, which informs the improvement directions of the model.

would like to visualize what they learn, e.g., local structural features that serve as puzzle pieces and lead to low-compliance structures when assembled. However, an investigation into the learned generators shows that such interpretable knowledge are not evident from the visualization of the network parameters. One potential reason is that the arbitrarily chosen network architecture may introduce confounding hidden nodes that decompose interpretable features that could have existed. One solution could be to impose regularization on all network parameters to be learned. Yet this will introduce more hyper-parameter tuning as a result.

5.12.4 Value of learning manifolds of optimal designs to designers

Last but not least, we shall come back to discuss and reiterate why and when learning manifolds of optimal designs has a value to designers. As we discussed in the introduction, there are cases where a large number of topology optimization problems (or other forms of optimization problems) need to be solved with only parametric differences. Such cases include when the TO is nested in a larger-scale system-level optimization, or when problem settings of the TO are required to be explored by human designers. In addition, designers may solve problems without knowing that similar ones have been previously solved by others. The proposed learning mechanism would not only allow a machine to accumulate and learn from solutions to similar problems, but also *effectively* practice by itself to reinforce its intuition at quickly solving common sets of problems, and thus may reduce the computational and energy cost of design tasks of growing complexity.

On the other hand, the learning itself requires solving more problems during adaptive sampling. Hence, it would be ideal to understand whether the expected total cost of solving a set of problems in the future surpasses the potential cost of learning. If so, the learning mechanism would have a value. The comparison between these two costs, however, will not be straight forward, as we do not know the learning cost beforehand for a certain performance threshold of the resultant generator. Therefore, a performance bound of the generator along the sample size will need to be developed to guide the decision on whether the learning has a practical value or not.

5.13 Conclusion

We were motivated by the lack of knowledge accumulation capability of existing computational solvers for design problems. This drawback of machines hampers them from quickly creating good solutions in response to changing requirements in real-world design processes. Our solution to this end was to create a solution generator that adaptively learns from true solutions from a distribution of problems and predicts solutions to other unseen problems from the same distribution. The generator was modeled by a feedforward neural network, and thus produces solutions in one-shot, as opposed to through iterations as in conventional approaches. Our key contribution was the introduction of problem-specific optimality conditions as a tractable validation measure to enable more effective learning. We highlight that computing the violation of a generated solution to the optimality conditions requires only a single FEA, while comparing the generated solution with the ground truth would require an entire TO process, which requires thousands of FEAs for the problems we studied. We showed through two case studies that the proposed learning algorithm achieves significantly better generalization performance than the benchmarks under the same computational cost.

Chapter 6

CONCLUSION

6.1 Summary

In this dissertation, two types of major issues in accelerating the design in the microstructure level of advanced material systems are explored - 1) the high resolution of material samples required by material property simulations causes the high dimensional design space and 2) the high cost of evaluating the design, making the searching and learning inefficient. As been discussed in Chapter 1, the key to addressing these two challenges is a). the learning of material representations and b). improve the learning and searching efficiency while staying in a small computational budget. Two corresponding tasks aim on the challenges showing that the material representations and the material microstructure distribution can be learned via convolutional neural network based generative model and a physics-based active learning framework can improve the network learning efficiency.

6.2 Contribution

Through the research, following contributions have been achieved. For task 1, 1) we showed that microstructure patterns can be learned and used to generate new material samples via CNN based generative model, such as the proposed Deep Belief Network and the VAE based Morphology Aware Network. 2) We showed that a low dimensional latent space can be characterized from the given high dimensional input with the learned feature. 3) We showed that generative models can be trained from the reduced latent space, and can produce morphologically and statistically similar samples compare to the limited given samples, which describes the distribution of input data. 4) We showed that the generated samples from the generative model can be applied to train predictive models to mimic the highly non-linear structure-property mapping, which allow us to bootstrap the limited material samples to improve model accuracy. For task 2., 1) We showed that a physics-based optimality condition can be used as a performance metric to evaluate the prediction by generative model. 2) We showed that the physics-based evaluation metric is able to guide the learning strategy to find an efficient way of searching and learning.

6.3 Future work

6.3.1 Improving the capability of microstructure representation learning

The built morphology-aware network is very efficient in characterizing and reconstructing heterogeneous material microstructure systems, reducing large microstructure dimensions (128×128 in the demonstrated case) to significantly smaller spaces (16 in the demonstration) from where similar material microstructures can be quickly generated. Two questions remain: 1) What has been learned by the low dimensional latent space? To figure out this problem, a physical property-based model should be applied to further tune the decoder to see whether the latent space is physically meaningful. 2) The proposed network is efficient on heterogeneous material microstructure systems, but for some other types of material systems such as Ti64, the generation quality remains unsatisfactory.

6.3.2 Reducing the computational cost in structure-property mapping

Predictive models for structure-property mapping is developed, which are very efficient in accelerating the the design process. Some remaining issues still need to be considered in the future: 1) the data-driven method is not applicable for all the material microstructure-property mapping. For example in Fig. 46, data-driven predictive results for the normalized fracture value of Ti64 alloy are shown. Three different regression models are applied: support vector regression model, VGG Net and the state-of-the-art Residual Network are trained on 1000 training data, but the prediction performance is still very poor according to the R^2 value; 2) the interpretation of the learned generator proposed in Chapter 6 requires more investigation. While the generator performs well, it is yet difficult to produce meaningful visualization of what it actually learned.

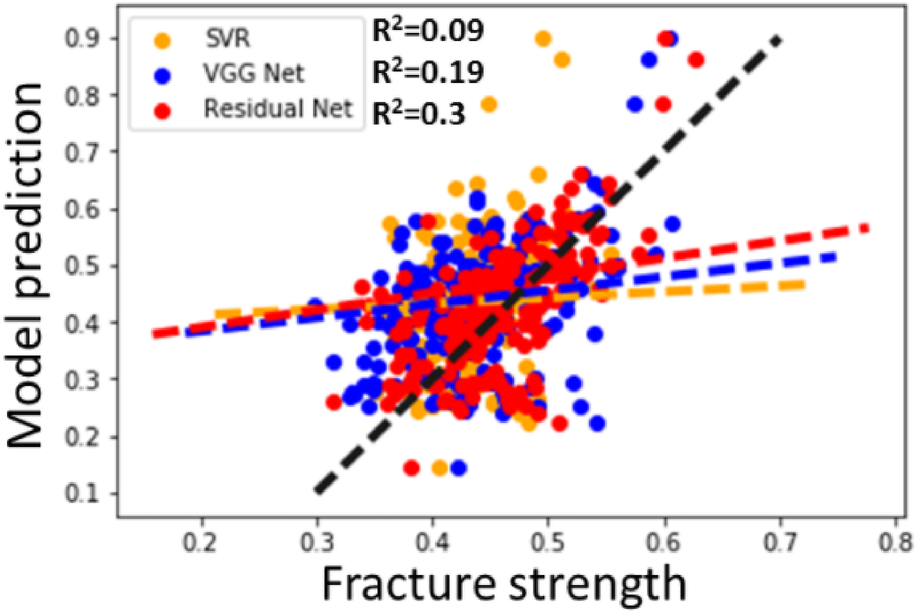


Figure 46: Data-driven predictive method on evaluating the normalized fracture value of Ti64 alloy; support vector regression (SVR), VGG Net and state-of-the-art Residual Network are used, but none of them can produce a good R^2 value

6.3.3 Interpretability of the meta-learner

Given the fact that the generators perform reasonably well in the topology optimization cases, so what do the learners learn from the problem would be very interesting. One would like to visualize what they learned, e.g., local structural features that serve as puzzle pieces and lead to low-compliance structures when assembled. However, an investigation into the learned generators shows that such interpretable knowledge is not evident from the visualization of the network parameters. One potential reason is that the arbitrarily chosen network architecture may introduce confounding hidden nodes that decompose interpretable features that could have existed.

