



Full length article



Characterization of porous membranes using artificial neural networks

Yinghan Zhao^a, Patrick Altschuh^{a,b}, Jay Santoki^a, Lars Griem^a, Giovanna Tosato^a,
Michael Selzer^{a,b}, Arnd Koeppe^{a,*}, Britta Nestler^{a,b}

^a Institute for Applied Materials-Microstructure Modelling and Simulation, Karlsruhe Institute of Technology, Strasse am Forum 7, Karlsruhe, 76131, Germany

^b Institute for Digital Materials, Karlsruhe University of Applied Sciences, Moltkestrasse 30, Karlsruhe, 76133, Germany

ARTICLE INFO

Keywords:

Porous membranes
Microstructure characterization
Machine learning
Variational autoencoder
Bayesian optimization

ABSTRACT

Porous membranes have been utilized intensively in a wide range of fields due to their special characteristics and a rigorous characterization of their microstructures is crucial for understanding their properties and improving the performance for target applications. A promising method for the quantitative analysis of porous structures leverages the physics-based generation of porous structures at the pore scale, which can be validated against real experimental microstructures, followed by building the process–structure–property relationships with data-driven algorithms such as artificial neural networks. In this study, a Variational AutoEncoder (VAE) neural network model is used to characterize the 3D structural information of porous materials and to represent them with low-dimensional latent variables, which further model the structure–property relationship and solve the inverse problem of process–structure linkage combined with the Bayesian optimization method. Our methods provide a quantitative way to learn structural descriptors in an unsupervised manner which can characterize porous microstructures robustly.

1. Introduction

Porous membranes are popular in a wide range of applications, including separation in water filtration as well as hemodialysis, diagnostic tools such as sensing platforms, supporting scaffold in tissue engineering, and energy harvesting/storage devices [1,2]. These applications are enabled by the performant and tunable characteristics of porous membranes, such as pore sizes, pore shapes, pore orientations, densities, surface-to-volume ratios, and their spatial distributions, which facilitate combinations of multiple, often conflicting functions. Nowadays, functional porous membranes usually have complex microstructural morphologies in order to satisfy different requirements, enabled by modern synthesis methods and fabrication techniques [3,4]. Hence, a rigorous characterization of the microstructures of porous membranes is crucial for understanding their properties and improving the performance for desired purposes.

To apply porous membranes for industrial or medical purposes, their microstructural morphologies, which are greatly influenced by the production process, need to be carefully considered. Hence, it would be beneficial to develop reliable methods for extracting the salient morphological features of porous membranes and relate them to both processing methods and properties of interest. Such kind of Process–Structure–Property (PSP) linkage is essential in the field of

material design [5,6]. Usually, more attention is paid to establish process–property relationships directly (the structure is bypassed), or to build structure–property linkages. Accordingly, problems in the context of process–structure linkage have received fewer investigations [7, 8], even though they allow important insights, particularly for reverse engineering. Therefore, in this work, the full process–structure–property chain is investigated in two parts: the process–structure and the structure–property linkage. The main challenge in these problems lies in the microstructure characterization, which involves the representation of microstructure in a set of comprehensive microstructure descriptors that can capture the morphologies of the microstructure in a robust way.

A qualitative and quantitative representation of the microstructure plays an essential role in building process–structure–property linkages. The most widely used conventional methods for characterizing microstructures are: (i) deterministic or statistical physical descriptors, which include composition descriptors, dispersion status descriptors and geometry descriptors [9]; (ii) statistical function descriptors, which usually employ correlation functions, for examples, n -point statistics, especially 2-point statistics [10,11] (a good balance between capturing rich microstructural information and feasible calculation and can be followed by statistical procedures such as principal component

* Corresponding author.

E-mail addresses: yinghan.zhao@kit.edu (Y. Zhao), patrick.altschuh@kit.edu (P. Altschuh), jaykumar.santoki2@kit.edu (J. Santoki), lars.griem@kit.edu (L. Griem), giovanna.tosato@kit.edu (G. Tosato), michael.selzer@kit.edu (M. Selzer), arnd.koeppe@kit.edu (A. Koeppe), britta.nestler@kit.edu (B. Nestler).

<https://doi.org/10.1016/j.actamat.2023.118922>

Received 15 November 2022; Received in revised form 27 March 2023; Accepted 7 April 2023

Available online 19 April 2023

1359-6454/© 2023 The Author(s). Published by Elsevier Ltd on behalf of Acta Materialia Inc. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

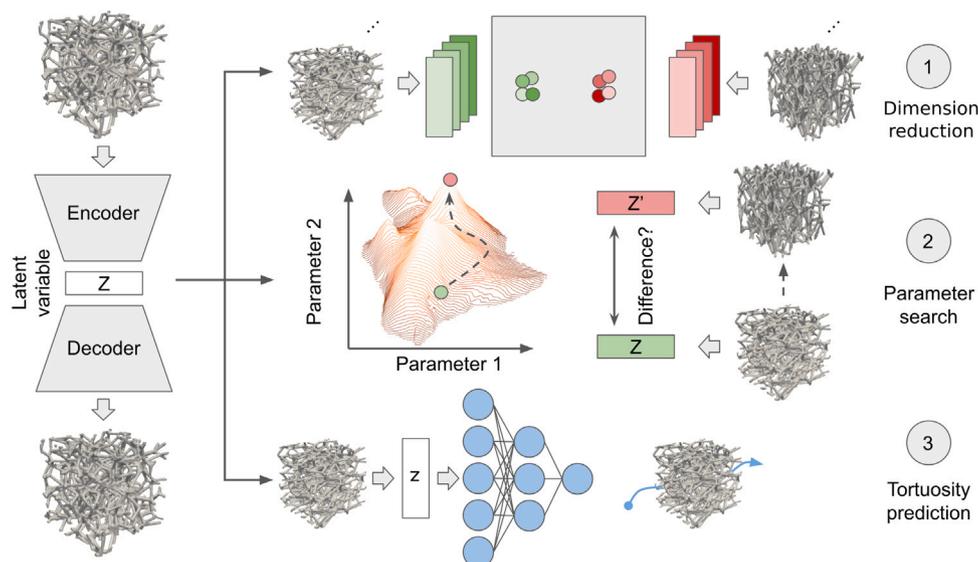


Fig. 1. Schematic diagram of the variational autoencoder model and its applications in this work. The generated 3D porous structure is represented by the model as a low-dimensional latent variable, which is used as a data-driven descriptor for three downstream tasks: estimating the properties of new unseen test sample by finding the nearest neighbor, solving the inverse problem of process–structure linkage, and establishing structure–property relationship.

analysis to extract the most important features [12–16]), lineal path functions [17], frequency domain-based statistics using fast Fourier transformation [18], random fields [19–21]. Apart from these methods, data-driven descriptors have recently gained immense popularity, which usually involve the use of latent variables to represent structures. These latent variables are usually learned from the data by adopting unsupervised machine learning algorithms. In particular, deep neural networks, such as Generative Adversarial Network (GAN) [22] and Variational AutoEncoder (VAE) [23,24], due to their powerful fitting and (for the most part) automatic feature extraction capabilities, have successfully solved many machine-learning problems, such as reconstructing/generating structures as well as linking structures and properties [25–33]. For more comprehensive reviews of characterization approaches, interested readers may refer to the works of Li [34] and Bostanabad [35].

In order to study the structures of porous membranes in a more qualitative and quantitative way on a larger scale, algorithmically generated structures on the pore scale are essential to establish a large database containing broad structural property information. Morphologies of structures can be controlled by varying the generation parameters, which allows the generated structures to be compared with real structures. Since the generation of structures by computational algorithms is rapid, a large number of structures with different morphologies can be produced in short time. Furthermore, these generated microstructures can provide the basis for data science methods that require large amounts of data to achieve good performance. In this work, a large number of digital structures, whose morphologies are based on reconstructed real samples, are generated by physics-based algorithms implemented in an in-house simulation software package PACE3D [36]. To characterize the structure, a neural network model will learn the corresponding data-driven descriptor in an unsupervised manner. Specifically, a Variational AutoEncoder (VAE) neural network is adopted, which is a deep learning model designed for learning latent representations and generating new samples. VAE models have been successfully adopted in several studies in both fields of materials science and physics [37,38]. They excel in nonlinear projections of high-dimensional microstructural information into low-dimensional latent representations in an unsupervised way. Thus, explicitly labeling the input data is not necessary, though it can extend the range of applications considerably. Finally, sampling from the latent space can yield new samples that obey the original data

distribution. These features facilitate many downstream applications such as building the process–structure–property relationships [39–42], microstructure generation [43,44], and inverse design [45–47].

This study explores the ability of the VAE model to characterize the structural information of porous membranes and to establish the PSP linkages. In particular, two types of problems related to PSP linkages are studied based on the low-dimensional structural descriptors learned from neural networks: building structure–property relationship and solving the inverse problem of process–structure linkage. A schematic diagram of these procedures is shown in Fig. 1 and the following contents are discussed in details: first, the capabilities of the model to extract microstructural features and to represent microstructures as low-dimensional latent variables, as well as their relationship to microstructural properties, are analyzed. Besides, properties of unseen microstructures are inferred by finding the most similar known microstructure (nearest neighbor) in low-dimensional space as they are clustered closer to each other. Both algorithm-generated porous microstructures and real experimental samples validate its effectiveness. Second, the model’s ability to build PSP linkages is discussed in two aspects: (i) solving the inverse problem of process–structure linkage, to be specific, given a target microstructure, the identification of its corresponding parameters for generation is of interest. The neural network model is used to represent microstructures as low-dimensional variables, and the difference of latent variables between target microstructure and newly generated microstructure is set as the indicator to be optimized. To solve this problem efficiently, Bayesian optimization method [48] is adopted; (ii) establishing structure–property relationship based on the previously obtained model using transfer learning techniques [49], specifically, the model is employed to predict the tortuosity of microstructures.

