



OPEN

Synthesizing controlled microstructures of porous media using generative adversarial networks and reinforcement learning

Phong C. H. Nguyen¹, Nikolaos N. Vlassis², Bahador Bahmani², WaiChing Sun², H. S. Udaykumar³ & Stephen S. Baek^{1,4}✉

For material modeling and discovery, synthetic microstructures play a critical role as digital twins. They provide stochastic samples upon which direct numerical simulations can be conducted to populate material databases. A large ensemble of simulation data on synthetic microstructures may provide supplemental data to inform and refine macroscopic material models, which might not be feasible from physical experiments alone. However, synthesizing realistic microstructures with realistic microstructural attributes is highly challenging. Thus, it is often oversimplified via rough approximations that may yield an inaccurate representation of the physical world. Here, we propose a novel deep learning method that can synthesize realistic three-dimensional microstructures with controlled structural properties using the combination of generative adversarial networks (GAN) and actor-critic (AC) reinforcement learning. The GAN-AC combination enables the generation of microstructures that not only resemble the appearances of real specimens but also yield user-defined physical quantities of interest (QoI). Our validation experiments confirm that the properties of synthetic microstructures generated by the GAN-AC framework are within a 5% error margin with respect to the target values. The scientific contribution of this paper resides in the novel design of the GAN-AC microstructure generator and the mathematical and algorithmic foundations therein. The proposed method will have a broad and substantive impact on the materials community by providing lenses for analyzing structure-property-performance linkages and for implementing the notion of 'materials-by-design'.

Developing digital twins of multi-phase microstructures is increasingly important in materials modeling and characterization¹⁻³. While physical experiments on real specimens may provide valuable ground-truth data for calibration and validation of material models, the process of acquiring and experimenting with physical specimens is often costly and laborious. In addition, the distribution of physical properties in nature-obtained specimens cannot be controlled easily, which forces materials scientists to rely on a numerous "cut-and-try" experiments. Furthermore, the discrepancies in microstructural heterogeneity⁴ and the limited reproducibility of experiments⁵ raise a fundamental question on the practicality of relying solely on experimental data to calibrate high-fidelity models⁶.

Instead, direct numerical simulations (DNS) can serve as a valuable complement to lab experiments. In DNS, computational experiments are conducted on microstructures inferred from microscopic imagery, such as micro computed tomography (μ CT)⁷ or generated from 2D scanning electron microscopy (SEM)⁸. However, the route of using DNS to obtain structure-property-performance (SPP) relationships is still hindered by the constraints placed by image acquisition. Typically, only a limited number of images spanning a limited sample size is available from images. The images may also pertain to specific formulations, as parameterized by the size distributions of

¹School of Data Science, University of Virginia, Charlottesville, VA 22903, USA. ²Department of Civil Engineering and Engineering Mechanics, Columbia University, New York, NY 10027, USA. ³Department of Mechanical Engineering, University of Iowa, Iowa City, IA 52242, USA. ⁴Department of Mechanical and Aerospace Engineering, University of Virginia, Charlottesville, VA 22903, USA. ✉email: baek@virginia.edu

particles, defects such as voids and cracks, and other important morphological features. Hence, from a material design point of view, it would be desirable to cover a large and diverse collection of microstructures, spanning a broad regime of the configuration space. However, in reality, it is practically impossible to obtain such diverse samples, as there is no viable way to control the micro-morphology but to “cut-and-try”.

To overcome these limitations placed by available imaged datasets, there has been an increasing in the number of research activities involving stochastic microstructure reconstruction in the materials research community⁹. “Synthetic” microstructures can be generated from a large variety of approaches, including N-point correlation functions¹⁰, shape descriptors¹¹, ellipsoid packing¹², or Gaussian random fields^{13,14} with different sets of advantages and limitations. High-order N-point correlation functions may theoretically generate microstructures with consistent statistics attributes; nevertheless, the high computational costs of those approaches are intractable. Meanwhile, Gaussian autocorrelation is often limited to the case where the Gaussian stochastic process can be completely characterized by their second-order statistics; as a result, constructing microstructures as non-Gaussian fields may lead to more realistic reconstructed microstructures¹⁵. However, such a task is challenging in practice as it requires all the joint probability density functions to be determined or estimated. In brief, DNS performed on idealized synthetic images may not lead to inferred effective mechanical properties that are physically realistic and hence may only be valid for simple trend analyses⁸.

Machine learning (ML) based approaches have rapidly emerged as promising alternatives to overcome the limitations of traditional microstructure reconstruction methods^{16–18}. In particular, convolutional neural networks (CNN) based deep learning (DL) approaches have been mostly investigated¹⁹. For example, Lubbers, Lookman, and Barros²⁰, and Li and colleagues²¹ utilized high-dimensional features encoded in a pre-trained CNNs to develop numerical representations of microstructure morphology. They employed the VGG network²², a type of CNN that is popularly utilized in computer vision applications, had been pre-trained using a generic image classification dataset. The authors discovered that the neural activations in response to a microstructural image input yielded accurate and detailed characterization of the complex morphology of microstructures (i.e. “neural style”) and, thus, could serve as a *texture vector* corresponding to a given microstructure image. Furthermore, by employing the optimization-based texture synthesis formulation of Gatys et al.²³, in which the objective of the optimization is to create a synthetic image yielding a texture vector similar to that of a real microstructure, they demonstrated that a CNN could be used to generate realistic synthetic microstructures. However, a significant drawback of such an approach is that the resultant synthetic microstructures can span only small variations in the texture space, as they are bounded to look similar to the reference image. Further, the user has no control over the material properties of generated microstructures, and the practical limitations of “cut-and-try” experiments still persist.

Recently, generative adversarial networks (GAN) based approaches for synthetic microstructure reconstruction have been drawing attention from the materials community. GAN is based on the competition between two neural network agents, namely *generator* and *discriminator*. In the context of synthetic microstructure reconstruction, the generator is a CNN that produces a synthetic microstructure image from a given stochastic noise, and the discriminator is another CNN that distinguishes if a microstructure image is synthetic or real. In principle, through the adversarial competition, the two randomly-initialized networks may eventually converge to a Nash equilibrium, in which synthetic microstructures generated by the generator are indistinguishable from real microstructures. Based on this idea, Chun et al.²⁴ proposed a GAN architecture to parameterize the generator inputs so that the morphology of generated 2D microstructures could be controlled parametrically. The authors demonstrated that the GAN-based synthetic microstructures were more realistic and contained fewer artifacts compared to the outcomes of other CNN-based methods. However, many physical phenomena, including the microstructure deformation, the dynamic of fluid flow within a porous media, or the anisotropic material properties, cannot be adequately modeled by 2D microstructure images alone, rendering a need for the development of GAN to produce 3D synthetic microstructures²⁵.

In the broader context of machine learning, there are 3D GAN solutions readily available to produce 3D models of common objects such as chairs, tables, and airplanes^{26–28}. However, compared to these common objects, 3D material microstructures contain more complex geometric and topological structures not only globally but also locally. In contrast to the fact that the global configuration or ‘style’ of geometry is predominantly the most important result, for the design of materials, not only the global morphology of the microstructure but also the local patterns that comprise the global morphology become critical. Therefore, unlike the common object GANs, microstructure GANs must be able to parameterize both local and global geometric and topological configurations.

There are a few approaches that were proposed to overcome the above issues of microstructure 3D GAN. For instance, Mosser et al.²⁹ stabilized the training of 3D GAN by introducing the Gaussian noise at the input of the discriminator and applying label-switching which can weaken the discriminator during the early stages. In another work, Hsu et al.³⁰ tried to address the above issues by applying Wasserstein GAN (WGAN) in which the Wasserstein loss function³¹ was used to train the discriminator. The Wasserstein distance (WD) has good training properties as it is more sensible than other common distance function applied for GAN loss, including the Total Variation (TV) distance, the Kullback-Leibler (KL) divergence, and Jensen-Shannon (JS) divergence³¹. Therefore, WD allows GAN to learn the probability distribution of complex shapes more efficiently. Moreover, the training of WGAN is also more stable as even at training completion of the discriminator, loss still can be provide to the generator; thus, the effort for balancing the generator and discriminator training is no longer required³¹. Kench and Cooper²⁵ overcome the difficulty in training microstructure 3D GAN differently by employing a discriminator that takes 2D microstructures images sliced from 3D synthetic model as the input instead of the fully 3D ones. The method has shown successes in producing high-quality 3D synthetic microstructures of different types of isotropic materials.

