

Denoising diffusion algorithm for inverse design of microstructures with fine-tuned nonlinear material properties

Nikolaos N. Vlassis, WaiChing Sun*

*Department of Civil Engineering and Engineering Mechanics, Columbia University, 614 SW Mudd, Mail Code: 4709,
New York, NY 10027, United States of America*

Received 6 March 2023; received in revised form 23 April 2023; accepted 12 May 2023

Available online 2 June 2023

Abstract

We introduce a denoising diffusion algorithm to discover microstructures with nonlinear fine-tuned properties. Denoising diffusion probabilistic models are generative models that use diffusion-based dynamics to gradually denoise images and generate realistic synthetic samples. By learning the reverse of a Markov diffusion process, we design an artificial intelligence to efficiently manipulate the topology of microstructures to generate a massive number of prototypes that exhibit constitutive responses sufficiently close to designated nonlinear constitutive behaviors. To identify the subset of microstructures with sufficiently precise fine-tuned properties, a convolutional neural network surrogate is trained to replace high-fidelity finite element simulations to filter out prototypes outside the admissible range. Results of this study indicate that the denoising diffusion process is capable of creating microstructures of fine-tuned nonlinear material properties within the latent space of the training data. More importantly, this denoising diffusion algorithm can be easily extended to incorporate additional topological and geometric modifications by introducing high-dimensional structures embedded in the latent space. Numerical experiments are conducted on the open-source mechanical MNIST data set (Lejeune, 2020). Consequently, this algorithm is not only capable of performing inverse design of nonlinear effective media, but also learns the nonlinear structure–property map to quantitatively understand the multiscale interplay among the geometry, topology, and their effective macroscopic properties. © 2023 Elsevier B.V. All rights reserved.

Keywords: Denoising diffusion; Inverse design; Latent space; Fine-tuning material properties

1. Introduction

Effective constitutive behaviors of manufactured and natural materials are governed by the material properties of the constituents, as well as the topology and geometry of microstructures [1–6]. Understanding this interplay between microstructures and macroscopic constitutive behaviors is crucial for optimizing microstructures to achieve desirable properties for a wide variety of engineering purposes across length scales. Conventionally, designing a microstructure with a designated set of material responses often requires one to solve optimization problems in the spatial domain where a level set or a phase field are used as indicator functions for the different constituents. This setup often leads to a multi-objective non-convex optimization problem which is non-trivial to solve or to

* Corresponding author.

E-mail address: wsun@columbia.edu (W. Sun).

locate the global optimal point. This difficulty motivates the use of machine learning techniques to either embed the design space onto a lower dimensional latent space or reduce the number of costly high-resolution simulations via surrogates [7–9]. On the other hand, Woldseth et al. [10] argue that generative methods such as generative adversarial networks (GAN) and variational autoencoders (VAE) could face challenges for small perturbations of noise, especially when these small changes may lead to profound differences in the optimized structures.

In this work, we introduce a fundamentally different approach in which the design of the microstructures is neither completed by solving a constrained optimization neither in the physical nor the latent space. Instead, the inverse design of a microstructure is completed through a reverse diffusion process learned by a denoising diffusion probabilistic model (DDPM) in which we utilize embedded feature vectors of the target attributes to guide the diffusion generative process. Compared to the generative adversarial networks, the gradual denoising diffusion process makes the training more stable and the performance more scalable with increasing data size [11]. DDPMs also do not suffer from the unstable training commonly found in the adversarial training of GAN. While these salient features could potentially be important for material design, the possibility of using the diffusion process to design microstructures has not been explored, according to the best knowledge of the authors. To fill this knowledge gap, we introduce context feature vectors to control both mechanical behaviors and topological features of the microstructures, allowing for the targeted inverse design of microstructures with desired nonlinear material properties. The relatively stable trajectory of the denoising process enables us to generate a large number of microstructure prototypes with desired macroscopic hyperelastic responses and topology characteristics such that the optimal microstructures can be searched by ranking the performance of the prototypes. To filter out the prototypes outside the desired property range, we develop a convolutional neural network surrogate that predicts the macroscopic hyperelastic energy functional behavior to efficiently rank the generated structures without the need of running full-scale finite element simulations.

To reduce the dimensions of the design space and to enable third-party validation, we use an open-source mechanical MNIST data set provided in Lejeune [12] as our training data set. We then demonstrate the capacity of the proposed approach to accurately generate microstructures with similar energy responses to those present in the dataset. Finally, we explore the algorithm's ability to generate microstructure twins for both the energy response and topology features. Even when given the topology features outside of the training dataset, the DDPM model is capable of designing microstructures that exhibit the target macroscopic elastic responses.

1.1. Past literature on generative methods

In recent years, there have been significant advances in algorithms generating synthetic data samples in a wide range of applications, from generating realistic images and videos to creating synthetic voices and music. The most popular include variational autoencoders [13,14] – learning to represent input data in a lower-dimensional latent space as probabilistic distribution and sampling from it to generate new samples, autoregressive models [15–17] – learning to generate new data samples by sequentially predicting each element of the sample conditioned on the previously generated elements, and generative adversarial networks [18–20] – a generator model learns to generate samples while a discriminator model decides if they are realistic. For instance, Kench and Cooper [21] synthesize high-fidelity 3D datasets using a single representative 2D image, and enable the generated samples to be of arbitrarily large volumes. Chun et al. [22] use GANs to generate ensembles of synthetic microstructures of heterogeneous energetic materials, which can be used to quantify hot spot ignition and growth and to engineer energetic materials for targeted performance. Henkes and Wessels [23] introduce a lightweight GAN formulation for the unconditional generation of 3D microstructures that emulate the macroscale attributes of the original data set distribution. A comprehensive hyperparameter tuning study is conducted to analyze how some of the hyperparameter choices may affect the effective properties of the synthetic microstructures.

In many applications, however, the generative models are often trained to generate new data content that exhibits target properties/attributes associated with the image data. To enforce these constraints, the architecture and the design of these generative models must be amended [24–26]. Several previous works have utilized artificial intelligence methods to guide the microstructure generation, including the use of variational autoencoders and Gaussian process regression, generative adversarial networks, and reinforcement learning. [27] utilize a variational autoencoder and a regressor for property prediction that provide a distance metric to measure shape similarity, enable interpolation between microstructures, and encode patterns of variation in geometries and properties. [28]

design a variational autoencoder to generate a continuous microstructure space. Using Gaussian process regression, they explore the structure–property relationships and predict the mechanical properties of dual-phase steels with a high accuracy.

Many studies have used GANs to generate synthetic images, perhaps due to the great flexibility to incorporate additional conditions in the discriminator component of the GAN such that images may exhibit the targeted properties. Nguyen et al. [29] train unconditional GANs to synthesize realistic three-dimensional microstructures of sandstones while employing actor-critic reinforcement learning to guide the GAN to generate the synthetic images exhibiting effective hydraulic properties consistent with the benchmark specimens. [30] use a style-based generative adversarial network with an adaptive discriminator augmentation mechanism that can successfully generate authentic patterns, which can then be used to augment the training dataset for finite element simulations of soft-tissue materials. Singh et al. [31] utilize a Wasserstein GAN to generate 2D microstructures which follow both explicit physics invariances as well as implicit constraints learned from the image data set. Lee et al. [32] compare the results for deep convolutional GAN (DCGAN), a cycle-consistent GAN (Cycle GAN), and a conditional GAN-based image to image translation (Pix2Pix) for the generation of 2D steel microstructures.

However, GANs can be difficult to train due to issues such as mode collapse, instability, and sensitivity to hyperparameters [33,34]. GANs also require a trade-off that sacrifices diversity for fidelity and hence might not have good coverage of the entire data distribution. These limitations have provided diffusion models the opportunity to surpass GANs as the new state-of-the-art technique for image synthesis on several metrics and data sets [35]. As a result, there has been a great influx of denoising diffusion probabilistic models [11,36–40] – learning the reverse of a diffusion process modeled as a Markov chain to iteratively denoise a data structure and generate new samples, that are replacing many of these state-of-the-art models.

1.2. Organization of this article

The rest of the paper is organized as follows. In Section 2, we discuss the concept of denoising diffusion probabilistic models for image generation, formulate the problem definition of microstructure generation using denoising diffusion algorithms, and describe the neural network setup that performs the microstructure generation task. In Section 3, we describe a CNN architecture that predicts the hyperelastic energy functional behavior under uniaxial extension, using the microstructure’s image as input. In Section 4, we demonstrate how the denoising diffusion algorithm can be trained to generate targeted microstructures with desired constitutive responses by conditioning the microstructure generation process with the hyperelastic energy functional curves. In Section 5, we introduce another context module neural network architecture that provides a feature vector for the targeted topology, which, along with the behavior context module, guides the microstructure generation. We show its capacity to generate microstructure twins with similar energy responses and topology features, as well as test its ability to generate microstructures of topologies outside of the training data set. In Section 6, we provide concluding remarks. For completeness, the open-source database we used to generate the latent design space is described in [Appendix](#).

2. Conditional diffusion for microstructure generation

In this section, we provide a detailed account of how we extend the denoising diffusion probabilistic model, originally proposed for image synthesis, to generate microstructures with fine-tuned nonlinear material properties. We first explain the key ideas and mechanisms of the denoising diffusion probabilistic models and formulate the problem definition for the unconditional microstructure synthesis (Section 2.1). We then formulate a learning problem with the specific task of the conditional generation microstructures with target nonlinear material properties (Section 2.2). Finally, we describe the neural network architecture used to learn the denoising generative process and highlight how we incorporate the embedded feature vectors that control the mechanical responses and the topology (Section 2.3).