REFERENCES

Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, “Tensorflow: Large-scale machine learning on heterogeneous distributed systems”, arXiv preprint arXiv:1603.04467 (2016).

Al-Raoush, R. and C. Willson, “Extraction of physically realistic pore network properties from three-dimensional synchrotron x-ray microtomography images of unconsolidated porous media systems”, Journal of hydrology **300**, 1, 44–64 (2005).

Allaire, G., F. Jouve and A.-M. Toader, “Structural optimization using sensitivity analysis and a level-set method”, Journal of computational physics **194**, 1, 363–393 (2004).

Andrychowicz, M., M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford and N. De Freitas, “Learning to learn by gradient descent by gradient descent”, in “Advances in Neural Information Processing Systems”, pp. 3981–3989 (2016).

Antonellini, M., A. Aydin, D. Pollard and P. d’Onfro, “Petrophysical study of faults in sandstone using petrographic image analysis and x-ray computerized tomography”, Pure and Applied Geophysics **143**, 1-3, 181–201 (1994).

Appoloni, C. R., Á. Macedo, C. P. Fernandes and P. C. Philippi, “Characterization of porous microstructure by x-ray microtomography”, X-Ray Spectrometry **31**, 2, 124–127 (2002).

Ashby, M. F., “Materials selection in mechanical design”, MRS Bull **30**, 12, 995 (2005).

Ashby, M. F. and D. Cebon, “Materials selection in mechanical design”, Le Journal de Physique IV,DOI:10.1051/jp4:1993701 **3**, C7, C7–1 (1993).

Baldwin, A. F., T. D. Huan, R. Ma, A. Mannodi-Kanakkithodi, M. Tefferi, N. Katz, Y. Cao, R. Ramprasad and G. A. Sotzing, “Rational design of organotin polyesters”, Macromolecules **48**, 8, 2422–2428 (2015).

Basanta, D., M. A. Miodownik, E. A. Holm and P. J. Bentley, “Using genetic algorithms to evolve three-dimensional microstructures from two-dimensional micrographs”, Metallurgical and Materials Transactions A **36**, 7, 1643–1652 (2005).

Bendsøe, M. P. and N. Kikuchi, “Generating optimal topologies in structural design using a homogenization method”, Computer methods in applied mechanics and engineering **71**, 2, 197–224 (1988).

Bengio, S., Y. Bengio, J. Cloutier and J. Gecsei, “On the optimization of a synaptic learning rule”, in “Preprints Conf. Optimality in Artificial and Biological Neural Networks”, pp. 6–8 (Univ. of Texas, 1992).

Bengio, Y., "Learning deep architectures for AI", Foundations and trends® in Machine Learning **2**, 1, 1–127 (2009).

Bengio, Y., P. Lamblin, D. Popovici and H. Larochelle, "Greedy layer-wise training of deep networks", in "Advances in neural information processing systems", pp. 153–160 (2007).

Berthelot, D., T. Schumm and L. Metz, "Began: boundary equilibrium generative adversarial networks", arXiv preprint arXiv:1703.10717 (2017).

Blair, S. C., P. A. Berge and J. G. Berryman, "Using two-point correlation functions to characterize microgeometry and estimate permeabilities of sandstones and porous glass", Journal of Geophysical Research: Solid Earth **101**, B9, 20359–20375 (1996).

Borbely, A., F. Csikor, S. Zabler, P. Cloetens and H. Biermann, "Three-dimensional characterization of the microstructure of a metal–matrix composite by holotomography", Materials Science and Engineering: A **367**, 1, 40–50 (2004).

Bostanabad, R., A. T. Bui, W. Xie, D. W. Apley and W. Chen, "Stochastic microstructure characterization and reconstruction via supervised learning", Acta Materialia **103**, 89–102 (2015).

Bostanabad, R., A. T. Bui, W. Xie, D. W. Apley and W. Chen, "Stochastic microstructure characterization and reconstruction via supervised learning", Acta Materialia **103**, 89–102 (2016).

Bostanabad, R., Y. Zhang, X. Li, T. Kearney, L. C. Brinson, D. W. Apley, W. K. Liu and W. Chen, "Computational microstructure characterization and reconstruction: Review of the state-of-the-art techniques", Progress in Materials Science **95**, 1 – 41, URL `http://www.sciencedirect.com/science/article/pii/S0079642518300112` (2018).

Bottou, L., "Large-scale machine learning with stochastic gradient descent", in "Proceedings of COMPSTAT'2010", pp. 177–186 (Springer, 2010).

Boureau, Y.-l., Y. L. Cun *et al.*, "Sparse feature learning for deep belief networks", in "Advances in neural information processing systems", pp. 1185–1192 (2008).

Bousquet, O. and L. Bottou, "The tradeoffs of large scale learning", in "Advances in neural information processing systems", pp. 161–168 (2008).

Broderick, S., C. Suh, J. Nowers, B. Vogel, S. Mallapragada, B. Narasimhan and K. Rajan, "Informatics for combinatorial materials science", JOM Journal of the Minerals, Metals and Materials Society **60**, 3, 56–59 (2008a).

Broderick, S., C. Suh, J. Nowers, B. Vogel, S. Mallapragada, B. Narasimhan and K. Rajan, "Informatics for combinatorial materials science", JOM **60**, 3, 56–59, URL `http://dx.doi.org/10.1007/s11837-008-0035-x` (2008b).

Cang, R., H. Li, H. Yao, Y. Jiao and Y. Ren, "Improving direct physical properties prediction of heterogeneous materials from imaging data via convolutional neural network and a morphology-aware generative model", Computational Materials Science **150**, 212 – 221, URL http://www.sciencedirect.com/science/article/pii/S0927025618302337 (2018).

Cang, R. and M. Y. Ren, "Deep network-based feature extraction and reconstruction of complex material microstructures", in "ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference", pp. V02BT03A008–V02BT03A008 (American Society of Mechanical Engineers, 2016).

Cang, R., A. Vipradas and Y. Ren, "Scalable microstructure reconstruction with multi-scale pattern preservation", in "ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference", pp. V02BT03A010–V02BT03A010 (American Society of Mechanical Engineers, 2017a).

Cang, R., Y. Xu, S. Chen, Y. Liu, Y. Jiao and M. Y. Ren, "Microstructure representation and reconstruction of heterogeneous materials via deep belief network for computational material design", arXiv preprint arXiv:1612.07401 (2016).

Cang, R., Y. Xu, S. Chen, Y. Liu, Y. Jiao and M. Y. Ren, "Microstructure representation and reconstruction of heterogeneous materials via deep belief network for computational material design", Journal of Mechanical Design **139**, 7, 071404 (2017b).

Caruana, R., "Learning many related tasks at the same time with backpropagation", in "Advances in neural information processing systems", pp. 657–664 (1995).

Chen, H., E. Lin, Y. Jiao and Y. Liu, "A generalized 2D non-local lattice spring model for fracture simulation", Computational Mechanics **54**, 6, 1541–1558 (2014).