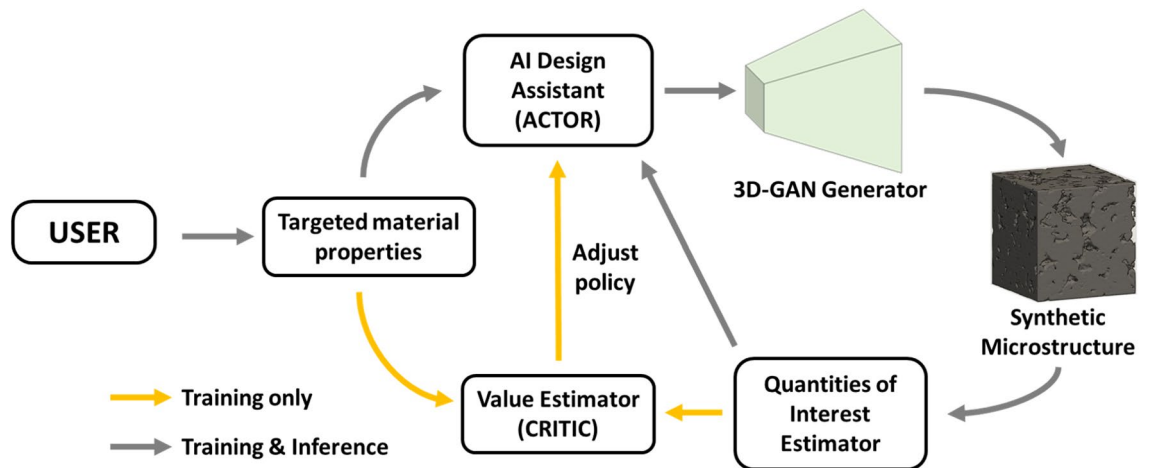


Figure 1. An overview of the proposed GAN-AC-based method. Initially, targeted QoI are provided by users. Consequently, the AI design assistant (the actor) starts to interact with the design environment by tuning input morphology parameters of the 3D-GAN generator for synthetic microstructure reconstruction. The properties of synthetic microstructures are estimated by DNS and sent to the value estimator (the critic) along with the targeted one to evaluate how good the design decision made the actor is. Based on the evaluation result, the actor's policy will be refined to make better design decisions in the subsequent iteration. The training of the GAN-AC framework is completed with two steps: (1) train the 3D-GAN generator and (2) train the AC model. After training, the actor is retained for inference, while the critic is discarded.

Despite of few achievements, previous works on 3D GAN microstructures are short in several aspects. First, the output 3D microstructures are not scalable beyond their fixed size by designed architecture³⁰. Few works has proposed the employment of fully convolution architectures to overcome the scalability problems^{29,32}. However, as mentioned by Kench and Cooper²⁵, during the training, the spatial dimension of the input latent vector is set to 1; therefore, there is no kernel overlap at the first layer of the generator network. As a result, the quality of the output synthetic microstructures is questioned due to distortion. The author proposed expanding the spatial dimension of input latent space during training, as similar to the approach by Chun et al.²⁴, so as the correlation between latent variables at different spatial locations can be properly modeled. Nevertheless, due to the training strategy, the approach by Kench and Cooper is only applicable for isotropic materials. Moreover, in most previous works, the quantitative linkages between the GAN parameters and physical properties are remained unknown. As a result, solving the inverse microstructure design problem, i.e. tuning of GAN morphology parameters that yields synthetic microstructures with realistic effective mechanical properties, is still a time-consuming task and difficult to accomplish manually. There are a few existing gradient-free optimization approaches to overcome such limitations of the GAN-based generative design methods³³. However, the efficiency and the robustness of these gradient-free optimization approaches are often limited by the complexity of the design space^{34,35}. Alternatively, conditional generative networks, which formulates the design constraints as additional inputs to the network, could also be considered as a viable option to address the limitations of the previous GAN approaches^{36–38}. However, the critical drawback of conditional generative networks is that it does not guarantee the design constraints, as there is no feedback loop for ensuring the constrained properties.

This research extends our prior GAN-based approach²⁴ for 3D microstructures along with the introduction of Actor-Critic (AC) reinforcement learning to tune the morphology parameters. We first make a substantive extension to the GAN method of Chun et al.²⁴ to enable the generation of realistic 3D microstructures with arbitrary size and introduce several solutions to overcome the instability in the training of 3D GAN. Consequently, we augment the GAN model using the AC reinforcement learning model to create a new capacity to produce microstructures with desired target properties. The proposed method is validated on μ CT scans of Bentheim sandstone, upon which we demonstrate that the GAN-AC framework is capable of generating visually and physically realistic microstructures. In addition, we also compare our GAN-AC framework with the Bayesian optimization (BO) to demonstrate its practical benefits over other conventional gradient-free design optimization approaches.

Methods

Overview. As illustrated in Fig. 1, the goal of the proposed method is to generate a microstructure that yields targeted physical properties provided by the user. In our design framework, an AI design assistant, i.e. the *actor network*, tunes the input parameters for the 3D GAN, in an attempt to achieve targeted physical properties. Consequently, the 3D GAN generates a 3D synthetic microstructure accordingly and OpenPNM, an open-source package for pore-network modeling³⁹, performs analysis and evaluates the physical quantities of interest (QoI) of the synthetic 3D microstructure. The computed QoI are combined with targeted ones to be utilized by the *critic network* to evaluate how successful the actor's action was in conforming to the targeted design goal. Finally, the *actor network* modifies its behavior, i.e. parameter tuning policy, based on the critic's feedback, and

the iteration continues until the actor becomes capable of generating microstructures with physical properties close to the target.

Training of the proposed algorithm is accomplished in two stages. In the first stage, the 3D-GAN is trained on samples of real microstructure images. As will be discussed later, the *generator network* learns to generate synthetic microstructure images and the *discriminator network* learns to distinguish real microstructure images from synthetic microstructure images. After a number of iterations, when these networks reach to the stage when the *discriminator network* is no longer able to distinguish between synthetic and real microstructure, the training process is complete. Consequently, the *discriminator network* is discarded while the *generator network* is retained to be used for synthetic microstructure reconstruction.

In the second training stage, the AC model is trained to solve the inverse microstructure design problem. Over a number of training episodes, the actor learns and refines its policy to generate morphology parameters for the 3D-GAN generator based on given targeted properties, while the critic learns to assess the quality of synthetic microstructures more accurately. Once trained, the critic network is discarded at the time of deployment.

Synthetic microstructure generation using GAN. GAN has demonstrated abilities in generating more realistic microstructures compared to other CNN-based approaches²⁴. Generally, GAN is trained via an adversarial competition (minimax game) of two neural networks, namely the *generator* \mathcal{G} and the *discriminator* \mathcal{D} . During the training process, the generator \mathcal{G} learns to generate realistic images to deceive the discriminator and the discriminator \mathcal{D} attempts to distinguish real images from generated images. The training objective, i.e. the loss function, can be written as follows:

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

Here, p_{data} denotes the distribution of the real images and z is the input parameter given to the generator. In principle, \mathcal{G} and \mathcal{D} should converge to a Nash equilibrium, in which the discriminator \mathcal{D} is no longer capable of distinguishing real images from generated images. However, in practice, termination criteria are empirically determined for different problems.

Chun et al.²⁴ advanced the above idea of GAN into a spatially parameterized GAN architecture for generating 2D synthetic microstructure images. Despite of their successes with 2D microstructure images, it is not straightforward to extend the architecture to 3D microstructures, mainly due to the morphological complexity of 3D microstructures compared to 2D. First, the 3D convolution layers has a significantly larger number of weights (i.e. network parameters to train) than the 2D version, which consequently increases the dimensionality of the optimization problem as well as the computational burden. In addition, the heavier network forces the use of a smaller batch size, which results in a longer training time and unstable convergence due to inaccurate approximation of the gradients. Furthermore, because 3D microstructures are much more sophisticated, 3D GAN requires a deeper architecture (i.e. more layers) for an increased expressiveness, worsening the numerical complexity even further.

As a result, in the current work, we introduce adjustments to the GAN architecture by Chun et al.²⁴, to address these issues when extending the method for 3D (Fig. 2). First, the number of convolutional blocks in both the generator, \mathcal{G} , and the discriminator, \mathcal{D} , is reduced from five to four to reduce the complexity of the architecture and thereby to enhance the computational efficiency. In addition, for the first three convolutional blocks, we added a stride-1 convolutional layer after each half-stride (stride-2 for the discriminator) convolution layer to enhance the expressiveness of the 3D-GAN. Moreover, batch normalization layers are added at the end of the first three building blocks in both the generator and the discriminator networks, beside the use of leaky ReLU layers to avoid vanishing/exploding gradients and enhance the stability of the training process. Finally, kernels of size $3 \times 3 \times 3$ are used for all convolutional layers. With the new proposed architecture, each block of the generator network scales the dimension of the incoming tensor by a factor of 2, resulting in $128 \times 128 \times 128$ voxels as the final output for a given $8 \times 8 \times 8$ input tensor. Meanwhile, the discriminator is a ‘mirror image’ of the generator and produces a $8 \times 8 \times 8$ output tensor from a given $128 \times 128 \times 128$ microstructure image.

Similar to the original architecture²⁴, our generator interfaces with two types of input parameters, namely the *global morphology* parameters, $\lambda \in \mathbb{R}^{15}$, and the *local stochasticity* parameters, $\rho \in \mathbb{R}^{30}$, defined at each location of an $8 \times 8 \times 8$ grid, resulting in a $8 \times 8 \times 8 \times (15 + 30)$ input tensor. During the training time, λ , is kept constant across all $8 \times 8 \times 8$ grid locations of the input tensor, while ρ , randomly varies across different grid locations. Since the input $8 \times 8 \times 8$ grid is associated with $8 \times 8 \times 8$ overlapping regions in the output image (receptive fields), the microstructure morphology within each output region is controlled by the first 15 global morphology parameters, and the consequent 30, randomly varying local stochasticity parameters. This setting promotes the GAN to generate the same morphology or ‘style’ across all output regions according to λ , while the local details could vary according to the randomly varying ρ .

In terms of training, ADAM⁴⁰ optimizer with the learning rate of 0.0002 are used and all of data is normalized from $[-1, 1]$. Furthermore, unlike the traditional GAN loss function, we evaluate Eq. (1) at each voxel and average them over the spatial dimension. Hence, the new loss function is defined as:

$$\min_{\mathcal{G}} \max_{\mathcal{D}} V(\mathcal{D}, \mathcal{G}) = \frac{1}{whd} \sum_{j=1}^h \sum_{i=1}^w \sum_{k=1}^d [\mathbb{E}_{x \sim p_{\text{data}}(x)} \{ \log \mathcal{D}_{j,i,k}(x) \} + \mathbb{E}_{z \sim p_z(z)} \{ \log (1 - \mathcal{D}_{j,i,k}(\mathcal{G}(z))) \}], \quad (2)$$

where, $\mathcal{D}_{j,i,k}$ refers to discriminator’s prediction at voxel location (j, i, k) . The application of the loss function as in Eq. (2) incentivizes the discriminator and the generator to scrutinize the details of microstructural patterns.