2.1. Unconditional generation of microstructures

Diffusion probabilistic models, first introduced by Sohl-Dickstein et al. [36], are a class of generative models that aim to match a given data distribution by learning to reverse a gradual, multi-step noising process. Ho et al. [38] propose a specific parameterization of the generative models that simplifies the training process and establishes

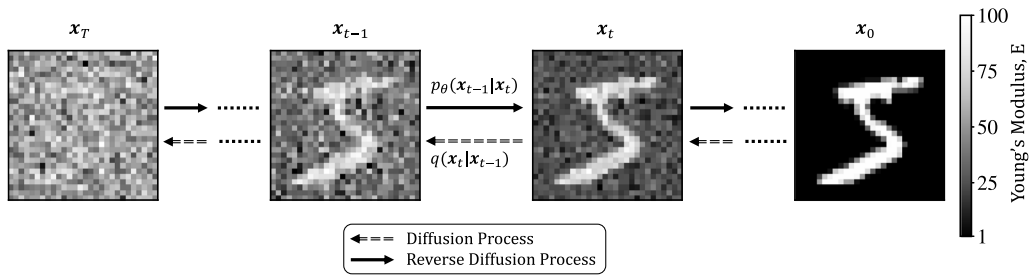


Fig. 1. The diffusion process and its reverse for the elastic property distribution of a microstructure. Noise is added to completely destroy the signal of the original distribution. The forward noising process q is fixed. The denoising process p_θ is learned by a neural network.

the equivalence between denoising diffusion probabilistic models (DDPMs) and score-based generative models, which learn a gradient of the log-density of the data distribution using denoising score matching [41]. Several related research works have shown that DDPMs can generate highly realistic images [39,40,42], audio [43,44], and forecast time series [45]. The outstanding performance could be attributed to the gradual and iterative nature of the diffusion (see Fig. 1).

A denoising diffusion probabilistic model for image generation can be defined as a generative model that combines the properties of denoising autoencoders and diffusion processes to generate new images. These models learn a Markov chain process — where the probability of each transition depends only on the current state and time elapsed. They are trained in a supervised learning setting in which a variational inference is learned to produce samples that match the training data after a finite number of time steps. The transitions of the chain are learned to reverse a diffusion process, which adds noise to the data in the opposite direction of sampling until the signal is destroyed. In each diffusion step, the diffusion probabilistic model adds a small amount of Gaussian noise, and allows a simple neural network parameterization of the new data with the added noise. This diffusion provides a progressive trajectory from a complex data sample to Gaussian noise and thus eases the learning of the reverse denoising process by breaking it down with multiple intermediate steps as shown in Fig. 1. Using a continuous-time approximation of the reverse of the Markov noising process, the model smoothly traverses the latent space and generates new images.

It is highlighted that adding and removing noise in the classical image synthesis context refers to manipulating the color channels of the original image. In the context of microstructure generation for this work, a channel represents the distribution of a material property in a specified material parameter range. For example, for the data set used in this work, the grayscale images have one channel that represents the distribution of Young's modulus (Fig. 1) as described in Appendix. The dimensionality of the problem can be increased by expanding the noise dimensions to represent more material parameters to facilitate the concurrent solution for multiple design targets and will be considered in future work.

In this work, we incorporate the denoising diffusion probabilistic model first introduced by [11] to guide the generation of microstructures. For completeness, we provide an overview of the background for the unconditional diffusion process as described in [11,38]. The unconditional diffusion process can be used to generate realistic microstructures in the general data range. For instance, it can be used to generate new microstructures that are considered consistent with the microstructures provided to train the AI [29,46,47].

Unconditional generation of microstructures alone can be useful in populating the data set with synthetic data for other machine learning tasks, such as establishing a response surface when real data are scarce. Our focus here is nevertheless on the conditional microstructure generation tasks, where the goal is not to create just any microstructures but ones that hold a particular set of topological and constitutive features demanded by the users of the DDPM. In practice, the conditional and unconditional generation strategy is not necessarily employed in a mutually exclusive manner. In Sections 4 & 5, we will formulate a technique in which the guided diffusion algorithm may drop the guidance context vectors (the vectors that provide additional constraints) by a percentage to enrich the generative capacity of our algorithm. For a more complete implementation of the algorithm and the detailed differences between formulations, readers are referred to Nichol and Dhariwal [11], Sohl-Dickstein et al. [36], Ho et al. [38].

2.1.1. Forward process

Consider a data distribution $x_0 \sim q(x_0)$, where x_0 is the original material property distribution sample. In the forward noising process q , we produce noised samples $x_1, \dots, x_T$ for T diffusion time steps by adding Gaussian noise in each time step. The noising is a Markov process where every noised sample x_t is given by:

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t \mathbf{I}), \quad (1)$$

where $\beta_t \in (0, 1)$ is a variance schedule. By setting $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=0}^t \alpha_s$, it is shown that an arbitrary step of the distribution in Eq. (1) can be written as:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)\mathbf{I}), \quad (2)$$

and a sample can be given by:

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \epsilon\sqrt{1 - \bar{\alpha}_t}, \quad (3)$$

where $\epsilon \sim (0, \mathbf{I})$. Using Bayes theorem, [38] find the posterior

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}(x_t, x_0), \tilde{\beta}_t \mathbf{I}), \quad (4)$$

where the mean of the Gaussian is:

$$\tilde{\mu}(x_t, x_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t}x_0 + \frac{\sqrt{\bar{\alpha}_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}x_t, \quad (5)$$

and the variance is:

$$\tilde{\beta}_t := \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t}\beta_t. \quad (6)$$

2.1.2. Reverse diffusion process

If we can learn the reverse distribution $q(x_{t-1}|x_t)$, we can sample backward from step $t = T$ to $t = 0$ to gradually remove the noise and generate a microstructure sample. [36] show that this is feasible since when $T \rightarrow \infty$ and $\beta_t \rightarrow 0$, $q(x_{t-1}|x_t)$ approaches a diagonal Gaussian distribution and a network can learn the mean μ_θ and the diagonal covariance Σ_θ such that:

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \quad (7)$$

While [38] set to only learn μ_θ and select a constant Σ_θ , we follow the improved formulation by [11] to parametrize both as this was observed to achieve better log-likelihoods. Specifically, in this formulation, we learn $\epsilon_\theta(x_t, t)$ such that:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right), \quad (8)$$

and additional to sample x_t the model outputs a vector v to learn $\Sigma_\theta(x_t, t)$ such that:

$$\Sigma_\theta(x_t, t) = \exp(v \log \beta_t + (1 - v) \log \tilde{\beta}_t). \quad (9)$$

Thus, q and p_θ constitute a variational autoencoder similar to [13] to be trained by optimizing a variational lower bound for p_θ . The training of the model optimizes two objectives. For the first learning objective, we randomly sample t and optimize μ_θ using the expectation:

$$L_\mu = E_{t, x_0, \epsilon} [\|\epsilon - \epsilon_\theta(x_t, t)\|^2], \quad (10)$$

to estimate a variational lower bound. The second learning objective to only optimize the learned Σ_θ is:

$$L_{\text{vib}} := L_0 + L_1 + \dots + L_{T-1} + L_T, \quad (11)$$

where:

$$L_t := \begin{cases} -\log p_\theta(x_0|x_1) & \text{if } t = 0 \\ D_{KL}(q(x_{t-1}|x_t, x_0) \| p_\theta(x_{t-1}|x_t)) & \text{if } 0 < t < T \\ D_{KL}(q(x_T|x_0) \| p(x_T)) & \text{if } t = T. \end{cases} \quad (12)$$

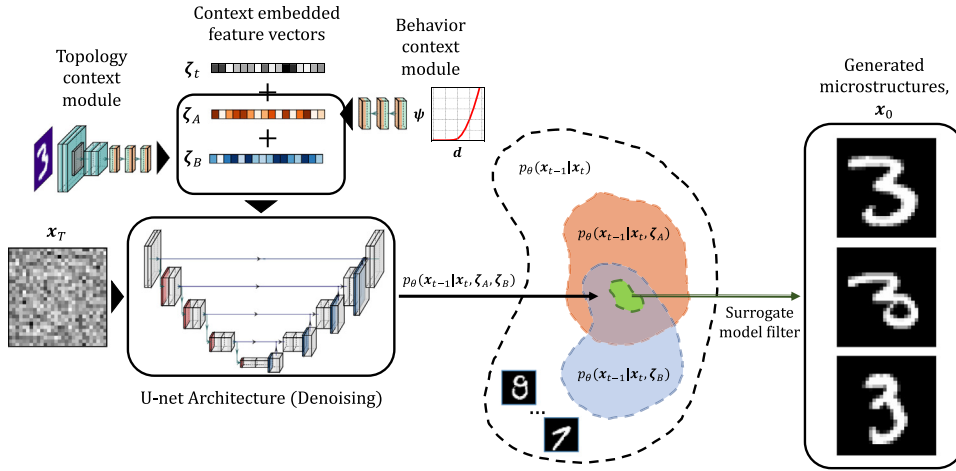


Fig. 2. Schematic of the sampling framework for microstructure generation conditioned by the summation of context feature vectors ζ_t , ζ_A , and ζ_B . Context feature vectors ζ_A and ζ_B are embedded by the behavior and context module neural network architectures respectively. The generated microstructures are further filtered by a surrogate model to demonstrate macroscopic behaviors in a desired admissible range.

In the above, each term is a KL divergence between the Gaussian distributions of two successive denoising steps. The learning objective to learn both μ_θ and Σ_θ is defined as:

$$L_{\text{hybrid}} = L_\mu + \lambda L_{\text{vib}}, \quad (13)$$

where $\lambda = 0.001$ is a scaling factor selected to weigh the two objectives similar to [11].

2.2. Conditional diffusion with embedded feature vectors

While the unconditional diffusion described in the previous section can readily generate microstructures consistent with the training data set, our goal is to design microstructures that exhibit prescribed mechanical behaviors. To achieve this goal, we use a conditional diffusion process which fine-tunes the resultant microstructures via feature vectors.

We enforce the designated material properties by introducing context into the synthesis problem, following a relevant approach that utilizes image classes or textual descriptions to provide context to target specific images [39]. Our approach, however, differs in that we aim to introduce context to control both targeted mechanical behaviors and targeted topologies, and we do so by incorporating context in the form of feature vectors into the denoising diffusion algorithm. Along the time embedding feature vector that is common in diffusion algorithms, our algorithm additionally employs a mechanical feature vector for every desired target property. This treatment allows us to modularize the problem and control different properties of the microstructures in a separate and distinct manner. For every desired target property, we introduce a feature vector and a corresponding neural network context module that will be trained in tandem with the diffusion algorithm.