Chen, Y., M. W. Hoffman, S. G. Colmenarejo, M. Denil, T. P. Lillicrap, M. Botvinick and N. de Freitas, "Learning to learn without gradient descent by gradient descent", arXiv preprint arXiv:1611.03824 (2016).

Chu, M. and N. Thuerey, "Data-driven synthesis of smoke flows with cnn-based feature descriptors", arXiv preprint arXiv:1705.01425 (2017).

Coates, A., H. Lee and A. Y. Ng, "An analysis of single-layer networks in unsupervised feature learning", Ann Arbor **1001**, 48109, 2 (2010).

Coker, D. A., S. Torquato and J. H. Dunsmuir, "Morphology and physical properties of fontainebleau sandstone via a tomographic analysis", Journal of Geophysical Research: Solid Earth **101**, B8, 17497–17506 (1996).

Curtarolo, S., G. L. Hart, M. B. Nardelli, N. Mingo, S. Sanvito and O. Levy, "The high-throughput highway to computational materials design", Nature materials **12**, 3, 191–201 (2013).

Dahl, G. E., T. N. Sainath and G. E. Hinton, “Improving deep neural networks for lvcsr using rectified linear units and dropout”, in “Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on”, pp. 8609–8613 (IEEE, 2013).

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database”, in “CVPR09”, (2009).

Denil, M., P. Agrawal, T. D. Kulkarni, T. Erez, P. Battaglia and N. de Freitas, “Learning to perform physics experiments via deep reinforcement learning”, arXiv preprint arXiv:1611.01843 (2016).

Dosovitskiy, A. and T. Brox, “Inverting convolutional networks with convolutional networks”, arXiv preprint arXiv:1506.02753 (2015).

Finn, C. and S. Levine, “Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm”, in “International Conference on Learning Representations”, (2018), URL `https://openreview.net/forum?id=HyjC5yWCW`.

Fukushima, K., “Neural network model for a mechanism of pattern recognition unaffected by shift in position- neocognitron”, Electron. & Commun. Japan **62**, 10, 11–18 (1979).

Fullwood, D., S. Kalidindi, S. Niezgoda, A. Fast and N. Hampson, “Gradient-based microstructure reconstructions from distributions using fast fourier transforms”, Materials Science and Engineering: A **494**, 1, 68–72 (2008a).

Fullwood, D. T., S. R. Niezgoda and S. R. Kalidindi, “Microstructure reconstructions from 2-point statistics using phase-recovery algorithms”, Acta Materialia **56**, 5, 942–948 (2008b).

Ganin, Y., E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand and V. Lempitsky, “Domain-adversarial training of neural networks”, The Journal of Machine Learning Research **17**, 1, 2096–2030 (2016).

Gatys, L. A., A. S. Ecker and M. Bethge, “Image style transfer using convolutional neural networks”, in “Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition”, pp. 2414–2423 (2016).

Giryes, R., Y. C. Eldar, A. Bronstein and G. Sapiro, “Tradeoffs between convergence speed and reconstruction accuracy in inverse problems”, IEEE Transactions on Signal Processing (2018).

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, “Generative adversarial nets”, in “Advances in Neural Information Processing Systems”, pp. 2672–2680 (2014).

Graham, B., “Fractional max-pooling”, arXiv preprint arXiv:1412.6071 (2014).

Grigoriu, M., “Random field models for two-phase microstructures”, Journal of Applied Physics **94**, 6, 3762–3770 (2003).

Gruber, J., A. Rollett and G. Rohrer, “Misorientation texture development during grain growth. Part II: Theory”, Acta Materialia **58**, 1, 14–19 (2010).

Guo, T., D. J. Lohan, R. Cang, M. Y. Ren and J. T. Allison, “An indirect design representation for topology optimization using variational autoencoder and style transfer”, in “2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference”, p. 0804 (2018).

Hajizadeh, A., A. Safekordi and F. A. Farhadpour, “A multiple-point statistics algorithm for 3D pore space reconstruction from 2D images”, Advances in water Resources **34**, 10, 1256–1267 (2011).

Hautier, G., C. Fischer, V. Ehrlacher, A. Jain and G. Ceder, “Data mined ionic substitutions for the discovery of new compounds”, Inorganic chemistry **50**, 2, 656–663 (2010).

He, K., X. Zhang, S. Ren and J. Sun, “Deep residual learning for image recognition”, in “Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition”, pp. 770–778 (2016).

Hinton, G., L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups”, IEEE Signal Processing Magazine **29**, 6, 82–97 (2012).

Hinton, G. E., “Training products of experts by minimizing contrastive divergence”, Neural computation **14**, 8, 1771–1800 (2002).

Hochreiter, S., A. S. Younger and P. R. Conwell, “Learning to learn using gradient descent”, in “International Conference on Artificial Neural Networks”, pp. 87–94 (Springer, 2001).

Hollister, S. J., “Porous scaffold design for tissue engineering”, Nature materials **4**, 7, 518–524 (2005).

Holotescu, S. and F. Stoian, “Prediction of particle size distribution effects on thermal conductivity of particulate composites”, Materialwissenschaft Und Werkstofftechnik **42**, 5, 379–385 (2011).

Hsiao, S.-W. and C.-H. Chen, “A semantic and shape grammar based approach for product design”, Design studies **18**, 3, 275–296 (1997).

Huan, T. D., A. Mannodi-Kanakkithodi, C. Kim, V. Sharma, G. Pilania and R. Ramprasad, “A polymer dataset for accelerated property prediction and design”, Scientific data **3** (2016).

Huang, S., N. Papernot, I. Goodfellow, Y. Duan and P. Abbeel, “Adversarial attacks on neural network policies”, arXiv preprint arXiv:1702.02284 (2017).

Hubel, D. H. and T. N. Wiesel, “Receptive fields and functional architecture of monkey striate cortex”, The Journal of physiology **195**, 1, 215–243 (1968).

Ioffe, S. and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift”, arXiv preprint arXiv:1502.03167 (2015).

Jain, A., S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder *et al.*, “Commentary: The materials project: A materials genome approach to accelerating materials innovation”, Apl Materials **1**, 1, 011002 (2013).

Jiang, Z., W. Chen and C. Burkhart, “Efficient 3D porous microstructure reconstruction via gaussian random field and hybrid optimization”, Journal of microscopy **252**, 2, 135–148 (2013).

Jiao, Y., E. Padilla and N. Chawla, “Modeling and predicting microstructure evolution in lead/tin alloy via correlation functions and stochastic material reconstruction”, Acta Materialia **61**, 9, 3370–3377 (2013).

Jiao, Y., F. Stillinger and S. Torquato, “Modeling heterogeneous materials via two-point correlation functions. ii. algorithmic details and applications”, Physical Review E **77**, 3, 031135 (2008).

Jiao, Y., F. Stillinger and S. Torquato, “A superior descriptor of random textures and its predictive capacity”, Proceedings of the National Academy of Sciences **106**, 42, 17634–17639 (2009).

Jiao, Y. and S. Torquato, “Quantitative characterization of the microstructure and transport properties of biopolymer networks”, Physical biology **9**, 3, 036009 (2012).

Jonschkowski, R., R. Hafner, J. Scholz and M. Riedmiller, “Pves: Position-velocity encoders for unsupervised learning of structured state representations”, arXiv preprint arXiv:1705.09805 (2017).

Jung, D. and H. C. Gea, “Topology optimization of nonlinear structures”, Finite Elements in Analysis and Design **40**, 11, 1417–1427 (2004).