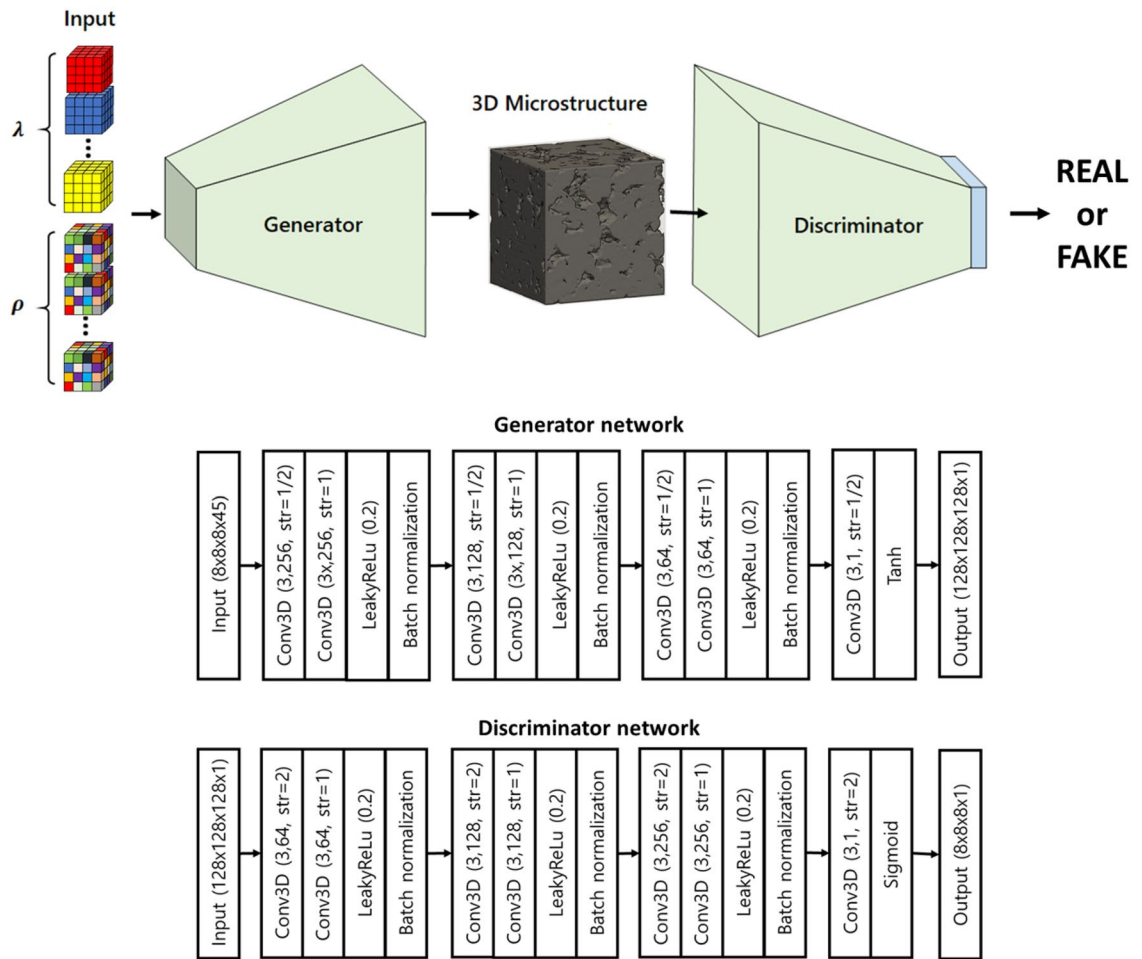


Figure 2. An overview of the 3D GAN architecture.

Inverse synthetic microstructure reconstruction using actor-critic reinforcement learning. The inverse synthetic microstructure reconstruction task can be modeled as a reinforcement learning problem, whereby agents learn to take optimal actions to maximize long-term expected reward from given states of the environment. In the case of inverse synthetic microstructure reconstruction, the agents’ action is a series of adjustments in design parameters with the target of achieving desired QoI. Additionally, a given state of the environment should include information on the current design parameters, the current and the targeted QoI. Generally, the environment is modeled as a Markov decision process (MDP), denoted as $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$. Here, \mathcal{S} is the set of states s_t that the agent can experience and \mathcal{A} is the set of actions a_t that the agent can take during its interaction with the environment. $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is a function which returns the probability over the state space, describing the likelihood of transition from the current state s_t to the future state s_{t+1} under an action a_t . Finally, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is a reward function that defines rewards the agent can receive from the state s_t transitioning under a particular action a_t . In the current microstructure reconstruction framework, the state s_t consists of the current global morphology parameters of the generator, λ_t , the properties of the current synthetic microstructure, $\mathbf{D}_t(\lambda_t)$, and the targeted properties, \mathbf{D}^* . Also, the action a_t determines the adjustment values $\Delta\lambda_t$ that are added to tune the material generator. The consequent state of the environment caused by action a_t under state s_t is $s_{t+1} = (\lambda_t + \Delta\lambda_t, \mathbf{D}_{t+1}(\lambda_t + \Delta\lambda_t), \mathbf{D}^*)$. The reward function gives the total difference between targeted and synthetic normalized properties:

$$r_t(a_t, s_t) = \sum (D - D^*) \tag{3}$$

In Eq. (3), D and D^* are normalized QoI of synthetic and targeted microstructures with values varying between 0 and 1. Initially, relations between the states, the actor’s policy, and the critic are unknown. The goal of the reinforcement learning is to deduce those relations through repeated trial-and-error such that the optimal morphology parameters that yields the most realistic effective properties can be estimated.

The AC model we employ in this paper is comprised of two components. The ‘actor’ determines the action a_t to interact with the environment based on the observed state s_t via the policy function $\pi_\theta(a|s)$. The ‘critic’, on the other hand, estimates the value function with given action (Q-value), $Q^\pi(s_t, a_t)$. Both actor and critic are modeled with deep neural networks. The objective is to learn an optimal policy which maximizes the cumulative

reward, particularly in this case, minimizing the difference between synthetic QoI and the targeted ones, which can be written as:

$$J(\theta) = \sum_{s \in \mathcal{S}} d^\pi(s) \sum_{a \in \mathcal{A}} \pi_\theta(a|s) Q^\pi(s, a) \quad (4)$$

where $d^\pi(s)$ is the stationary distribution of Markov chain for π_θ . The problem of finding the optimal policy can be solved by utilizing policy gradient algorithm:

$$\nabla_\theta J(\theta) \propto \sum_{s \in \mathcal{S}} d^\pi(s) \sum_{a \in \mathcal{A}} Q^\pi(s, a) \nabla_\theta \pi_\theta(a|s) \quad (5)$$

The AC algorithm is a well-known RL framework derived from a temporal different version of policy gradient where the strengths of both policy-based and value-based approaches are leveraged⁴¹. In the AC method, the two networks, $\pi_\theta(s)$ and $Q_w(a, s)$, are updated simultaneously during each training episode. While the actor adjusts parameters to find the optimal policy using Eq. (5), the critic adjust its parameters to minimize the difference between the function approximator $Q_w(a, s)$ and the true action-value function $Q^\pi(s, a)$ ⁴², such that the following is minimized.

$$\mathbb{E}_{s \sim d^\pi, a \sim \pi_\theta} [(Q_w(a, s) - Q^\pi(s, a))^2] \quad (6)$$

Algorithm 1: Off-policy deterministic AC algorithm for inverse material design

```

Initialize behavior critic network  $Q_w(s, a)$  and behavior actor network  $\mu_\theta(s)$ 
Initialize target critic network  $Q'$  and target actor network  $\mu'$ 
Make behavior and target network similar:  $w' \leftarrow w; \theta' \leftarrow \theta$ 
Initialize a replay buffer  $R$ 
for  $episode = 1 \dots M$  do
  Set targeted QoI  $\mathbf{D}^*$ 
  Generate a microstructure with random  $\lambda_t$ 
  Compute generated QoI  $\mathbf{D}_t$ 
  Compute reward  $r_t = \sum(\mathbf{D} - \mathbf{D}^*)$ 
  Receive initial state  $s_t$ 
  for  $t = 1 \dots T$  do
    Perform action  $a_t = \mu_\theta(s_t) + \mathcal{N}(0, 1)$ 
    if  $r_t \geq 0.5$  then
      |  $\Delta = 0.4$ 
    else
      | if  $r_t \geq 0.15$  then
        | |  $\Delta = 0.2$ 
      | else
        | |  $\Delta = 0.1$ 
    Generate new morphology parameter  $\lambda_t = \lambda_t + \Delta a_t$ 
    Generate new microstructures
    Compute QoI  $\mathbf{D}_{t+1}$ 
    Form the next state  $s_{t+1}$ 
    Compute reward  $r_t$  using Equation 3
    Store the state transition  $(s_t, a_t, s_{t+1}, r_t)$  to  $R$ 
    Sample a mini-batch of  $N$  sets  $(s_t, a_t, s_{t+1}, r_t)$  from  $R$ 
    Update the critic network using Equation 9 with  $\alpha_w = 0.002$ 
    Update the actor network using Equation 10 with  $\alpha_\theta = 0.002$ 
    Update target critic and actor parameters
       $w' \leftarrow \tau w + (1 - \tau)w'$ 
       $\theta' \leftarrow \tau \theta + (1 - \tau)\theta'$ 
  Result: Target critic  $Q'$  and target actor  $\mu'$ 

