

Zijiang Yang¹

Department of Electrical Engineering
and Computer Science,
Northwestern University,
Evanston, IL 60208
e-mail: zijiangyang2016@u.northwestern.edu

Xiaolin Li¹

Theoretical and Applied Mechanics,
Northwestern University,
Evanston, IL 60208
e-mail: xiaolinli2018@u.northwestern.edu

L. Catherine Brinson

Department of Mechanical Engineering
and Materials Science,
Duke University,
Durham, NC 27708
e-mail: cate.brinson@duke.edu

Alok N. Choudhary

Department of Electrical Engineering and
Computer Science,
Northwestern University,
Evanston, IL 60208
e-mail: a-choudhary@northwestern.edu

Wei Chen²

Department of Mechanical Engineering,
Northwestern University,
Evanston, IL 60208
e-mail: weichen@northwestern.edu

Ankit Agrawal²

Department of Electrical Engineering and
Computer Science,
Northwestern University,
Evanston, IL 60208
e-mail: ankitag@eecs.northwestern.edu

Microstructural Materials Design Via Deep Adversarial Learning Methodology

Identifying the key microstructure representations is crucial for computational materials design (CMD). However, existing microstructure characterization and reconstruction (MCR) techniques have limitations to be applied for microstructural materials design. Some MCR approaches are not applicable for microstructural materials design because no parameters are available to serve as design variables, while others introduce significant information loss in either microstructure representation and/or dimensionality reduction. In this work, we present a deep adversarial learning methodology that overcomes the limitations of existing MCR techniques. In the proposed methodology, generative adversarial networks (GAN) are trained to learn the mapping between latent variables and microstructures. Thereafter, the low-dimensional latent variables serve as design variables, and a Bayesian optimization framework is applied to obtain microstructures with desired material property. Due to the special design of the network architecture, the proposed methodology is able to identify the latent (design) variables with desired dimensionality, as well as capturing complex material microstructural characteristics. The validity of the proposed methodology is tested numerically on a synthetic microstructure dataset and its effectiveness for microstructural materials design is evaluated through a case study of optimizing optical performance for energy absorption. Additional features, such as scalability and transferability, are also demonstrated in this work. In essence, the proposed methodology provides an end-to-end solution for microstructural materials design, in which GAN reduces information loss and preserves more microstructural characteristics, and the GP-Hedge optimization improves the efficiency of design exploration. [DOI: 10.1115/1.4041371]

Keywords: microstructural materials design, microstructural analysis, deep learning, generative adversarial network, Bayesian optimization, scalability, transfer learning

1 Introduction

To date, computational materials design (CMD) has revolutionarily changed the way advanced materials are developed [1–8]. In the plethora of successes in CMD [9–15], microstructure-sensitive design [16] has shown its significance in driving the rapid discovery and manufacturing of new materials. In designing material microstructures, the appropriate design representation of microstructures determines its ultimate success. A common practice of selecting microstructural design variables is to choose key microstructure characteristics from existing microstructure characterization and reconstruction techniques (MCR). A comprehensive review of existing MCR techniques is provided by Bostanaband et al. [17]. Together with some recent works using deep learning, the existing techniques are classified into the following categories:

- (1) Correlation function-based methods [18]
- (2) Physical descriptor-based methods [19]
- (3) Gaussian random field (GRF)-based methods [20]
- (4) Markovian random field-based methods [21]

(5) Deep belief network-based methods [22]

(6) Spectral density function-based methods [15], and

(7) Transfer learning-based methods [23,24]

However, not all existing MCR techniques are applicable for microstructural materials design. Two major limitations exist: (1) Some MCR methods (methods 3, 4, 5 and 7) are not applicable for microstructural materials design, because no parameters are available to serve as design variables for generating new microstructure designs. (2) While methods 1, 2, and 6 are applicable for microstructural materials design, their efficacy is limited by the potential information loss (i.e., loss of either dispersive or geometrical characteristics) in microstructure representation and/or dimensionality reduction. In microstructure representations, some approximations such as taking radial averages in method 1 and 6 or approximating cluster shapes with ellipses in method 2 could result in the loss of microstructural characteristics. Dimension reduction is often needed in microstructure optimization due to the high-dimensional representation of microstructures. A common practice is to conduct a transformation of microstructure representations (e.g., using principal component analysis (PCA)) and remove some insignificant dimensions. Information loss would also occur in the removal process. For instance, Paulson et al. [25] use spatial correlation function as the microstructure representation, and conduct a PCA transformation. It is shown in their work that removing some principal components could lead to a

¹Z. Yang and X. Li contributed equally to this work.

²Corresponding authors.

Contributed by the Design Automation Committee of ASME for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received March 24, 2018; final manuscript received July 25, 2018; published online October 1, 2018. Special Editor: Carolyn Seepersad.

significant reduction in explained structural variance. Another example is the use of descriptor-based approach. After obtaining the full list of descriptors, a supervised learning-based feature selection step is often used to remove the lower-ranked descriptors [26], wherein some geometric or higher order dispersive information is lost. It should be noted that the aforementioned dimensionality reduction techniques do not guarantee the capability of generating new microstructural designs using the reduced dimension. For example, while the principal components learned by PCA are capable of identifying new dimensions that are not linearly correlated, it is not clear how to generate a new microstructure by sampling in the learned principal dimensions.

Compared to the existing MCR techniques, generative models are promising alternatives to address the problems in microstructural materials design. Instead of identifying characteristics from microstructures, generative models emphasize the ability of using a low-dimensional latent variables Z to generate high-dimensional data X through a generative mapping $G: Z \rightarrow X$ to approximate the real data probability density $P_{\text{data}}(x)$. In other words, the evaluation criterion for generative models is whether it is capable of producing very realistic samples, which are indistinguishable from real samples. The latent variables learned in the generative model can therefore serve as design variables for microstructural materials design. In addition, generative models are especially powerful for microstructural materials design because the approach is model-based and it can rapidly generate new microstructures by changing the values of latent variables, while existing MCR approaches often need tedious optimization for microstructure reconstructions (e.g., simulated annealing is used in correlation function-based reconstruction).

In the realm of deep learning, variational auto-encoder (VAE) [27] and generative adversarial networks (GAN) [28,29] are two major categories of generative models. It is well recognized that VAE suffers from the issue of “maximum likelihood training paradigm” when combined with a conditional independence assumption on the output given the latent variables, and they tend to distribute probability mass diffusely over the data space and generate blurry samples [30]. Despite these theoretical disadvantages, both Cang et al. [31] and Guo et al. [32] developed VAE-based models for representing sandstone material microstructures and topology optimization, respectively. However, their generative capability is bottlenecked at images of size 40×40 , and it is impossible to scale up because fully connected layers are involved in their network architecture.

In contrast to VAE, GAN is a better choice to bypass these problems. Different from VAE, GAN identifies the latent variables of data by training a generator-discriminator model pair in adversarial manner. In Refs. [33] and [34], GAN is used for reconstructing different types of microstructures, but their applications in computational materials design are unexplored. In this work, as illustrated in Fig. 1, we apply a fully scalable GAN-based approach to determine the latent variables of a set of microstructures once its dimensionality is prespecified. The latent variables are then treated as design variables in microstructure optimization. Thereafter, the material property for the latent variables is obtained by propagating the latent variables through the generator in GAN, followed by physical simulations of structure–property or structure–performance relations. Considering that physical simulations are usually computationally costly, we also want to minimize the number of property evaluations. Therefore, we pursue a

response surface model-based GP-Hedge Bayesian optimization framework to optimize microstructure with desired material property/performance.

The proposed deep adversarial learning methodology provides an end-to-end solution that offers a low-dimensional and nonlinear embedding of microstructures for microstructural materials design. Compared with the existing methods, which cannot fully capture microstructural characteristics (e.g., two-point correlation function in method 1 and physical descriptors in method 2), the proposed method does not make any geometrical or dispersive approximations and thus there is no information loss. In addition, the nonlinear embedding of microstructures in the proposed method avoids the removal of insignificant dimensions of microstructure representations (e.g., physical descriptors in method 2 and principal components in method 1) so that more microstructural information is preserved. Moreover, the proposed method is also beneficial for microstructural materials design because the dimensionality of latent variables can be prespecified as needed. Meanwhile, since the GAN is implemented by deep neural networks with large model capacity, it is able to capture very complex microstructural characteristics. In addition to the contribution of the proposed approach to microstructural materials design, we also demonstrate that the proposed approach is advantageous in: (1) *scalability*: the proposed approach is capable of converting microstructures into reasonable and computationally affordable low-dimensional representations as needed, and the generator in proposed model is scalable to produce arbitrary sized microstructures; (2) *transferability*: the discriminator in the proposed approach could be reused to serve as a pretrained model to facilitate the development of structure–property predictive models. To the best of the authors’ knowledge, this work is the first that applies adversarial learning in computational design of materials microstructure.