2. Materials and methods

2.1. Data set

In this section, 3D digital twins of porous membranes are introduced. The data sets representing the digital twins are obtained by applying a Voronoi-tessellation based generation algorithm and by converting reconstructed computer tomography scans from the real membrane microstructure into 3D data of voxels.

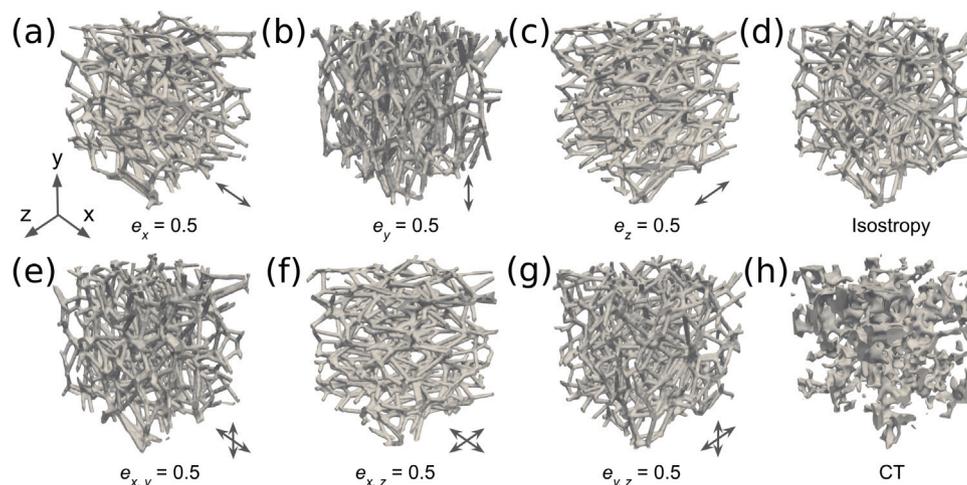


Fig. 2. Examples of generated porous microstructure stretched in different directions. Samples (a)–(c) are stretched (with 0.5 stretching intensity) only in x-, y-, and z-directions, respectively, while the sample in image (d) is an isotropic one (no stretch). For samples (e)–(g) they are stretched in two directions at the same time. Sample (h) is a real reconstructed structure from the experiment.

2.1.1. Generated porous microstructure

3D porous microstructures are generated by an in-house implemented generator based on our previous study [12]. The generation process is performed by two main steps: (i) random points are dispersed inside the simulation volume for the construction of the Voronoi diagram, which will divide the domain of interest in Voronoi regions (cells) according to the nearest neighbor rule. Coordinates of these points are determined using a deterministic working random function which can recreate the same random distribution of points by using the (random) seed parameter; (ii) porous microstructure is generated on the framework constructed by edges of the polyhedral Voronoi cells, with stanchions and pores being formed parametrically during the generation process. After these two steps, certain post-processing can be followed such as placing multiple spheres with possible varying radii by obeying a logarithmic distribution function, and smoothing the structure by appropriate Gaussian filters. Porous membranes created using the algorithm described above can hold diverse features. The most influencing factors that determine the morphology of porous microstructure are: the density (number) of Voronoi seed points, the prescribed thickness of ligaments, the resulting overall porosity as well as the pore size (directly related to the number of Voronoi points), and the stretching factor for the x-, y- and z-directions, which orients the ligaments in certain directions to introduce anisotropy to the microstructure. For stretched microstructures, the produced Voronoi diagram is manipulated according to the stretching factor followed by performing the microstructure generation. As a result, the stretching process only leads to a distortion of the Voronoi region while the primary features of the microstructure are remained unaffected.

Features of generated microstructures can be controlled by varying parameters for digital generation, which can yield microstructures with a wide variety of morphology and properties, e.g., the pore size of the membrane structure is indirectly defined by the number of Voronoi points. A smaller number of Voronoi points leads to larger pores and thus also to a higher porosity. In this work, the most important parameters for the generated microstructures are selected as: (i) the number of Voronoi points n : 40–170; (ii) stretching factor e : 0.5–1.0 (only one direction at a time, smaller value stands for stronger stretch); (iii) stanchion radius r : 3.4. Choices of these parameters are made so to cover various morphologies and to match the algorithm-generated microstructures as closely as possible with the real experimental structures (detailed in the next section). For real membranes, their microstructural characteristics can be controlled by processing parameters during the production process, such as temperature, humidity, evaporation speed and composition of the added solvents, etc. [50]. In this manuscript,

the research interest is how to use optimization algorithms (detailed in section Bayesian optimization) to quickly find the parameters for the generation of given digital membranes, however, this principle is promising to be adapted and transferred to real production scenarios. Since the parameters used for digital data generation do not directly translate into real processing parameters, further studies are required, e.g. a quantitative and qualitative analysis of their linkages and influences on resulting microstructures.

In this work, microstructures are characterized by using the data-driven descriptors, *i.e.*, features of microstructures are learned by the neural network model automatically in an unsupervised manner and microstructures are represented as low-dimensional latent variables. The analysis of latent variables and their association with microstructural properties are discussed in the results sections. For the training of the Variational AutoEncoder (VAE) neural network model used for data-driven characterization, a data set which consists of a total of 10000 porous microstructures (generated by uniformly sampling within the selected range of parameters for generation) with a domain size of 200^3 cells (about $30 \times 30 \times 30 \mu\text{m}$ corresponding to the real physical world) is produced by algorithm-based generation (a few representative ones are depicted in Fig. 2a–d). For generating digital membranes, the in-house developed PACE3D simulations tool was adopted and it took about 15 s (averaged on 100 generations on a platform with CPU Intel Xeon Gold 6146 \times 2, 24 cores, 48 threads) to generate a single sample (and in total about 42 h for 10000 samples as data set). These microstructures serve as the inputs for training the model. To meet the input requirements of the proposed VAE model, generated microstructures are resized to 128^3 using spline interpolation [51]. In addition to the generated microstructures which are stretched in single direction, another batch of microstructures (~ 3000) which are stretched in two directions (a few representative ones are depicted in Fig. 2e–g) are also generated to serve as the additional test data set for validating the VAE model's ability to characterize porous microstructures.

2.1.2. Real experimental porous microstructure

The geometric anisotropy is one of the most interested properties, as this can reflect many microstructural properties of the porous microstructure, such as the permeability in different flow directions. Hence, quantification of the degree of average anisotropy in the real porous microstructure is one of the main interests. In this study, real 3D images of three different microstructure samples are obtained by high resolution X-ray Computer Tomography (XCT), whose anisotropy is of the most interest. The XCT scans are carried out with ID16B-NA beamline at the European Synchrotron Radiation Facility (ESRF) in

Grenoble [52]. The beamline has a minimum and maximum beam size of $50.0 \times 50.0 \text{ nm}^2$ up to $1.0 \times 1.0 \text{ }\mu\text{m}$ using an energy range of 6.0 keV to 65.0 keV. The reconstructed 3D microstructure samples are composed of 900 equally spaced 2D images with the resolution of $150 \text{ nm pixel}^{-1}$. The size of the resulting segment of the real membranes is $900 \times 500 \times 500$ voxels (approximately $75 \times 135 \times 75 \text{ }\mu\text{m}$) for each sample. The resulting 3D data are obtained as a stack of 8-bit (256 intensity levels) gray-scale images in XCT imaging, which is further required to be filtered and then segmented to obtain a clear separation between the microstructure and the pore space. To reduce the noise in the images, a Gaussian 3D filter with a sigma of 3 voxels is applied first. Then the threshold for segmenting the images into zeros and ones (representing the pore space and the membrane material, respectively) is continuously adjusted until the experimentally obtained porosity values match up with the porosity of the segmented microstructures [53]. To meet the need for the input of the model, the original large microstructures are cropped into smaller cubes with size of 200^3 and then are resized to 128^3 . Among them, 15 structures are randomly selected to validate the effectiveness of the neural network model to extract microstructural features and to relate them to interested properties (porosity and anisotropy). These properties are estimated by finding the closest neighbor in the low-dimensional space, to be more specific, their properties are inferred from the most similar generated microstructure found in the training data set. For more details about these real porous membranes, interested readers can refer to our previous work [54]. Fig. 2h displays one of the real reconstructed experimental 3D porous microstructures.

2.2. Machine learning model

A Variational AutoEncoder (VAE) is a generative neural network model that can learn complex probabilistic distributions of data characterized by latent variables. It is usually seen as an extension to the AutoEncoder (AE) model for their architectural affinity though VAE actually belongs to the families of probabilistic graphical models and variational Bayesian methods whose goal and mathematical formulation are significantly different from the standard autoencoder. Both the VAE and AE are composed of two sub-models: an encoder and a decoder. The encoder $\mathbf{z} = f(\mathbf{x})$ compresses the (high-dimensional) input data \mathbf{x} into a low-dimensional multivariate latent variable \mathbf{z} ; the decoder $\mathbf{x}' = g(\mathbf{z})$ reconstructs the latent representation \mathbf{z} back to the original data space to generate the output data \mathbf{x}' . An AE is trained to minimize the discrepancy between inputs \mathbf{x} and their corresponding reconstructions \mathbf{x}' , however, it has the problem of being prone to generate new outputs with unrealistic or invalid topology. VAE was introduced to address this issue by constraining the distribution of the latent variables \mathbf{z} encoded from the input data to the distribution used for output generation, indirectly forcing a match between the distributions of the model outputs and the input data. By sampling the latent variables from the learned distribution, the model can generate new samples that share the same salient features as those in the original data set. This is why the model is commonly referred to as generative model (its performance on generating new porous microstructures is discussed in Supplementary Note 3).