```

Algorithm 2: Inverse microstructure reconstruction

```

Set targeted QoI  $D^*$ 
Initialize a random morphology parameters  $\lambda^*$ 
Initialize an extremely big reward value  $r^*$ 
Generate a microstructure with random  $\lambda$ 
Compute generated QoI  $D$ 
Compute current reward  $r_t$  using Equation 3
Receive initial state  $s$ 
for  $t = 1 \dots T$  do
  Perform action  $a = \mu'_\theta(s)$ 
  if  $r_t \geq 0.5$  then
     $\Delta = 0.4$ 
  else
    if  $r_t \geq 0.15$  then
       $\Delta = 0.2$ 
    else
       $\Delta = 0.1$ 
    Generate new morphology parameter  $\lambda = \lambda + \Delta a_t$ 
    Generate new microstructures with  $\lambda$ 
    Compute QoI  $D$ 
    Compute reward  $r$  using Equation 3
  if  $r_t < \varepsilon$  then
     $\lambda^* = \lambda$ 
    Terminate the process
  if  $r_t < r^*$  then
     $\lambda^* = \lambda$ 
Result: Optimal morphology parameter  $\lambda^*$ 

```

As studied in the work of Konda and Tsitsiklis⁴¹, the AC method tends to converge more smoothly and have better training performance for a system with a large state space.

In addition, since the synthesized microstructures are controlled with continuous morphology variables, a deterministic policy gradient (DPG) algorithm^{42,43} is applied. Particularly, in DPG, the actor decides the action to interact with the environment by a deterministic policy, such as $a = \mu(s)$. As a result, the performance objective and the update scheme for both actor and critic are as follows:

$$J(\theta) = \int_S \rho^\mu(s) Q(s, \mu_\theta(s)) ds \quad (7)$$

$$\delta_t = [r_t + \gamma Q_w(s_{t+1}, a_{t+1}) - Q_w(s_t, a_t)]^2 \quad (8)$$

$$w_{t+1} = w_t + \alpha_w \delta_t \nabla_w Q_w(s_t, a_t) \quad (9)$$

$$\theta_{t+1} = \theta_t + \alpha_\theta \nabla_a Q_w(s_t, a_t) \nabla_\theta \mu_\theta(s) |_{a=\mu_\theta(s)} \quad (10)$$

Further, in this work, an off-policy AC method, which employs two different sets of actor-critic networks (behavior and target), is used. In off-policy AC, behavior networks take the role of exploration while the target ones take the role of learning to find the optimal morphology parameters. The application of off-policy AC brings two main advantages: (1) full trajectories are not required and the “experience replay” can be performed, and (2) the action sampling follows a behavior policy which is different from the target policy, providing better exploration. The training and inference processes are described clearly in Algorithms 1 and 2. In addition, the architecture of the proposed properties-driven morphology control utilizing AC algorithm is as in Fig. 3 and Gym⁴⁴ is utilized for implementation. During training, the number of training episodes is set to 1000 with each episode containing 32 design iterations. Also, a total of 32,000 “experiences” can be saved in the replay buffer. A mini-batch size of 64 is used to update both actor and critic network using the ADAM optimizer⁴⁰. Once the training is complete, the ‘target actor’ and ‘target critic’ are used for inference. Moreover, as similar to the training settings, the maximum of 32 design iterations are also set for the inference described in Algorithm 2.

Results and discussion

Datasets. To test our microstructure reconstruction framework, we used a micro computed tomography (CT) scans data of Bentheim sandstones (*BM_B1*) provided by our colleague (Prof. Teng-fong Wong) as the training/validation data. The voxel size of the original data was $704 \times 678 \times 500$, from which $256 \times 256 \times 256$ sub-volumes were sampled at random positions and then coarsen into $128 \times 128 \times 128$. The resolution of the coarsened data is 8 micron per voxel and the size of representative elementary volume is $1024 \mu m \times 1024 \mu m \times 1024 \mu m$.

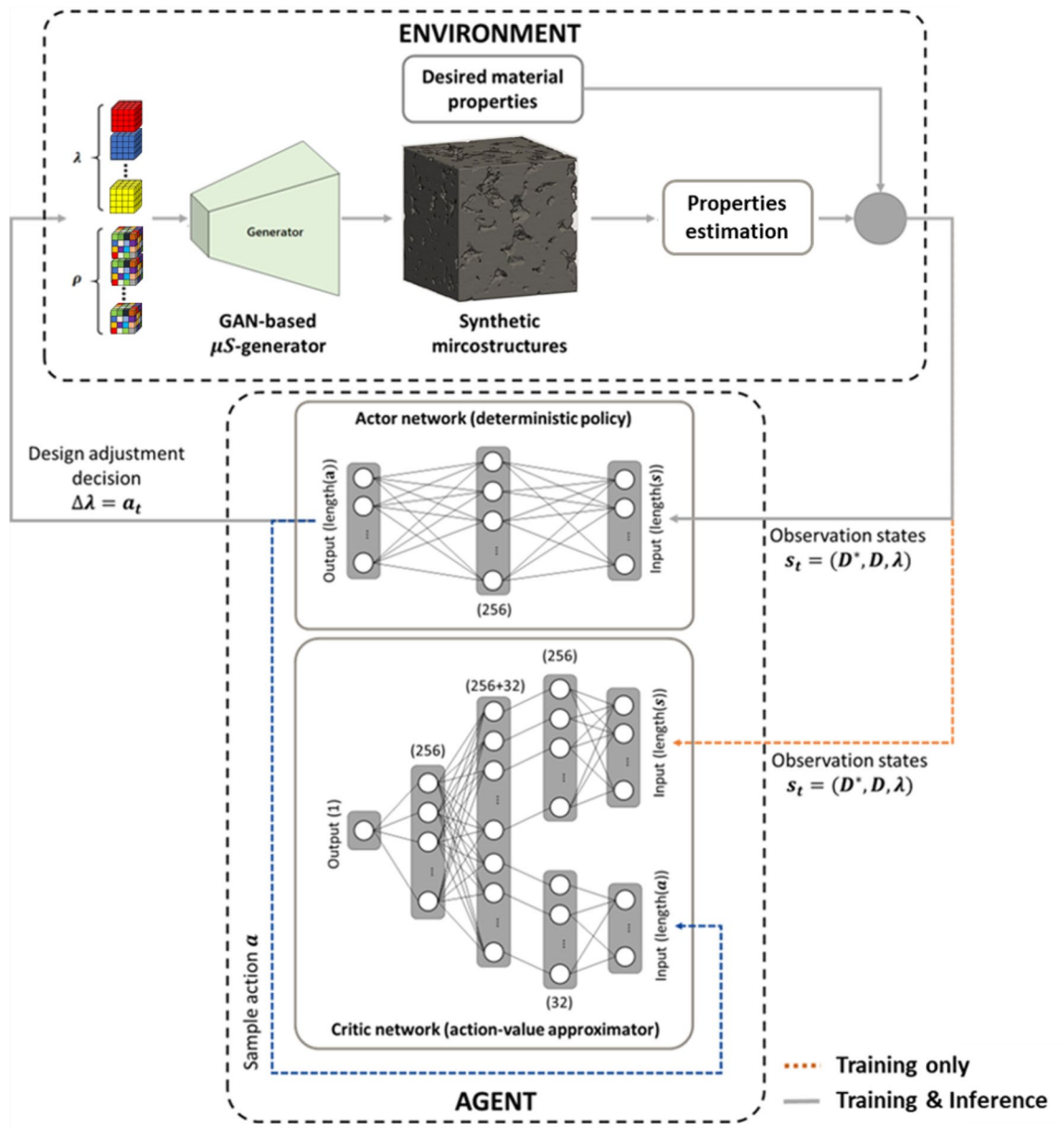


Figure 3. A schematic overview of the morphology control via an AC model. The design problem is formulated as a sequential process, adjusting global parameters at each step so that the synthetic microstructure transforms the morphology as desired. During the training process, *actor network* repeatedly gives design decisions based on its policy to interact with the *environment*. Physical properties of synthetic microstructures are computed in the *environment* and given as feedback to both *actor network* and *critic network*. The *critic network* then evaluates how successful the *actor network's* action was in conforming the targeted properties. Finally, the *actor network* utilizes the evaluation from the *critic network* to adjust its policy in order to generate synthetic microstructures with physical properties closer to the targeted ones. Once the training is finished, the *critic network* is discarded and only the *actor network* is retained to perform the design task.

Evaluation metrics. We use the following statistical, geometric, and topological attributes of the generated images and the real microstructures to measure whether the synthetic microstructures possess topological and physical properties consistent with the real specimen.

Kullback–Leibler divergence (KLD)⁴⁵ is a statistical distance that can be used to measure how the distribution of synthetic data $y \sim P_g$ differs from that of real data $x \sim P_r$:

$$KLD(P||Q) = \mathbb{E}_{x \sim P_r} \left[\log \frac{P_r(x)}{P_g(x)} \right], \tag{11}$$

where $P_r(x)$ and $P_g(x)$ are derived empirically using e.g., Kernel Density Estimation:

$$p(x) = \sum_{i=1}^n K(\mathbf{x}, \mathbf{x}_i; h) \quad (12)$$

where $K(\mathbf{x}, \mathbf{x}'; h) \propto \frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2h^2}$ is the Gaussian kernel function and $\mathbf{x}_i \in \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ is a sample in the data set.

