



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

## **Interpretability versus performance of analytical and neural-network-based permeability prediction models: Exploring separability, monotonicity, and**

Downloaded from: <https://research.chalmers.se>, 2025-05-30 03:13 UTC

Citation for the original published paper (version of record):

Jansson, E., Röding, M. (2025). Interpretability versus performance of analytical and neural-network-based permeability prediction models: Exploring separability, monotonicity, and dimensional consistency. *Physical Review E*, 111(4). <http://dx.doi.org/10.1103/PhysRevE.111.045509>

N.B. When citing this work, cite the original published paper.

# Interpretability versus performance of analytical and neural-network-based permeability prediction models: Exploring separability, monotonicity, and dimensional consistency

Erik Jansson <sup>1,\*</sup> and Magnus Röding <sup>1,2,3</sup>

<sup>1</sup>Department of Mathematical Sciences, Chalmers University of Technology and University of Gothenburg, 41296 Göteborg, Sweden

<sup>2</sup>RISE Research Institutes of Sweden, Bioeconomy and Health, Agriculture and Food, 41276 Göteborg, Sweden

<sup>3</sup>Sustainable Innovation and Transformational Excellence, Pharmaceutical Technology and Development, Operations, AstraZeneca Gothenburg, 43183 Mölndal, Sweden



(Received 26 September 2024; revised 6 February 2025; accepted 18 March 2025; published 25 April 2025)

Effective mass transport properties of porous materials, such as permeability, are heavily influenced by their three-dimensional microstructure. There are numerous models developed for the prediction of permeability from microstructural characteristics, ranging from straightforward analytical relationships to high-performing machine learning models based on neural networks. There is an inherent tradeoff between predictive performance and interpretability; analytical models do not provide the best predictive performance but are relatively simple to understand. Neural networks, on the other hand, provide better predictive performance but are harder to interpret. In this paper, we investigate a multitude of models on the performance-versus-interpretability spectrum. Specifically, we use a dataset of 90 000 microstructures developed elsewhere and consider the prediction of permeability using the microstructural descriptors porosity, specific surface area, and geodesic tortuosity. At the respective ends of the spectrum, we study analytical, power-law-type models and fully connected neural networks. In between, we study neural networks that are either separable, monotonic, or both separable and monotonic. Establishing monotonic relationships is particularly interesting considering the potential for solving the inverse microstructure design problem using gradient-based methods. In addition, we study versions of these models that are consistent and inconsistent in terms of physical dimension.

DOI: [10.1103/PhysRevE.111.045509](https://doi.org/10.1103/PhysRevE.111.045509)

## I. INTRODUCTION

Understanding how the three-dimensional (3D) microstructural geometry influences the effective properties of a material is crucial for property prediction and ultimately for materials design and optimization [1]. For the case of mass transport properties in random porous materials, there are numerous analytical structure-property relationships derived from physical and geometrical assumptions. One of the most classical examples is the well-known Kozeny-Carman equation for permeability [2,3]. The space of possible 3D porous microstructures is enormous and beyond the reach of both being accurately described by the simplest physical models and being explored experimentally. However, for several years, it has been computationally feasible to explore the microstructural space *in silico*: by no means exhaustively, but extensively to the point where general structure-property relationships can be pursued using a more data-driven type of approach.

Computational exploration of the microstructural space requires mathematical models for virtual microstructures, based on, e.g., mathematical morphology and stochastic geometry [4,5]. There are plenty of mathematical models for real microstructures, including for lithium-ion batteries [6–8], solid oxide fuel cells [9,10], gas diffusion electrodes [11], fiber structures [12,13], polymeric coatings [14], mesoporous silica [15], open-cell foams [16], and granular structures [17,18]. In addition to "conventional" mathematical methods, generative AI approaches have also been used to generate microstructures lately [19,20].

There are numerous microstructural descriptors that have proven useful for mass transport property prediction. The most straightforward ones are porosity and specific surface area, but there are others such as pore size distributions, geodesic tortuosity [14,21], constrictivity [22], and two-point and three-point correlation functions [23,24]. Simple microstructural descriptors have been used to establish physics-based relationships for effective mass transport properties [25]. As a natural, more data-driven extension to the physics-based models, different nonlinear regression and machine learning approaches including random forests, artificial neural networks (ANNs), and convolutional neural networks (CNNs) have been trained on large datasets [26–31]. Of particular interest for this paper is Prifling *et al.* [32], where a dataset of 90 000 microstructures was used to develop prediction models for both permeability and effective diffusivity using conventional nonlinear regression, ANNs, and CNNs.

\*Contact author: erikjans@chalmers.se

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI. Funded by Bibsam.