To incorporate the feature vector conditioning, we follow [35] by introducing a conditional Markov process $\hat{q}$ that adds noise similar to q described in the previous section. We define the conditional noising process to:

$$\begin{aligned} \hat{q}(x_0) &:= q(x_0), \quad \hat{q}(x_{t+1}|x_t, \zeta_1, \dots, \zeta_N) := q(x_{t+1}|x_t), \\ \text{and } \hat{q}(x_1, \dots, x_T|x_0, \zeta_1, \dots, \zeta_N) &:= \prod_{t=1}^T \hat{q}(x_t|x_{t-1}, \zeta_1, \dots, \zeta_N), \end{aligned} \quad (14)$$

where $\{\zeta_1, \dots, \zeta_N\}$ are the corresponding context feature vector embeddings for $i = 1, \dots, N$ target microstructure properties — be it behavior or topology. [35] refer to a class conditioning given supervised labels (image classes) but we perform the conditioning with unsupervised labels that are embedded during training, more similar to the [39] approach. Thus, we are also given $\hat{q}(\zeta_1, \dots, \zeta_N|x_0)$ to be the embedding generated during the training process. It is noted that these context vectors are constant and independent of the diffusion process time. [35] show

that the conditional noising process with a diffusion time-independent class behaves similarly to the unconditional counterpart described in the previous section and can be used to guide image synthesis.

Thus, in order to perform the conditional sampling, we will need to train two neural network approximations. The first is the approximation $p_\theta(x_{t-1}|x_t)$ of $q(x_{t-1}|x_t)$ to perform the denoising as described in the previous section. The second is $p_\phi(\xi_1, \dots, \xi_N|x_t)$ of $\hat{q}(\xi_1, \dots, \xi_N|x_t)$ for the generation of the unsupervised labels. The two will be optimized simultaneously using the training objectives in Eqs. (10) and (11).

In Fig. 2, we demonstrate a schematic of the sampling process for the conditional microstructure generation for two sample context feature vectors ξ_A and ξ_B . Each context vector corresponds to a different type of conditioning (e.g., macroscopic response and topology respectively) and comes from the corresponding context module neural network that was trained along the diffusion algorithm. At the core of the algorithm, there is a neural network architecture that learns the reverse of the diffusion process. Starting with a random Gaussian noise input, it can iteratively denoise the sample from time $t = T$ to $t = 0$ to generate a synthetic microstructure. Without the conditioning by a feature vector, the neural network will be able to sample microstructures from the entire data set latent space representation $p_\theta(x_{t-1}|x_t)$. By introducing the context feature vectors, we can fine-tune the latent space the model samples from to $p_\theta(x_{t-1}|x_t, \xi_A, \xi_B)$. To further fine-tune the microstructure generation to the desired material nonlinear properties, we introduce an additional step to filter the mass generated microstructures. We train a surrogate model that predicts the macroscopic response of the generated microstructures, ranks the samples, and filters out the ones that do not satisfy a desired admissible range. More details on this process are discussed in Section 3.

2.3. Denoising diffusion neural network model architecture

In this section, we explain (1) the neural network architecture used to perform the denoising diffusion and (2) how the context feature vectors to perform the conditioning are introduced into the diffusion process. The architecture we used is modified from an implementation of Nichol and Dhariwal [11]. They have adopted and improved on the U-net architecture used by [38], which have adapted the PixelCNN++ [48] and Wide ResNet [49] for their image synthesis tasks. This architecture is used to approximate the denoising process as described in the previous section.

We acknowledge that the dimensionality and resolution of the images that represent the microstructure in this work — the MNIST data set [50], are smaller than those studied in [11], such as the CIFAR-10 [51] and ImageNet 64×64 [52]. Thus, we adapt a smaller architecture version of the original U-net that was deemed adequate to perform our synthesis tasks. Specifically, in our U-net implementation, there are two stacks that perform the downsampling and upsampling in four steps respectively. Each step has one residual block. We use one attention head with 32×32 , 16×16 , and 8×8 attention resolutions. We modify the output of the architecture to have one output image channel to produce the microstructure grayscale material property maps. The U-net utilizes $[C, 2C, 4C, 8C]$ channel widths from higher to lower resolutions with the base model channel size selected as $C = 32$. The architecture is conditioned by the diffusion time step t through an embedded feature vector ξ_t . This vector is produced by a sequential two-layer network — it has two-dense layers of 128 neurons, each with a Sigmoid Linear Unit (SiLU) and a Linear activation function respectively.

At this point of the architecture, we inject our context feature vectors to perform the microstructure conditioning. Specifically, given context data structures Z_i for $i = 1, \dots, N$ target properties to condition the microstructure generation with, we define neural network architectures c_i that produce context feature vector embeddings $\xi_i = c_i(Z_i)$. In this framework, the context data structures are curves (hyperelastic energy functional responses) and 2D images (topologies). However, any other type of data structure can be embedded to control the generation process. The context feature vector embeddings are introduced in the architecture by defining an embedding vector ξ_{emb} for both the diffusion time and the microstructure properties as:

$$\xi_{\text{emb}} = \xi_t + \xi_i + \dots + \xi_N \text{ for } i = 1, \dots, N. \quad (15)$$

In this work, we will introduce two types of context embeddings, by (1) material behavior and (2) topology, in Section 4 and Section 5, respectively. The corresponding neural network architectures c_i are defined in these sections. We highlight that we introduce the conditioning as the summation of embedded feature vectors to allow a modular structure of the framework independent of the diffusion algorithm. A summary of all the neural network models used in this study, as well as the corresponding inputs, outputs, and functions is presented in Table 1. More types of conditioning will be considered in future work.

Table 1

Summary of neural network model architectures used in the study.

Model type	Model inputs	Model outputs	Model function
U-net	Noisy signal and context feature vector ζ_{emb}	Generated microstructure	Denoising diffusion
Feed-forward behavior context model	Hyperelastic energy functional ψ context curve	Behavior context feature vector ζ_A	Behavior conditioning
Convolutional topology context model	Topology context image	Topology context feature vector ζ_B	Topology conditioning
Convolutional surrogate model	Generated microstructure	Hyperelastic energy functional curve ψ	Filtering and ranking of generated microstructures

3. Neural network for material behavior prediction

In order to further optimize the microstructure generation for desired material nonlinear properties, we train a surrogate model which predicts the macroscopic response of the generated microstructures and, subsequently, ranks and filters out samples which do not fall within the desired admissible hyperelastic energy functional range. In this section, we describe the Convolutional Neural Network (CNN) architecture to perform this task. Given the distribution of a microstructure's Young's modulus, the model can predict the hyperelastic energy functional behavior under uniaxial extension. This CNN model can be used to predict the behavior of a new microstructure during a uniaxial extension simulation, eliminating the need to perform a complete finite element simulation. These predictions can then be used in place of the output of a FEM simulation, allowing for faster prediction of the microstructure's behavior with a reasonable trade-off in accuracy. In [12], along with introducing the Mechanical MNIST data set that will be used in this work ([Appendix](#)), it was also demonstrated that a surrogate CNN model can be trained with a reasonable extrapolation capacity for samples unseen during training. We are training a CNN surrogate model for a similar purpose in a similar fashion, except the following difference. Instead of predicting the energy increment $\Delta\psi$ given a displacement, we are predicting the entire constitutive response of the material under uniaxial extension. It is noted that full-scale finite element simulations will eventually be used to validate the generated microstructures in the following sections. Here the CNN counterpart will be used to quickly assess the estimated behavior of a generated microstructure before it is confirmed by the FEM results.

Remark 1. While only the conventional CNN is used to generate surrogate models for performance predictions (see [Table 1](#)), a more sophisticated surrogate model that is generative, such as GANs [53], diffusion [36] or flow models [54] as well as variational autoencoders [55], can also be trained and deployed. At the expense of a potentially more difficult and costly training, these generative adversarial networks may help disentangle factors of variations [56] and hence further improve the robustness of the out-of-distribution performance predictions [53]. Incorporating these generative approaches for performance prediction will be considered in future studies, but is out of the scope of this paper.

The neural network implemented follows a standard CNN architecture and consists of two Convolutional layers with a kernel size of 3, stride of 1, and padding of 1. The first CNN layer has 1 input channel and 16 output channels, and the second Convolutional layer has 16 input channels and 16 output channels. The output of the first Convolutional layer is passed through a Max Pooling layer with a kernel size of 2 and stride of 2, which down-samples the input by taking the maximum value over a 2×2 window. The output of the second Convolutional layer is also passed through a Max Pooling layer with the same configuration. The output of the second Max Pooling layer is then flattened to a single dimension, representing the features extracted by the CNN. This flattened output is then passed through three fully connected (Dense) layers with 120, 84, and 13 neurons, respectively. The output size of 13 corresponds to the number of time steps in the Mechanical MNIST uniaxial extension curves. The ReLU activation function is applied to the output of the CNN and Dense hidden layers, while the output uses a Linear activation function. The kernel weight matrices of the layers are initialized with the default Glorot uniform distribution, and the bias vectors are initialized with a zero distribution. The model was trained for 200 epochs with

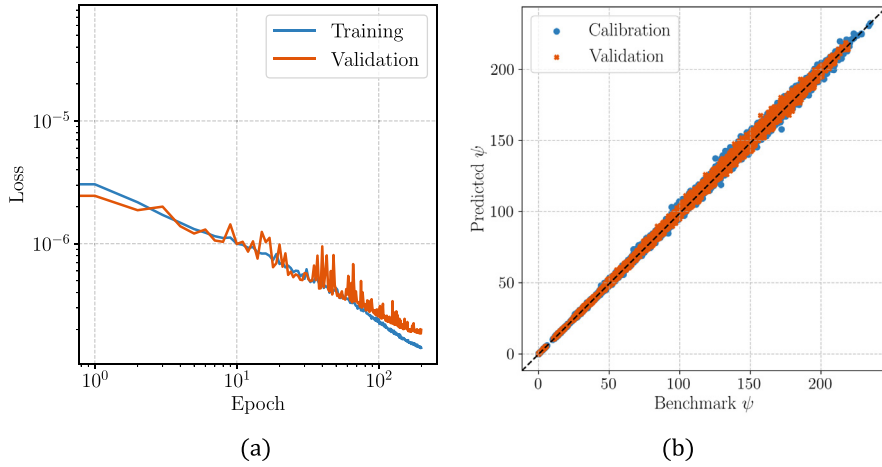


Fig. 3. (a) Training and validation loss function values and (b) predictions of the energy functional response of all the training and testing samples of the MNIST dataset for the CNN architecture.

a batch size of 256 using the Adam optimization algorithm with a learning rate of 10^{-3} and a mean squared error (MSE) loss function. The learning rate was reduced by a factor of 0.9 when the validation loss plateaued for 5 epochs. The network was trained on the 60,000 and validated on the 10,000 MNIST training and test samples and their corresponding energy functional ψ curves of the database described in [Appendix](#). The results of the training are demonstrated in [Fig. 3](#), where the network is shown to perform well within the training set and demonstrate a good capacity to generalize outside the samples used for the calibration.