Mean Maximum Discrepancy (MMD)⁴⁶ measures the dissimilarity between two probability distributions using samples drawn independently from each distribution. Given a finite set of synthetic microstructures $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\} \sim P_r$ and a finite set of real microstructures $\mathbf{y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n\} \sim P_g$, the MMD of the two sets is defined as:

$$MMD = \left[\mathbb{E}_{\mathbf{x} \sim P_r, \mathbf{y} \sim P_g} [K(\mathbf{x}, \mathbf{x}') - 2K(\mathbf{x}, \mathbf{y}) + K(\mathbf{y}, \mathbf{y}')] \right]^{1/2}. \quad (13)$$

Earth Mover's Distance (EMD)³¹ is the minimum mass displacement to transform one distribution into the other, which is also known as the Wasserstein distance:

$$W(P_r, P_g) \propto \max_f \mathbb{E}_{\mathbf{x} \sim P_r} [f(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim P_g} [f(\mathbf{x})], \quad (14)$$

where $f: \mathbb{R}^D \rightarrow \mathbb{R}$ is a Lipschitz continuous function, called the Wasserstein critic. In practice, f can be modeled as a neural network with clipped weights to have bounded derivatives, which is trained to derive high values at real samples and low values at generated samples, i.e. maximize:

$$\hat{W}(\mathbf{x}_{test}, \mathbf{x}_g) = \frac{1}{N} \sum_{i=1}^N \hat{f}(\mathbf{x}_{test}[i]) - \frac{1}{N} \sum_{j=1}^N \hat{f}(\mathbf{x}_g[j]) \quad (15)$$

Pore space measurements The pore space metrics measures the size and the tortuosity of the pore space and include the *total porosity* (ratio between the size of the void space and total volume) the *specific surface area* (the total area divided by the total volume). The total porosity is computed by the total number of void voxels in the cubic image divided by the size of the cube. The specific surface area is estimated by counting the total number of voxels that are at the interface between the void and solid constituent divided by the total volume of the cubic image.

Effective permeability The effective permeability of the GAN-generated micro-structure is the last and the most important metric that measures whether the GAN-generated micro-structures mimics the real micro structures. In brief, the effective permeability of a given porous medium to a fluid phase (oil, water, gas) is the ability of that phase to flow inside that medium given a hydraulic gradient⁴⁷. In this work, the effective permeability is estimated with `OpenPNM`³⁹.

Topology measurements The topology metrics measure is designed to measure the topological similarity of the pore connectivity graph generated from the GAN and those obtained from micro-CT imaging. To generate the connectivity graph, an open source software called `PORESpy`⁴⁸ is used to convert the binary images into weighted graphs and the properties of the weighted graphs are measured. The definition of these graph measures are listed below. They are calculated using the open-source software `NetworkX`⁴⁹ for exploration and analysis of graph networks.

- **Degree assortativity** The degree assortativity coefficient measures the similarity of the connections in a graph with respect to the node degree.
- **Graph transitivity** The graph transitivity is the fraction of all possible triangles present in the graph over the number of triads. Possible triangles are identified by the number of triads—two edges with a shared vertex.
- **Graph density** The density for undirected graphs is defined as $d = \frac{2m}{n(n-1)}$, where n is the number of nodes and m is the number of edges of the graph.
- **Average clustering** The average clustering coefficient of the graph is defined as $C = \frac{1}{n} \sum_{v \in G} c_n$, where n is the number of nodes. The clustering coefficient c_n of node n is defined as $c_n = \frac{2T(n)}{\deg(n)(\deg(n)-1)}$, where $T(n)$ is the number of triangles passing through node n and $\deg(n)$ is the degree of node n .
- **Efficiency** The efficiency of a pair of nodes is defined as the reciprocal of the shortest path distance between the nodes. The *local efficiency* of a node in the graph is the average *global efficiency* of the subgraph induced by the neighbours of the node. The *average local efficiency*, used in this work, is the average of the local efficiency calculated for every node in the graph.

Evaluation of 3D-GAN synthetic microstructures. Figure 4 illustrates the comparison between the real and synthetic microstructures. By observing both 3D models and corresponding 2D slices, the synthetic microstructure is similar to the real microstructure as it is difficult to distinguish between them. Moreover, the shapes of voids and crystals of synthetic microstructures are also diverse and realistic, thus, showing that the proposed 3D-GAN-based material generator is capable of emulating the complex micro-geometry of natural materials. In addition, Fig. 5 shows the effects of the global morphology parameters and the local stochasticity parameters. As shown in the figure, the global morphology parameters result in the change in overall “style” of the synthetic microstructures, whereas the local stochasticity parameters create minor local variations in the microstructure morphology.

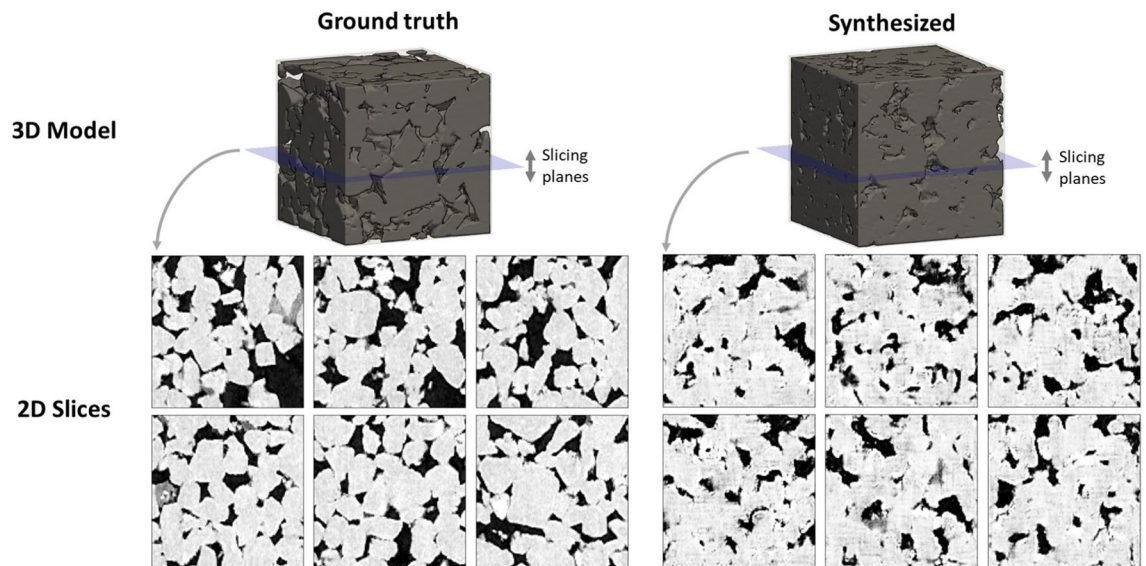


Figure 4. Comparison of 2D slices and 3D volume renders of ground truth microstructures and GAN-generated microstructures. In terms of the visual quality, the synthetic microstructures look realistic such that it is difficult to distinguish between the real and synthetic microstructures. Further, the synthetic voids and crystals also look realistic and diverse.

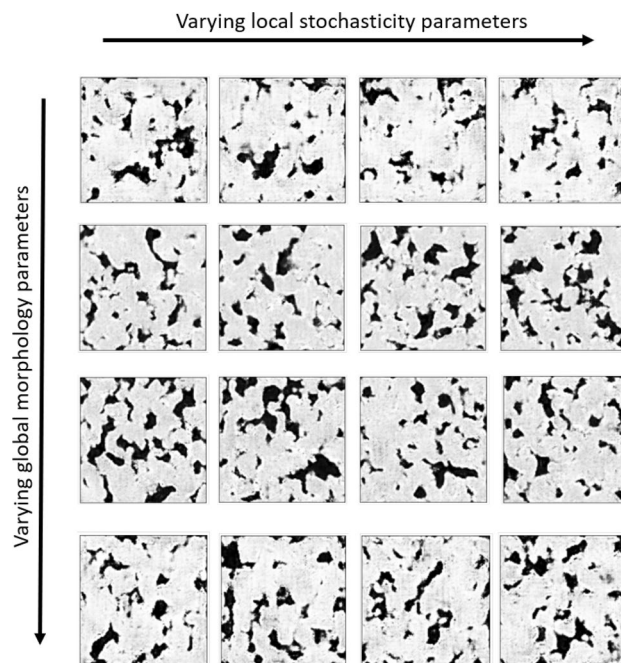


Figure 5. Effects of the local stochasticity parameter ρ . All images are 2D slices of synthetic microstructures taken by cutting planes at the same height position. Microstructures in the same row have the same morphology parameters but different local stochasticity parameters. As can be seen, their overall “style” is maintained and there are only subtle differences between microstructures slice images.

In addition, Table 1 quantitatively evaluates the quality of GAN-generated microstructures. For this comparison, we prepared two disjoint sets of real microstructures and a set of synthetic microstructures. Each of those three sets had 100 randomly selected samples. We computed the KLD, MMD, and EMD distances between the statistical distributions of real microstructures and synthetic microstructures. The statistical distance between the two sets of real microstructures were also computed to be used as the baseline for comparison. Results reported in Table 1 indicate that there is only negligible difference between real and synthetic microstructures, in terms of their statistical distributions.

Metric	Synthetic–real	Real–real
Kullback–Leibler divergence (KLD)	0.0235	0.0197
Maximum mean discrepancy (MMD)	0.0190	0.0175
Earth mover’s distance (EMD)	0.0799	0.0899

Table 1. Evaluation of synthetic microstructures generated by GAN.