In the remainder of this paper, we break our presentation of the deep adversarial learning design methodology into five sections. In the first part (Sec. 2—design representation), we present the technical fundamentals of the deep adversarial learning approach, and show how the latent variables of microstructures are learned using the proposed approach. The latent variables are then treated as design variables in the latter sections. In the second part (Sec. 3—design evaluation), we demonstrate how material properties are evaluated from design variables using the proposed model. This demonstration is then followed by Sec. 4—design synthesis, in which Gaussian Process metamodeling is used to create a surrogate response surface between the latent variables and the objective property/performance, and a GP-Hedge Bayesian optimization is applied to optimize the microstructure to achieve the target material property. After that, we elaborate two additional features of the proposed methodology—scalability, which provides flexibility in taking arbitrary sized input/output, and transferability, which makes it possible to utilize the trained weights to build a more accurate structure–property predictive model (Sec. 5). Last but not the least, we draw conclusions and discuss potential directions to further extend this proposed methodology.³

2 Microstructural Design Representation Using Deep Adversarial Learning

In the proposed methodology, the deep adversarial learning approach, specifically GAN, is first used to identify a set of latent variables as microstructure design variables based on microstructure images collected for the same material system. In this section, the fundamentals of GAN are first introduced. It is then followed by a presentation of the proposed network architecture and designated loss function. Finally, we specify some training details of the proposed deep adversarial learning model.

³The GAN model is available for download at: https://github.com/zyz293/GAN_Materials_Design

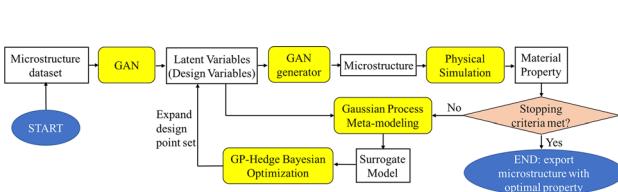


Fig. 1 The flowchart of the proposed design methodology

2.1 Fundamentals of Generative Adversarial Network.

Generative Adversarial Network is a type of deep generative neural network first proposed by Goodfellow et al. [28,29]. Originated from game theory, the training process of GAN is essentially a two-player competitive game. Specifically, GAN trains a generator network $G(z; \theta^{(G)})$ that produces samples x_G from latent variables z to approximate real samples x_{data} , and a discriminator network $D(x)$ that distinguishes the generated samples from the real samples. This competitive game would eventually lead to a Nash Equilibrium [35] between the generator G and the discriminator D . A more vivid analogy of GAN is given by Goodfellow et al. [28]: in this adversary scenario, the generator can be thought of a group of counterfeiters who tries to produce fake currency, while the discriminator is analogous to a team of police, trying to detect the counterfeit currency from the real money. Competitions in this adversary game would keep pushing both sides to the equilibrium in which the counterfeits are indistinguishable. When the generator is capable of producing realistic samples at the equilibrium, the latent variables z would be naturally taken as the “code” of the data. In the context of proposed generative microstructural design framework, the “code” will serve as the design variables to create new microstructure designs.

An illustration of GAN is shown in Fig. 2. The latent variable space is denoted as Z while the microstructure data space is represented by X . On the left hand side, to learn the generator distribution p_g that approximates the data distribution, a prior distribution of the latent variables is defined by $Z \sim p_z(z)$. z is then propagated through a deep neural network to create a differentiable mapping $G(z; \theta^{(G)})$ from the latent variable space Z to microstructure data space X . On the right-hand side, we also define a discriminator network that takes x , either generated or real microstructures, and produces a scalar label that indicates if x is from real data. In other words, we train discriminator (D) to maximize the probability of assigning the correct label to both real (label = 1) and generated samples (label = 0), while we train generator (G) to maximize the number of occurrences that the labels are incorrectly assigned by D . Essentially, D and G play a two-player minimax game, which can be expressed as the following equation:

$$\min_G \max_D V(D, G) = \mathbb{E}_{X \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (1)$$

2.2 Network Architecture. In this work, the architecture of deep convolutional generative adversarial network in Ref. [36] is adopted except that we use convolutional layers to replace the fully connected layers in both generator and discriminator for the sake of scalability (this will be introduced in Sec. 5). The generator and the discriminator have the same number of layers, and the number of the convolutional filters are aligned symmetrically in the generator and the discriminator. In the generator, the last deconvolutional layer is associated with a tanh activation function to produce images with bounded pixel values, while the other deconvolutional layers are attached with batch normalization operations [37] and rectified linear unit (ReLU) activations [38]. In the discriminator, the last convolutional layer has a sigmoid activation function appended to produce probabilities between 0 and 1, while the other convolutional layers are all associated with batch normalization operations [37] and leaky rectified linear unit (Leak ReLU) activations [39].

Figure 2 is a simple demonstration of the proposed architecture with five layers in both generator and discriminator. It should be noted that arbitrary number of layers could be applied in the proposed architecture, as long as the symmetry is kept.

2.3 Loss Function. While the optimality of GAN model is Nash equilibrium theoretically, in practice, the global optimality or sufficiently good local optimality is not guaranteed [40]. A common example of failure is the model collapse, in which the generator converges to a state that consistently produces identical samples. Therefore, in order to produce morphologically and statistically equivalent microstructures from the generator, we carefully design the loss function which can be generalized to different applications (Sec. 2.3) and training parameters (Sec. 2.4). Specifically, the total loss consists of three major components: (1) *adversarial loss* (aka. *GAN loss*) that combinatorially evaluates the performance of generator and discriminator, (2) *Style transfer loss* that imposes morphological constraints to the generated microstructures, and (3) *Model collapse loss* that prevents the training from collapsing.

2.3.1 Generative Adversarial Networks Adversarial Loss. The GAN Adversarial Loss is essentially the optimization objective in the vanilla version of GAN (Eq. (1)), expressed as

$$L_{\text{GAN}} = \mathbb{E}_{X \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (2)$$

Note again that, in the min-max training, $\min_G \max_D L_{\text{GAN}}$ essentially wants the generator G to minimize this loss and let D maximizes it. In practice, we follow [40] to alter the loss of $\min(\log(1 - D))$ to $\max(\log D)$ when optimizing G .

2.3.2 Style Transfer Loss. This loss essentially imposes morphology constraints to the generated samples. The style transfer loss, namely Gram-matrix loss, is originated from a work by Gatys et al. [41] for the purpose of texture synthesis. In the field of material science, Cang et al. [31] included the style transfer loss into the total loss function as a penalty term when training a Variational Auto-Encoder network [29]. In our early work, Li et al. [23] take the style transfer loss as an optimization objective and uses its gradients with respect to each entry in the microstructure image to reconstruct statistically equivalent microstructures. They also discover an interesting intrinsic relationship between the layers included in the calculation of style transfer loss and the reconstructed microstructure: higher level convolutional layers could be dropped to reduce the computational cost while preserving the reconstruction accuracy. Recognizing this intrinsic relationship, in this work, we only retain the first four lowest convolutional layers

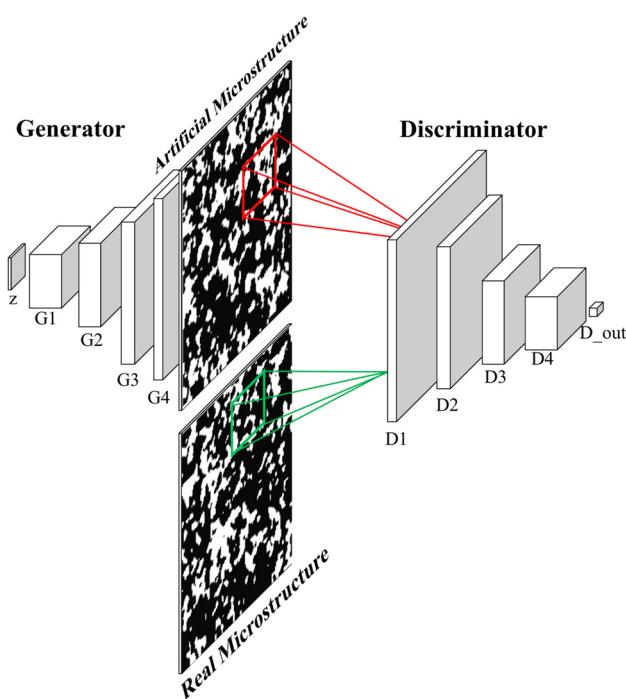


Fig. 2 The architecture of the proposed GAN

in the VGG-16 model [42] and compute their Gram-matrix as the style representations. The style transfer loss [41] can be expressed as

$$L_{\text{style}} = \sum_l \sum_{i,j} \frac{1}{4N_l^2 M_l^2} \left(G_{ij}^l - A_{ij}^l \right)^2 \quad (l = 1, 2, 3, 4) \quad (3)$$

which measures the distance between style representations of generated images and real images. In Eq. (3), N_l and M_l are number of feature maps and size of each feature map (i.e., height \times width) of the l th convolutional layer. G^l and A^l are the Gram-matrix of generated images and real images, respectively. The formula of Gram-matrix is

$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l \quad (4)$$

which calculates the inner product between the i th and j th vectorized feature maps of the l th convolutional layer.