4. Generation of hyperelastic microstructures conditioned by material behaviors

In this section, we will demonstrate that the denoising diffusion algorithm described in [Section 2](#) can be trained on the Mechanical MNIST data set to generate targeted microstructures with desired constitutive responses by conditioning the microstructure generation process with the hyperelastic energy functional curves. In [Section 4.1](#), we will first describe the hyperelastic curve context module of the diffusion algorithm, the training of the architecture, and the capacity of the algorithm to generate synthetic microstructures with similar statistical properties as the given data set. In [Section 4.2](#), we will input random polynomials to design targeted microstructures outside of the training range and explore the capacity of the algorithm to generate microstructures for unseen behaviors.

4.1. Training of behavior-conditioned architecture

In this section, we introduce a context module neural network architecture that extracts the behavior feature vector of the hyperelastic energy functional curves from the Mechanical MNIST dataset to guide the microstructure generation process. We detail the training procedure and demonstrate the capacity of the architecture to accurately generate microstructures with similar energy responses to these present in the Mechanical MNIST dataset.

The context module in this section embeds the hyperelastic energy functional curves described in [Appendix](#) into feature vectors. The module follows a simple feed-forward neural network architecture. It inputs the energy functional values ψ of the 13 time steps of the uniaxial extension experiments of the Mechanical MNIST data set. It is noted that here we are only using the energy functional values since the displacement inputs for all the FEM simulations are the same and this input information would be redundant as a context for this case of microstructure generation. In the case the input strain was variable or there existed different types of tests in the database, we would have to input additional identifiers to properly control the microstructure generation. The sequential module has two Dense layers of 128 neurons each. The hidden layer has a SiLU activation function and the output feature vector layer has a Linear activation function. We opt for an embedded feature vector of 128 neurons that matches the dimensions of the time embedding vector. The predicted behavior feature vector is added to the time embedding

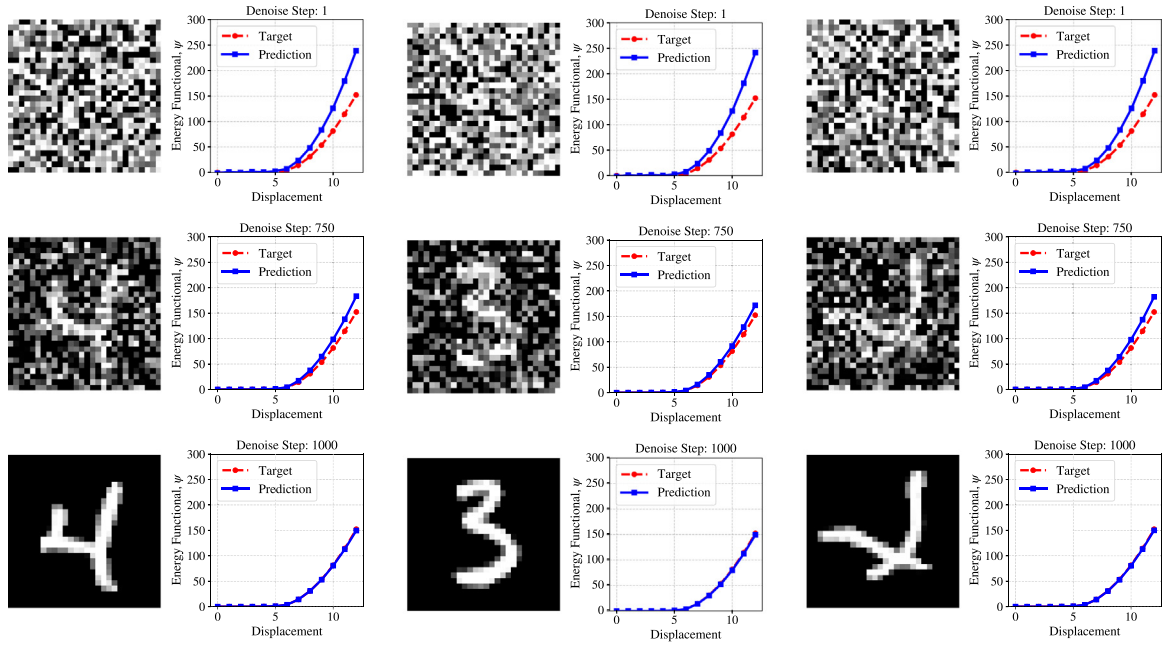


Fig. 4. Microstructure generation and neural network prediction results for the same hyperelastic energy functional response from the Mechanical MNIST testing dataset. The results are shown for denoising steps 1, 750, and 1000.

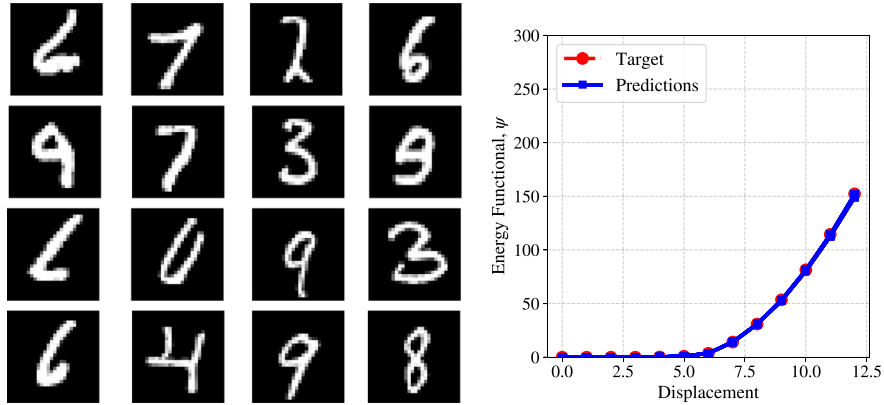


Fig. 5. Microstructure generation and neural network prediction results of 16 samples for the same hyperelastic energy functional response from the mechanical MNIST testing data set. The samples were batch generated and filtered to have an energy functional prediction with $MSE < 10^{-4}$.

feature vector as described in Section 2.3. The kernel weight matrices of the layers are initialized with the default Glorot uniform distribution, and the bias vectors are initialized with a zero distribution.

After assembling the diffusion architecture with the behavior context module, we train the diffusion model for 50,000 steps on the 60,000 sample pairs of MNIST handwritten digits and energy functional curves. We also set the context to be dropped with a 10% chance – the feature vector used is $\zeta_{\text{emb}} = \zeta_t$ – to improve the algorithm’s capacity to perform unconditional synthesis. We opt for 1000 diffusion time steps with a linear noise schedule as used in [38]. We use an Adam optimizer [57] with a learning rate of 10^{-4} with a batch size of 128.

In Fig. 4, we observe that the algorithm can create twins for a specific behavior that do not look alike or do not resemble the topologies in the training data set. All three microstructures are generated for the same mechanical

response from the mechanical MNIST testing data set. The algorithm is trained on images of handwritten digits and does not have any limitations or context guidance on the generated images to fall within a specific image class. This is an important aspect of the behavior guided diffusion model. It means that the algorithm is not constrained to generate microstructures that resemble the images it was trained on but rather has the ability to generate a wide variety of microstructures that may have different shapes, forms, and other properties. This feature allows for a greater degree of flexibility in the design of microstructures and has the potential to enable the generation of microstructures with specific properties that may not be found in the training dataset. Additionally, it is capable of generating microstructures with shapes or forms that are not seen during the training phase. This ability to generalize and adapt to new behavior context inputs is a key advantage of this approach to microstructure generation, as it enables the discovery of a diverse range that can meet specific design criteria.

It should be noticed that the generated microstructures may not always exhibit the target energy response with sufficient precision. This could be attributed to the fact that the generation process is probabilistic and performed using the denoising diffusion algorithm, which inherently introduces a degree of uncertainty in the generative sampling path that causes departure from the data manifold (cf. Chung et al. [58]). To ensure robustness and improve the precision of the constitutive responses of the generated microstructures, leverage the efficiency of the DDPM to generate a sufficient number of prototypes and simply filter out the ones that do not satisfy a target MSE error limit of $\text{MSE} < 10^{-4}$ using the CNN architecture previously described in Section 3. Thus, we can batch generate multiple microstructures and quickly select the one with the desired attributes without spending a significant amount of time to run the slower finite element simulations, just to filter out the unneeded ones.

In Fig. 5, we demonstrate 16 microstructure twins that were batch generated in this way for the same mechanical MNIST testing curve. The error range of the generation process is further explored in the following section.

4.2. Generate hyperelastic microstructures with polynomial energy functional

In this section, we will be further testing the generation for energy functional forms not included in the data set. Instead of using the hyperelastic energy functionals of the Mechanical MNIST data set as a guide, we will be using a set of random energy functional polynomials as the target behaviors to generate microstructures. These random polynomials will be selected from the coefficient ranges present in the Mechanical MNIST data set and validate that the generated microstructures are physically realistic and representative of the range of behaviors present in the data set. By using these random polynomials as the target behaviors, we can generate a diverse set of microstructures with a wide range of hyperelastic properties, rather than being limited to the specific behaviors present in the original dataset. This allows for the exploration of the behavior of microstructures with a wider range of properties and to more fully understand the relationship between microstructure and behavior.

To generate the target polynomials for the microstructure generation process, we first fit all the energy functional curves in both the training and testing data sets with third-order polynomials. The general form of the polynomials is $ax^3 + bx^2 + cx$. After fitting the curves, we find the ranges of the coefficients $a = [-0.0162, -0.0047]$, $b = [0.6346, 1.3968]$, and $c = [0.00532, 0.4114]$. We sample three random polynomials from these coefficient ranges to explore the entire energy data range of the data set. From a higher to a lower energy functional range, we sample $-0.014x^3 + 1.368x^2 + 0.057x$, $-0.011x^3 + 0.919x^2 + 0.062x$, and $-0.013x^3 + 0.709x^2 + 0.276x$. The data range, the input range of all the random polynomials, and the sampled energy functionals are shown in Fig. 7(a).