Kaczmarowski, A., S. Yang, I. Szlufarska and D. Morgan, “Genetic algorithm optimization of defect clusters in crystalline materials”, Computational Materials Science **98**, 234–244 (2015).

Kalidindi, S. R. and M. De Graef, “Materials data science: current status and future outlook”, Annual Review of Materials Research **45**, 171–193 (2015).

Karasek, L. and M. Sumita, “Characterization of dispersion state of filler and polymer-filler interactions in rubber carbon black composites”, Mater. Sci. **31**, 281–289 (1996).

Karsanina, M. V., K. M. Gerke, E. B. Skvortsova and D. Mallants, “Universal spatial correlation functions for describing and reconstructing soil microstructure”, PloS one **10**, 5, e0126515 (2015).

Kharmanda, G., N. Olhoff, A. Mohamed and M. Lemaire, “Reliability-based topology optimization”, Structural and Multidisciplinary Optimization **26**, 5, 295–307 (2004).

Kingma, D. and J. Ba, “Adam: A method for stochastic optimization”, arXiv preprint arXiv:1412.6980 (2014).

Kingma, D. P. and M. Welling, “Auto-encoding variational bayes”, in “Proceedings of the 2nd International Conference on Learning Representations, Arizona, US”, (ICLP, 2013).

Kirklin, S., J. E. Saal, V. I. Hegde and C. Wolverton, “High-throughput computational search for strengthening precipitates in alloys”, Acta Materialia **102**, 125–135 (2016).

Klaysom, C., S.-H. Moon, B. P. Ladewig, G. M. Lu and L. Wang, “The effects of aspect ratio of inorganic fillers on the structure and property of composite ion-exchange membranes”, Journal of colloid and interface science **363**, 2, 431–439 (2011).

Kondo, R., S. Yamakawa, Y. Masuoka, S. Tajima and R. Asahi, “Microstructure recognition using convolutional neural networks for prediction of ionic conductivity in ceramics”, Acta Materialia (2017).

Krizhevsky, A., I. Sutskever and G. E. Hinton, “ImageNet classification with deep convolutional neural networks”, in “Advances in neural information processing systems, Nevada, US”, pp. 1097–1105 (2012).

Kurakin, A., I. Goodfellow and S. Bengio, “Adversarial examples in the physical world”, arXiv preprint arXiv:1607.02533 (2016).

Le, Q. V. *et al.*, “A tutorial on deep learning part 2: Autoencoders, convolutional neural networks and recurrent neural networks”, Google Brain (2015).

LeCun, Y., Y. Bengio and G. Hinton, “Deep learning”, Nature **521**, 7553, 436–444 (2015).

LeCun, Y., B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard and L. D. Jackel, “Handwritten digit recognition with a back-propagation network”, in “Advances in neural information processing systems”, pp. 396–404 (1990).

Lee, H., R. Grosse, R. Ranganath and A. Y. Ng, “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations”, in “Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, Canada”, pp. 609–616 (ACM, 2009).

Lei, N., Z. Luo, S.-T. Yau and D. X. Gu, “Geometric understanding of deep learning”, arXiv preprint arXiv:1805.10451 (2018).

Levine, S., C. Finn, T. Darrell and P. Abbeel, “End-to-end training of deep visuomotor policies”, arXiv preprint arXiv:1504.00702 (2015).

Lewis, D. D. and J. Catlett, “Heterogeneous uncertainty sampling for supervised learning”, in “Machine Learning Proceedings 1994”, pp. 148–156 (Elsevier, 1994).

Li, H., N. Chawla and Y. Jiao, “Reconstruction of heterogeneous materials via stochastic optimization of limited-angle x-ray tomographic projections”, Scripta Materialia **86**, 48–51 (2014).

Li, H., P.-E. Chen and Y. Jiao, “Accurate reconstruction of porous materials via stochastic fusion of limited bimodal microstructural data”, Transport in Porous Media pp. 1–18 (2017).

Li, H., S. Kaira, J. Mertens, N. Chawla and Y. Jiao, “Accurate stochastic reconstruction of heterogeneous microstructures by limited x-ray tomographic projections”, Journal of Microscopy **264**, 3, 339–350 (2016).

Li, K. and J. Malik, “Learning to optimize”, arXiv preprint arXiv:1606.01885 (2016).

Ling, J., M. Hutchinson, E. Antono, S. Paradiso and B. Meredig, “High-dimensional materials and process optimization using data-driven experimental design with well-calibrated uncertainty estimates”, arXiv preprint arXiv:1704.07423 (2017).

Liu, X. and V. Shapiro, “Random heterogeneous materials via texture synthesis”, Computational Materials Science **99**, 177–189 (2015).

Liu, X. and S. Vadim, “Random heterogeneous materials via texture synthesis”, Computational Materials Science **99**, 177–189 (2015).

Liu, Y., M. S. Greene, W. Chen, D. A. Dikin and W. K. Liu, “Computational microstructure characterization and reconstruction for stochastic multiscale material design”, Computer-Aided Design **45**, 1, 65–76 (2013).

Lookman, T., F. J. Alexander and A. R. Bishop, “Perspective: Codesign for materials science: An optimal learning approach”, APL Materials **4**, 5, 053501 (2016).

Ma, R., V. Sharma, A. F. Baldwin, M. Tefferi, I. Offenbach, M. Cakmak, R. Weiss, Y. Cao, R. Ramprasad and G. A. Sotzing, “Rational design and synthesis of polythioureas as capacitor dielectrics”, Journal of Materials Chemistry A **3**, 28, 14845–14852 (2015).

MacQueen, J. *et al.*, “Some methods for classification and analysis of multivariate observations”, in “Proceedings of the fifth Berkeley symposium on mathematical statistics and probability”, vol. 1, pp. 281–297 (Oakland, CA, USA., 1967).

Mahendran, A. and A. Vedaldi, “Understanding deep image representations by inverting them”, in “Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on Bostion, US”, pp. 5188–5196 (IEEE, 2015).

Mallick, P. K., *Fiber-reinforced composites: materials, manufacturing, and design* (CRC press, 2007).

Matthews, J., T. Klatt, C. Morris, C. C. Seepersad, M. Haberman and D. Shahan, "Hierarchical design of negative stiffness metamaterials using a bayesian network classifier", Journal of Mechanical Design **138**, 4, 041404 (2016).

McDowell, D. L. and G. Olson, "Concurrent design of hierarchical materials and structures", in "Scientific Modeling and Simulations", pp. 207–240 (Springer, 2008a).

McDowell, D. L. and G. B. Olson, "Concurrent design of hierarchical materials and structures", Scientific Modeling and Simulations **15**, 207–240 (2008b).

Milliken, K. L. and S. E. Laubach, "Brittle deformation in sandstone diagenesis as revealed by scanned cathodoluminescence imaging with application to characterization of fractured reservoirs", in "Cathodoluminescence in geosciences", pp. 225–243 (Springer, 2000).

Mishra, N., M. Rohaninejad, X. Chen and P. Abbeel, "A simple neural attentive meta-learner", (2018).

Mnih, V., K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra and M. Riedmiller, "Playing Atari with deep reinforcement learning", arXiv preprint arXiv:1312.5602 (2013).

Narayanan, H. and S. Mitter, "Sample complexity of testing the manifold hypothesis", in "Advances in Neural Information Processing Systems", pp. 1786–1794 (2010).

Ng, A., "Sparse autoencoder", CS294A Lecture notes **72**, 2011, 1–19 (2011).