2.3.3 Model Collapse Loss. Model collapse is a common problem of training a GAN model where the generated samples are clustered in only one or few modes of $p_{\text{data}}(x)$. Thus, model collapse loss [43]

$$L_{\text{collapse}} = \frac{1}{n(n-1)} \sum_i \sum_{j \neq i} \left(\frac{S_i^T S_j}{\|S_i\| \|S_j\|} \right)^2 \quad (5)$$

is introduced to prevent the training from getting into collapse mode. In this equation, n denotes the number of samples in a batch and S represents a batch of sample representations from outputs of the first four convolutional layers of VGG-16 model [42]. In other words, S is the concatenated vectorized feature maps of the first four convolutional layers of VGG-16 model [42].

2.3.4 The Total Loss. The total loss is a weighted combination of the three aforementioned losses

$$L(G, D) = L_{\text{GAN}} + \alpha L_{\text{style}} + \beta L_{\text{collapse}} \quad (6)$$

α and β are the moderating weights that prevent the style transfer loss and model collapse loss from diminishing to zero or overwhelming the GAN adversarial loss. The composition of loss

functions and the information flow in the proposed neural network architecture is depicted in Fig. 3.

2.4 Numerical Validation of Latent Variables. We apply the proposed deep adversarial learning approach to determine the latent variables for a dataset of material microstructures.

2.4.1 Training Data. To train the proposed GAN model, a dataset of material microstructure images that covers a variety of microstructural dispersions is required. In addition, it is also required that all the training microstructure images share the same size. In this work, to validate the proposed approach, 5000 synthetic microstructure images of size 128×128 are created using GRF method [20]. In order to reasonably cover the vast space of compositional and dispersive patterns that correspond to different processing conditions for the same material system, three parameters (mean, standard deviation, and volume fraction) are carefully controlled in the GRF model to produce microstructures with different dispersive status but sharing similar underlying characteristics of morphology. Figure 4 row 1 demonstrates some examples of the training microstructures. 5000 of these samples are used for training the GAN model. While 5000 seems to be an unrealistic number in material data gathering, we note that multiple images can be cropped from one microstructure image in practice. For example, for 1000×1000 sized microstructure images, thousands of 128×128 samples can be cropped with partial overlapping of the samples.

2.4.2 Network Architecture Specifications and Training Parameters. One advantage of the proposed methodology is the flexibility in assigning the dimension of latent variables. The generator network is essentially a mapping between latent variables and microstructure images, so the neural network architecture depends on both the dimensionality of latent variables prespecified and the size of microstructure images. Typically, lower dimensionality is desired for latent variables from the microstructural design perspective, because smaller number of design variables helps to reduce the computational cost in microstructure optimization. However, smaller dimensionality of latent variables will increase the depth of neural network or increase the stride parameter in the convolutional layers, which makes the training of GAN more difficult. Therefore, a trade-off between the latent variables' dimensionality and the training difficulty needs to be considered. After several experiments, it is discovered that the five-layer architecture with stride 2×2 as illustrated in Fig. 2 is practically easy to stabilize and converge while providing sufficiently low dimensionality for the latent variables. The 2×2 stride configuration essentially results in a scaling factor of 2 on each dimension in each layer, thus five stacked layers would scaling down the microstructures by a factor of 32 (i.e., 2^5) on each dimension. For the aforementioned dataset, the 128×128 images are converted to a 4×4 latent variable tensor, which is flattened to a 16-dimensional latent variable vector z .

In addition to the dimensionality of z , a bounded latent variable space is defined by setting each entry of z to be independent and uniformly distributed between -1 and 1 . For generator network, four (deconvolutional)-batch normalization-ReLU layers are appended to z sequentially, which is then followed by a (deconvolutional)-tanh layer to produce $128 \times 128 \times 1$ sized microstructure images. In contrast, the discriminator network is composed by four sequentially connected convolution-batch normalization-leaky ReLU layers. A convolutional-sigmoid layer is appended to the end of the discriminator network to produce a scalar valued between 0 and 1 to represent the probability of classifying if the image given to the discriminator is from real microstructure dataset (instead of artificially generated ones). A detailed specification of the dimensionality in each layer is illustrated in Table 1. Note that to achieve the specified dimensionality, in both deconvolutional and convolutional layers, the filter size is set as 4×4 and

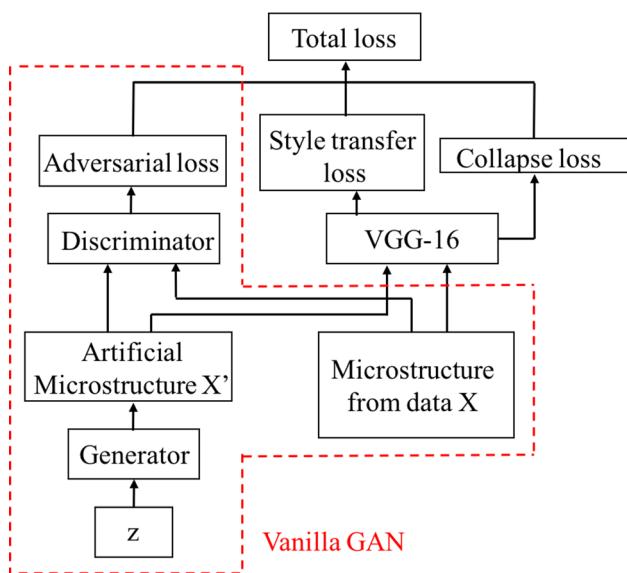


Fig. 3 The composition of loss function and information flow in the proposed architecture

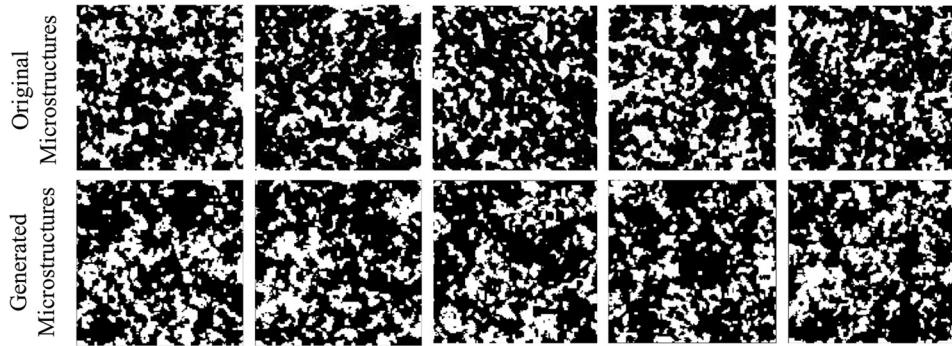


Fig. 4 Examples of original (training) microstructures and microstructures produced by the generator

Table 1 The dimensionality of each layer in the proposed network architecture. (bs. is the abbreviation of batch size)

Layer	Dimension
Random tensor z	$bs. \times 4 \times 4 \times 1$
Generator layer 1	$bs. \times 8 \times 8 \times 128$
Generator layer 2	$bs. \times 16 \times 16 \times 64$
Generator layer 3	$bs. \times 32 \times 32 \times 32$
Generator layer 4	$bs. \times 64 \times 64 \times 16$
Image X	$bs. \times 128 \times 128 \times 1$
Discriminator layer 1	$bs. \times 64 \times 64 \times 16$
Discriminator layer 2	$bs. \times 32 \times 32 \times 32$
Discriminator layer 3	$bs. \times 16 \times 16 \times 64$
Discriminator layer 4	$bs. \times 8 \times 8 \times 128$
Discriminator layer 5	$bs. \times 1 \times 1 \times 1$

strides are all 2×2 (The only exception is that we use 8×8 filter with stride 1×1 between discriminator layers 4 and 5).