In all these and other approaches, there is an inherent trade-off between predictive performance and interpretability. This tradeoff applies to prediction models as such; further, it also applies to microstructural descriptors, where some have more intuitive meaning than others.

A disadvantage of neural networks is they can produce physically implausible predictions. For example, the output can be a nonmonotonic function of the input(s). In contrast, it is very natural that mass transport properties depend on each microstructural descriptor in a monotonic (increasing or decreasing) fashion. There are numerous approaches to attaining monotonicity in neural networks, including architectures that are monotonic by construction (by, e.g., imposing constraints on the weights) or monotonic by regularization (by, e.g., modifying the loss function) [33–43]. On top of providing a more physically plausible prediction model, monotonicity can make the model generalize better (including extrapolation) and may require less data. Further, if the model is used as a surrogate model, i.e., a computationally efficient replacement for the actual simulation, in combination with gradient-based methods to solve inverse microstructure design problems, violations of monotonicity may become problematic.

In this paper, we investigate a multitude of models on the performance versus interpretability spectrum. At the respective ends of the spectrum, we study analytical, power-law type models and fully connected neural networks. In between, we study different flavors of these models that are constrained in one of several ways: they are either monotonic, separable, or both monotonic and separable, and further they are dimensionally consistent or inconsistent. Monotonicity (i.e., if the prediction is increasing or decreasing as a function of each individual microstructural descriptor) is attained by construction, following the approach of Runje and Shankaranarayana [43]. Separability (i.e., the prediction is expressed as a product of functions, each depending on only one microstructural descriptor) is also attained by construction, explicitly representing the different parts as individual functions learned by neural networks. Separability does not necessarily make the model more physically plausible, but is, on the other hand, more interpretable. In addition, we study versions of the different models that are dimensionally consistent and inconsistent (i.e., the physical units of the predicted quantity and the prediction models match or do not match). We use parts of the dataset created by Prifling *et al.* [32] and investigate the special case of predicting permeability using the microstructural descriptors porosity, specific surface area, and geodesic tortuosity. We emphasize that this is merely an example, and the principle is valid for other sets of microstructural descriptors, and for other properties, mass transport or otherwise. All investigated models provide good to excellent prediction results, but there are indeed tradeoffs: the more constrained the model, the less expressive it is, and this has both advantages and disadvantages.

## II. RESULTS AND DISCUSSION

### A. Dataset

We use a dataset created by Prifling *et al.* [32], consisting of all the microstructures and simulated permeabilities as well

as a subset of the microstructural descriptors. We refer to that earlier work for more details, but brief descriptions are provided below.

### 1. Microstructures

The set of microstructures is generated from nine different stochastic geometry models. For each model class, 10 000 microstructures with size  $192^3$  voxels are generated. The porosity is approximately uniformly distributed in  $[0.3, 0.95]$ , and the length scales are approximately equally distributed in all model classes. The microstructures are periodic in all directions. The microstructure models are denoted as (i) fiber systems, (ii) channel systems, (iii) spatial stochastic graphs, (iv) Gaussian random fields, (v) spinodal decompositions, (vi) hard ellipsoids, (vii) smoothed hard ellipsoids, (viii) soft ellipsoids, and (ix) smoothed soft ellipsoids. Examples from the microstructure models are provided in Fig. 1.

### 2. Microstructural descriptors

Three microstructural descriptors are used in this paper: porosity, specific surface area, and geodesic tortuosity. First, the porosity  $\varepsilon$  ( $0 \leq \varepsilon \leq 1$ ) is the volume fraction of the pore space, and hence dimensionless. Second, the specific surface area  $s$  ( $s \geq 0$ ) is the (interfacial) surface area (between the pore space and the solid) per unit volume, with physical dimension  $\text{length}^{-1}$  (here, in units of  $\text{voxel}^{-1}$ ). It can be approximated by estimating local contributions in all  $2 \times 2 \times 2$  voxel configurations, computed according to Schladitz *et al.* [44]. Third, the geodesic tortuosity  $\tau$  ( $\tau \geq 1$ ), is a dimensionless measure of the average length of paths through the pore space relative to the size of the microstructure in the transport direction, computed according to Neumann *et al.* [45]. Even though the structures are statistically isotropic, there are random variations between the directions which are to some extent accounted for by including the geodesic tortuosity. The periodicity of the microstructures is taken into account in the computations. It should be noted that geodesic tortuosity for a real microstructure cannot be computed without access to complete 3D geometric information, making the descriptor dependent on high-quality imaging and image segmentation for real-world applications. The porosity is on average computed in  $<1$  ms, the specific surface area is on average computed in approximately 0.3 s, and the tortuosity is on average computed in approximately 2 s.