Nguyen, A., J. Yosinski and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images", in "Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on Boston, US", pp. 427–436 (IEEE, 2015).

Okabe, H. and M. J. Blunt, "Pore space reconstruction using multiple-point statistics", Journal of Petroleum Science and Engineering **46**, 1, 121–137 (2005).

Orbay, G., L. Fu and L. B. Kara, "Deciphering the influence of product shape on consumer judgments through geometric abstraction", Journal of Mechanical Design **137**, 8, 081103 (2015).

Øren, P.-E. and S. Bakke, "Reconstruction of berea sandstone and pore-scale modelling of wettability effects", Journal of Petroleum Science and Engineering **39**, 3, 177–199 (2003).

Ostoja-Starzewski, M., "Random field models of heterogeneous materials", International Journal of Solids and Structures **35**, 19, 2429–2455 (1998).

Pan, S. J. and Q. Yang, "A survey on transfer learning", IEEE Transactions on knowledge and data engineering **22**, 10, 1345–1359 (2010).

Panchal, J. H., S. R. Kalidindi and D. L. McDowell, "Key computational modeling issues in integrated computational materials engineering", Computer-Aided Design **45**, 1, 4–25 (2013).

Prasad, A. M., L. R. Iverson and A. Liaw, "Newer classification and regression tree techniques: bagging and random forests for ecological prediction", Ecosystems **9**, 2, 181–199 (2006).

Pytz, R., "Microstructure description of composites, statistical methods, mechanics of microstructure materials", CISM Courses and Lectures (2004).

Querin, O., G. Steven and Y. Xie, "Evolutionary structural optimisation (eso) using a bidirectional algorithm", Engineering computations **15**, 8, 1031–1048 (1998).

Quiblier, J. A., "A new three-dimensional modeling technique for studying porous media", Journal of Colloid and Interface Science **98**, 1, 84–102 (1984).

Radlinski, A., M. Ioannidis, A. Hinde, M. Hainbuchner, M. Baron, H. Rauch and S. Kline, "Angstrom-to-millimeter characterization of sedimentary rock microstructure", Journal of colloid and interface science **274**, 2, 607–612 (2004).

Rao, R. V., V. J. Savsani and D. Vakharia, "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems", Computer-Aided Design **43**, 3, 303–315 (2011).

Ravi, S. and H. Larochelle, "Optimization as a model for few-shot learning", (2016).

Reed, S. E., Y. Zhang, Y. Zhang and H. Lee, "Deep visual analogy-making", in "Advances in Neural Information Processing Systems", pp. 1252–1260 (2015).

Ren, Y. and P. Y. Papalambros, "A design preference elicitation query as an optimization process", Journal of Mechanical Design **133**, 11, 111004 (2011).

Rifai, S., P. Vincent, X. Muller, X. Glorot and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction", in "Proceedings of the 28th international conference on machine learning (ICML-11)", pp. 833–840 (2011).

Rintoul, M. D. and S. Torquato, "Reconstruction of the structure of dispersions", Journal of Colloid and Interface Science **186**, 2, 467–476 (1997).

Roberts, A. P., "Statistical reconstruction of three-dimensional porous media from two-dimensional images", Physical Review E **56**, 3, 3203 (1997).

Rollett, A. D., S.-B. Lee, R. Campman and G. Rohrer, "Three-dimensional characterization of microstructure by electron back-scatter diffraction", Annu. Rev. Mater. Res. **37**, 627–658 (2007).

Roy, N. and A. McCallum, "Toward optimal active learning through monte carlo estimation of error reduction", ICML, Williamstown pp. 441–448 (2001).

Rumelhart, D. E., G. E. Hinton and R. J. Williams, "Learning representations by back-propagating errors", Cognitive modeling, DOI:10.1038/323533a0 **5**, 3, 1 (1988).

Sabour, S., N. Frosst and G. E. Hinton, “Dynamic routing between capsules”, in “Advances in Neural Information Processing Systems”, pp. 3859–3869 (2017).

Sahimi, M., *Flow and transport in porous media and fractured rock: from classical methods to modern approaches* (John Wiley and Sons, 2011).

Santoro, A., S. Bartunov, M. Botvinick, D. Wierstra and T. Lillicrap, “Meta-learning with memory-augmented neural networks”, in “International conference on machine learning”, pp. 1842–1850 (2016).

Scalon, J., N. Fieller, E. Stillman and H. Atkinson, “Spatial pattern analysis of second-phase particles in composite materials”, Materials Science and Engineering: A **356**, 1, 245–257 (2003).

Schmidhuber, J., *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*, Ph.D. thesis, Technische Universität München (1987).

Schmidhuber, J., “Deep learning in neural networks: An overview”, Neural Networks **61**, 85–117 (2015).

Schölkopf, B. and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond* (MIT press, 2002).

Settles, B., “Active learning literature survey”, University of Wisconsin, Madison **52**, 55-66, 11 (2010).

Settles, B., “Active learning”, Synthesis Lectures on Artificial Intelligence and Machine Learning **6**, 1, 1–114 (2012).

Sharma, V., C. Wang, R. G. Lorenzini, R. Ma, Q. Zhu, D. W. Sinkovits, G. Pilania, A. R. Oganov, S. Kumar and G. A. Sotzing, “Rational design of all organic polymer dielectrics”, Nature communications **5** (2014).

Sigmund, O., “Design of multiphysics actuators using topology optimization–part ii: Two-material structures”, Computer methods in applied mechanics and engineering **190**, 49-50, 6605–6627 (2001).

Simonyan, K. and A. Zisserman, “Very deep convolutional networks for large-scale image recognition”, arXiv preprint arXiv:1409.1556 (2014a).

Simonyan, K. and A. Zisserman, “Very deep convolutional networks for large-scale image recognition”, CoRR **abs/1409.1556** (2014b).

Sohn, K. and H. Lee, “Learning invariant representations with local transformations”, arXiv preprint arXiv:1206.6418 (2012).

Sosnovik, I. and I. Oseledets, “Neural networks for topology optimization”, arXiv preprint arXiv:1709.09578 (2017).

Steinzig, M. and F. Harlow, “Probability distribution function evolution for binary alloy solidification”, in “Solidification, Proceedings of the Minerals, Metals, Materials Society Annual Meeting, San Diego, CA”, pp. 197–206 (Citeseer, 1999).

Stewart, R. and S. Ermon, “Label-free supervision of neural networks with physics and domain knowledge.”, in “AAAI”, pp. 2576–2582 (2017).

Stiny, G., “Introduction to shape and shape grammars”, Environment and planning B: planning and design **7**, 3, 343–351 (1980).

Stöcker, M., “Biofuels and biomass-to-liquid fuels in the biorefinery: Catalytic conversion of lignocellulosic biomass using porous materials”, Angewandte Chemie International Edition **47**, 48, 9200–9211 (2008).

Sundararaghavan, V., “Reconstruction of three-dimensional anisotropic microstructures from two-dimensional micrographs imaged on orthogonal planes”, Integrating Materials and Manufacturing Innovation **3(1)**, 19 (2014).

Sundararaghavan, V. and N. Zabaras, “Classification and reconstruction of three-dimensional microstructures using support vector machines”, Computational Materials Science **32**, 2, 223–239 (2005).

Svanberg, K., “The method of moving asymptotesa new method for structural optimization”, International journal for numerical methods in engineering **24**, 2, 359–373 (1987).