The α and β parameters discussed in Sec. 2.3 are set as 0.03 and 0.03 for optimal balance between the three components of losses, respectively. Adam optimizer [44] is applied in training by setting the learning rate as 0.0005, β_1 value as 0.5 and β_2 value as 0.99. In the alternating training of the generator G and the discriminator D , it is found that it is optimal to set the ratio of network optimization for discriminator and generator to 3:1 (i.e., update discriminator three times and then update generator once) to achieve stability and convergence.

Some other significant training parameters include: number of epochs – 15,000; batch size – 30 and the α parameter in leaky ReLU – 0.2.

2.4.3 Validation of the Latent Variables. The validity of the latent variables and the amount of information loss are evaluated by comparing the original microstructure set and a set of microstructures produced by randomly sampling latent variables z and propagating through the generator network. Specifically, we compare the two-point correlation functions [17,18] and lineal-path correlation functions [45] of the 5000 original microstructures and 5000 generated ones produced by the generative model trained in GAN. Figure 4 shows that the generator in GAN is capable of producing visually similar microstructures as the original image data used for training. Figure 5 shows the two-point and lineal-path correlation functions of original microstructures and microstructures generated by the proposed generator. Figures 5(a) and 5(b) show that the mean correlation functions of the 5000 training samples matches those of the 5000 generated ones. In addition, the two-point correlation functions' envelop of the generated samples overlaps with all possible regions that the original data covers and its slightly broadened envelop suggests that the proposed model might be capable of extrapolating the range of microstructures (by exploring more possibilities of the microstructures) while retaining the morphological characteristics of the collected samples.

3 Microstructure Design Evaluation

In the context of microstructural materials design, design evaluation is the process of evaluating the material properties of interest for a generated microstructure controlled by the design variables. In the proposed methodology, it includes two steps: (1) Latent variables (design variables) to microstructures: the GAN generator learned in the deep adversarial learning is used to propagate the values of latent variables to obtain microstructure images. (2) Microstructure to

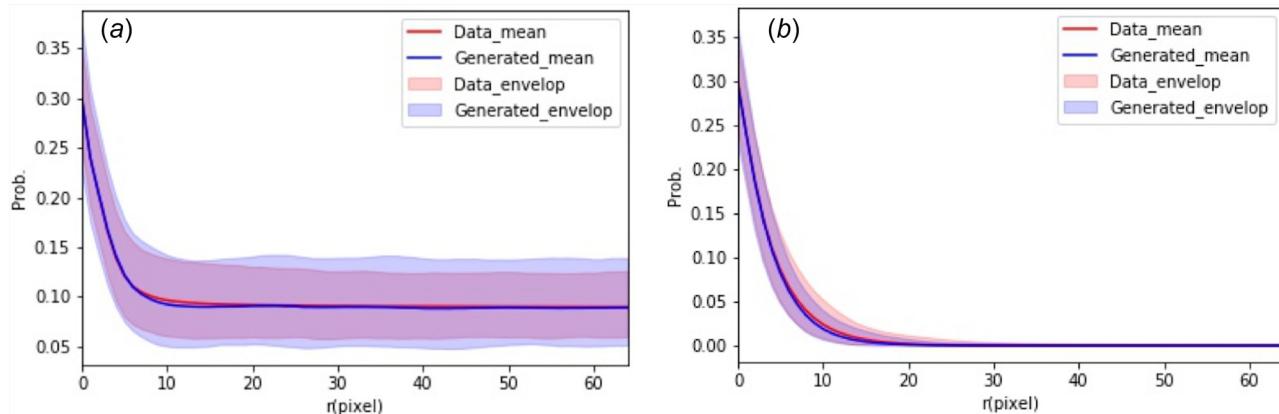


Fig. 5 Comparison of correlation functions of original microstructures and microstructures generated by the proposed generator: (a) two-point correlation function and (b) lineal-path correlation function

material property: For a generated microstructure, physics-based simulation is used to obtain the corresponding material property or structure performance. For the case study in this work, the rigorous coupled wave analysis [15] is used to simulate the optical absorption performance of the given microstructure.

4 Microstructure Design Synthesis

Each entry of the latent variables vector z identified by GAN is independent and bounded in $[-1, 1]$. They serve as the microstructure design variables in design synthesis, which is accomplished through simulation-based optimization. Since the structure–property or structure–performance evaluation is often computationally expensive, a Bayesian optimization approach is applied to search for the optimal microstructure with desired material behavior through sequential adaptive sampling. The design optimization problem is formulated as

$$\begin{aligned} z &= \underset{z}{\operatorname{argmin}} -f(G(z)) \\ \text{s.t. } z_i &\in [-1, 1] \end{aligned}$$

where $G(\cdot)$ is the generator mapping in GAN, and $f(\cdot)$ is the physical simulation. After obtaining the optimal value of z , the optimal microstructure can be generated rapidly by generator $G(z)$.

In the remaining part of this section, we illustrate the use of response surface-based Bayesian optimization through a microstructural materials design case study. The 2D metamaterial structures being explored have similar morphological characteristics as the ones used in Sec. 2 (Fig. 4 row 1), but a smaller size of 96×96 pixels. The design objective is to obtain the microstructure that maximizes the optical absorption simulated by rigorous coupled wave analysis, a desirable performance in applications such as solar cell design. The learned model in Sec. 2 is applied in this case study, and the dimensionality scaling factor is still $\times 32$ in each dimension. In other words, the 96×96 microstructure images would be represented by 3×3 dimensional tensor (i.e., nine-dimensional vector).

4.1 Exploration of Design Variable Space Using Design of Experiments. To create the response surface model between the design variables and the objective material property, a set of design of experiments are sampled. In this work, Latin hyper-cube sampling (LHS) [46] is applied to sample 250 points in the nine-dimensional space. Then the material optical performance for these designs, denoted as y , is obtained by following the design evaluation process described in Sec. 3. The dataset of 250 samples (\mathbf{z} , y) are used to create the initial response surface model for Bayesian optimization.

4.2 Gaussian Process Metamodeling and GP-Hedge Bayesian Optimization. After the initial sampling using LHS, metamodel-based Bayesian optimization is conducted to iteratively explore the potentially optimal design point. Compared to stochastic optimization approaches such as genetic algorithm and simulated annealing, Bayesian optimization is a much more efficient global optimization technique as it encourages both exploration and exploitation in the optimization search process. In each optimization iteration, we fit a metamodel (aka. surrogate model or response surface model) using *Gaussian process metamodeling* [47] to statistically approximate the relationship between design variables and the design performance. The dataset (\mathbf{z} , y) is expanded by one more sampling point in each iteration using the *GP-Hedge* criteria [48]. Figure 6 illustrates how Gaussian process metamodeling and the GP-Hedge optimization strategy are integrated in this work.

Gaussian process model [47], also known as Kriging model, is a statistical model that interpolates the observations and supplies uncertainty for the metamodel prediction at each estimation point.

In essence, Gaussian process models the data points $\{\mathbf{X}, \mathbf{y}\}$ and the estimations $\{\mathbf{X}', \mathbf{y}'\}$ using

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}' \end{bmatrix} \sim \mathcal{N} \left(0, \begin{bmatrix} \text{Cov}(\mathbf{X}, \mathbf{X}) & \text{Cov}(\mathbf{X}, \mathbf{X}') \\ \text{Cov}(\mathbf{X}', \mathbf{X}) & \text{Cov}(\mathbf{X}', \mathbf{X}') \end{bmatrix} \right) \quad (7)$$

where $\text{Cov}(\mathbf{A}, \mathbf{B})$ represents the covariance matrix between \mathbf{A} and \mathbf{B} , defined by $\text{Cov}(\mathbf{A}, \mathbf{B}) = \mathbb{E}(\mathbf{AB}^T) - \mathbb{E}(\mathbf{A})\mathbb{E}(\mathbf{B})^T$. Conditioning on the data $D = \{\mathbf{X}, \mathbf{y}\}$, the posterior $P(\mathbf{y}'|\mathbf{X}, \mathbf{X}', \mathbf{y})$ yields a Gaussian distribution in which,

$$\begin{aligned} \mu &= \text{Cov}(\mathbf{X}, \mathbf{X}')\text{Cov}(\mathbf{X}, \mathbf{X})^{-1} \\ \Sigma &= \text{Cov}(\mathbf{X}', \mathbf{X}') - \text{Cov}(\mathbf{X}, \mathbf{X}')\text{Cov}(\mathbf{X}, \mathbf{X})^{-1}\text{Cov}(\mathbf{X}', \mathbf{X}) \end{aligned} \quad (8)$$

Gaussian process metamodeling essentially gives a surrogate model that quantifies the statistical mean estimations and uncertainties at the unexplored design points. By using the mean estimations and the uncertainties, a smaller set of design points that could potentially improve the performance can be identified. In this case, expensive design evaluations only need to be conducted on these candidate design points, thereby eliminating redundant design evaluations. As a consequence, the overall computational cost of the design process is reduced tremendously.