In order to generate microstructures with sufficiently accurate targeted behaviors, we filter the generated microstructures using the CNN architecture previously described in Section 3. We continuously generate microstructures until 512 of them are predicted to have an $\text{MSE} < 10^{-4}$. The error range of the predictions can be quantified and is shown in Fig. 6(a). In this figure, we end up generating 2112 microstructures for the second target polynomial $-0.011x^3 + 0.919x^2 + 0.062x$ as a context, and plot the error range of every predicted behavior using the CNN. We also plot the MSE distribution curve for these predictions, shown in Fig. 6(b). This allows us to visualize the overall accuracy of the CNN in predicting the energy response of the generated microstructures, as well as the range of errors that we can expect in these predictions. In addition to visualizing the overall error distribution, we can also use the CNN to filter out and select only the microstructures that have a predicted MSE within a certain range. In Fig. 6(c), we plot the 2112 generated microstructures, and highlight the 512 that have a predicted MSE lower than a selected 10^{-4} admissible limit. We also replot the error range for this MSE limit in Fig. 6(d), to show how the errors are distributed among the selected microstructures. By selecting only the microstructures with a predicted

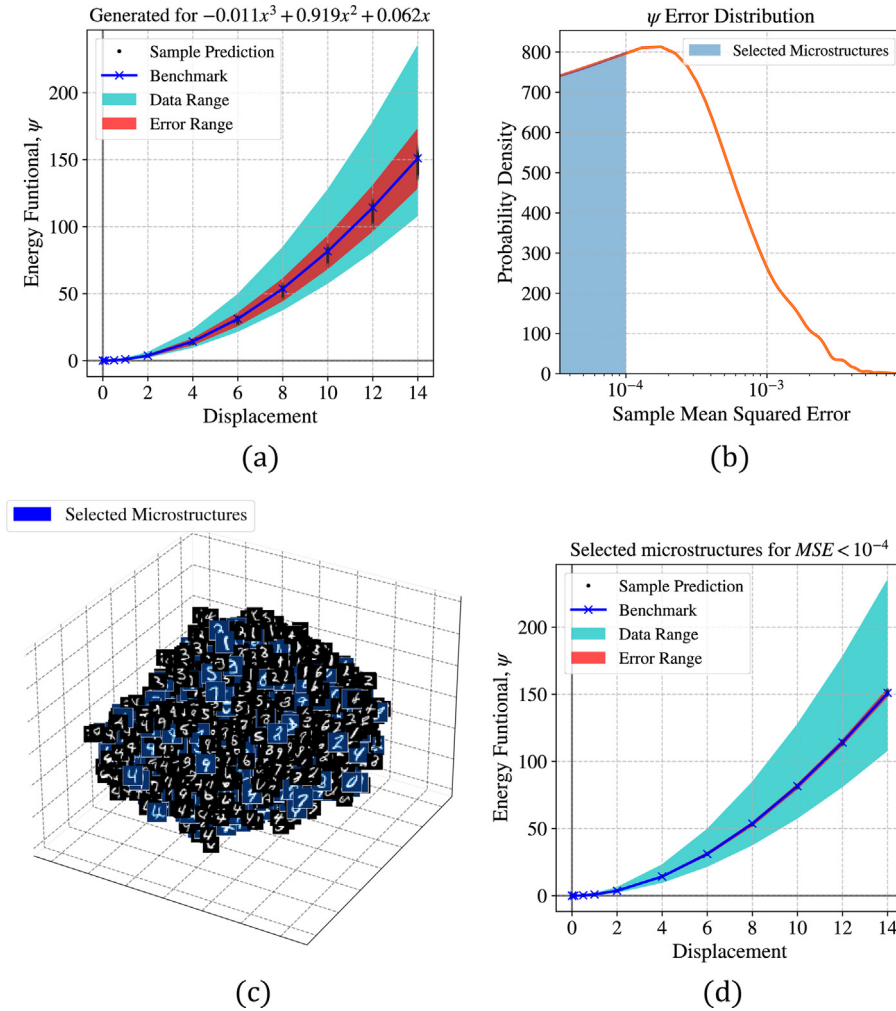


Fig. 6. (a) Energy functional neural network predictions for 2112 generated microstructures for the random target polynomial $-0.011x^3 + 0.919x^2 + 0.062x$. The data range and error range of the predictions are also demonstrated. (b) MSE distribution for the predicted energy functional of the 2112 microstructures. Microstructures with an MSE less than 10^{-4} are selected. (c) The 2112 generated microstructures and the selected microstructures that have an $MSE < 10^{-4}$. The microstructures are embedded using the T-distributed Stochastic Neighbor Embedding algorithm for easier visualization in three axes. (d) Energy functional neural network predictions for the selected generated microstructures that have an $MSE < 10^{-4}$.

MSE below a certain limit, we can effectively filter out the microstructures that have a higher degree of uncertainty, and focus on those with a higher degree of accuracy without having to run a full FEM simulation.

The filtering procedure is performed during the microstructure generation for all the target polynomials. The results for the generated microstructures are shown in Fig. 7(c), (d), and (e). The microstructures shown were first selected to have a MSE less than 10^{-4} . To validate the microstructure design, we also test the microstructure by performing a forward FEM simulation and validate that the generated microstructures indeed have the desired behaviors. We can also observe that the shapes are progressively thicker from lower to higher polynomial value ranges. This is expected as the higher final energy functional values represent increasingly stiffer materials — it is expected that more stiff inclusion material is predicted for the microstructures. By carefully selecting the target behavior, we can achieve a high-level control of the overall shape and topology of the generated microstructures with an accurate control of the target behavior. In the next section, along with the target behavior we will control the topology of the generated microstructures as well.

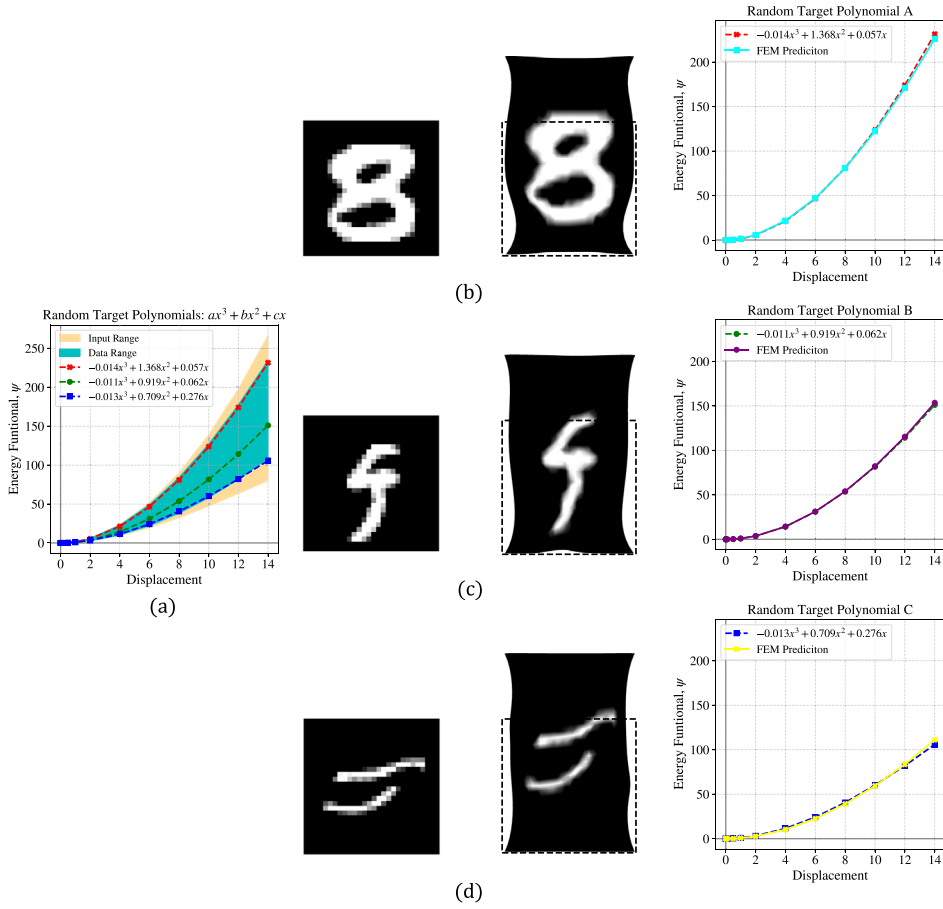


Fig. 7. Microstructure generation and FEM simulation results for three random target energy functional polynomials.

5. Generation of hyperelastic microstructures conditioned by both material behavior and topology

In this section, we introduce another context module neural network architecture that will provide another feature vector to control topology. Along with the behavior context module, it will be used to guide the microstructure generation for two concurrent tasks. We outline the training procedure and demonstrate the capacity of the architecture to generate microstructure twins now for both the energy response and topology features for the behaviors and microstructure present in the Mechanical MNIST data set. Additionally, we will test the capacity of the diffusion algorithm to generate microstructures of topologies outside of the training data set.

5.1. Training of behavior and topology conditioned architecture

In this section, we introduce an enhanced context module neural network architecture that extracts both behavior and topology feature vectors from the Mechanical MNIST data set to guide the microstructure generation process. In addition to the material behavior module introduced in the previous section, we now also include a topology module. We detail the training procedure and demonstrate the capacity of the enhanced architecture to accurately generate microstructures with similar energy responses and topologies to those present in the Mechanical MNIST data set.

The additional topology context module follows a convolutional neural network architecture. The architecture has two convolutional layers of kernel size 3, stride 1 and padding 1 with ReLU activations. The first layer inputs the grayscale map that describes the desired target material property distribution context. Both layers have 16 output channels. Both are followed by a max pooling layer of kernel size 2 and stride 2 to down-sample their output.