Szegedy, C., W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow and R. Fergus, “Intriguing properties of neural networks”, arXiv preprint arXiv:1312.6199 (2013).

Tahmasebi, P. and A. Hezarkhani, “A fast and independent architecture of artificial neural network for permeability prediction”, Journal of Petroleum Science and Engineering **86**, 118–126 (2012).

Tahmasebi, P. and M. Sahimi, “Cross-correlation function for accurate reconstruction of heterogeneous media”, Physical review letters **110**, 7, 078002 (2013).

Tahmasebi, P. and M. Sahimi, “Reconstruction of nonstationary disordered materials and media: Watershed transform and cross-correlation function”, Physical Review E **91**, 3, 032401 (2015).

Tang, Y., “Deep learning using linear support vector machines”, arXiv preprint arXiv:1306.0239 (2013).

Tewari, A. and A. Gokhale, “Nearest-neighbor distances between particles of finite size in three-dimensional uniform random microstructures”, Materials Science and Engineering: A **385**, 1, 332–341 (2004).

Thrun, S., “Is learning the n-th thing any easier than learning the first”, in “Advances in neural information processing systems”, pp. 640–646 (1996).

Tompson, J., K. Schlachter, P. Sprechmann and K. Perlin, “Accelerating Eulerian Fluid Simulation With Convolutional Networks”, ArXiv e-prints (2016).

Tong, S. and E. Chang, “Support vector machine active learning for image retrieval”, in “Proceedings of the ninth ACM international conference on Multimedia”, pp. 107–118 (ACM, 2001).

Tong, S. and D. Koller, “Support vector machine active learning with applications to text classification”, Journal of machine learning research **2**, Nov, 45–66 (2001).

Torquato, S., “Effective electrical conductivity of two-phase disordered composite media”, Journal of Applied Physics **58**, 10, 3790–3797 (1985).

Torquato, S., “Exact expression for the effective elastic tensor of disordered composites”, Physical review letters **79**, 4, 681 (1997).

Torquato, S., *Random heterogeneous materials: microstructure and macroscopic properties*, vol. 16 (Springer Science & Business Media, New York, US, 2013).

Torquato, S. and B. Lu, “Rigorous bounds on the fluid permeability: effect of polydispersivity in grain size”, Physics of Fluids A: Fluid Dynamics **2**, 4, 487–490 (1990).

Ulu, E., R. Zhang and L. B. Kara, “A data-driven investigation and estimation of optimal topologies under variable loading configurations”, Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization **4**, 2, 61–72 (2016).

Venter, J. C., M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, G. G. Sutton, H. O. Smith, M. Yandell, C. A. Evans, R. A. Holt *et al.*, “The sequence of the human genome”, science **291**, 5507, 1304–1351 (2001).

Vilalta, R. and Y. Drissi, “A perspective view and survey of meta-learning”, Artificial Intelligence Review **18**, 2, 77–95 (2002).

Vincent, P., H. Larochelle, Y. Bengio and P.-A. Manzagol, “Extracting and composing robust features with denoising autoencoders”, in “Proceedings of the 25th international conference on Machine learning”, pp. 1096–1103 (ACM, 2008).

Vincent, P., H. Larochelle, I. Lajoie, Y. Bengio and P.-A. Manzagol, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion”, Journal of Machine Learning Research **11**, Dec, 3371–3408 (2010).

Wadia, C., A. P. Alivisatos and D. M. Kammen, “Materials availability expands the opportunity for large-scale photovoltaics deployment”, Environmental science & technology **43**, 6, 2072–2077 (2009).

Wang, J. X., Z. Kurth-Nelson, D. Tirumala, H. Soyer, J. Z. Leibo, R. Munos, C. Blundell, D. Kumaran and M. Botvinick, “Learning to reinforcement learn”, arXiv preprint arXiv:1611.05763 (2016).

Wang, M., F. Sha and M. I. Jordan, "Unsupervised kernel dimension reduction", in "Advances in Neural Information Processing Systems 23", edited by J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel and A. Culotta, pp. 2379–2387 (Curran Associates, Inc., 2010), URL `http://papers.nips.cc/paper/4122-unsupervised-kernel-dimension-reduction.pdf`.

Wang, M. Y. and X. Wang, ""color" level sets: a multi-phase method for structural topology optimization with multiple materials", Computer Methods in Applied Mechanics and Engineering **193**, 6-8, 469–496 (2004).

Wijnhoven, R. G. and P. de With, "Fast training of object detection using stochastic gradient descent", in "Pattern Recognition (ICPR), 2010 20th International Conference on", pp. 424–427 (IEEE, 2010).

Wool, R. and X. S. Sun, *Bio-based polymers and composites* (Academic Press, 2011).

Wu, J., N. Aage, R. Westermann and O. Sigmund, "Infill optimization for additive manufacturingapproaching bone-like porous structures", IEEE transactions on visualization and computer graphics **24**, 2, 1127–1140 (2018).

Wu, J., J. J. Lim, H. Zhang, J. B. Tenenbaum and W. T. Freeman, "Physics 101: Learning physical object properties from unlabeled videos.", in "BMVC", vol. 2, p. 7 (2016).

Xie, Y. M. and G. P. Steven, "A simple evolutionary procedure for structural optimization", Computers & structures **49**, 5, 885–896 (1993).

Xing, C., L. Ma and X. Yang, "Stacked denoise autoencoder based feature extraction and classification for hyperspectral images", Journal of Sensors **2016** (2015).

Xu, H., D. A. Dikin, C. Burkhart and W. Chen, "Descriptor-based methodology for statistical characterization and 3D reconstruction of microstructural materials", Computational Materials Science **85**, 206–216 (2014).

Xu, H., R. Liu, A. Choudhary and W. Chen, "A machine learning-based design representation method for designing heterogeneous microstructures", Journal of Mechanical Design **137**, 5, 051403 (2015).

Y. Jiao, E. P. and N. Chawla, "Modeling and predicting microstructure evolution in lead/tin alloy via correlation functions and stochastic material reconstruction", (2013).

Yan, X., J. Yang, K. Sohn and H. Lee, "Attribute2Image: Conditional image generation from visual attributes", arXiv preprint arXiv:1512.00570 (2015).

Yeong, C. and S. Torquato, "Reconstructing random media", Physical Review E **57**, 1, 495 (1998).

Yu, S., Y. Zhang, C. Wang, W.-k. Lee, B. Dong, T. W. Odom, C. Sun and W. Chen, "Characterization and design of functional quasi-random nanostructured materials using spectral density function", Journal of Mechanical Design **139**, 7, 071401 (2017).

Yumer, M. E., P. Asente, R. Mech and L. B. Kara, “Procedural modeling using autoencoder networks”, in “Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology”, pp. 109–118 (ACM, 2015).

Zhao, J., M. Mathieu and Y. LeCun, “Energy-based generative adversarial network”, arXiv preprint arXiv:1609.03126 (2016).

Zhu, B., M. Skouras, D. Chen and W. Matusik, “Two-scale topology optimization with microstructures”, ACM Transactions on Graphics (TOG) **36**, 5, 164 (2017).

Zinkevich, M., M. Weimer, L. Li and A. J. Smola, “Parallelized stochastic gradient descent”, in “Advances in neural information processing systems”, pp. 2595–2603 (2010).