In each iteration of the Bayesian optimization, the Gaussian process metamodel is applied to determine the next sampling point. Typical criteria (aka. acquisition functions) to locate the next sampling point include expected improvement [49], probability of improvement [50], and lower confidence bound [51]. These criteria are different in how the trade-off is made between exploration (picking samples at locations with large uncertainty) and exploitation (choosing samples at locations close to the optimum based on the mean prediction). In this work, we apply the GP-Hedge mechanism to probabilistically choose one of the above three acquisition functions at every optimization iteration. The general procedure of GP-Hedge Bayesian optimization is illustrated in Algorithm 1.

Algorithm 1 GP-Hedge Bayesian optimization

```

1: Select parameter  $\eta \in \mathbb{R}^+$ 
2: Set the gains for acquisition function  $i$ ,  $g_0^i = 0$  for  $i = 1, 2, \dots, N$ 
3:  $t = 0$ 
4: while stopping criteria is not met do
5:    $t = t + 1$ 
6:   Each acquisition function propose a point  $\mathbf{x}_t^i$ 
7:   Set  $\mathbf{x}_t = \mathbf{x}_t^i$  with softmax probability
       $p_t^i = \exp(\eta g_{t-1}^i) / \sum_{l=1}^K \exp(\eta g_{t-1}^l)$ 
8:   Obtain the objective function value  $y_t = f(\mathbf{x}_t)$ 
9:   Augment data  $D_{1:t} = \{D_{1:t-1}, (\mathbf{x}_t, y_t)\}$ 
10:  Receive rewards  $r_t^i = \mu(\mathbf{x}_t^i)$  from the updated GP
11:  Update gains  $g_t^i = g_{t-1}^i + r_t^i$ 
12: end while

```

This GP-Hedge Bayesian optimization process is applied to our design case study beginning with the metamodel created using the 250 initial LHS samples, followed by 120 iterations of optimization. Throughout the optimization process, the values of the latent (design) variables are constrained between $[-1, 1]$ to retain the morphological characteristics learned from the sample images.

Figure 7 shows the optimization history with microstructure design solution indicated at a few iterations. A few observations can be made: (1) design performance is improved significantly at the very beginning of the Bayesian optimization, while the improvement becomes less as the number of iterations increases. (2) Design performance is not necessarily improved in the new iteration. This is reasonable because the new sampling point is chosen for both exploration and exploitation using the criterion

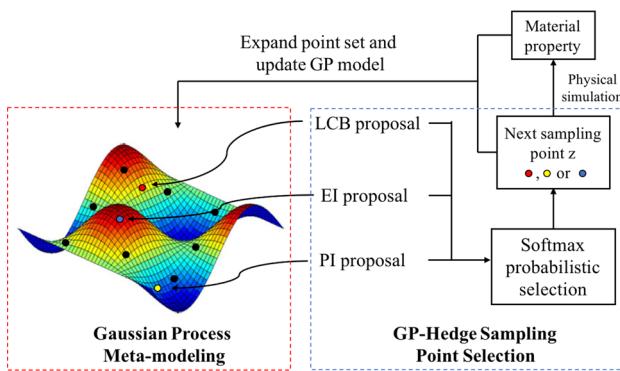


Fig. 6 The integration of Gaussian process metamodeling and GP-Hedge Bayesian optimization

that combines both the mean estimation and the uncertainty in the metamodel.

Figure 8 illustrates the comparison between the optical performance of three datasets: (a) 30 randomly sampled microstructures from training set, (b) 30 microstructures generated by randomly sampling latent variables z and propagating through the trained generator, and (c) the optimized microstructure. It should be noted that in order to make a fair comparison, we randomly sampled 30 microstructures from training set in each trial for dataset (a), and repeated this trial ten times. It is observed that the results of randomly sampled microstructures have the lowest

optical performance and the largest variance. It is found that the mean optical performance of the microstructures produced by the GAN generator (0.6827) is 4.8% ($0.6827/0.6509 - 1$) greater than that of the randomly sampled microstructures (0.6509), while the optimized microstructure's performance (0.7630) exceeds the mean performance of randomly sampled microstructures by 17.2% ($0.7630/0.6509 - 1$). It should be noted that the theoretical upper bound of the evaluated optical absorption property is 1.0, so the design solution provided by the proposed approach is reasonably good. These results verify the effectiveness of the proposed design optimization framework.

5 Scalability and Transferability

In the previous sections, we have discussed the process of applying the proposed deep adversarial learning model for identifying latent variables of microstructures and conducting microstructural materials design. With the proposed methodology, the dimensionality of latent (design) variables can be prescribed and the information loss is negligible even for complex microstructural geometries. In addition to these advantages, in this section a few additional useful features of the proposed deep adversarial learning model are elaborated.

5.1 Scalability of the Generator. Benefited from the exclusion of fully connected layers in the network architecture, the scalability of the generator provides the proposed GAN model the flexibility of taking arbitrary sized inputs (latent variables) and outputs (microstructures). This is a signature of the proposed model because confining the input dimensionality could lead to a low-dimensional microstructural design space, and varying the output size can consequentially produce different sized microstructures to serve different analytical purposes (e.g., analysis in statistical volume elements versus representative volume elements).

Specifically, the scalability is useful in two ways: (a) Flexibility in setting the dimensionality of latent variables. In the proposed network architecture, adding each additional convolutional layer increases the scaling factor between the generated image and the latent variables by a factor of 4 ($\times 2$ on each dimension). Therefore, in the aforementioned design case in Sec. 4, when the 9216-dimensional (96×96) microstructure is to be converted into 9-dimensional (3×3) latent variables, five network layers are stacked (i.e., $96/3 = 32 = 2^5$). In theory, stacking more neural network layers in the proposed model can enlarge the scaling factor, and the accuracy would be retained as long as the training is well handled. However, adding more layers inevitably increase the difficulty of training the GAN. In other words, while low dimensionality of the latent variables often leads to less microstructure design optimization cost because of less design variables, it increases the GAN training cost because of higher model complexity. Hence, a key consideration in choosing the number of latent variables is the trade-off between the optimization cost and the GAN training cost. When the computational resource for design optimization is limited (e.g., physics-based simulations are extremely expensive), it would be better to keep a lower dimensionality of the latent variables though more training time for GANs is needed. In contrast, if the design optimization is not limited by the computational resource, a reasonably higher dimensionality of the latent variables is acceptable so that the burden on training GANs can be reduced. (b) Generating arbitrary sized microstructures. While the deep learning network is trained by setting the dimension of z as $bs \times 4 \times 4 \times 1$, one may modify the dimensionality of latent variables z to control the size of the generated images without retraining the model. Figure 9 illustrates the generated images with different sizes using different dimensional settings of z . It demonstrates that the proposed generator is capable of generating arbitrary sized microstructures for the material system of interest. An alternative way of controlling the size of microstructure is to include/remove convolutional layers. For

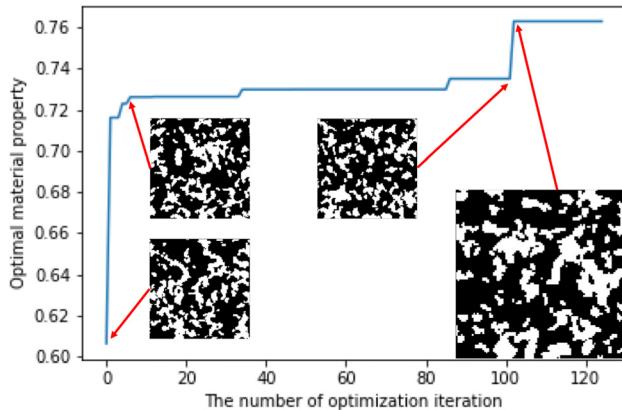


Fig. 7 The microstructure optimization history and microstructure designs indicated at selected iterations