Let the input data set be $\chi = \{\mathbf{x}_i\}_{i=1}^N$, which is characterized by true data distribution $p(\mathbf{x})$ and it is the objective to model. In practical applications, data can be very complicated and its mathematical expression is often unknown, making it difficult to model. In probabilistic graphical models, the observed data \mathbf{x} is assumed to be generated from some latent variables \mathbf{z} . This generation process can be divided into two steps: first, latent variables are sampled from their prior distribution $p_\theta(\mathbf{z})$ and second, data is generated from these sampled latent variables $p_\theta(\mathbf{x}|\mathbf{z})$ with θ as parameters. $p_\theta(\mathbf{z})$ and $p_\theta(\mathbf{x}|\mathbf{z})$ are usually assumed to be some parameterized family of distributions, such as Gaussian distribution. Hence, matching the model distribution with the data distribution is equivalent to maximizing the marginal likelihood $p_\theta(\mathbf{x})$:

$$p_\theta(\mathbf{x}) = \int_{\mathbf{z}} p_\theta(\mathbf{x}, \mathbf{z}) d\mathbf{z} = \int_{\mathbf{z}} p_\theta(\mathbf{x}|\mathbf{z}) p_\theta(\mathbf{z}) d\mathbf{z} = E_{\mathbf{z} \sim p_\theta(\mathbf{z})} p_\theta(\mathbf{x}|\mathbf{z}). \quad (1)$$

Given an observed data sample \mathbf{x} , its latent variable \mathbf{z} can be inferred from its posterior $p_\theta(\mathbf{z}|\mathbf{x})$. However, this distribution is usually hard to compute and can even be intractable. To make it feasible and speed up the calculus, in variational inference, another distribution $q_\phi(\mathbf{z}|\mathbf{x})$ with parameters ϕ from a known family is introduced to approximate the posterior distribution $p_\theta(\mathbf{z}|\mathbf{x})$ by minimizing the KL divergence [55] between these two distributions (generally assumed to be Gaussian): $D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z}|\mathbf{x}))$. In this way, the overall problem is translated into the autoencoder domain, in which the conditional likelihood distribution $p_\theta(\mathbf{x}|\mathbf{z})$ is carried by a decoder neural network model, while the approximated posterior distribution $q_\phi(\mathbf{z}|\mathbf{x})$ is computed by another encoder neural network model. However, the computation of the KL divergence still involves the same intractable integrals as in the posterior computation. To solve this, the original formula is deformed to derive an equivalent function to optimize, which is usually referred to as the Evidence Lower Bound (ELBO) and is written as:

$$L_{\theta, \phi} = \log p_\theta(\mathbf{x}) - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z}|\mathbf{x})) \\ = E_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z})). \quad (2)$$

Maximizing ELBO is equivalent to maximizing the log-likelihood of the observed data and minimizing the KL divergence at the same time. The above function can also be interpreted as two parts: the first term is usually called the reconstruction loss between the input and the output of the network, which is the expected log-likelihood of reconstructing the same input \mathbf{x} flowing through the model; the second term can be seen as a regularization posed on the posteriors which will penalize when it deviates from the prior. To solve the optimization problem of Eq. (2) using the universal gradient descent approach by backpropagation, a reparameterization trick is needed to remove the stochastic sampling from the formation and thus making the training process differentiable.

The architecture of the VAE model used in this work is summarized in Table 1. This architecture is obtained empirically by trial and error based on the performance on validation data (10% of the whole data set). An exhaustive systematic study of varying the architecture was not performed, which may be helpful for improving the performance. For the inputs, as the original data is already in binary format (pore: 0, structure: 1), no scaling is performed. Binary cross-entropy is chosen as the loss function for measuring the reconstruction difference. Besides, Adam optimizer (learning rate: 10^{-5} , moment terms β_1 : 0.9, β_2 : 0.999) is used as the optimization algorithm for stochastic gradient descent to train the neural network model with batch size of 64 for 1000 epochs. The models were trained on one NVIDIA A100-80 GB GPU and it took about 14 h 36 min (averaged on 10 trials of training) to finish the training process. Besides, the machine learning framework CIDS [56] is used to process the data and to build neural network models. The research data infrastructure Kadi4Mat [57] is adopted for efficiently storing and managing the data for the continuous development of the machine learning model.

2.3. Bayesian optimization

Inverse problems in materials science are frequently expressed as building the process–property or process–microstructure linkage, with materials properties or microstructures serving as outputs and manufacturing processes serving as inputs. In this work, solving the inverse problem of process–structure linkage is of interest, *i.e.*, given a target microstructure, how one can identify the corresponding parameters for its generation. The idea is to minimize the difference (here l^p -norm is adopted) between low-dimensional representations of the target microstructure and the generated microstructure with the help of the VAE model. The goal is to find the parameters for digital generation that can reproduce a microstructure that should be as similar as possible to the target structure. Hence, this process is represented as an optimization problem, and an active learning approach based on Bayesian Optimization (BO) is used to solve it effectively.

Table 1
Architecture of the VAE model.

Model	Functions	Parameters	Model dimensions
Encoder	Input	–	128, 128, 128, 1
	Conv3d + ReLU	filters = 16, kernel = 4, stride = 2	64, 64, 64, 16
	Conv3d + ReLU	filters = 32, kernel = 4, stride = 2	32, 32, 32, 32
	Conv3d + ReLU	filters = 64, kernel = 4, stride = 2	16, 16, 16, 64
	Conv3d + ReLU	filters = 128, kernel = 4, stride = 2	8, 8, 8, 128
	Flatten	–	65536, 1
	Dense + ReLU	2048	2048, 1
	Dense ×2	256 ×2	μ : 256, 1; σ : 256, 1
Decoder	Dense + ReLU	65536	65536, 1
	Reshape	–	8, 8, 8, 128
	ConvTranspose3d + ReLU	filters = 64, kernel = 4, stride = 2	16, 16, 16, 64
	ConvTranspose3d + ReLU	filters = 32, kernel = 4, stride = 2	32, 32, 32, 32
	ConvTranspose3d + ReLU	filters = 16, kernel = 4, stride = 2	64, 64, 64, 16
	ConvTranspose3d + Sigmoid	filters = 1, kernel = 4, stride = 2	128, 128, 128, 1

Active learning is a branch of machine learning methods that investigates the best allocation strategy with limited finite resources. Active learning techniques are primarily motivated by the wish to select the most informative points at which the objective function should be queried efficiently by balancing the trade-offs between exploration and exploitation. The goal is to reduce the required number of attempts to achieve an equal or better performance compared to an exhaustive exploration. The Bayesian optimization technique arises and is often considered as one of the most widely used active learning strategies in this context. Bayesian optimization is a class of optimization methods which focuses on solving the problem $\arg \min_{x \in \mathcal{X}} f(x)$ within an input domain $\mathcal{X} \subset \mathbb{R}^d$ as the bounding box. Its ability to optimize the expensive black-box derivative-free functions makes BO extremely versatile [48]. Recently, it has become extremely popular for tuning hyper-parameters in machine learning models, especially deep neural networks [58].

A typical BO algorithm involves two primary components: (i) a surrogate model for statistical inference, typically Gaussian Process (GP) [59,60], which is determined by a mean function μ , a covariance kernel K , and is represented as $f \sim \mathcal{GP}(\mu, K)$; (ii) an acquisition function which decides where to sample the next most promising candidate. The acquisition function is usually an inexpensive function that defines a balance between exploring new unseen areas in the objective space and exploiting areas which are already known to have favorable values [61]. Typical acquisition functions are Probability of Improvement (PI) [62], Expected Improvement (EI) [63,64], and Lower Confidence Bound (LCB) [65]. These functions are different in how the trade-off is made between exploration and exploitation. The decision strategy is important for finding the global optimum efficiently instead of being trapped in a local optimum.

In each iteration of BO, the surrogate model is fitted based on the known samples to obtain information of the statistical mean estimations and uncertainties across the input domain space. The information is then fed into the selected acquisition function to decide the next candidate points that may potentially improve the performance of the target function at best. New validations will be conducted on these new candidates and their results will serve as new inputs to update the BO model for the next iteration. This loop will continue until certain criteria are met, such as the achievement of sufficient performance of the target function. On this account, the merit is that expensive evaluations only need to be conducted on these promising candidates, which has the potential to reduce the number of required evaluations. Following this strategy, the overall computational cost of the optimization process is reduced considerably. In short, by adopting Bayesian optimization, the optimization problem of the original function is replaced with another optimization problem based on a much-cheaper acquisition function. The GP-based (with Matérn 5/2 kernel) Bayesian optimization with EI as the acquisition function is used to solve the inverse problem of building process–structure linkage described previously.

2.4. Tortuosity prediction

A structure–property linkage is built with neural network model in this work. Many properties are of interest for porous media, and among them, tortuosity (τ) is one of the parameters that plays an important role in describing and understanding the transport behavior through the porous microstructures, such as effective diffusivity, permeability, thermal conductivity and so on [66]. The tortuosity is the quantity to characterize paths of particles being transported through the medium [67]. Numerical calculation of tortuosity for the generated microstructures is determined according to the method described in Supplementary Note 4 and is set as the to be predicted quantity by training a neural network model. Transfer learning technique is used with the expectation of improving performance, that is, the new developed regression model is based on the pre-trained VAE model. For comparison, another neural network model with the same architecture will also be trained from scratch.