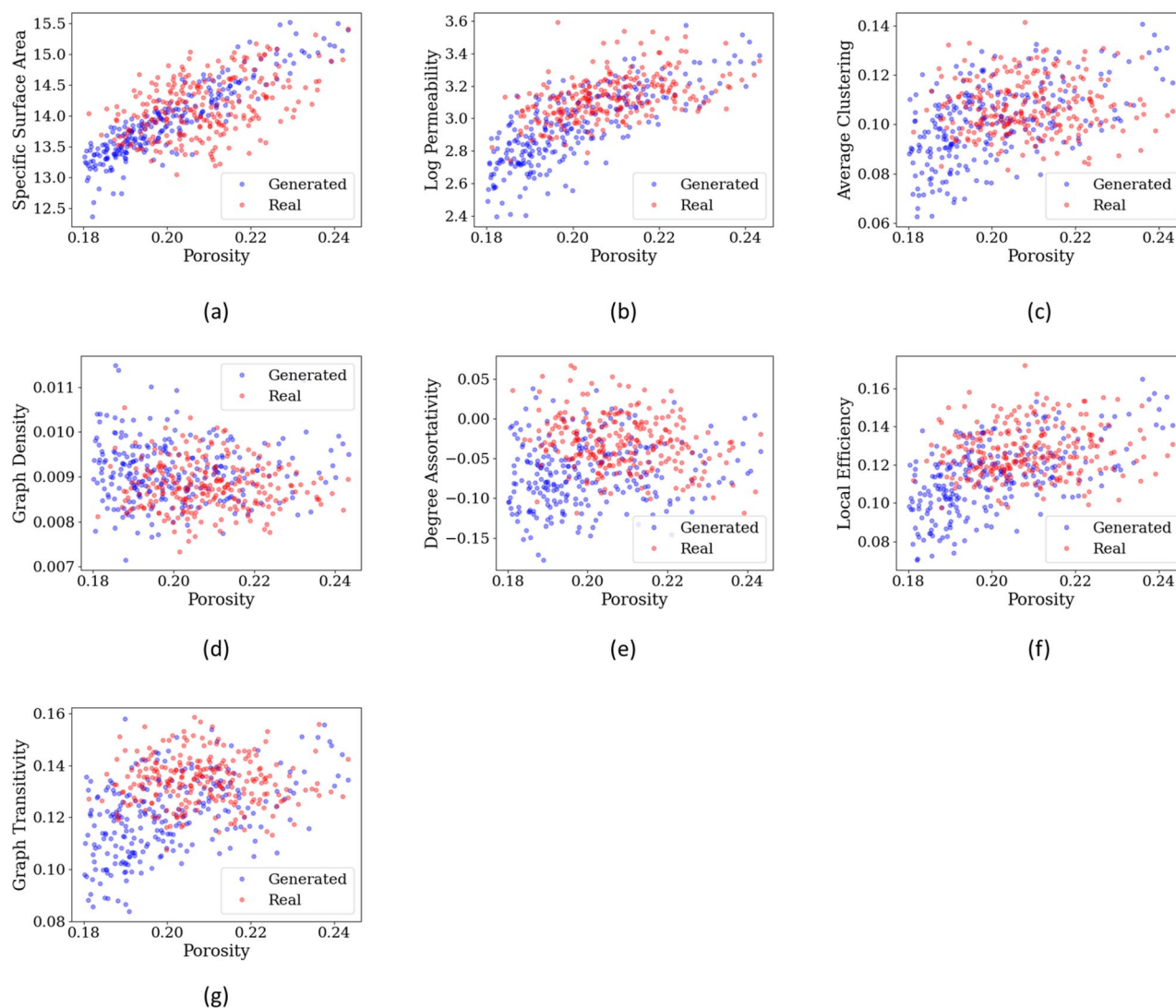


Figure 6. Quantitative evaluation of 3D synthetic microstructures. Pore space measurements, topology measurements, and permeability are plotted w.r.t porosity. **(a)** specific surface area, **(b)** log permeability (md), **(c)** average clustering, **(d)** graph density, **(e)** degree assortativity coefficient, **(f)** local efficiency, and **(g)** graph transitivity. As seen in the plots, there is an agreement between physical properties of ground truth microstructures and synthetic microstructures; thus, proving the validity of the proposed 3D-GAN microstructure generator.

Furthermore, we also validated GAN-generated microstructures using several physics-based metrics, including porosity, specific surface area, permeability, average clustering, graph density, degree assortativity coefficient, local efficiency, and graph transitivity. As reported in Fig. 6, physical properties of synthetic microstructures are, in general, in a good agreement with those of the real microstructure data and the distributions of the properties coincide with each other. Moreover, it is worth noting that the proposed 3D-GAN generator can generate synthesized microstructures with porosities that cover the range between 0.1 and 0.25 which is larger than that of the ground truth which only covers porosities between 0.18 to 0.25. This extrapolation capacity could be helpful on extending the material databases for characteristics provided that the micro-structural attributes of RVE outside the training data range remain sufficiently similar⁶.

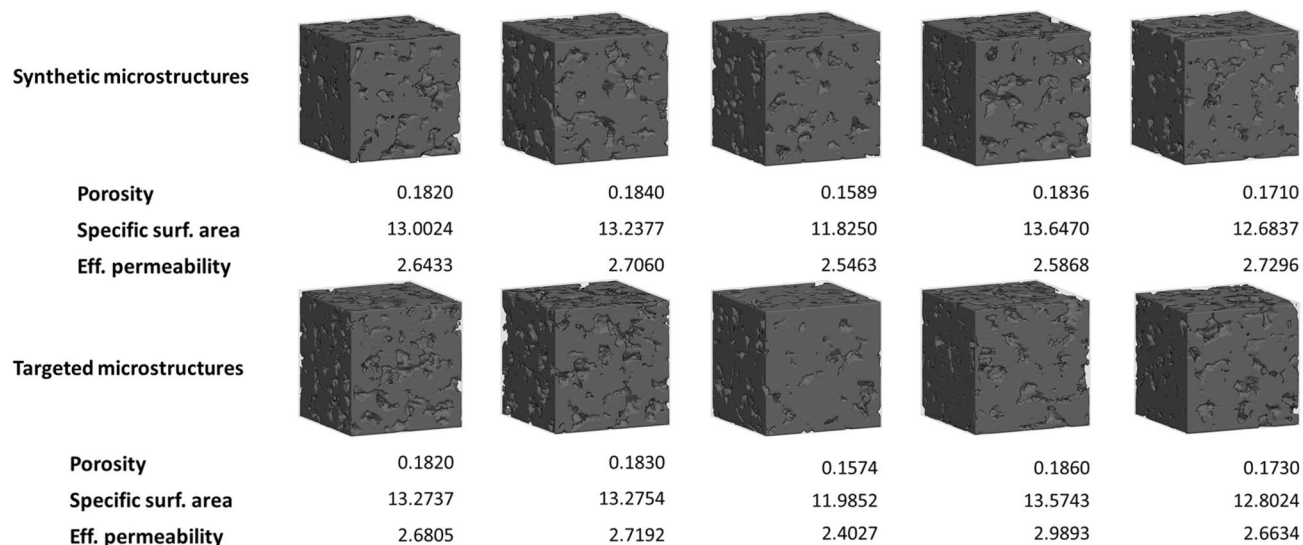


Figure 7. Visualization of synthetic microstructures generated by the proposed GAN-AC framework (top) and their corresponding targeted microstructures (bottom). The 3D voxelized data has been made available to public for inspection and third-party validation.

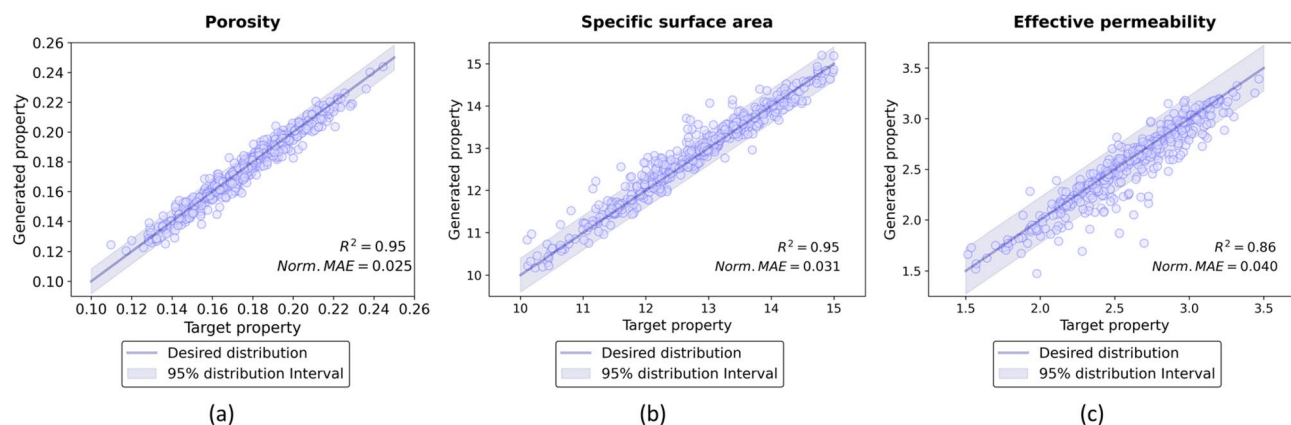


Figure 8. Comparison between targeted and generated properties. On the test conducted on 500 examples that were set aside during training, synthetic microstructures generated by the GAN-AC framework exhibited the targeted physical properties with extremely small difference, 2.54, 3.51, and 4.06% for porosity, specific surface area, and effective permeability, respectively.