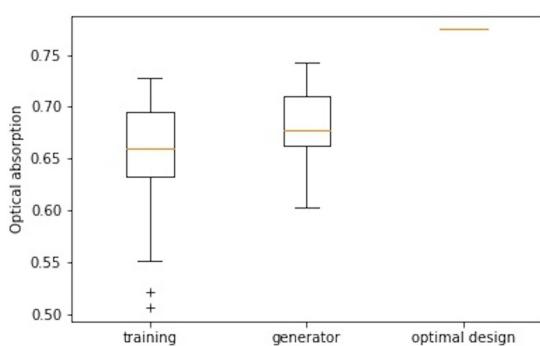


Fig. 8 The comparison of the optical absorption property between (1) 30 randomly generated microstructures, (2) 30 microstructures generated by the trained generator, and (3) optimal design

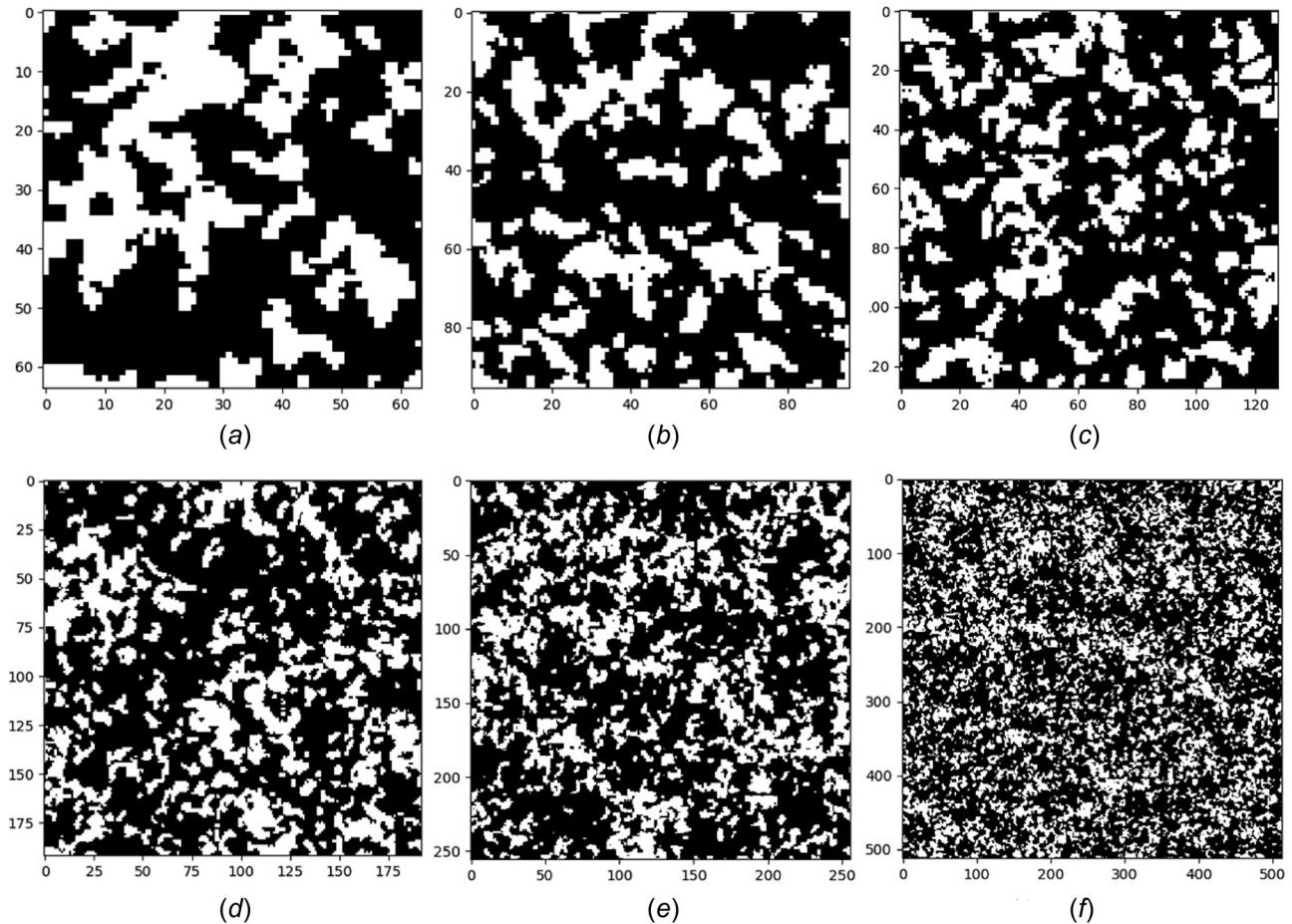


Fig. 9 An illustration of microstructures of different sizes generated by the scalable generator: (a) 64 × 64, (b) 96 × 96, (c) 128 × 128, (d) 192 × 192, (e) 256 × 356, and (f) 512 × 512

instance, 256 × 256 images could be generated by adding one more layer in both generator and discriminator before training and keep the size of z as $bs. \times 4 \times 4 \times 1$. However, the deeper the neural network is, the harder the training process would be. Moreover, retraining is required if model's architecture is changed. Hence, changing the dimensionality of z is often preferred for this reason.

5.2 Transferability of the Discriminator. In addition to the aforementioned microstructural materials design contributions of the proposed approach, we also discover an additional utility of the discriminator in improving structure–property predictions via transfer learning. While the generative capability is usually emphasized [28,52], the utilization of discriminator is more or less ignored. However, totally discarding the discriminator is wasteful as there is always significant “knowledge” about the data (in the context of this work, microstructures) learned by the discriminator. In this work, we propose to leverage the knowledge learned from the discriminator into the development of machine learning-based structure–property predictions via transfer learning. In training deep networks, stochastic gradient descent-based algorithms are the typical choices. Since stochastic gradient descent converges to local minimum, its optimized value is very sensitive to the initialization of the network. With transfer learning, instead of randomly selecting a starting point for the weights of the structure–property predictive network, the weights are initialized using ones obtained in the GAN discriminator trained on the microstructure dataset in Sec. 2, by analogy to Ref. [53].

In the context of this work, the discriminator is essentially a binary classifier trained together with the generator to distinguish

generated microstructure from real ones. Our objective is to utilize the trained weights in this classifier and transfer them into a structure–property regression model. It should be noted that, in training and testing this regression model, we use additional 250 samples (microstructures and their corresponding properties) exclusive from the 5000 samples used for training the GAN, and we randomly split them into 200/50 sets for training/testing. There are three primary steps in building the regression model:

- (1) *Transferring partial architecture and weights:* We borrow the first four convolutional layers of the trained discriminator (their architecture and the corresponding weights) as the basic building blocks.
- (2) *Appending full-connected layers at the end:* The output of the 4th convolutional layer is flattened and two fully connected layers of 2048 and 1024 neurons with ReLU activation are appended. Dropout normalization ($p=0.5$) is applied after each fully connected layer. A fully connected layer of 1 neuron is added at the end to produce the scalar output of the regressor. The weights of all these additional layers are initialized randomly.
- (3) *Fine-tuning weights using Adam:* Adam optimizer is applied to fine-tune the weights in some of the layers. As it is well recognized that the early convolutional layers (Convolutional layers 1–3 of the discriminator) usually contains general Gabor-like filters, we freeze these layers' weights from Adam optimization. The other layers are subject to the Adam optimization (learning rate = 0.0005, $\beta_1 = 0.5$, $\beta_2 = 0.99$) for 4000 epochs with batch size of 50.