3. Results and discussion

3.1. Characterization of generated microstructure

A total number of 10000 porous structures are generated using physic-based algorithm for evaluating the Variational AutoEncoder (VAE) model. This data set is divided into 80% of training part, 10% validation part and 10% test part. The training part is used to train the model and the validation part is used to select good hyper-parameters for the model and the training algorithm. After identifying the satisfying architecture of the model, the training and validation parts are merged as a larger training data set to retrain the model for further improving its performance. The remaining 10% test part is utilized for evaluating the final performance of the model. Fig. 3 shows the total loss of the VAE model during the training process of 1000 epochs. In the beginning, the loss decreases quickly until this changing rate eventually drops significantly after about 150 epochs. Then, the loss of test set declines slowly and gradually converges and finally levels off in the last 200 epochs, while the loss of training set is still slowly continuing to decrease. A gap between the loss of the training set (blue solid line) and the test set (gray dashed line) can be observed after about 250 epochs of the training, and it is speculated that the model appears to be slightly over-fitted.

The encoder part of the VAE model processes the algorithm-generated structures and thus represents them as low-dimensional latent variables. It is found that in this latent space, the most interested structural information is scattered across multiple dimensions and no single dimension is found to be obviously well disentangled (detailed in Supplementary Note 1). Hence, to further refine the information and to

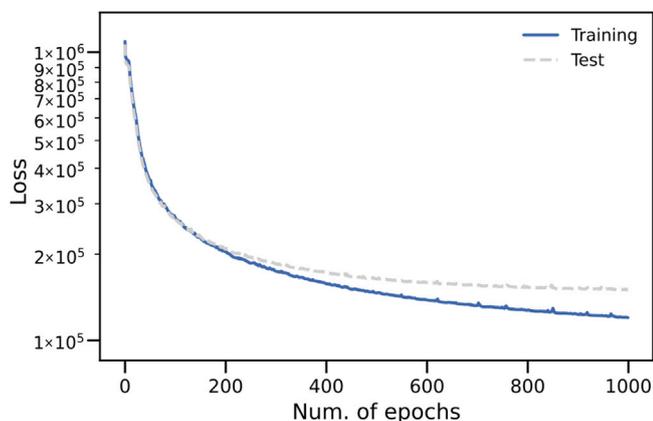


Fig. 3. Total loss of the neural network model during the training process. The loss of the training set is marked with blue solid line while the loss of the test set is marked with gray dashed line.

visualize the result, the Principal Component Analysis (PCA) [68,69] is performed on these latent variables. The distribution of the generated porous microstructures in the space of the first three principal components is shown in Fig. 4. The top two subplots (a and b) exhibit the space distribution of the training set (points without edges) and test set (points with black edges). Besides, the results obtained from AutoEncoder (AE) model is also shown in Supplementary Note 2 for comparison.

The distribution of the generated microstructures in PC1 and PC2 is presented in Fig. 4a and it can be inferred that the directional features of the microstructures correlate mainly to these two principal components. In this figure, the microstructures are color-coded according to their stretching directions. It can be seen that microstructures without preferred stretching direction are concentrated in the center, while structures with different stretching directions are distributed along their respective orientation branches and lie at an angle of about 120 degrees to each other. The embedded-subplot depicts the stretching strength of the microstructure in y -direction, with lighter colors implying a stronger degree of stretching. It can be observed that the color changes from darker to lighter from the center outward in the corresponding branch (microstructures stretched in y -direction) and the farther the microstructure is from the center, the stronger the stretching strength is. Besides, the top and right subplots (the density distribution of the stretching direction) also demonstrates that PC1 and PC2 together can clearly capture the directional information of microstructures. Furthermore, comparing the distribution of training set (points without edges) and validation set (points with black edges), the model performs well on both and can accurately characterize their stretching intensities and directions.

The magnitude of porosity is reflected by the color shade of the data points (darker the color, larger the value of porosity) and it can be found that the value of porosity gradually increases along the third principal component (PC3) axis (right to left). Hence, it is concluded that PC3 mainly captures the porosity information of microstructures. Besides, it can also be observed that the number of Voronoi points used in microstructure generation process decreases (indicated by the sizes of the data points) along the PC3 axis. This is in accordance with the properties of the generated microstructure as more skeletal structures are produced with increasing Voronoi points during generation process, which will take up more space and hence result in a smaller porosity value. In addition, the subplot on the top of Fig. 4b shows the distribution of stretching directions along PC3 axis and it is observed that all the microstructures are uniformly distributed along PC3 axis. Therefore, PC3 is inferred to contain less amount of directional information and only capture information related to porosity.

To further demonstrate the model's ability to extract microstructural information, an additional test data set which contains microstructures stretched in two directions is generated. These new microstructures are also processed by the VAE model to extract structural information, represented as low-dimensional variables and further projected to principal component space. Similar results are shown in Fig. 4c and d. Fig. 4c (PC1 vs. PC2) shows the distribution of stretching directions of all generated microstructures. From the figure it can be known that the model show good performance in extracting the directional information of new microstructures which have not shown in the training period and can project them to reasonable positions according to their stretching direction. For example, the model projects microstructures stretched in both x - and y -directions (purple) to the place between microstructures stretched in only single direction (x or y , blue or orange, respectively). From Fig. 4d (PC2 vs. PC3) it is known that PC3 can also capture the information related to porosity (and the associated number of Voronoi points). Along the PC3 axis, the color of the data points changes continuously from dark to light, implying the decrease of porosity values of microstructures, and it corresponds to an increase in the number of Voronoi points set in the microstructure generation process. This finding is consistent with the conclusions from the previous analysis based on the microstructures stretched in single direction. These results further demonstrate the model's robust ability to extract microstructural information and shows its robust performance on new unseen data.

3.2. Characterization of experimental microstructure

The VAE model is further applied to the 3D experimental microstructures which are reconstructed from CT images. As before, the trained neural network model is used to extract microstructural features from the reconstructed experimental microstructures and to represent them as low-dimensional latent variables. Together with the generated microstructures, PCA is performed on these latent variables. In the principal component space, microstructures with similar properties will clustered closer to each other and it is intended to analyze the interested properties of the experimental microstructure according to the meanings represented by PC1–3. These results are shown in Fig. 5.

First, the stretching direction of reconstructed experimental microstructures is estimated according to PC1–2 as they represent directional information of microstructures. From Fig. 5a it can be clearly seen that experimental microstructures (marked with white stars) are concentrated near the center and biased in the y -axis. Similarly, the distance (similarity) can be calculated in PC1 and PC2 to find the closest algorithm-generated microstructures to estimate the stretching direction of the reconstructed experimental microstructures. Besides, the anisotropy of the microstructure can be judged accordingly. In the present work, the tortuosity is numerically calculated by the simulation method, which is used to approximate the anisotropy of the microstructure, as it can indirectly represent the directional orientation of the microstructure. More specifically, if a microstructure is stretched in a certain direction, the tortuosity value should be small, which means that the microstructure shows a high anisotropy in this direction. The calculated tortuosity values in x -, y -, z -directions of the reconstructed experimental structures are regarded as their true values and they are listed in Table 2. It can be seen that all reconstructed experimental microstructures have the smallest values in the y -direction, implying that they all tend to have a relatively large stretch along the y -direction. It is worth noting that tortuosity values in each direction do not differ much, which indicates the reconstructed experimental structures show only weak anisotropy. The average of the ratio of tortuosity in the y -direction to the x - and z -directions, respectively, is used to approximate the degree of anisotropy in the y -direction. Estimated stretching intensity values of the reconstructed experimental structures according to their closest generated microstructures are given in Table 2. It can

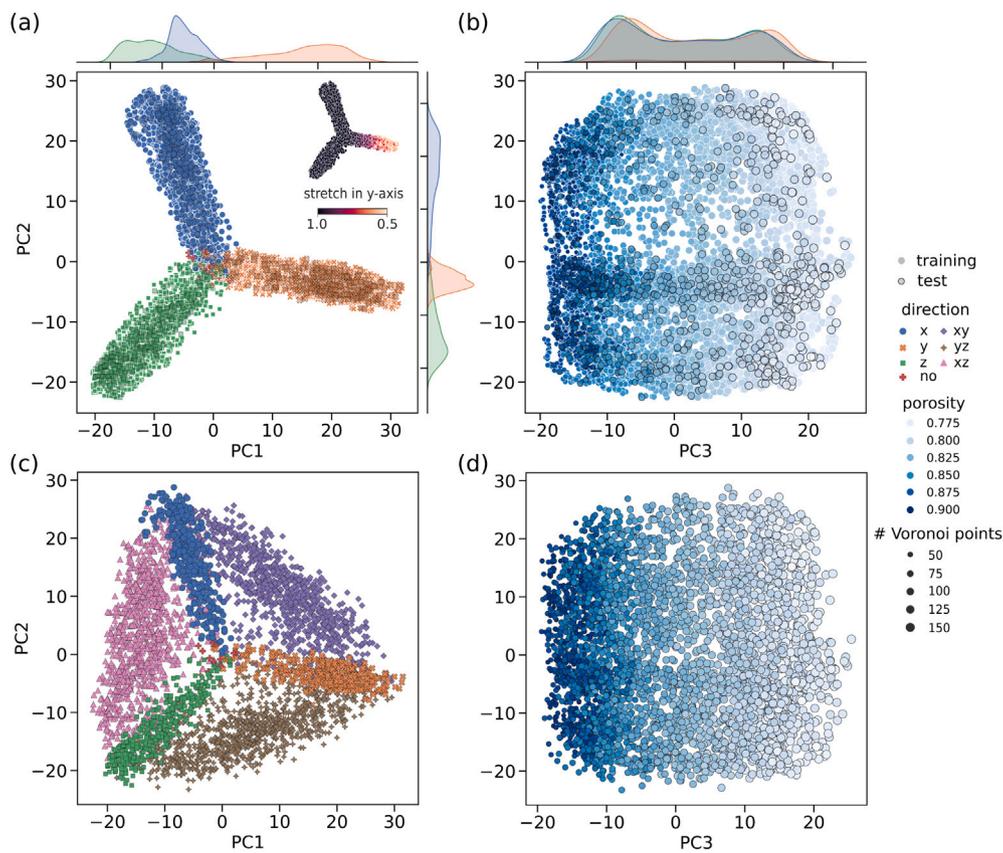


Fig. 4. The distribution of porous microstructures in the first three principal component space (PC1–3). (a) and (b) exhibit microstructures with single stretching direction while (c) and (d) show microstructures having at most two stretching directions. It is found that the first two principal component (PC1–2) together capture the information of stretching direction and strength while PC3 mainly contains information related to porosity/Voronoi points of microstructures.