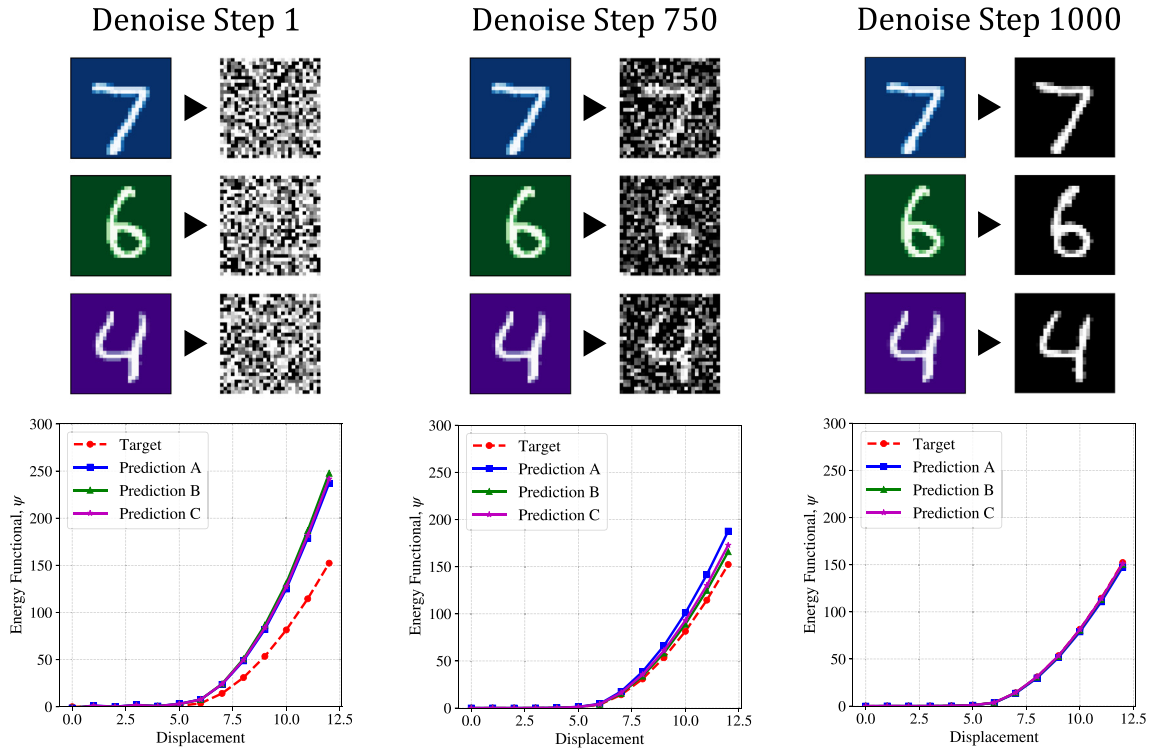


Fig. 8. Generation of topology and behavior twins from a microstructure from the MNIST test set. The results are shown for steps 0, 750, and 1000 of the denoising process. The color and grayscale images show the topology context input and the diffusion output respectively.

The output of the second pooling layer is then flattened and passed through three fully-connected layers with 128 neurons each with two ReLU activations and a Linear one respectively. This choice of 128 neurons is made as it aligns with the size of time and behavior context embedding vectors from the previous sections. Similarly to the previous section, the context vector containing the time, behavior, and topology contexts comes from the summation of the three embedded vector counterparts. The training procedure is identical to the one described in Section 4.1 with the difference that along every sample we also input a topology context map.

With the addition of a topology module, along with the material behavior module, we are now able to perform more controlled microstructure generation tasks. In Fig. 8, we demonstrate the generation of microstructures from image and curve pairs from the Mechanical MNIST testing set. In this example, we attempt to create twins for a randomly sampled microstructure and its corresponding hyperelastic behavior. We first check if the algorithm can reproduce a microstructure with the same topology context (the digit 7) and behavior. We then test the capacity to generate microstructures that have the same behavior context but different topology context inputs randomly sampled from the MNIST test set (the digits 6 and 4). We can see the algorithm modifies the material property map inputs to synthesize structures to adjust to the specific target behavior. These microstructures not only look similar to the images they were generated from but also have the same hyperelastic behavior as the target energy functional curve. Thus, we show that we can perform the creation of both behavior and topology twins. While in this example the alterations from the input context images may seem slight, we test some more apparent topology changes in the following section.

5.2. Generation of topologies outside the training dataset

In this section, we will investigate the capability of the behavior and topology-guided diffusion model to generate microstructures for topologies that are not represented in the Mechanical MNIST data set. Despite the algorithm being trained on images of handwritten digits, there are no limitations or context guidance imposed on the generated

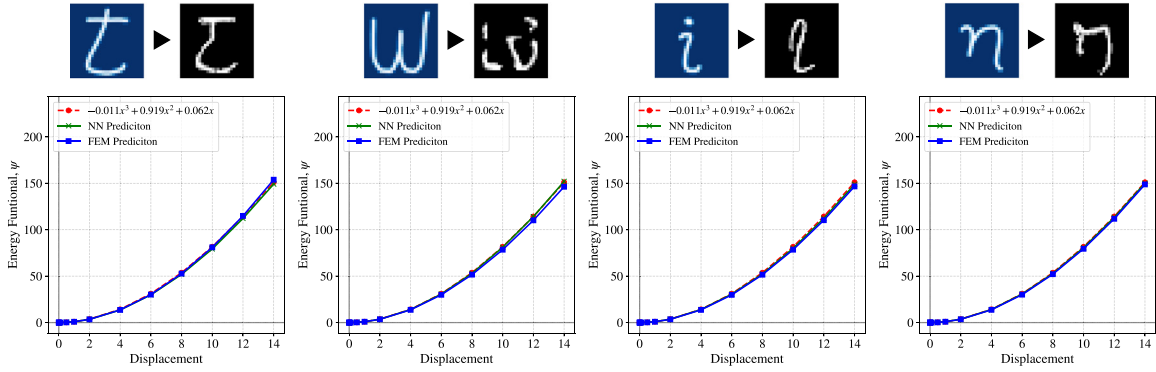


Fig. 9. Microstructure generation and FEM simulation results for the target energy functional polynomial $-0.011x^3 + 0.919x^2 + 0.062x$ and target images that spell the word “twin”. The color and grayscale images show the topology context input and the diffusion output respectively.

images to fall within a specific image class, as previously demonstrated in Section 4.1. This suggests that the algorithm has the potential to generate microstructures of other shapes and geometries as well. However, it is observed that while the topology context input is dissimilar to the MNIST data set, the images generated in this section tend to resemble the MNIST style.

By focusing on the topology context aspect of image generation, we will create microstructures that share the same energy response, represented by the energy functional $-0.011x^3 + 0.919x^2 + 0.062x$ — all the microstructures in this section have the same energy response. Additionally, to improve the accuracy of the generated microstructures, we utilize a mass generation method of prototypes and employ a CNN model from Section 3 to filter out those microstructures that have a predicted $MSE > 10^{-4}$. We were able to utilize the proposed method to generate microstructures that spell the word “twin”, as illustrated in Fig. 9. Furthermore, we demonstrate the denoising process, in which the letters α , β , and γ were generated, as depicted in Fig. 10. Despite of the fact that these new characters had never been included in the training data set, the proposed algorithm is capable of generating microstructures conforming to the target energy functional behavior while preserving the key features of the characters.

The results generated by the behavior and topology-guided diffusion model indicates that there are several potential benefits of this guided approach for the synthesis of microstructures. These advantages include the ability to create microstructures of specific shapes and forms, as well as the potential to design microstructures that exhibit a specific set of properties, such as material symmetry, volumes of materials, and ratios between constituents. Nevertheless, a current challenge in this application is the limited similarity between the topology context inputs used and the Mechanical MNIST data set. This has resulted in difficulty in efficiently identifying suitable microstructures, as well as instances in which it is not possible to discover microstructures with a predicted energy response with $MSE < 10^{-4}$ at all. Specifically, when topology context inputs are dissimilar to the MNIST data set, the error range of the generated microstructures may be larger, deeming it harder to find a suitable microstructure. For example, inputting a very stiff topology context image and an energy curve in the lower range of the material behaviors, it is expected that the algorithm would have difficulty in designing a microstructure that satisfies both criteria simultaneously.

One potential solution to this obstacle in discovering microstructures with dissimilar topology context inputs would be to augment the training data set. Specifically, by selecting a more heterogeneous data set that includes a diverse range of microstructure shapes and forms, the algorithm would have access to a wider range of reference examples upon which to base its generative process. Furthermore, in cases where a specific inclusion shape is desired, incorporating additional examples of that shape within the training data set could also prove beneficial in terms of improving the algorithm’s ability to generate microstructures featuring that specific shape. Additionally, employing other forms of guidance within the microstructure generation process, such as through the use of metrics (e.g., a symmetry loss metric, a target ratio of soft to stiff material) or neural networks, could also help to improve the quality of the generated microstructures. Such additional guidance mechanisms can provide a filter for the