To demonstrate the advantage of applying this transfer learning strategy for building the structure–property model, we also

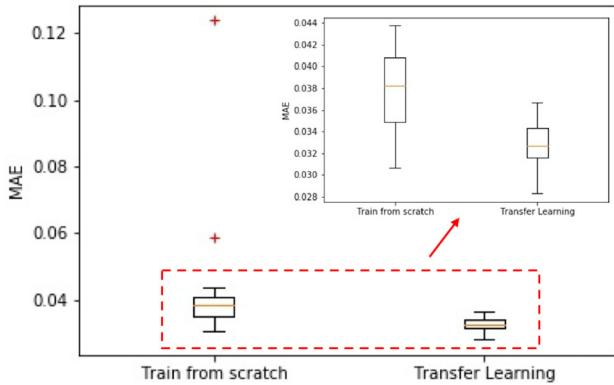


Fig. 10 The comparison of the MAE for training from scratch and transfer learning. Outliers, higher error and variance are observed from the results of “training from scratch” control group.

conduct another training process with exactly the same network architecture but initializing all weights randomly (instead of using pretrained weights) as a control group. This control group is named “training from scratch” in the remainder of this section. We compute the mean-squared-errors and the mean-absolute-errors (MAE) on the 50 reserved testing data with 30 repetitive trials. Since both error metrics measure the same fundamental error phenomena, we only show the result of MAE comparison in Fig. 10. From the results, it is found that, compared to training from scratch, transfer learning strategy can facilitate the development of structure-property predictive model by improving its accuracy and stability. This finding is consistent with our intuition that prior knowledge learned by the discriminator network could help in building a more accurate predictive model.

6 Conclusion and Future Work

In this work, we proposed a deep adversarial learning methodology for microstructural material design. In the proposed methodology, the dimensionality of latent variables for microstructures is prescribed first. Then a GAN consisting of a generator and a discriminator is trained on a dataset of microstructures being studied. The latent variables are then taken as design variables in a Bayesian optimization framework to obtain the microstructure with desired material property. Gaussian Process metamodeling is used at each optimization iteration to update the relationship between the design variables and the microstructure performance, and GP-Hedge criterion is used for proposing the next candidate sampling point. The proposed methodology features several contributions: First, the proposed methodology provides an end-to-end solution for microstructural materials design, which reduces information loss and preserves more microstructural characteristics. Second, this work is to extend the use of GAN to be a part of the design loop. The GP-Hedge Bayesian optimization incorporates Gaussian Process metamodeling to reduce the number of design evaluations and thus decreases the computational cost while improving the design performance. Third, a customized loss function with the proper moderating parameters is presented for generating new microstructural design with similar characteristics. Finally, the deep learning network architecture and the training parameters obtained in this work could be re-used as a starting point for other applications of deep learning in materials science (e.g., transfer learning).

While this work demonstrates the benefits of the proposed methodology, a few technical details can be further examined in future work. First, this work could make a boarder impact on other material microstructures such as ones with very sharp features (e.g., pointy edges), crystalline structures, or grain boundary maps, multiphase or continuous phase microstructures. Next, the

processing or manufacturing constraints are not considered in the design optimization. In order to take the processing conditions as design variables, the processing-structure-property linkage needs to be established. Similar to our earlier work [11,15,54], we will study the relationship between latent variables and such processing or manufacturing parameters, including appropriate constraints in the optimization process. Attempts would be also made to associate physical meanings to the learned latent variables so that materials scientists could explicitly control some characteristics of the optimized microstructure. In addition, the choice of dimensionality of latent variables can be guided through detailed numerical studies to better understand the impact of low dimensionality on network training. Special attention needs to be paid toward the network theory and practice for stabilizing the training process. Other potential directions for improving network modeling include but are not limited to utilizing Wasserstein GAN [40] for solving model collapse problem, introducing ResNet structure [55] for higher learning capability, or investigating visual attention mechanism [56] for better interpretation of the model.

Acknowledgment

The Rigorous Couple Wave Analysis simulation is supported by Professor Cheng Sun’s lab at Northwestern University. This work is primarily supported by the Center of Hierarchical Materials Design (NIST ChiMaD 70 NANB14H012) and Predictive Science and Engineering Design Cluster (PS&ED, Northwestern University). Partial support from NSF awards DMREF-1818574, DMREF-1729743, DIBBS-1640840, CCF-1409601; DOE awards DE-SC0007456, DE-SC0014330; AFOSR award FA9550-12-1-0458; and Northwestern Data Science Initiative is also acknowledged.

Funding Data

- National Science Foundation (CCF-1409601, DIBBS-1640840, DMREF-1729743, and DMREF-1818574).
- U.S. Department of Energy (DE-SC0007456 and DE-SC0014330).

References

- [1] Thornton, K., Nola, S., Garcia, R. E., Asta, M., and Olson, G., 2009, “Computational Materials Science and Engineering Education: A Survey of Trends and Needs,” *JOM*, **61**(10), p. 12.
- [2] Olson, G. B., 1997, “Computational Design of Hierarchically Structured Materials,” *Science*, **277**(5330), pp. 1237–1242.
- [3] Saito, T., 2013, *Computational Materials Design*, Vol. 34, Springer Science & Business Media, Berlin, Germany.
- [4] Kuehmann, C., and Olson, G., 2009, “Computational Materials Design and Engineering,” *Mater. Sci. Technol.*, **25**(4), pp. 472–478.
- [5] Vickers, J., 2015, “Materials Genome Initiative,” *Energy Efficiency & Renewable Energy*, Washington, DC.
- [6] Agrawal, A., and Choudhary, A., 2016, “Perspective: Materials Informatics and Big Data: Realization of the “Fourth Paradigm” of Science in Materials Science,” *APL Mater.*, **4**(5), p. 053208.
- [7] Zhao, H., Li, X., Zhang, Y., Schadler, L. S., Chen, W., and Brinson, L. C., 2016, “Perspective: Nanomine: A Material Genome Approach for Polymer Nanocomposites Analysis and Design,” *APL Mater.*, **4**(5), p. 053204.
- [8] Wang, Y., Zhang, Y., Zhao, H., Li, X., Huang, Y., Schadler, L. S., Chen, W., and Brinson, L. C., 2018, “Identifying Interphase Properties in Polymer Nanocomposites Using Adaptive Optimization,” *Compos. Sci. Technol.*, **162**(7), pp. 146–155.
- [9] Paul, D., and Robeson, L. M., 2008, “Polymer Nanotechnology: Nanocomposites,” *Polymers*, **49**(15), pp. 3187–3204.
- [10] Natarajan, B., Li, Y., Deng, H., Brinson, L. C., and Schadler, L. S., 2013, “Effect of Interfacial Energetics on Dispersion and Glass Transition Temperature in Polymer Nanocomposites,” *Macromolecules*, **46**(7), pp. 2833–2841.
- [11] Hassinger, I., Li, X., Zhao, H., Xu, H., Huang, Y., Prasad, A., Schadler, L., Chen, W., and Brinson, L. C., 2016, “Toward the Development of a Quantitative Tool for Predicting Dispersion of Nanocomposites Under Non-Equilibrium Processing Conditions,” *J. Mater. Sci.*, **51**(9), pp. 4238–4249.
- [12] Brough, D. B., Wheeler, D., Warren, J. A., and Kalidindi, S. R., 2017, “Microstructure-Based Knowledge Systems for Capturing Process-Structure Evolution Linkages,” *Curr. Opin. Solid State Mater. Sci.*, **21**(3), pp. 129–140.

[13] Liu, R., Yabansu, Y. C., Yang, Z., Choudhary, A. N., Kalidindi, S. R., and Agrawal, A., 2017, "Context Aware Machine Learning Approaches for Modeling Elastic Localization in Three-Dimensional Composite Microstructures," *Integr. Mater. Manuf. Innov.*, **6**, p. 160.

[14] Yang, Z., Yabansu, Y. C., Al-Bahrani, R., Liao, W.-K., Choudhary, A. N., Kalidindi, S. R., and Agrawal, A., 2018, "Deep Learning Approaches for Mining Structure-Property Linkages in High Contrast Composites From Simulation Datasets," *Comput. Mater. Sci.*, **151**, pp. 278–287.

[15] Yu, S., Zhang, Y., Wang, C., Lee, W.-K., Dong, B., Odom, T. W., Sun, C., and Chen, W., 2017, "Characterization and Design of Functional Quasi-Random Nanostructured Materials Using Spectral Density Function," *ASME J. Mech. Des.*, **139**(7), p. 071401.

[16] Fullwood, D. T., Niezgoda, S. R., Adams, B. L., and Kalidindi, S. R., 2010, "Microstructure Sensitive Design for Performance Optimization," *Prog. Mater. Sci.*, **55**(6), pp. 477–562.

[17] Bostanabad, R., Zhang, Y., Li, X., Kearney, T., Brinson, L. C., Apley, D. W., Liu, W. K., and Chen, W., "Computational Microstructure Characterization and Reconstruction: Review of the State-of-the-Art Techniques," *Prog. Mater. Sci.*, **95**, pp. 1–41.