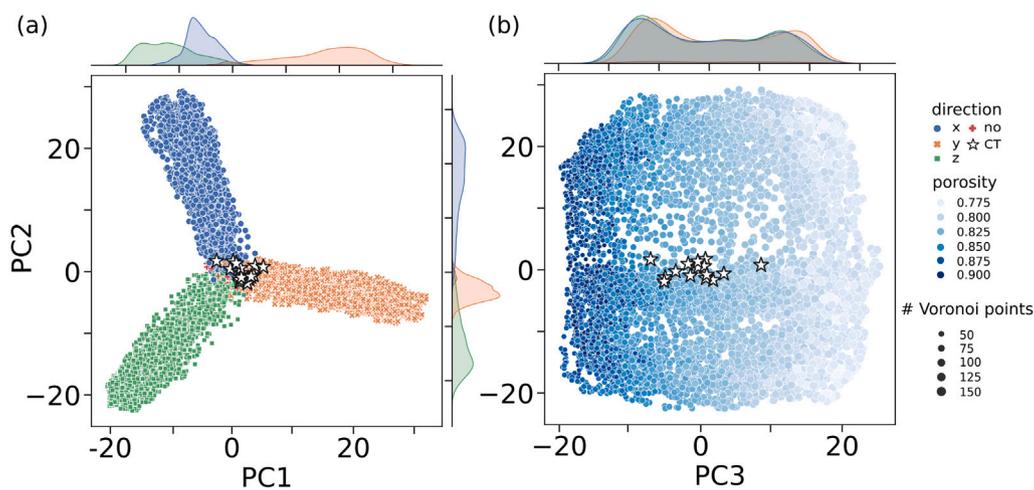


Fig. 5. The distribution of algorithm-generated porous microstructures and real reconstructed microstructures from experiments (marked as CT). In (a) (PC1 vs. PC2) the reconstructed experimental microstructures are concentrated near the center and biased in the *y*-axis. Their microstructural anisotropy (measured in stretching direction) are estimated by finding the closest algorithm-generated microstructures in PC1–2. The porosity value of reconstructed experimental microstructures are estimated in PC3 (b) as it is found to correlate with the information about porosity.

be known that for most of the real experimental microstructures, the estimated stretching directions and intensity values are satisfyingly accurate. However, two wrong estimations of the stretching direction for No. 4 and No. 6 sample are made. This discrepancy is understandable, as the reconstructed experimental microstructures possess only very weak anisotropy (slightly stretched) so inherently it is hard to infer them accurately. Such a phenomena can also be observed in Fig. 5a (PC1 vs. PC2) where the microstructures tend to cluster closely to

the center, and a slight deviation may lead to misjudge the stretching direction.

In addition, it is possible to estimate the porosity value of the experimental microstructure in PC3, which correlates most with the information about porosity. From Fig. 5b it can be seen that the reconstructed experimental microstructures are concentrated in the central place of all the data points, locating at the position in PC3 axis where it should correspond to a porosity value of about 0.84. The porosity value of

Table 2
Numerically calculated property values of experimental microstructures and their estimated (predicted) values.

Sample no.	Porosity			Tortuosity			Geometric anisotropy				
	C	P	Diff. %	Axis			Direction		Intensity		
				x	y	z	C	P	C	P	Diff. %
1	0.846	0.822	-2.791	1.162	1.121	1.161	y	y	0.965	0.980	-1.554
2	0.894	0.836	-6.441	1.101	1.077	1.105	y	y	0.976	0.960	-1.639
3	0.842	0.839	-0.291	1.165	1.119	1.162	y	y	0.962	0.980	1.871
4	0.866	0.858	-0.862	1.127	1.104	1.134	y	x	0.977	0.980	0.307
5	0.850	0.825	-2.919	1.162	1.111	1.159	y	y	0.958	0.960	0.208
6	0.857	0.824	-3.740	1.143	1.125	1.137	y	no	0.987	1.000	1.317
7	0.887	0.835	-5.755	1.106	1.086	1.115	y	y	0.978	0.980	0.204
8	0.861	0.833	-3.207	1.140	1.109	1.139	y	y	0.973	0.990	1.747
9	0.827	0.829	0.242	1.184	1.147	1.188	y	y	0.967	0.950	-1.758
10	0.845	0.839	-0.702	1.167	1.130	1.157	y	y	0.972	0.970	-0.205
11	0.790	0.806	2.046	1.227	1.192	1.250	y	y	0.962	0.950	-1.247
12	0.836	0.829	-0.834	1.169	1.122	1.184	y	y	0.953	0.980	2.833
13	0.898	0.827	-7.869	1.096	1.073	1.105	y	y	0.976	0.980	0.409
14	0.844	0.830	-1.641	1.156	1.115	1.173	y	y	0.957	0.950	-0.731
15	0.842	0.825	-1.967	1.169	1.140	1.158	y	y	0.980	0.970	-1.020

C: calculated value, P: predicted value. Geometric anisotropy is measured in stretching strength.

Table 3
Results of Bayesian optimization in finding parameters for digital generation of target microstructures.

Sample no.		Stanchion radius	Num. of Voronoi points	Stretch			Diff. in z space
				x	y	z	
A	Real	3.40	100	1.00	0.75	1.00	
	Median	3.58	100	1.00	0.75	1.00	3.82
	Best	3.39	100	1.00	0.75	1.00	0.18
B	Real	3.40	180	1.00	0.75	1.00	
	Median	3.57	183	1.00	0.75	1.00	4.14
	Best	3.28	179	1.00	0.74	1.00	1.51
C	Real	5.40	180	1.00	0.75	1.00	
	Median	5.35	184	1.00	0.75	1.00	8.03
	Best	5.57	179	1.00	0.75	1.00	2.02
D	Real	3.40	180	0.65	0.55	1.00	
	Median	3.77	179	0.61	0.50	1.00	8.23
	Best	3.40	178	0.63	0.55	1.00	3.12

experimental microstructures can be estimated by finding the most similar algorithm-generated microstructures (*i.e.*, nearest neighbor) where the “similarity” is measured by calculating the difference in PC3. The calculated porosity of reconstructed experimental microstructures, as well as the estimated (predicted) porosity value according to the closest found algorithm-generated microstructures, are listed in Table 2. It is found that there are some differences between the numerically calculated porosity value and the estimated value. By comparing the reconstructed experimental and algorithm-generated microstructures, it is speculated that such kind of discrepancy is mainly caused by some dissimilarities of real reconstructed experimental and algorithm-generated microstructures. Though our generated microstructures can resemble the morphology of real microstructures, artifacts caused by *e.g.*, noise during the reconstruction process of experimental microstructures, can lead to some deviations. As a result, this impairs the performance of the model’s ability to characterize microstructures to a certain extent. In addition, the limited amount of the data used to train the model somehow restricts model’s ability: only about 10000 microstructures are adopted and they may not be able to completely cover all possible configurations and morphologies of the real experimental microstructures. It is therefore understandable that for some microstructures (for examples, No. 2 and No. 11, whose porosity values locate at the marginal place of the porosity distribution of the training set), no good corresponding algorithm-generated microstructures are identified to estimate the porosity accurately. In general, the VAE model perform well

on extracting directional information of the reconstructed experimental microstructure and their stretching direction (anisotropy) as well as stretching strength can be estimated with high accuracy.

3.3. Inverse problem of the process–structure linkage

The VAE model is used to extract structural information and represent microstructures as latent variables. With these low-dimensional variables, mathematical metrics such as l^2 -norm, can be calculated to measure the difference between various microstructures. Taking advantage of this kind of data-driven descriptors of microstructures, an inverse design problem of the process–structure linkage is explored in this section. In order to verify the validity of the model’s performance on unseen microstructures, one microstructure (marked as A) whose morphology is similar to the training data, and three microstructures (B, C and D) with morphologies that are quite different from the training data, are intentionally generated and their corresponding real parameters for generation are listed in Table 3. The corresponding generation parameters to produce these microstructures are assumed unknown to us and the goal is to find these values. Such a problem is treated as an optimization task, and an active learning approach based on Bayesian Optimization (BO) is used to solve it effectively.