### 3. Permeability

The (fluid) permeability is computed through solving the Navier-Stokes equations for pressure-driven flow for the steady state. The fluid flow is simulated using the lattice Boltzmann method with no-slip, bounce-back boundary conditions [46,47]. The flow is driven by a constant pressure difference across the structure in the transport direction [48]. After convergence to the steady state, the permeability  $\kappa$  (with dimension  $\text{length}^2$ ) is extracted from Darcy's law:

$$\bar{u} = -\frac{\kappa \Delta p}{\mu d}, \quad (1)$$

where  $\bar{u}$  is the average velocity,  $\Delta p$  is the applied pressure difference across the structure,  $\mu$  is the dynamic viscosity of

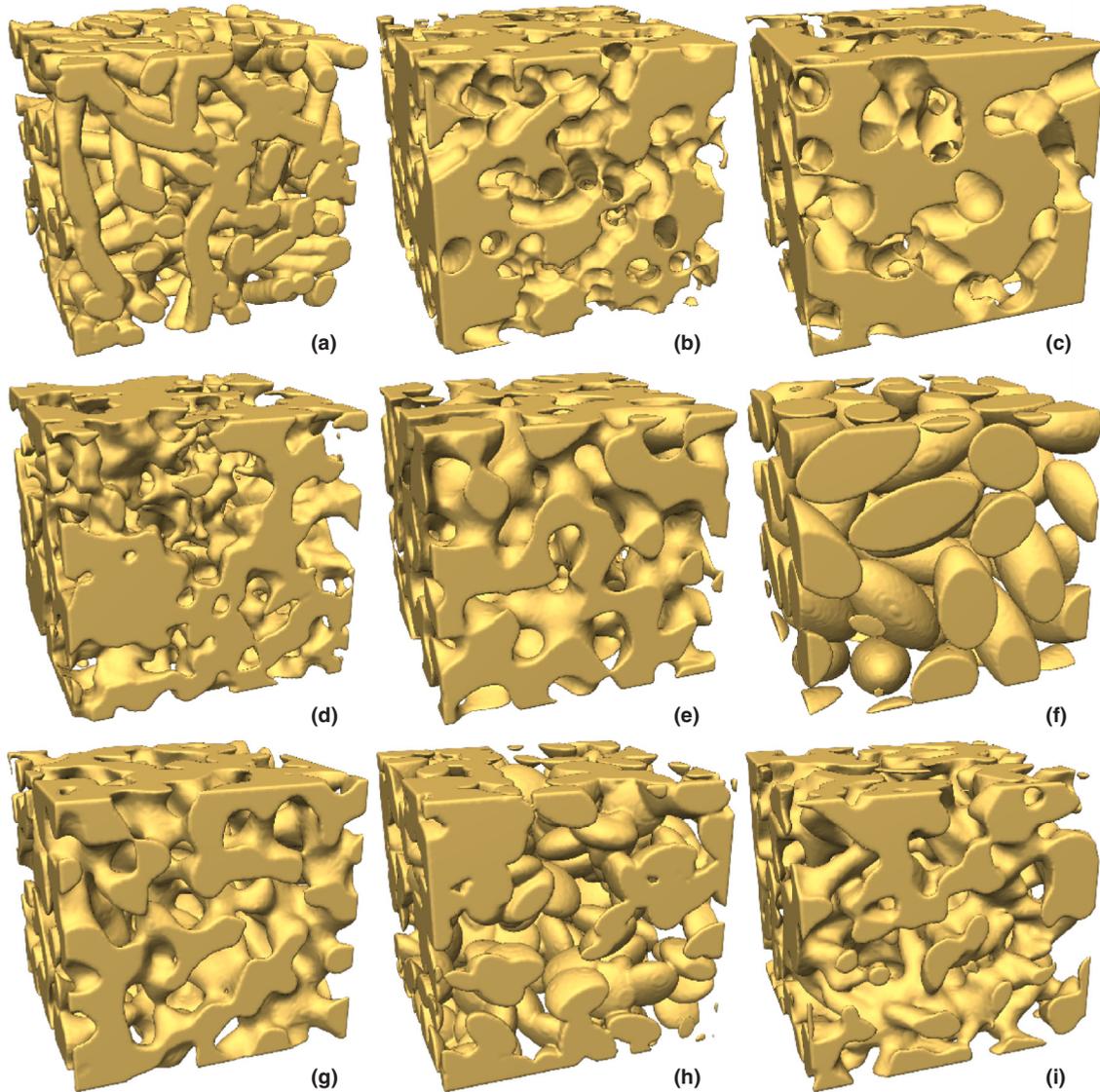


FIG. 1. Microstructures from the nine model classes, showing examples of (a) fiber systems, (b) channel systems, (c) spatial stochastic graphs, (d) Gaussian random fields, (e) spinodal decompositions, (f) hard ellipsoids, (g) smoothed hard ellipsoids, (h) soft ellipsoids, and (i) smoothed soft ellipsoids.

the fluid, and  $d$  is the size of the microstructure in the transport direction. Given that the Reynolds number is sufficiently small ( $Re < 0.01$ ), the permeability  $\kappa$  is independent of both the fluid (viscosity) and the pressure difference. Therefore, it is determined only by the microstructure. The permeability is on average computed in approximately 100 s.