Appendix A

Four different material systems input samples (Chapter3, Data collected in 2016)

There are four different material systems are tested with the proposed Convolutional Deep Belief Network. The training samples are provided:

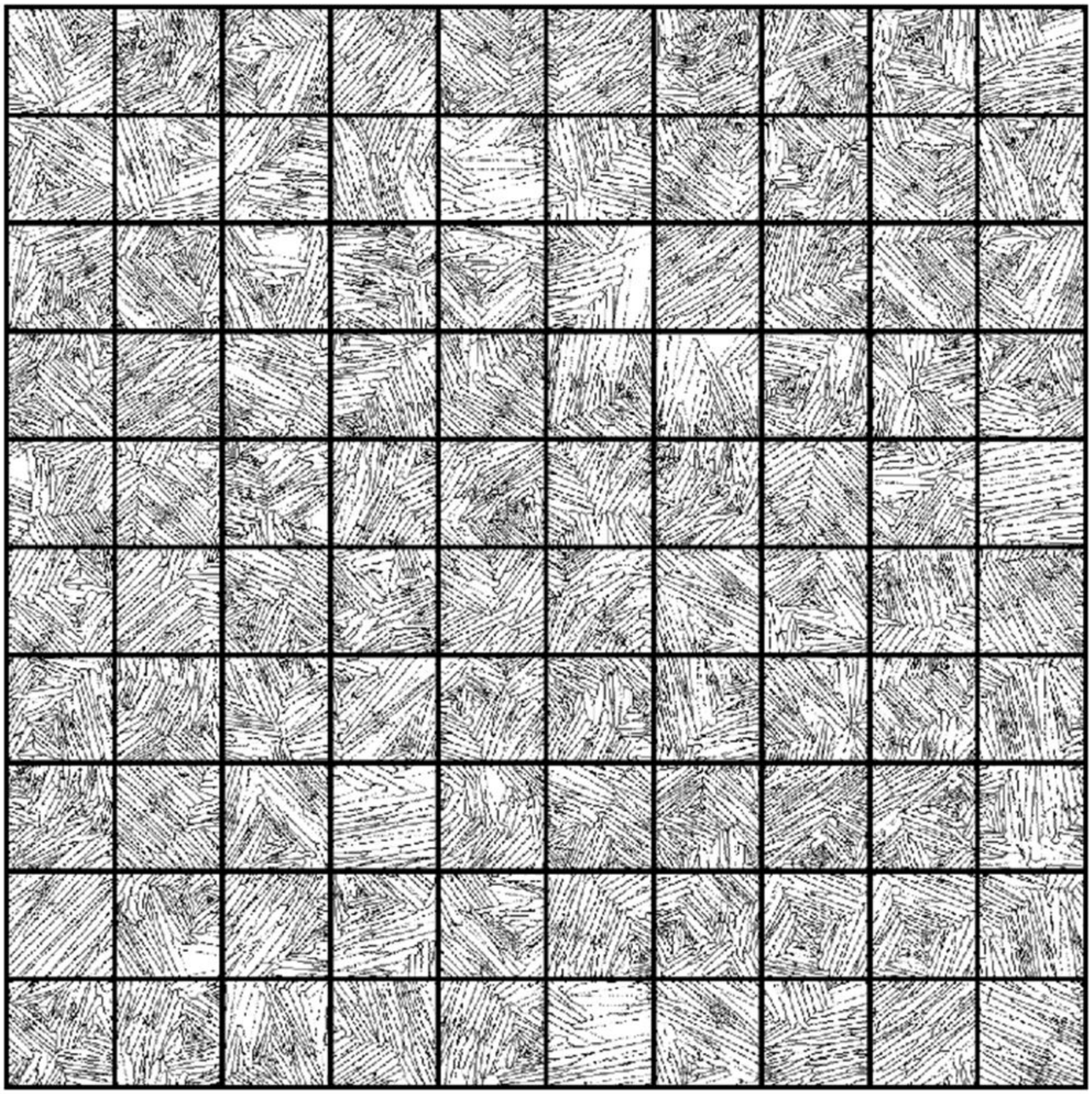


Figure 47: Ti-6Al-4V alloy

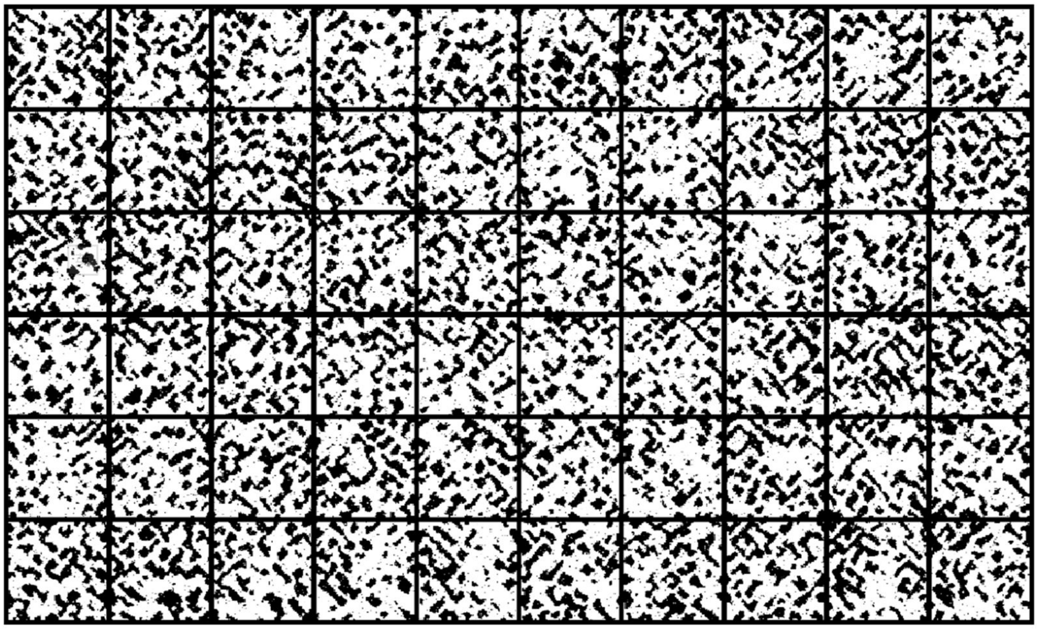


Figure 48: Pb-Sn (lead-tin) alloy

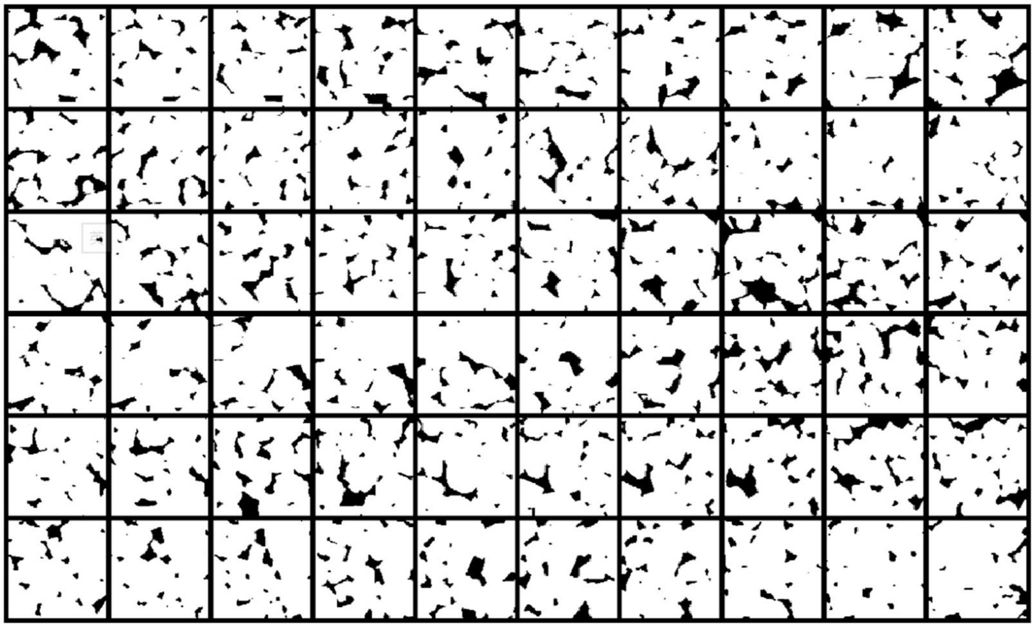


Figure 49: Pore structure of Fontainebleau sandstone

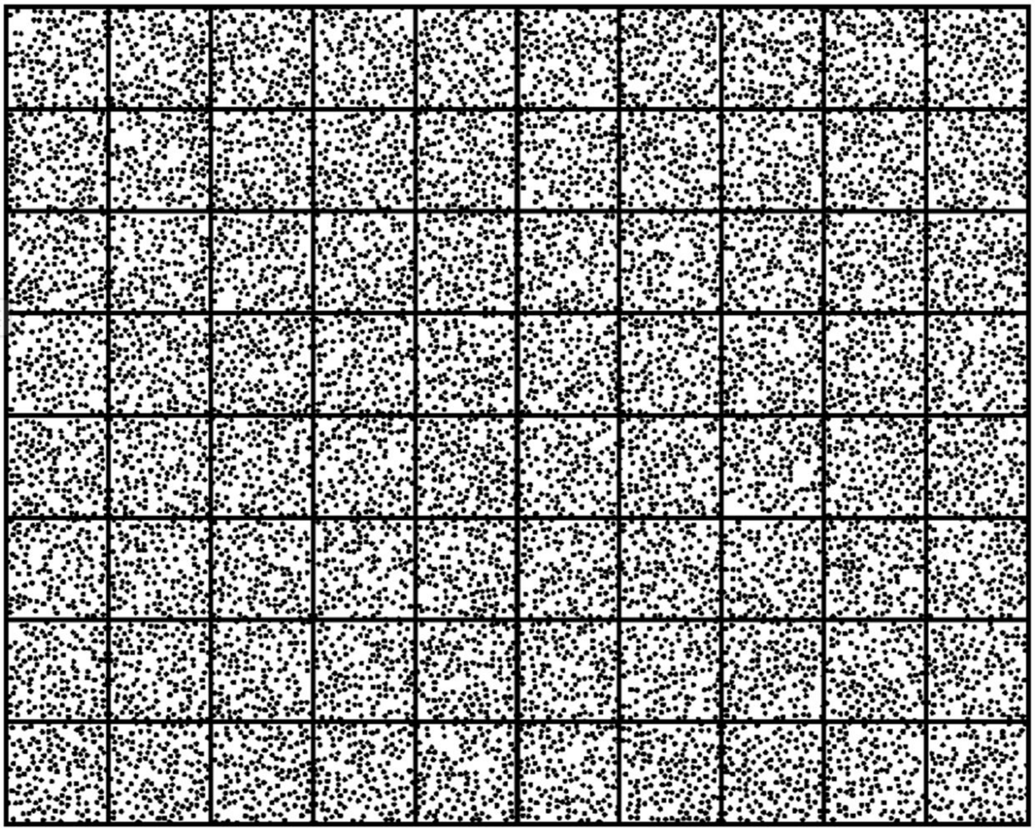


Figure 50: 2D suspension of spherical colloids

Appendix B

Topology optimization through augmented Lagrangian (Chapter 6)

The topology optimization problem of Eq. (TO) in Chapter6 is solved using Alg. 2. We choose this algorithm because it can be readily implemented in TensorFlow. Note that other algorithms, such as the method of moving asymptotes Svanberg (1987), are also applicable.

Algorithm 2: Topology optimization through Augmented Lagrangian (AL)

```
input  : Problem parameters $\alpha$, $\beta$, $p$, $r_e$, $R_e$, $\beta_t$, $\mathbf{s}$
output: Design variable $\mathbf{x}$
Set algorithmic parameters $\epsilon_{al} = 1$, $\epsilon_{opt} = 10^{-3}$, initial guess $\mathbf{x} = \alpha\mathbf{1}$;
Pre-compute neighbourhood $\mathcal{M}_e$ and $\mathcal{N}_e$, and filter weights $\omega_{i,j}$;
while $\beta < \beta_t$ do
    /* gradually change the problem */
    Set AL parameters: $r_0 = 1$, $r_1 = 1$, $\mu_0 = \mu_1 = 0$, $\eta_0 = \eta_1 = 0.1$, $\epsilon = 1$,
      $\delta\mathbf{x} = 10^6\mathbf{1}$;
    Compute $\tilde{\mathbf{x}}$, $\boldsymbol{\rho}$, $\bar{\boldsymbol{\rho}}$, $\mathbf{K}$, $\mathbf{u}$, $f$, $g_0$, and $g_1$;