The convergence plot of the difference (l^2 -norm of the 256-dimensional latent variables μ) between target microstructure and generated microstructure being minimized during the optimization process

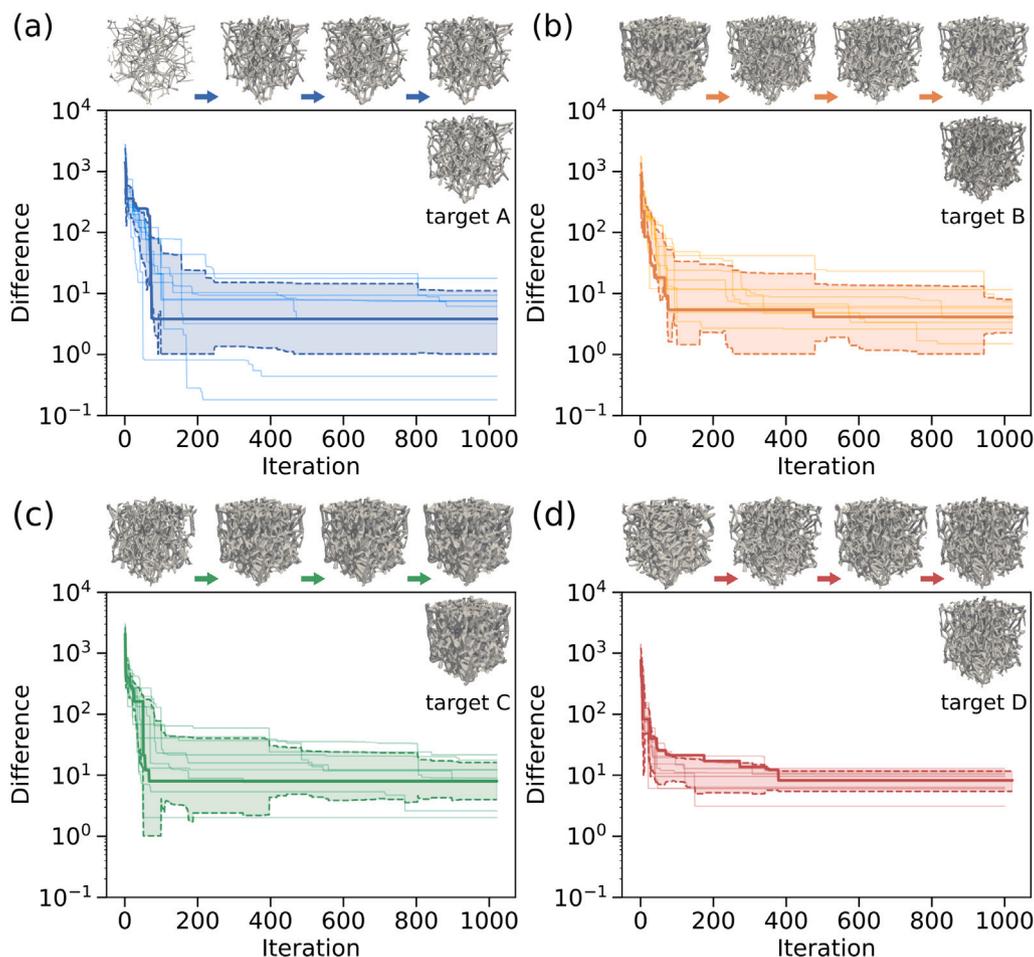


Fig. 6. Iteration history of the Bayesian optimization in solving the inverse problem of process–structure linkage for four different target microstructures (a–d). For each target microstructure (A–D), the optimization process is repeated 10 times and the result corresponding to the median of final differences is marked with darker color. The difference between the target microstructure and the generated one is measured by calculating the Euclidean distance in the latent space.

is shown in Fig. 6. In the beginning of the optimization, 20 random microstructures are generated to serve as the input to initialize the Bayesian optimizer and the number of iterations is relaxed to 1000 to ensure that the optimal parameters can be reached as close as possible. For each target microstructure, the optimization process is repeated 10 times to draw basic statistical information for better evaluating the optimization results. Among the results (marked in lighter lines), the round of iteration result corresponding to the median is marked in darker color. For better visualization, a filled area of one standard deviation is also plotted for each sample to see how the error (difference) drops as BO iteration proceeds and the variation between different iteration trials. Besides, to visually display the optimization process, above each subplot, the corresponding microstructures generated by the found parameters for generation at the first trial after the initialization, at one-half and one-fourth states of initial difference values, and final found value, are also drawn respectively. True parameters for generation that generate the structures and results of the BO are summarized in Table 3 and the corresponding time cost is given in Supplementary Note 5.

From the BO results it is found that the corresponding parameters for generation of the target microstructures can be satisfyingly estimated despite a certain amount of discrepancy. Besides, it can be observed from Fig. 6 that during the optimization process, similarity of the generated microstructure to the corresponding target is improved significantly at the very beginning part of the optimization process and the degree of improvement gradually slows down as the number of iterations grows. It is noticeable that the optimization process have also experienced several plateaus, which is due to the nature of BO

as it takes both the mean estimation and the uncertainty in the meta-model into consideration (the so-called exploration vs. exploitation). Comparing the optimization results of the four target microstructures, it is found that for microstructures (A, B in Table 3) that are more similar to the original training set, the VAE model can better characterize them and hence can better distinguish them, resulting in better performance and fewer iterations needed to find the optimum. On the other hand, for those microstructures (C, D in Table 3) that are more different from the original training set, the model is also able to extract their structural information and differentiate between them, though the performance is slightly poorer. Thus, more trials may be required in these cases for BO to find the optimal value. Overall, the process of combining data-driven microstructural descriptor learned by the VAE model with the Bayesian optimization can provide a satisfying solution to the inverse problem of structure–process linkage, *i.e.*, finding the parameters for generation of unknown target microstructure. This further validates the ability of the VAE model in characterizing the porous microstructures and proves the applicability of the Bayesian optimization method in solving the considered problem effectively. It should be noted that in this work, the BO is conducted in a single-step manner, *i.e.*, only one new candidate is generated and evaluated in each iteration, other more advanced BO methods, such as batch-parallel or asynchronous parallel BO, are promising to accelerate the whole process [70,71]. Besides, finding the appropriate parameters that can generate microstructure resembling real experimental microstructure will be an interesting extension of this strategy and is remained for future study.

3.4. Transfer learning for predicting tortuosity

The performance of neural network model is often related to the setting of the initial weights of the model. Hence, good choices of setting these initial weights play an important role in obtaining a good neural network model. The VAE model was trained on generated porous microstructures in an unsupervised manner and the model has learned some knowledge about extracting the structural information. Hence, it is expected that re-using parts of the trained model can show better performance on prediction tasks such as building the structure–property linkage. To be specific, the encoder part of the VAE model is taken out to serve as the basic part of building a neural network-based regressor, that is, these pre-trained weights are re-used and are believed to be able to yield better results than random initialization of the weights, especially when only limited amount of labeled data is provided. This idea is one of the key points in Transfer Learning (TL).

In this work, a structure–property linkage is built by training another neural network model, that is, the relationship between porous microstructures and their corresponding tortuosity values (described in Materials and Methods: tortuosity prediction). For this purpose, an additional 300 new unseen structures (250/50 training/test split) with large differences (number of Voronoi points $n = 180\text{--}250$, stanchion radius $r = 2.8$) from training set in previous sections are generated and simulations are performed to calculate tortuosity values (labels). Fine-tuning is adopted as a means to achieve transfer learning. More specifically, the whole process is divided into three steps: (i) transferring architecture and weights of the encoder: the pre-trained encoder part of the VAE model is taken out to serve as the microstructural information extractor and it is the basic building block for the new neural network model. It should be noted that no sampling process is performed on the latent variables (unlike that during the previous training of the VAE model); (ii) adding dense layers at the end of the encoder: immediately following the output of the encoder, two more dense layers (64 ReLU) are connected and a final linear output neuron is added to yield the predicted scalar value (tortuosity). The weights of these layers are randomly initialized using the Xavier uniform initialization method [72]. Furthermore, a weak L_2 constraint (10^{-5}) is imposed as regularization term to mitigate the over-fitting phenomenon; (iii) fine-tuning weights of the model: the Mean Squared Error (MSE) is chosen as the loss function for this regression task. Besides, the Adam optimizer [73] is adopted as the optimization algorithm for the training and the parameters are set to be the same as those used in training the VAE model. The optimization is set to last for 100 epochs to ensure convergence. The time cost for calculating the tortuosity, training the regressor model as well as predicting on samples, is summarized in Supplementary Note 5. During the training process, the encoder part is frozen so only the weights of fully-connected layers are updated. It is noted that for the sake of brevity, other network architectures and training strategies are not exhaustively investigated in this study.

To demonstrate the benefit of employing TL technique for developing the structure–property linkage model, another neural network model, which is built with exactly the same architecture but with all weights initialized randomly (rather than using the pre-trained weights of encoder), is also trained for comparison (referred to as the “training from scratch”). The mean-squared-errors are calculated on both the training and testing data to evaluate the performance of these two models trained with different strategies. To better assess the models’ performance, the training process and evaluation are repeated 20 times and the results are shown in Fig. 7. It can be noticed that results obtained from the TL method (averaged MSE: 0.00401 on training data and 0.00502 on test data) are better than those from the “training from scratch” method (averaged MSE: 0.00460 on training data and 0.00567 on test data). Besides, results from the TL method are also significantly more stable as they exhibit smaller variance. For both models, the errors on the test set are slightly larger than those on the training set, but the difference between TL strategy is smaller

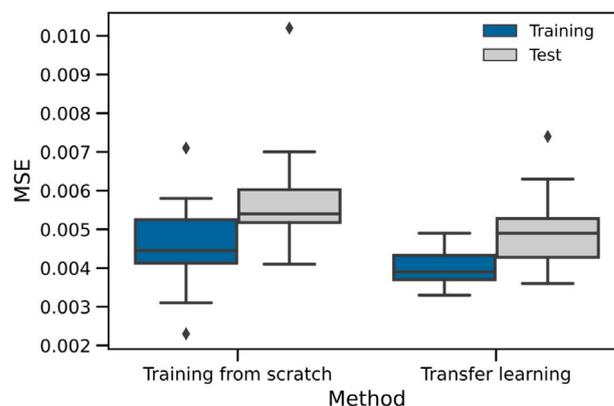


Fig. 7. Mean square errors (MSE) of predicted tortuosity on both training set and test set for two neural network models with the same architecture but trained with different strategies: one is trained from the scratch and another adopts the transfer learning strategy.

than that with the “training from scratch” method, indicating that models obtained with transfer learning strategy are more robust in extrapolation. It is speculated that such a privilege may come from the anti-over-fitting effect introduced by freezing the weights in the pre-trained encoder model. Furthermore, although occasionally the model trained from scratch can achieve smaller value of MSE on the training set, its performance on the test set is apparently worse than that of TL strategy. Based on these results, it is inferred that the transfer learning strategy can facilitate the development of structure–property prediction model in many aspects such as improving its accuracy and stability, especially when only limited amount of data is provided. These results are consistent with the previous hypothesis that the model trained in an unsupervised manner have gained useful knowledge from the data and can be transferred to other downstream tasks which helps achieve better performance. Moreover, it should also be mentioned that there are many other model architecture designs and training strategies. Their impacts on the results of transfer learning, and how useful transfer learning can be in different application scenarios, are worth exploring and is of interest for future work.