Evaluation of the GAN-AC framework for synthetic microstructure reconstruction. The validation of the GAN-AC framework is conducted via a microstructure reconstruction problem with targeted QoI including porosity, specific surface area, and effective permeability. The 3D-GAN model is utilized as the microstructure generator and an AC model, trained by using Algorithm 1, is attached to control the morphology parameters of the generator. The inverse design process follows Algorithm 2.

Figure 7 shows some examples of synthetic microstructures generated by the proposed GAN-AC framework with their physical properties. Additionally, the referenced microstructures with targeted physical properties were added for comparison. Generally, although synthetic microstructures presented in Fig. 7 are visually different with the targeted ones, their physical QoI are relatively similar. Moreover, Fig. 8 illustrates the quantitative evaluation of 500 synthetic microstructures by comparing the synthetic and targeted QoI. As illustrated, the synthetic microstructures can exhibit the properties of targeted ones with the average error of 2.54, 3.51, and 4.06% for porosity, specific surface area, and effective permeability, respectively. These evaluation results demonstrate the validity of the GAN-AC framework in terms of inverse microstructure design. We have also made 10 synthetic 3D microstructures available via an open-access data repository, Mendeley data.

Comparison with other design optimization approaches. Bayesian optimization (BO) is a well-known gradient-free optimization technique that is popularly applied for solving inverse design problems in material science^{33,50}. Acknowledging its state of the art performance, we use BO as a reference of comparison to further investigate the performance of our proposed GAN-AC framework. For the implementation of BO with

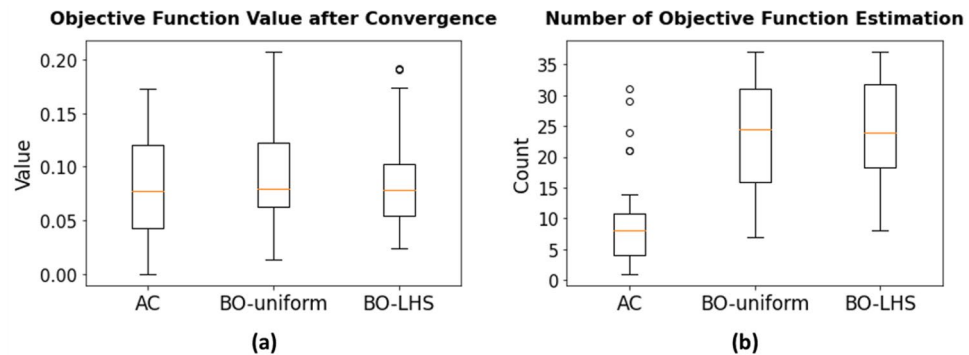


Figure 9. Performance comparison between our GAN-AC framework and a conventional optimization approach (Bayesian optimization; BO). **(a)** While the quality of the solution is on par compared to BO with Latin Hypercube sampling and marginally better than BO with random uniform sampling, **(b)** the AC method is significantly more efficient than BO with both initial sampling strategies (about three times) in terms of the number of objective function evaluation.

the GAN microstructure generator, we employ the following steps. First, an initial dataset is created by sampling within the 15-dimensional (global parameters) GAN latent space. Second, the QoIs of the initial microstructures are computed and the Gaussian process model is fit to the initial data. From the predicted uncertainty of the Gaussian process model (acquisition function), the next sampling point is determined. Finally, the QoIs of the new sample are estimated and added to the database for updating the Gaussian process model and the current solution. The same process is repeated until the convergence criteria is met or the algorithm reach the maximum number of iterations.

As similar to the AC method, BO also targets to minimize the objective function in Eq. (3) which measures the difference between desired and generated QoIs. We selected expected improvement (EI) as our acquisition function to balance between exploitation and exploration⁵⁰. In addition, the maximum number of iterations is set to 32 to make its working condition similar to that of the GAN-AC model during inference. Finally, since the performance of BO is highly affected by initial sampling, we examined two different sampling strategies, including Latin Hypercube sampling (BO-LHS) and random uniform sampling (BO-uniform). In addition, the number of initial samples is set to 5 empirically, to assure the best quality of BO solution with the lowest number of objective function estimations.

Figure 9 shows the performance comparison between AC and BO. As can be observed from Fig. 9a, both the AC and BO exhibited a similar performance in terms of the converged minimum. In particular, while AC-computed minima were 0.0836 ± 0.0479 , the BO-uniform-computed minima were 0.0934 ± 0.0450 and BO-LHS-computed minima were 0.0842 ± 0.0407 , for 50 different optimization tasks with various target QoIs. In addition, despite of requiring 1000 training optimization instances, AC was significantly (about 3 times) faster in terms of the number of objective function evaluation, in which AC only required the average of 8.7 function evaluations whereas BO-uniform required 23.7 function evaluations and BO-LHS required 24.4 function evaluations on average.

In fact, it is widely accepted that reinforcement learning approaches in general are more efficient and scalable than conventional design optimization approaches (e.g., BO or genetic algorithm)^{34,35,51}. The problem of generating microstructures is highly complex because the wide variation of grain/pore sizes, aspect ratios, orientations, and many other morphological attributes spans a vastly large design space. While traditional optimization methods are often limited by the complexity of the design problems³⁴, the proposed GAN-AC framework can be significantly faster than traditional optimization algorithms during the inference time, even though the training could be computationally demanding and data-intensive. Therefore, for design optimization applications where the same type of material needs to be repetitively optimized for different design targets, reinforcement learning approaches provide a substantial advantage over traditional optimization algorithms^{34,35}. In addition, reinforcement learning approaches are problem-aware, as opposed to the traditional optimization approaches that are problem agnostic. Reinforcement learning (RL) algorithms can learn the “landscape” of the design space, and they become better at searching for the nearest optimum as they accumulate more experiences, making it more generalized and is suitable for repetitive optimization tasks⁵². Also, it is worth to note that, the generalizability of the proposed GAN-AC method is only stopped at unseen optimization tasks where only targeted QoI values are different. For the design problems whereby different types of targeted QoI or different materials are required, BO is still more beneficial as RL requires a retraining of the model. Although methods such as transfer learning can reduce the effort of retraining RL algorithms, their effectiveness is still need to be further investigated.

Conclusions

In this work, a novel method for inverse synthetic microstructure reconstruction utilizing 3D-GAN and off-policy deterministic AC reinforcement learning is proposed. Experimental studies on a dataset collected from X-ray CT scans of a Bentheim sandstone justified the validity of the proposed method. As reported, the synthetic microstructures are diverse and realistic in qualitative and quantitative comparison with the real microstructures.

Further, the application of the AC model also enables the controllability on the physical properties of synthetic microstructures. The quantitative analysis of three given physical properties, including porosity, specific surface area, and effective permeability, results in a good agreement between synthetic and targeted physical QoI. The results demonstrate the capability of the proposed method in mimicking original microstructures both geometrically and physically.

For our future work, there are a few extensions to be made to make the GAN-AC framework more practical and usable. For example, a multi-agent AC model with multiple actors could be adopted to accelerate the explorations. Multi-actor models with decentralized policies and shared experience, for instance, may enable more effective exploration-exploitation than the current single-actor counterpart. Furthermore, we may consider capturing the alestoric and epistemic uncertainties of real microstructures. Uncertainty quantification is necessary in practice, as the process-structure-property relationships of real-world materials incorporates a great deal of stochasticity, arising from manufacturing, measurement, and modeling errors⁵³. Such randomness could be addressed by solving a stochastic inverse problem whereby a stochastic solution, instead of a deterministic one, is derived. In addition, the manufacturability, or more broadly, the process-structure relationships were not included in the scope of this current work, which must be further explored as an immediate future work. Finally, physics-informed machine learning, in which the neural network architecture is designed to fulfill the relevant physical constraints (e.g. material symmetry, invariance properties), can also be applied to further assure the validity of the synthetic microstructures.

Data availability

The data set generated and analyzed during the current study are available in Mendeley Data (<https://data.mendeley.com/datasets/tp9nynzc34/1>).