[18] Jiao, Y., Stillinger, F., and Torquato, S., 2007, "Modeling Heterogeneous Materials Via Two-Point Correlation Functions: Basic Principles," *Phys. Rev. E*, **76**(3), p. 031110.

[19] Xu, H., Dikin, D., Burkhardt, C., and Chen, W., 2014, "Descriptor-Based Methodology for Statistical Characterization and 3d Reconstruction of Microstructural Materials," *Comput. Mater. Sci.*, **85**, pp. 206–216.

[20] Jiang, Z., Chen, W., and Burkhardt, C., 2013, "Efficient 3d Porous Microstructure Reconstruction Via Gaussian Random Field and Hybrid Optimization," *J. Microsc.*, **252**(2), pp. 135–148.

[21] Bostanabad, R., Bui, A. T., Xie, W., Apley, D. W., and Chen, W., 2016, "Stochastic Microstructure Characterization and Reconstruction Via Supervised Learning," *Acta Mater.*, **103**, pp. 89–102.

[22] Cang, R., Xu, Y., Chen, S., Liu, Y., Jiao, Y., and Ren, M. Y., 2017, "Microstructure Representation and Reconstruction of Heterogeneous Materials Via Deep Belief Network for Computational Material Design," *ASME J. Mech. Des.*, **139**(7), p. 071404.

[23] Li, X., Zhang, Y., Zhao, H., Burkhardt, C., Brinson, L. C., and Chen, W., 2018, "A Transfer Learning Approach for Microstructure Reconstruction and Structure-Property Predictions," preprint [arXiv: 02784](https://arxiv.org/abs/1804.02784).

[24] Lubbers, N., Lookman, T., and Barros, K., 2017, "Inferring Low-Dimensional Microstructure Representations Using Convolutional Neural Networks," *Phys. Rev. E*, **96**(5), p. 052111.

[25] Paulson, N. H., Priddy, M. W., McDowell, D. L., and Kalidindi, S. R., 2017, "Reduced-Order Structure-Property Linkages for Polycrystalline Microstructures Based on 2-Point Statistics," *Acta Mater.*, **129**, pp. 428–438.

[26] Xu, H., Liu, R., Choudhary, A., and Chen, W., 2015, "A Machine Learning-Based Design Representation Method for Designing Heterogeneous Microstructures," *ASME J. Mech. Des.*, **137**(5), p. 051403.

[27] Kingma, D. P., and Welling, M., 2014, "Auto-Encoding Variational Bayes," preprint [arXiv: 1312.6114](https://arxiv.org/abs/1312.6114).

[28] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., 2014, "Generative Adversarial Nets," *Advances in Neural Information Processing Systems*, Neural Information Processing Systems, Montreal, Canada, pp. 2672–2680.

[29] Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B., 2016, "Adversarial Autoencoders," preprint [arXiv:1511.05644](https://arxiv.org/abs/1511.05644).

[30] Dumoulin, V., Belghazi, I., Poole, B., Lamb, A., Arjovsky, M., Mastropietro, O., and Courville, A., 2017, "Adversarially Learned Inference," preprint [arXiv:1606.00704](https://arxiv.org/abs/1606.00704).

[31] Cang, R., Li, H., Yao, H., Jiao, Y., and Ren, Y., 2018, "Improving Direct Physical Properties Prediction of Heterogeneous Materials From Imaging Data Via Convolutional Neural Network and a Morphology-Aware Generative Model," preprint [arXiv:1712.03811](https://arxiv.org/abs/1712.03811).

[32] Guo, T., Lohan, D. J., Cang, R., Ren, M. Y., and Allison, J. T., 2018, "An Indirect Design Representation for Topology Optimization Using Variational Autoencoder and Style Transfer," *AIAA Paper No. AIAA 2018-0804*.

[33] Mosser, L., Dubrule, O., and Blunt, M. J., 2017, "Reconstruction of Three-Dimensional Porous Media Using Generative Adversarial Neural Networks," *Phys. Rev. E*, **96**(4), p. 043309.

[34] Mosser, L., Dubrule, O., and Blunt, M. J., 2017, "Stochastic Reconstruction of an Oolitic Limestone by Generative Adversarial Networks," preprint [arXiv:1712.02854](https://arxiv.org/abs/1712.02854).

[35] Osborne, M. J., and Rubinstein, A., 1994, *A Course in Game Theory*, MIT Press, Cambridge, MA.

[36] Radford, A., Metz, L., and Chintala, S., 2016, "Unsupervised Representation Learning With Deep Convolutional Generative Adversarial Networks," preprint [arXiv:1511.06434](https://arxiv.org/abs/1511.06434).

[37] Ioffe, S., and Szegedy, C., 2015, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *International Conference on Machine Learning*, Lille, France, July 6–11, pp. 448–456.

[38] Nair, V., and Hinton, G. E., 2010, "Rectified Linear Units Improve Restricted Boltzmann Machines," *27th International Conference on Machine Learning (ICML-10)*, Haifa, Israel, June 21–24, pp. 807–814.

[39] Maas, A. L., Hannun, A. Y., and Ng, A. Y., 2013, "Rectifier Nonlinearities Improve Neural Network Acoustic Models," *30th International Conference on Machine Learning (ICML)*, Atlanta, Georgia, June 16–21.

[40] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X., 2016, "Improved Techniques for Training Gans," *30th Conference on Neural Information Processing Systems (NIPS)*, Barcelona, Spain, Dec. 5–8, pp. 2234–2242.

[41] Gatys, L., Ecker, A. S., and Bethge, M., 2015, "Texture Synthesis Using Convolutional Neural Networks," *28th International Conference on Neural Information Processing Systems*, Montreal, QC, Canada, Dec. 7–12, pp. 262–270.

[42] Simonyan, K., and Zisserman, A., 2015, "Very Deep Convolutional Networks for Large-Scale Image Recognition," preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556).

[43] Zhao, J., Mathieu, M., and LeCun, Y., 2017, "Energy-Based Generative Adversarial Network," preprint [arXiv:1609.03126](https://arxiv.org/abs/1609.03126).

[44] Kingma, D. P., and Ba, J., 2017, "Adam: A Method for Stochastic Optimization," preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).

[45] Lu, B., and Torquato, S., 1992, "Lineal-Path Function for Random Heterogeneous Materials," *Phys. Rev. A*, **45**(2), p. 922.

[46] McKay, M. D., Beckman, R. J., and Conover, W. J., 2000, "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a Computer Code," *Technometrics*, **42**(1), pp. 55–61.

[47] Rasmussen, C. E., 2004, "Gaussian Processes in Machine Learning," *Advanced Lectures on Machine Learning*, Springer, Berlin, pp. 63–71.

[48] Hoffman, M. D., Brochu, E., and de Freitas, N., 2011, "Portfolio Allocation for Bayesian Optimization," *Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, Barcelona, Spain, July 14–17, pp. 327–336.

[49] Jones, D. R., Schonlau, M., and Welch, W. J., 1998, "Efficient Global Optimization of Expensive Black-Box Functions," *J. Global Optim.*, **13**(4), pp. 455–492.

[50] Kushner, H. J., 1964, "A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise," *J. Basic Eng.*, **86**(1), pp. 97–106.

[51] Auer, P., 2002, "Using Confidence Bounds for Exploitation-Exploration Trade-Offs," *J. Mach. Learn. Res.*, **3**, pp. 397–422.

[52] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P., 2016, "Infogan: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets," *Advances in Neural Information Processing Systems*, Barcelona, Spain, pp. 2172–2180.

[53] Yosinski, J., Clune, J., Bengio, Y., and Lipson, H., 2014, "How Transferable Are Features in Deep Neural Networks?," *27th International Conference on Neural Information Processing Systems*, Montreal, QC, Canada, Dec. 8–13, pp. 3320–3328.

[54] Lee, W.-K., Yu, S., Engel, C. J., Reese, T., Rhee, D., Chen, W., and Odom, T. W., 2017, "Concurrent Design of Quasi-Random Photonic Nanostructures," *Proc. Natl. Acad. Sci.*, **114**(33), pp. 8734–8739.

[55] He, K., Zhang, X., Ren, S., and Sun, J., 2016, "Deep Residual Learning for Image Recognition," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, June 27–30, pp. 770–778.

[56] Mnih, V., Heess, N., Graves, A., et al., 2014, "Recurrent Models of Visual Attention," *27th International Conference on Neural Information Processing Systems*, Montreal, QC, Canada, Dec. 8–13, pp. 2204–2212.