    while $\max|\delta\boldsymbol{x}| > \epsilon_{al}$ or $g_0 > 0$ or $g_1 > 0$ do
        /* solve the constrained problem */
        Set $\delta\mathbf{x} = 10^6\mathbf{1}$;
        while $\max|\delta\boldsymbol{x}| > \epsilon$ do
            /* solve the unconstrained problem */
            Set learning rate $a = 10^{-3}$;
            Compute $\nabla_{\mathbf{x}}f$, $\nabla_{\mathbf{x}}g_0$, $\nabla_{\mathbf{x}}g_1$, and
              $\delta\mathbf{x} = \nabla_{\mathbf{x}}f + (\mu_0 + 2g_0/r_0)\nabla_{\mathbf{x}}g_0(g_0 > 0) + (\mu_1 + 2g_1/r_1)\nabla_{\mathbf{x}}g_1(g_1 > 0)$;
            while 1 do
                /* line search */
                Set $\Delta\mathbf{x} = -a\delta\mathbf{x}$, clip each element of $\Delta x$ to $[-0.1, 0.1]$;
                Set $\mathbf{x}' = \mathbf{x} + \Delta\mathbf{x}$;
                Compute $\boldsymbol{\rho}'$, $\bar{\boldsymbol{\rho}}'$, $\mathbf{K}'$, $\mathbf{u}'$, $f'$, $g_0'$, and $g_1'$ based on $\mathbf{x}'$;
                Compute $L = f + \mu_0 g_0 + 0.5g_0^2/r_0 + \mu_1 g_1 + 0.5g_1^2/r_1$;
                Compute $L' = f' + \mu_0 g_0' + 0.5g_0'^2/r_0 + \mu_1 g_1' + 0.5g_1'^2/r_1$;
                if $L' - L > 0$ then
                    /* if learning rate is too high */
                    Set $a = 0.5a$;
                else
                    $\mathbf{x} = \mathbf{x}'$;
                end
            end
        end
        /* update augmented Lagrangian parameters */
        for $i = 0, 1$ do
            if $g_i < \eta_i$ then
                Set $\mu_i = \mu_i + 2g_i/r_i$, $\eta_i = 0.5\eta_i$;
            else
                Set $r_i = 0.5r_i$;
            end
        end
    end
    Set $\beta = 2\beta$;
end
```