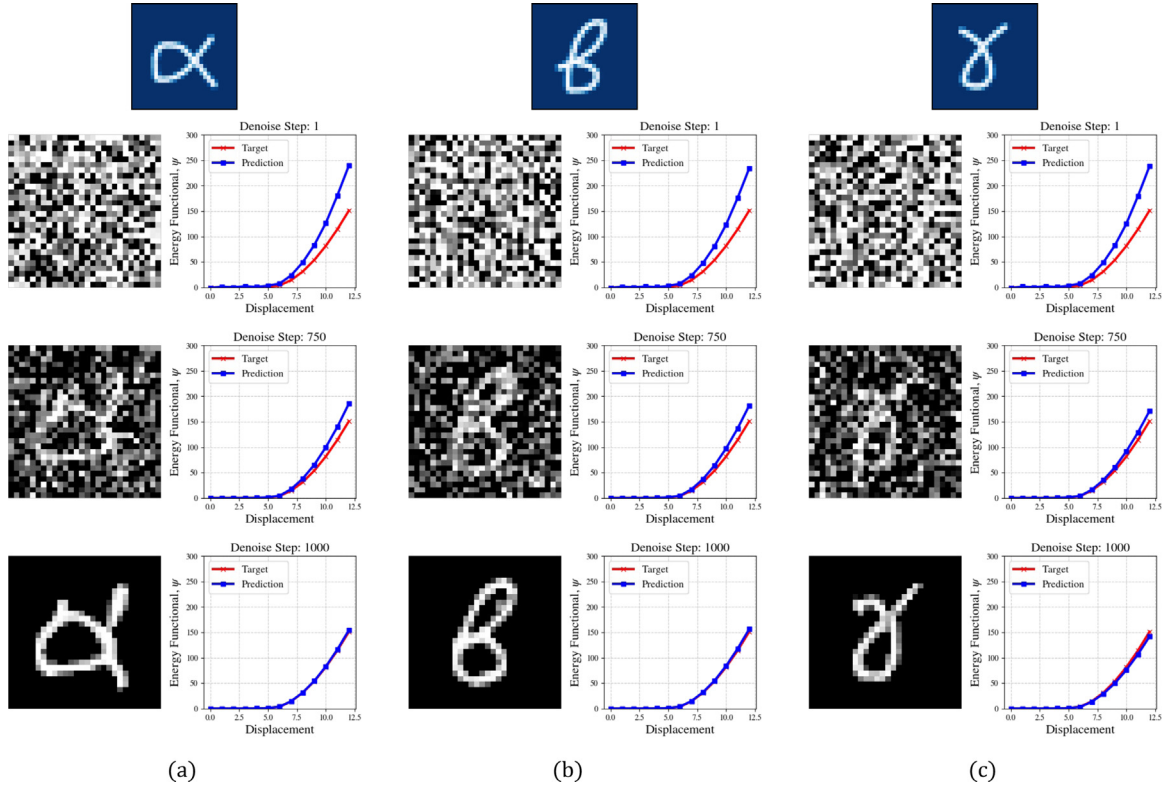


Fig. 10. Microstructure generation and neural network prediction results for the target energy functional polynomial $-0.011x^3 + 0.919x^2 + 0.062x$ and targeted images for the letters α , β , and γ . The results are shown for denoising steps 1, 750, and 1000. The color and grayscale images show the topology context input and the diffusion output respectively.

generated microstructures, allowing the algorithm to focus on those with the desired characteristics and eliminate those that are not similar to the target topology.

Remark 2. Note that, while generative models, such as DDPMs and GANs, may generate new content from the training data, the training data may still implicitly restrict the possible mechanical behaviors of the generated microstructures (compared to the alternative approach where an inverse problem is conducted in a high-dimensional design space Woldseth et al. [10]). Quantitatively estimating the distribution of the performance for a given training data set is not trivial but possible. In particular, we may estimate the boundary of the data manifold through a guided diffusion in the reverse diffusion process where the focus shifted from enforcing a particular set of mechanical behaviors to maximizing/minimizing a given performance metric and therefore establishing the performance bound. For instance, instead of guiding the microstructures to match a prescribed energy function, one may modify the guidance to maximize/minimize the energy it stored for a given loading. Other feasible approaches to improve the quantitative interpretability of the generative model may include post-hoc methods, such as model distillation (e.g., Tan et al. [59]), in which an interpretable model is built based on the black-box counterpart, as well as the use of feature attribution, and counterfactual explanation (e.g., Kommiya Mothilal et al. [60]) which may convey the important features of microstructures that change the performance outcomes. The investigations of these research directions are out of the scope of this study but will be considered in the future.

6. Conclusion

We present a novel approach to design microstructures with desired mechanical behaviors and topologies by introducing context feature vectors that control targeted properties in a DDPM model. We demonstrate the capacity of the architecture to accurately generate microstructures with similar energy responses and topologies to those

present in the Mechanical MNIST data set, while demonstrating a reasonable extrapolation capacity. Our method is based on a modular framework where a feature vector is assigned to every desired target property. This allows for the incorporation of different types of target properties and their corresponding embeddings, rendering the problem more scalable and easier to interpret. Future work will explore the design of microstructures with additional behavior controls such as behaviors in the microscale along the macroscopic responses and more precise topology constraints, such as symmetry. This could be achieved by introducing more complex microstructure databases as well as additional constraints in the diffusion algorithm. The flexibility and speed of the design of microstructures using diffusion algorithms will not only facilitate the generation of digital twins but could also be extended to generate synthetic databases in usually data-scarce engineering applications.

CRedit authorship contribution statement

Nikolaos N. Vlassis: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft. **WaiChing Sun:** Methodology, Validation, Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors thank the two anonymous reviewers for the helpful suggestions. The authors are supported by the National Science Foundation, United States under grant contracts CMMI-1846875 and OAC-1940203, and the Dynamic Materials and Interactions Program from the Air Force Office of Scientific Research, United States under grant contracts FA9550-19-1-0318, FA9550-21-1-0391 and FA9550-21-1-0027, with additional support provided to WCS by the Department of Energy DE-NA0003962, United States. These supports are gratefully acknowledged. The views and conclusions contained in this document are those of the authors, and should not be interpreted as representing the official policies, either expressed or implied, of the sponsors, including the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

Appendix. Mechanical MNIST database

This work uses the Mechanical MNIST data set [12] that contains pairs of heterogeneous microstructures and their corresponding hyperelastic responses. Mechanical MNIST is a data set of finite element simulation results generated from images in the MNIST database of handwritten digits [50]. The MNIST dataset is a large database of handwritten digits that is commonly used for training, evaluating, and benchmarking image classification algorithms. It consists of 60,000 training images and 10,000 test images, each of which is a 28×28 pixel grayscale image of a handwritten digit from 0 to 9.

The images are transformed into material properties using a compressible Neo-Hookean material model, where the strain energy functional ψ is given by

$$\psi = \frac{1}{2}\mu[\mathbf{F} : \mathbf{F} - 3 - 2\ln(\det \mathbf{F})] + \frac{1}{2}\lambda \left[\frac{1}{2}((\det \mathbf{F})^2 - 1) - \ln(\det \mathbf{F}) \right], \quad (16)$$

where $\mathbf{F}$ is the deformation gradient and μ, λ are the Lamé coefficients. The Lamé coefficients at every pixel are calculated through the corresponding Young's modulus:

$$E = \frac{\beta}{255.0}(100.0 - 1.0) + 1.0, \quad (17)$$

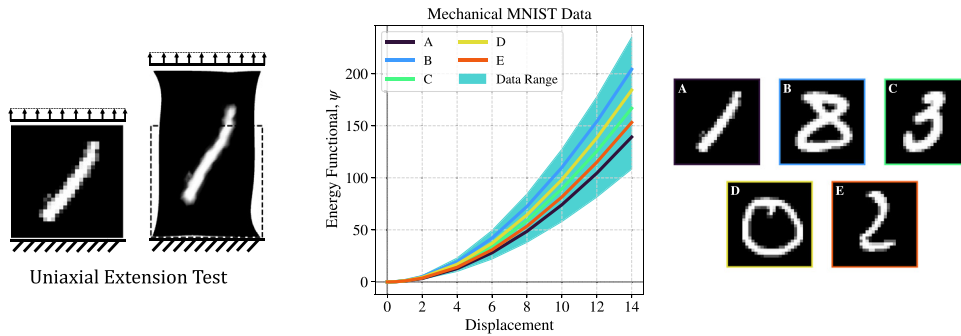


Fig. 11. The Mechanical MNIST data set includes 60,000 training and 10,000 testing microstructures based on the MNIST handwritten digits data set and their corresponding total strain energy response curves in uniaxial extension.

where β is the value of the grayscale bitmap and a constant Poisson's ratio of $\nu = 0.3$, such that:

$$\lambda = \frac{Ev}{(1 + \nu)(1 - 2\nu)} \quad \text{and} \quad \mu = \frac{E}{2(1 + \nu)}. \quad (18)$$

Thus, the resulting microstructures in the data set have stiff elastic inclusions in a two orders of magnitude softer elastic domain.

Finite element simulations are then run on the resulting material domains, with the bottom of the domain fixed and the top displaced according to a set of prescribed displacements. The applied displacements range from 0 to half the initial size of the domain. The finite element simulations are run using the FEniCS platform [61] with a mesh size of 39,200 quadratic triangular elements which corresponds to approximately 140 elements per pixel in the MNIST image. The Mechanical MNIST data set includes load cases in addition to the uniaxial extension used in this work, such as shear, equibiaxial extension, and confined compression. Fig. 11 shows 5 samples from the training set as well as the range of the responses in the entire Mechanical MNIST data set.

References

- [1] Salvatore Torquato, Statistical description of microstructures, *Annu. Rev. Mater. Res.* 32 (1) (2002) 77–111.
- [2] Ole Sigmund, Kurt Maute, Topology optimization approaches: A comparative review, *Struct. Multidiscip. Optim.* 48 (6) (2013) 1031–1055.
- [3] Siddhant Kumar, Stephanie Tan, Li Zheng, Dennis M. Kochmann, Inverse-designed spinodoid metamaterials, *npj Comput. Mater.* 6 (1) (2020) 73.
- [4] Ulrike G.K. Wegst, Hao Bai, Eduardo Saiz, Antoni P. Tomsia, Robert O. Ritchie, Bioinspired structural materials, *Nature Mater.* 14 (1) (2015) 23–36.
- [5] Kun Wang, WaiChing Sun, Meta-modeling game for deriving theory-consistent, microstructure-based traction–separation laws via deep reinforcement learning, *Comput. Methods Appl. Mech. Engrg.* 346 (2019) 216–241.
- [6] Xiao Sun, Bahador Bahmani, Nikolaos N. Vlassis, WaiChing Sun, Yanxun Xu, Data-driven discovery of interpretable causal relations for deep learning material laws with uncertainty propagation, *Granul. Matter* 24 (1) (2022) 1–32.
- [7] Heng Chi, Yuyu Zhang, Tsz Ling Elaine Tang, Lucia Mirabella, Livio Dalloro, Le Song, Glaucio H. Paulino, Universal machine learning for topology optimization, *Comput. Methods Appl. Mech. Engrg.* 375 (2021) 112739.
- [8] Zeyu Zhang, Yu Li, Weien Zhou, Xiaoqian Chen, Wen Yao, Yong Zhao, TONR: An exploration for a novel way combining neural network with topology optimization, *Comput. Methods Appl. Mech. Engrg.* 386 (2021) 114083.
- [9] Fernando V. Senhora, Heng Chi, Yuyu Zhang, Lucia Mirabella, Tsz Ling Elaine Tang, Glaucio H. Paulino, Machine learning for topology optimization: Physics-based learning through an independent training strategy, *Comput. Methods Appl. Mech. Engrg.* 398 (2022) 115116.
- [10] Rebekka V. Woldseth, Niels Aage, J. Andreas Bærentzen, Ole Sigmund, On the use of artificial neural networks in topology optimisation, *Struct. Multidiscip. Optim.* 65 (10) (2022) 294.
- [11] Alexander Quinn Nichol, Prafulla Dhariwal, Improved denoising diffusion probabilistic models, in: *International Conference on Machine Learning*, PMLR, 2021, pp. 8162–8171.
- [12] Emma Lejeune, Mechanical MNIST: A benchmark dataset for mechanical metamaterials, *Extreme Mech. Lett.* 36 (2020) 100659.
- [13] Diederik P. Kingma, Max Welling, Auto-encoding variational bayes, 2013, arXiv preprint arXiv:1312.6114.
- [14] Arash Vahdat, Jan Kautz, NVAE: A deep hierarchical variational autoencoder, *Adv. Neural Inf. Process. Syst.* 33 (2020) 19667–19679.
- [15] Nal Kalchbrenner, Aaron Oord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves, Koray Kavukcuoglu, Video pixel networks, in: *International Conference on Machine Learning*, PMLR, 2017, pp. 1771–1779.