4. Conclusion

The Variational AutoEncoder (VAE) neural network model is used to characterize the three-dimensional microstructures of porous membranes. The model is used to extract structural information and represent it with low-dimensional latent variables as data-driven descriptors. The ability of the model to characterize porous structural information is verified in three aspects regarding the establishment of process–structure–property linkages. First, Principal Component Analysis (PCA) is performed to refine information and the meanings of the first 3 principal components are interpreted, namely porosity and stretching direction as well as factor. These properties can be estimated by finding the most similar known microstructures in the corresponding principal component space and were validated on both unseen algorithm-generated porous microstructures and reconstructed experimental microstructures. Second, Bayesian Optimization (BO) is utilized to efficiently solve the inverse problem of process–structure linkage, *i.e.*, finding the parameters for generation of a given target microstructure. The results were validated on several microstructures whose morphologies are different from the previous training data. Third, following the transfer learning strategy, a structure–property linkage (predicting the tortuosity of microstructure) is established by re-using the parts of the VAE model. The results show that its performance is better and more stable than the model trained from scratch, especially when only a small amount of data is provided. The above

studies demonstrate the robust ability of the VAE model to characterize microstructures of porous membranes and has the potential for other geometries.

CRedit authorship contribution statement

Yinghan Zhao: Conceptualization, Methodology. **Patrick Altschuh:** Conceptualization, Generated the data. **Jay Santoki:** Performed calculations. **Lars Griem:** Generated the data. **Giovanna Tosato:** Methodology. **Michael Selzer:** Conceptualization, Funding acquisition, Supervision. **Arnd Koeppe:** Conceptualization, Methodology. **Britta Nestler:** Funding acquisition, Supervision.

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.

Acknowledgment

The supports provided by the following organizations and institutions are gratefully appreciated: data of the porous microstructures as well as specific microstructure characterization methods are supported by the Helmholtz Artificial Intelligence Cooperation Unit (Helmholtz-AI) and funded by the Initiative and Networking Fund (IVF) of the Helmholtz Association (grant number: ZT-I-PF-5-075); results and programming interfaces within the research data infrastructure Kadi4Mat are funded by the German Federal Ministry of Education and Research (BMBF) as part of the Excellence Strategy of the German Federal and State Governments in the FutureFields project Kadi4X and by the Ministry of Science, Research and the Arts Baden-Württemberg (MWK-BW), Germany, in the project MoMaF-Science Data Center (grant number: 34-7547.222); contributions of data processing and workflow applications are funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), under project ID 390874152, POLiS Cluster of Excellence (grant number: UP 33/1), within the “FESTBATT” consortium (grant number: 03XP0435D) by BMBF, Germany and through the MTET program No. 38.04.04 supported by the Helmholtz association, Germany. Besides, this work contributes to the research performed at CELEST (Center for Electrochemical Energy Storage Ulm-Karlsruhe) and was partly performed on the HoreKa supercomputer funded by MWK-BW and BMBF, Germany]. All authors actively discussed the results and contributed to the manuscript.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.actamat.2023.118922>.

References

- [1] M. Ulbricht, Advanced functional polymer membranes, *Polymer* 47 (7) (2006) 2217–2262.
- [2] K. Wang, K. Amin, Z. An, Z. Cai, H. Chen, H. Chen, Y. Dong, X. Feng, W. Fu, J. Gu, et al., Advanced functional polymer materials, *Mater. Chem. Front.* 4 (7) (2020) 1803–1915.
- [3] F. Wang, P. Altschuh, L. Ratke, H. Zhang, M. Selzer, B. Nestler, Progress report on phase separation in polymer solutions, *Adv. Mater.* 31 (26) (2019) 1806733.
- [4] A. Shiohara, B. Prieto-Simon, N.H. Voelcker, Porous polymeric membranes: Fabrication techniques and biomedical applications, *J. Mater. Chem. B* 9 (9) (2021) 2129–2154.
- [5] G.B. Olson, Computational design of hierarchically structured materials, *Science* 277 (5330) (1997) 1237–1242.
- [6] S.R. Kalidindi, Feature engineering of material structure for AI-based materials knowledge systems, *J. Appl. Phys.* 128 (4) (2020) 041103.
- [7] A. Tran, J.A. Mitchell, L.P. Swiler, T. Wildey, An active learning high-throughput microstructure calibration framework for solving inverse structure–process problems in materials informatics, *Acta Mater.* 194 (2020) 80–92.
- [8] P. Honarmandi, V. Attari, R. Arroyave, Accelerated materials design using batch Bayesian optimization: A case study for solving the inverse problem from materials microstructure to process specification, *Comput. Mater. Sci.* 210 (2022) 111417.
- [9] H. Xu, D.A. Dikin, C. Burkhart, W. Chen, Descriptor-based methodology for statistical characterization and 3D reconstruction of microstructural materials, *Comput. Mater. Sci.* 85 (2014) 206–216.
- [10] P.B. Corson, Correlation functions for predicting properties of heterogeneous materials. I. Experimental measurement of spatial correlation functions in multiphase solids, *J. Appl. Phys.* 45 (7) (1974) 3159–3164.
- [11] S. Torquato, G. Stell, Microstructure of two-phase random media. I. The n-point probability functions, *J. Chem. Phys.* 77 (4) (1982) 2071–2077.
- [12] P. Altschuh, Y.C. Yabansu, J. Hötzer, M. Selzer, B. Nestler, S.R. Kalidindi, Data science approaches for microstructure quantification and feature identification in porous membranes, *J. Membr. Sci.* 540 (2017) 88–97.
- [13] Y.C. Yabansu, P. Altschuh, J. Hötzer, M. Selzer, B. Nestler, S.R. Kalidindi, A digital workflow for learning the reduced-order structure-property linkages for permeability of porous membranes, *Acta Mater.* 195 (2020) 668–680.
- [14] S.R. Niezgoda, A.K. Kanjarla, S.R. Kalidindi, Novel microstructure quantification framework for databasing, visualization, and analysis of microstructure data, *Integrating Mater. Manuf. Innov.* 2 (1) (2013) 54–80.
- [15] S.R. Niezgoda, Y.C. Yabansu, S.R. Kalidindi, Understanding and visualizing microstructure and microstructure variance as a stochastic process, *Acta Mater.* 59 (16) (2011) 6387–6400.
- [16] Y.C. Yabansu, P. Steinmetz, J. Hötzer, S.R. Kalidindi, B. Nestler, Extraction of reduced-order process-structure linkages from phase-field simulations, *Acta Mater.* 124 (2017) 182–194.
- [17] P.-E. Øren, S. Bakke, Reconstruction of Berea sandstone and pore-scale modelling of wettability effects, *J. Pet. Sci. Eng.* 39 (3–4) (2003) 177–199.
- [18] D. Fullwood, S. Kalidindi, S. Niezgoda, A. Fast, N. Hampson, Gradient-based microstructure reconstructions from distributions using fast Fourier transforms, *Mater. Sci. Eng. A* 494 (1–2) (2008) 68–72.
- [19] A.P. Roberts, Statistical reconstruction of three-dimensional porous media from two-dimensional images, *Phys. Rev. E* 56 (3) (1997) 3203.
- [20] R. Bostanabad, A.T. Bui, W. Xie, D.W. Apley, W. Chen, Stochastic microstructure characterization and reconstruction via supervised learning, *Acta Mater.* 103 (2016) 89–102.
- [21] X. Liu, V. Shapiro, Random heterogeneous materials via texture synthesis, *Comput. Mater. Sci.* 99 (2015) 177–189.
- [22] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, *Adv. Neural Inf. Process. Syst.* 27 (2014).
- [23] D.P. Kingma, M. Welling, Auto-encoding variational bayes, 2013, arXiv preprint arXiv:1312.6114.
- [24] D.J. Rezende, S. Mohamed, D. Wierstra, Stochastic backpropagation and approximate inference in deep generative models, in: *International Conference on Machine Learning*, PMLR, 2014, pp. 1278–1286.
- [25] R. Cang, Y. Xu, S. Chen, Y. Liu, Y. Jiao, M. Yi Ren, Microstructure representation and reconstruction of heterogeneous materials via deep belief network for computational material design, *J. Mech. Des.* 139 (7) (2017) 071404.
- [26] Z. Yang, Y.C. Yabansu, R. Al-Bahrani, W.-k. Liao, A.N. Choudhary, S.R. Kalidindi, A. Agrawal, Deep learning approaches for mining structure-property linkages in high contrast composites from simulation datasets, *Comput. Mater. Sci.* 151 (2018) 278–287.
- [27] Z. Yang, Y.C. Yabansu, D. Jha, W.-k. Liao, A.N. Choudhary, S.R. Kalidindi, A. Agrawal, Establishing structure-property localization linkages for elastic deformation of three-dimensional high contrast composites using deep learning approaches, *Acta Mater.* 166 (2019) 335–345.
- [28] A. Cecen, H. Dai, Y.C. Yabansu, S.R. Kalidindi, L. Song, Material structure-property linkages using three-dimensional convolutional neural networks, *Acta Mater.* 146 (2018) 76–84.
- [29] N. Kouraytem, X. Li, W. Tan, B. Kappes, A.D. Spear, Modeling process-structure–property relationships in metal additive manufacturing: A review on physics-driven versus data-driven approaches, *J. Phys.: Mater.* 4 (3) (2021) 032002.
- [30] H. Wei, S. Zhao, Q. Rong, H. Bao, Predicting the effective thermal conductivities of composite materials and porous media by machine learning methods, *Int. J. Heat Mass Transfer* 127 (2018) 908–916.
- [31] N. Lubbers, T. Lookman, K. Barros, Inferring low-dimensional microstructure representations using convolutional neural networks, *Phys. Rev. E* 96 (5) (2017) 052111.
- [32] S. Kamrava, M. Sahimi, P. Tahmasebi, Simulating fluid flow in complex porous materials by integrating the governing equations with deep-layered machines, *Npj Comput. Mater.* 7 (1) (2021) 1–9.
- [33] A. Gayon-Lombardo, L. Mosser, N.P. Brandon, S.J. Cooper, Pores for thought: Generative adversarial networks for stochastic reconstruction of 3D multi-phase electrode microstructures with periodic boundaries, *Npj Comput. Mater.* 6 (1) (2020) 1–11.