Received: 29 November 2021; Accepted: 17 May 2022

Published online: 31 May 2022

References

1. Torquato, S. & Haslach, H. Jr. Random heterogeneous materials: microstructure and macroscopic properties. *Appl. Mech. Rev.* **55**, B62–B63 (2002).
2. Glaessgen, E. & Stargel, D. The digital twin paradigm for future nasa and us air force vehicles. In *53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA*, 1818 (2012).
3. Zohdi, T. A machine-learning framework for rapid adaptive digital-twin based fire-propagation simulation in complex environments. *Comput. Methods Appl. Mech. Eng.* **363**, 112907 (2020).
4. Gupta, R., Salager, S., Wang, K. & Sun, W. Open-source support toward validating and falsifying discrete mechanics models using synthetic granular materials-Part I: Experimental tests with particles manufactured by a 3D printer. *Acta Geotech.* **14**, 923–937 (2019).
5. Tropsha, A., Mills, K. C. & Hickey, A. J. Reproducibility, sharing and progress in nanomaterial databases. *Nat. Nanotechnol.* **12**, 1111–1114 (2017).
6. Liu, Y., Sun, W. & Fish, J. Determining material parameters for critical state plasticity models based on multilevel extended digital database. *J. Appl. Mech.* **83**, 1–10 (2016).
7. Fredrich, J., DiGiovanni, A. & Noble, D. Predicting macroscopic transport properties using microscopic image data. *J. Geophys. Res. Solid Earth* **111**, 123 (2006).
8. Valsecchi, A., Damas, S., Tubilleja, C. & Arechalde, J. Stochastic reconstruction of 3d porous media from 2d images using generative adversarial networks. *Neurocomputing* **399**, 227–236 (2020).
9. Li, X. *et al.* A transfer learning approach for microstructure reconstruction and structure-property predictions. *Sci. Rep.* **8**, 1–13 (2018).
10. Jiao, Y., Stillinger, F. & Torquato, S. Modeling heterogeneous materials via two-point correlation functions: Basic principles. *Phys. Rev. E* **76**, 031110. <https://doi.org/10.1103/PhysRevE.76.031110> (2007).
11. Xu, H., Dikin, D. A., Burkhart, C. & Chen, W. Descriptor-based methodology for statistical characterization and 3d reconstruction of microstructural materials. *Comput. Mater. Sci.* **85**, 206–216. <https://doi.org/10.1016/j.commatsci.2013.12.046> (2014).
12. Hart, K. A. & Rimoli, J. J. Generation of statistically representative microstructures with direct grain geometry control. *Comput. Methods Appl. Mech. Eng.* **370**, 113242. <https://doi.org/10.1016/j.cma.2020.113242> (2020).
13. Jiang, Z., Chen, W. & Burkhart, C. Efficient 3D porous microstructure reconstruction via Gaussian random field and hybrid optimization. *J. Microsc.* **252**, 135–148. <https://doi.org/10.1111/jmi.12077> (2013).
14. Koutsourelakis, P. S. & Deodatis, G. Simulation of multidimensional binary random fields with application to modeling of two-phase random media. *J. Eng. Mech.* **132**, 619–631 (2006).
15. Sakamoto, S. & Ghanem, R. Simulation of multi-dimensional non-gaussian non-stationary random fields. *Probab. Eng. Mech.* **17**, 167–176 (2002).
16. Ma, W. *et al.* Image-driven discriminative and generative machine learning algorithms for establishing microstructure-processing relationships. *J. Appl. Phys.* **128**, 134901 (2020).
17. Wang, Y. *et al.* Mining structure-property relationships in polymer nanocomposites using data driven finite element analysis and multi-task convolutional neural networks. *Mol. Syst. Des. Eng.* **5**, 962–975 (2020).
18. Bostanabad, R. *et al.* Computational microstructure characterization and reconstruction: Review of the state-of-the-art techniques. *Progress Mater. Sci.* **95**, 1–41 (2018).
19. Yang, Z. *et al.* Microstructural materials design via deep adversarial learning methodology. *J. Mech. Des.* **140**, 1–12 (2018).
20. Lubbers, N., Lookman, T. & Barros, K. Inferring low-dimensional microstructure representations using convolutional neural networks. *Phys. Rev. E* **96**, 052111 (2017).
21. Li, X. *et al.* A transfer learning approach for microstructure reconstruction and structure-property predictions. *Sci. Rep.* <https://doi.org/10.1038/s41598-018-31571-7> (2018).
22. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. (2014). <http://arxiv.org/abs/1409.1556>.
23. Gatys, L., Ecker, A. S. & Bethge, M. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems* (eds Cortes, C. *et al.*) 262–270 (Curran Associates Inc, 2015).
24. Chun, S. *et al.* Deep learning for synthetic microstructure generation in a materials-by-design framework for heterogeneous energetic materials. *Sci. Rep.* **10**, 13307 (2020).

25. Kench, S. & Cooper, S. J. Generating three-dimensional structures from a two-dimensional slice with generative adversarial network-based dimensionality expansion. *Nat. Mach. Intell.* **3**, 299–305. <https://doi.org/10.1038/s42256-021-00322-1> (2021).
26. Wu, J., Zhang, C., Xue, T., Freeman, B. & Tenenbaum, J. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems* (eds Lee, D. et al.) (Curran Associates Inc, 2016).
27. Smith, E. J. & Meger, D. Improved adversarial systems for 3d object generation and reconstruction. In Levine, S., Vanhoucke, V. & Goldberg, K. (eds.) *Proceedings of the 1st Annual Conference on Robot Learning*, vol. 78 of *Proceedings of Machine Learning Research*, 87–96 (PMLR, 2017).
28. Shu, D. et al. 3D design using generative adversarial networks and physics-based validation. *J. Mech. Des.* **142**, 071701. <https://doi.org/10.1115/1.4045419> (2019).
29. Mosser, L., Dubrule, O. & Blunt, M. J. Stochastic reconstruction of an oolitic limestone by generative adversarial networks. *Transp. Porous Media* **125**, 81–103. <https://doi.org/10.1007/s11242-018-1039-9> (2018).
30. Hsu, T. et al. Microstructure generation via generative adversarial network for heterogeneous, topologically complex 3d materials. *JOM* **73**, 90–102. <https://doi.org/10.1007/s11837-020-04484-y> (2021).
31. Arjovsky, M., Chintala, S. & Bottou, L. *Wasserstein gan* **1701**, 07875 (2017).
32. Gayon-Lombardo, A., Mosser, L., Brandon, N. P. & Cooper, S. J. Pores for thought: The use of generative adversarial networks for the stochastic reconstruction of 3d multi-phase electrode microstructures with periodic boundaries. <https://doi.org/10.48550/ARXIV.2003.11632> (2020)
33. Yang, Z. et al. Microstructural materials design via deep adversarial learning methodology. *J. Mech. Des.* **140**, 111416 (2018).
34. Gongora, A. E. et al. Designing composites with target effective young's modulus using reinforcement learning. in *Symposium on Computational Fabrication* (Association for Computing Machinery, 2021).
35. Hui, X. et al. Multi-object aerodynamic design optimization using deep reinforcement learning. *AIP Adv.* **11**, 085311. <https://doi.org/10.1063/5.0058088> (2021).
36. Chen, Q., Wang, J., Pope, P., Chen, W. & Fuge, M. Inverse design of two-dimensional airfoils using conditional generative models and surrogate log-likelihoods. *J. Mech. Des.* **144**, 021712. <https://doi.org/10.1115/1.4052846> (2021).
37. Yilmaz, E. & German, B. Conditional generative adversarial network framework for airfoil inverse design. in *AIAA AVIATION 2020 FORUM* (2020). <https://arc.aiaa.org/doi/pdf/10.2514/6.2020-3185>.
38. Achour, G., Sung, W. J., Pinon-Fischer, O. J. & Mavris, D. N. Development of a conditional generative adversarial network for airfoil shape optimization. in *AIAA Scitech 2020 Forum* (2020). <https://doi.org/10.2514/6.2020-2261>. <https://arc.aiaa.org/doi/pdf/10.2514/6.2020-2261>.
39. Gostick, J. et al. OpenPNM: A pore network modeling package. *Comput. Sci. Eng.* **18**, 60–74 (2016).
40. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization (2017). <http://arxiv.org/abs/1412.6980>.
41. Konda, V. R. & Tsitsiklis, J. N. On actor-critic algorithms. *SIAM J. Control Optim.* **42**, 1143–1166. <https://doi.org/10.1137/S0363012901385691> (2003).
42. Silver, D. et al. Deterministic policy gradient algorithms. in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, 1–387–I–395 (JMLR.org, 2014).
43. Lillicrap, T. P. et al. Continuous control with deep reinforcement learning (2019). <http://arxiv.org/abs/1509.02971>.
44. Brockman, G. et al. OpenAI Gym. (2016). <http://arxiv.org/abs/1606.01540>.
45. Joyce, J. M. *Kullback–Leibler Divergence* 720–722 (Springer, 2011).
46. Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B. & Smola, A. A kernel two-sample test. *J. Mach. Learn. Res.* **13**, 723–773 (2012).
47. Abdus, S. & Ghulam, M. I. Chapter 3: Reservoir rock properties. In *Reservoir Engineering*, 29–79 (Gulf Professional Publishing, 2016). <https://doi.org/10.1016/B978-0-12-800219-3.00003-6>.
48. Gostick, J. T. et al. PoreSpy: A python toolkit for quantitative analysis of porous media images. *J. Open Source Softw.* **4**, 1296 (2019).
49. Hagberg, A., Swart, P. & S Chult, D. Exploring network structure, dynamics, and function using networkx. Tech. Rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States) (2008).
50. Zhang, Y., Apley, D. W. & Chen, W. Bayesian optimization for materials design with mixed quantitative and qualitative variables. *Sci. Rep.* **10**, 4924 (2020).
51. Sui, F., Guo, R., Zhang, Z., Gu, G. X. & Lin, L. Deep reinforcement learning for digital materials design. *ACS Mater. Lett.* **3**, 1433–1439 (2021).
52. Yonekura, K. & Hattori, H. Framework for design optimization using deep reinforcement learning. *Struct. Multidiscipl. Optim.* **60**, 1709–1713. <https://doi.org/10.1007/s00158-019-02276-w> (2019).
53. Tran, A. & Wildey, T. Solving stochastic inverse problems for property–structure linkages using data-consistent inversion and machine learning. *JOM* **73**, 72–89. <https://doi.org/10.1007/s11837-020-04432-w> (2021).

Acknowledgements

The authors are grateful for the support by the U.S. Air Force Office of Scientific Research (AFOSR) Multidisciplinary University Research Initiative (MURI) program (Grant No. FA9550-19-1-0318; PM: Dr. Martin Schmidt, Dynamic Materials program). The images of the Bentheim sandstone is provided by Professor Teng-fong Wong from Stony Brook University.

Author contributions

W.S., H.U., and S.B. conceived the methods, W.S. and S.B. designed and managed the experiments, P.N., N.V., and B.B. conducted and analyzed the experiments, and P.N. and S.B. wrote the main manuscript text. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to S.S.B.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2022