- [16] Jacob Menick, Nal Kalchbrenner, Generating high fidelity images with subscale pixel networks and multidimensional upscaling, 2018, arXiv preprint [arXiv:1812.01608](#).
- [17] Ali Razavi, Aaron Van den Oord, Oriol Vinyals, Generating diverse high-fidelity images with vq-vae-2, *Adv. Neural Inf. Process. Syst.* 32 (2019).
- [18] Aaron Courville, Yoshua Bengio, Generative adversarial nets, *Adv. Neural* (2014).
- [19] Tero Karras, Timo Aila, Samuli Laine, Jaakko Lehtinen, Progressive growing of gans for improved quality, stability, and variation, 2017, arXiv preprint [arXiv:1710.10196](#).
- [20] Andrew Brock, Jeff Donahue, Karen Simonyan, Large scale GAN training for high fidelity natural image synthesis, 2018, arXiv preprint [arXiv:1809.11096](#).
- [21] Steve Kench, Samuel J. Cooper, Generating 3D structures from a 2D slice with GAN-based dimensionality expansion, 2021, arXiv preprint [arXiv:2102.07708](#).
- [22] Sehyun Chun, Sidhartha Roy, Yen Thi Nguyen, Joseph B. Choi, H.S. Udaykumar, Stephen S. Baek, Deep learning for synthetic microstructure generation in a materials-by-design framework for heterogeneous energetic materials, *Sci. Rep.* 10 (1) (2020) 13307.
- [23] Alexander Henkes, Henning Wessels, Three-dimensional microstructure generation using generative adversarial neural networks in the context of continuum micromechanics, *Comput. Methods Appl. Mech. Engrg.* 400 (2022) 115497.
- [24] Yan Wang, Biting Yu, Lei Wang, Chen Zu, David S Lalush, Weili Lin, Xi Wu, Jiliu Zhou, Dinggang Shen, Luping Zhou, 3D conditional generative adversarial networks for high-quality PET image estimation at low dose, *Neuroimage* 174 (2018) 550–562.
- [25] Albert Pumarola, Stefan Popov, Francesc Moreno-Noguer, Vittorio Ferrari, C-flow: Conditional generative flow models for images and 3d point clouds, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7949–7958.
- [26] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, Sungroh Yoon, Ilvr: Conditioning method for denoising diffusion probabilistic models, 2021, arXiv preprint [arXiv:2108.02938](#).
- [27] Liwei Wang, Yu-Chin Chan, Faez Ahmed, Zhao Liu, Ping Zhu, Wei Chen, Deep generative modeling for mechanistic-based learning and design of metamaterial systems, *Comput. Methods Appl. Mech. Engrg.* 372 (2020) 113377.
- [28] Yongju Kim, Hyung Keun Park, Jaimyun Jung, Peyman Asghari-Rad, Seungchul Lee, Jin You Kim, Hwan Gyo Jung, Hyoung Seop Kim, Exploration of optimal microstructure and mechanical properties in continuous microstructure space using a variational autoencoder, *Mater. Des.* 202 (2021) 109544.
- [29] Phong C.H. Nguyen, Nikolaos N. Vlassis, Bahador Bahmani, WaiChing Sun, H.S. Udaykumar, Stephen S. Baek, Synthesizing controlled microstructures of porous media using generative adversarial networks and reinforcement learning, *Sci. Rep.* 12 (1) (2022) 9034.
- [30] Hiba Kobeissi, Saeed Mohammadzadeh, Emma Lejeune, Enhancing mechanical metamaterials with a generative model-based augmented training dataset, *J. Biomech. Eng.* 144 (12) (2022) 121002.
- [31] Rahul Singh, Viraj Shah, Balaji Pokuri, Soumik Sarkar, Baskar Ganapathysubramanian, Chinmay Hegde, Physics-aware deep generative models for creating synthetic microstructures, 2018, arXiv preprint [arXiv:1811.09669](#).
- [32] Jin-Woong Lee, Nam Hoon Goo, Woon Bae Park, Myungho Pyo, Kee-Sun Sohn, Virtual microstructure design for steels using generative adversarial networks, *Eng. Rep.* 3 (1) (2021) e12274.
- [33] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, Improved techniques for training gans, *Adv. Neural Inf. Process. Syst.* 29 (2016).
- [34] Martin Arjovsky, Léon Bottou, Towards principled methods for training generative adversarial networks, 2017, arXiv preprint [arXiv:1701.04862](#).
- [35] Prafulla Dhariwal, Alexander Nichol, Diffusion models beat gans on image synthesis, *Adv. Neural Inf. Process. Syst.* 34 (2021) 8780–8794.
- [36] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, Surya Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, in: *International Conference on Machine Learning*, PMLR, 2015, pp. 2256–2265.
- [37] Yang Song, Stefano Ermon, Improved techniques for training score-based generative models, *Adv. Neural Inf. Process. Syst.* 33 (2020) 12438–12448.
- [38] Jonathan Ho, Ajay Jain, Pieter Abbeel, Denoising diffusion probabilistic models, *Adv. Neural Inf. Process. Syst.* 33 (2020) 6840–6851.
- [39] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, Mark Chen, Hierarchical text-conditional image generation with clip latents, 2022, arXiv preprint [arXiv:2204.06125](#).
- [40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, Björn Ommer, High-resolution image synthesis with latent diffusion models, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10684–10695.
- [41] Aapo Hyvärinen, Peter Dayan, Estimation of non-normalized statistical models by score matching, *J. Mach. Learn. Res.* 6 (4) (2005).
- [42] Jiaming Song, Chenlin Meng, Stefano Ermon, Denoising diffusion implicit models, 2020, arXiv preprint [arXiv:2010.02502](#).
- [43] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J. Weiss, Mohammad Norouzi, William Chan, Wavegrad: Estimating gradients for waveform generation, 2020, arXiv preprint [arXiv:2009.00713](#).
- [44] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, Bryan Catanzaro, Diffwave: A versatile diffusion model for audio synthesis, 2020, arXiv preprint [arXiv:2009.09761](#).
- [45] Kashif Rasul, Calvin Seward, Ingmar Schuster, Roland Vollgraf, Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting, in: *International Conference on Machine Learning*, PMLR, 2021, pp. 8857–8868.
- [46] Xiaolin Li, Yichi Zhang, He Zhao, Craig Burkhart, L. Catherine Brinson, Wei Chen, A transfer learning approach for microstructure reconstruction and structure-property predictions, *Sci. Rep.* 8 (1) (2018) 13461.
- [47] Yixing Wang, Min Zhang, Anqi Lin, Akshay Iyer, Aditya Shanker Prasad, Xiaolin Li, Yichi Zhang, Linda S. Schadler, Wei Chen, L. Catherine Brinson, Mining structure–property relationships in polymer nanocomposites using data driven finite element analysis and multi-task convolutional neural networks, *Mol. Syst. Des. Eng.* 5 (5) (2020) 962–975.

- [48] Tim Salimans, Andrej Karpathy, Xi Chen, Diederik P. Kingma, Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications, 2017, arXiv preprint [arXiv:1701.05517](#).
- [49] Sergey Zagoruyko, Nikos Komodakis, Wide residual networks, 2016, arXiv preprint [arXiv:1605.07146](#).
- [50] Li Deng, The mnist database of handwritten digit images for machine learning research, *IEEE Signal Process. Mag.* 29 (6) (2012) 141–142.
- [51] Alex Krizhevsky, Geoffrey Hinton, et al., Learning Multiple Layers of Features from Tiny Images, Toronto, ON, Canada, 2009.
- [52] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, Li Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 248–255.
- [53] Ian J. Goodfellow, Jonathon Shlens, Christian Szegedy, Explaining and harnessing adversarial examples, 2014, arXiv preprint [arXiv:1412.6572](#).
- [54] Durk P. Kingma, Prafulla Dhariwal, Glow: Generative flow with invertible 1x1 convolutions, *Adv. Neural Inf. Process. Syst.* 31 (2018).
- [55] Lars Mescheder, Sebastian Nowozin, Andreas Geiger, Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 2391–2400.
- [56] Yoshua Bengio, Aaron Courville, Pascal Vincent, Representation learning: A review and new perspectives, *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (8) (2013) 1798–1828.
- [57] Diederik P. Kingma, Jimmy Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint [arXiv:1412.6980](#).
- [58] Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, Jong Chul Ye, Improving diffusion models for inverse problems using manifold constraints, 2022, arXiv preprint [arXiv:2206.00941](#).
- [59] Sarah Tan, Rich Caruana, Giles Hooker, Yin Lou, Distill-and-compare: Auditing black-box models using transparent model distillation, in: Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, 2018, pp. 303–310.
- [60] Ramaravind Kommiya Mothilal, Divyat Mahajan, Chenhao Tan, Amit Sharma, Towards unifying feature attribution and counterfactual explanations: Different means to the same end, in: Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society, 2021, pp. 652–663.
- [61] Martin Alnæs, Jan Blechta, Johan Hake, August Johansson, Benjamin Kehlet, Anders Logg, Chris Richardson, Johannes Ring, Marie E. Rognes, Garth N. Wells, The FEniCS project version 1.5, *Arch. Numer. Softw.* 3 (100) (2015).