- [34] D. Li, Review of structure representation and reconstruction on mesoscale and microscale, *Jom* 66 (3) (2014) 444–454.
- [35] R. Bostanabad, Y. Zhang, X. Li, T. Kearney, L.C. Brinson, D.W. Apley, W.K. Liu, W. Chen, Computational microstructure characterization and reconstruction: Review of the state-of-the-art techniques, *Prog. Mater. Sci.* 95 (2018) 1–41.
- [36] J. Hötzer, A. Reiter, H. Hierl, P. Steinmetz, M. Selzer, B. Nestler, The parallel multi-physics phase-field framework Pace3D, *J. Comput. Sci.* 26 (2018) 1–12.
- [37] H.S. Stein, D. Guevarra, P.F. Newhouse, E. Soedarmadji, J.M. Gregoire, Machine learning of optical properties of materials—predicting spectra from images and images from spectra, *Chem. Sci.* 10 (1) (2019) 47–55.
- [38] D. Bondarenko, P. Feldmann, Quantum autoencoders to denoise quantum data, *Phys. Rev. Lett.* 124 (13) (2020) 130502.
- [39] Y. Kim, H.K. Park, J. Jung, P. Aghari-Rad, S. Lee, J.Y. Kim, H.G. Jung, H.S. Kim, Exploration of optimal microstructure and mechanical properties in continuous microstructure space using a variational autoencoder, *Mater. Des.* 202 (2021) 109544.
- [40] L. Wang, Y.-C. Chan, F. Ahmed, Z. Liu, P. Zhu, W. Chen, Deep generative modeling for mechanistic-based learning and design of metamaterial systems, *Comput. Methods Appl. Mech. Engrg.* 372 (2020) 113377.
- [41] A. Tran, T. Wildey, Solving stochastic inverse problems for property–structure linkages using data-consistent inversion and machine learning, *JOM* 73 (1) (2021) 72–89.
- [42] M. Rixner, P.-S. Koutsourelakis, Self-supervised optimization of random material microstructures in the small-data regime, *Npj Comput. Mater.* 8 (1) (2022) 46.
- [43] S. Noguchi, J. Inoue, Stochastic characterization and reconstruction of material microstructures for establishment of process-structure-property linkage using the deep generative model, *Phys. Rev. E* 104 (2) (2021) 025302.
- [44] R. Cang, H. Li, H. Yao, Y. Jiao, Y. Ren, Improving direct physical properties prediction of heterogeneous materials from imaging data via convolutional neural network and a morphology-aware generative model, *Comput. Mater. Sci.* 150 (2018) 212–221.
- [45] Z. Pei, K.A. Rozman, Ö.N. Doğan, Y. Wen, N. Gao, E.A. Holm, J.A. Hawk, D.E. Alman, M.C. Gao, Machine-learning microstructure for inverse material design, *Adv. Sci.* 8 (23) (2021) 2101207.
- [46] T. Long, Y. Zhang, N.M. Fortunato, C. Shen, M. Dai, H. Zhang, Inverse design of crystal structures for multicomponent systems, *Acta Mater.* 231 (2022) 117898.
- [47] Z. Yang, X. Li, L. Catherine Brinson, A.N. Choudhary, W. Chen, A. Agrawal, Microstructural materials design via deep adversarial learning methodology, *J. Mech. Des.* 140 (11) (2018).
- [48] B. Shahriari, K. Swersky, Z. Wang, R.P. Adams, N. De Freitas, Taking the human out of the loop: A review of Bayesian optimization, *Proc. IEEE* 104 (1) (2015) 148–175.
- [49] K. Weiss, T.M. Khoshgoftaar, D. Wang, A survey of transfer learning, *J. Big Data* 3 (1) (2016) 1–40.
- [50] G.E. Fridley, C.A. Holstein, S.B. Oza, P. Yager, The evolution of nitrocellulose as a material for bioassays, *MRS Bull.* 38 (4) (2013) 326–330.
- [51] C. De Boor, C. De Boor, *A Practical Guide to Splines*, Vol. 27, Springer-Verlag, New York, 1978.
- [52] G. Martínez-Criado, J. Villanova, R. Tucoulou, D. Salomon, J.-P. Suuronen, S. Labouré, C. Guilloud, V. Valls, R. Barrett, E. Gagliardini, et al., ID16B: A hard X-ray nanoprobe beamline at the ESRF for nano-analysis, *J. Synchrotron Radiat.* 23 (1) (2016) 344–352, <http://dx.doi.org/10.1107/S1600577515019839>.
- [53] P. Altschuh, W. Kunz, M. Bremerich, A. Reiter, M. Selzer, B. Nestler, Wicking in porous polymeric membranes: Determination of an effective capillary radius to predict the flow behavior in lateral flow assays, *Membranes* 12 (7) (2022) 638.
- [54] P. Altschuh, *Skalenübergreifende Analyse makroporöser Membranen im Kontext digitaler Zwillinge* (Ph.D. thesis), Karlsruhe Institut für Technologie (KIT), 2020, <http://dx.doi.org/10.5445/IR/1000122904>.
- [55] S. Kullback, R.A. Leibler, On information and sufficiency, *Ann. Math. Stat.* 22 (1) (1951) 79–86.
- [56] A. Koepe, *Deep Learning in the Finite Element Method* (Ph.D. thesis), Dissertation, Rheinisch-Westfälische Technische Hochschule Aachen, 2021, 2021.
- [57] N. Brandt, L. Griem, C. Herrmann, E. Schoof, G. Tosato, Y. Zhao, P. Zschumme, M. Selzer, Kadi4Mat: A research data infrastructure for materials science, *Data Sci. J.* 20 (1) (2021).
- [58] J. Snoek, H. Larochelle, R.P. Adams, Practical bayesian optimization of machine learning algorithms, *Adv. Neural Inf. Process. Syst.* 25 (2012).
- [59] C.E. Rasmussen, *Gaussian processes in machine learning*, in: *Summer School on Machine Learning*, Springer, 2003, pp. 63–71.
- [60] C.K. Williams, C.E. Rasmussen, *Gaussian Processes for Machine Learning*, Vol. 2, MIT press, Cambridge, MA, 2006.
- [61] P.I. Frazier, *A tutorial on Bayesian optimization*, 2018, arXiv preprint arXiv:1807.02811.
- [62] H.J. Kushner, A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise, *J. Basic Eng.* 8 (1964).
- [63] J. Moćkus, On Bayesian methods for seeking the extremum, in: *Optimization Techniques IFIP Technical Conference*, Springer, 1975, pp. 400–404.
- [64] D.R. Jones, M. Schonlau, W.J. Welch, Efficient global optimization of expensive black-box functions, *J. Global Optim.* 13 (4) (1998) 455–492.
- [65] P. Auer, Using confidence bounds for exploitation-exploration trade-offs, *J. Mach. Learn. Res.* 3 (Nov) (2002) 397–422.
- [66] L. Chen, L. Zhang, Q. Kang, H.S. Viswanathan, J. Yao, W. Tao, Nanoscale simulation of shale transport properties using the lattice Boltzmann method: Permeability and diffusivity, *Sci. Rep.* 5 (1) (2015) 1–8.
- [67] N.R. Backeberg, F. Iacoviello, M. Rittner, T.M. Mitchell, A.P. Jones, R. Day, J. Wheeler, P.R. Shearing, P. Vermeesch, A. Striolo, Quantifying the anisotropy and tortuosity of permeable pathways in clay-rich mudstones using models based on X-ray tomography, *Sci. Rep.* 7 (1) (2017) 1–12.
- [68] K. Pearson, LIII. On lines and planes of closest fit to systems of points in space, *Lond Edinb Dublin Philos Mag J Sci* 2 (11) (1901) 559–572.
- [69] H. Abdi, L.J. Williams, *Principal component analysis*, Wiley Interdiscip. Rev. Comput. Stat. 2 (4) (2010) 433–459.
- [70] A. Tran, M. Eldred, T. Wildey, S. McCann, J. Sun, R.J. Visintainer, aphBO-2GP-3B: A budgeted asynchronous parallel multi-acquisition functions for constrained Bayesian optimization on high-performing computing architecture, *Struct. Multidiscip. Optim.* 65 (4) (2022) 132.
- [71] R. Couperthwaite, D. Khatamsaz, A. Molkeri, J. James, A. Srivastava, D. Allaire, R. Arróyave, The BAREFOOT optimization framework, *Integr. Mater. Manuf. Innov.* 10 (2021) 644–660.
- [72] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings*, 2010, pp. 249–256.
- [73] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.