**B. Prediction models**

*1. Analytical models*

Our point of departure is a particular Kozeny-Carman type of equation, where the permeability is modeled as a function of porosity  $\varepsilon$ , specific surface area  $s$ , and tortuosity  $\tau$ :

$$\kappa = c\varepsilon^{c_\varepsilon} s^{c_s} \tau^{c_\tau}. \tag{2}$$

Note that this model is separable because it is the product of functions that depend on only  $\varepsilon$ , only  $s$ , and only  $\tau$ . A physically plausible constraint is that the permeability should

increase monotonically with increasing  $\varepsilon$  (more space for the fluid) and decrease monotonically with increasing  $s$  (more frictional forces between the fluid and the surface) or increasing  $\tau$  (longer transport paths). This will be satisfied if  $c_\varepsilon > 0$ ,  $c_s < 0$ , and  $c_\tau < 0$  (and  $c > 0$ ). Further, the model is dimensionally consistent if and only if  $c_s = -2$  (because the dimension of  $\kappa$  is  $\text{length}^2$  and the dimension of  $s$  is  $\text{length}^{-1}$ ).

We consider both the dimensionally consistent and inconsistent cases of the model, as summarized in Table I.

*2. Neural networks*

A conventional, fully connected (dense) ANN is an arbitrarily complex nonlinear mapping implemented as a composition of linear and nonlinear operations. The building blocks are fully connected (dense) layers consisting of some nodes. In each node, a weighted sum of the outputs from

TABLE I. Summary of the two analytical prediction models, covering dimensionally consistent and inconsistent cases.

Model	Type	Separable	Monotone	Dimensionally consistent	Equation
1	Analytical	Yes	Yes	No	$\kappa = c\varepsilon^{c\varepsilon} s^{c\varepsilon} \tau^{c\varepsilon}$
2	Analytical	Yes	Yes	Yes	$\kappa = c\varepsilon^{c\varepsilon} s^{-2} \tau^{c\varepsilon}$

the nodes in the previous layer is computed, and a nonlinear activation function  $\sigma$  is applied to produce an output. As a model for permeability, the ANN produces a mapping  $f$ ,

$$\kappa = f(\varepsilon, s, \tau), \quad (3)$$

that depends on, e.g., the number of layers, the number of nodes in each layer, the activation function  $\sigma$ , the training procedure, and the data. In this setting, we use the exponential linear unit (ELU) activation function [49]:

$$\sigma(x) = \begin{cases} e^x - 1, & x < 0 \\ x, & x \geq 0. \end{cases} \quad (4)$$

Generally, a fully connected ANN is neither monotonic nor separable with respect to any of the inputs.

We also consider ANNs that are, by construction, either monotonic, separable, or both, inspired by Eq. (2). First, we consider the setting

$$\kappa = f_\varepsilon(\varepsilon)f_s(s)f_\tau(\tau), \quad (5)$$

The network architectures are illustrated in Fig. 2. Where  $f_\varepsilon$ ,  $f_s$ , and  $f_\tau$  are separate fully connected ANNs. This construction provides separability by definition. Second, we use the approach of Runje and Shankaranarayana [43] as described below to obtain monotonicity by construction. A simple method to obtain monotonicity is to constrain the weights to be only non-negative or nonpositive, combined with a monotonic activation function [33]. However, such ANNs can, on the one hand, be difficult to train with bounded activation functions and, on the other hand, only represent convex mappings if the activation function is convex, such as rectified linear unit (ReLU) or ELU [41]. The solution in Runje and Shankaranarayana [43] is to constrain the sign of the weights and use a set of three monotonic activation functions derived from one original activation function  $\sigma_1$  by defining

$$\sigma_2(x) = -\sigma_1(-x) \quad (6)$$

and

$$\sigma_3(x) = \begin{cases} \sigma_1(x+1) - \sigma_1(1), & x < 0 \\ \sigma_2(x-1) + \sigma_1(1), & x \geq 0. \end{cases} \quad (7)$$

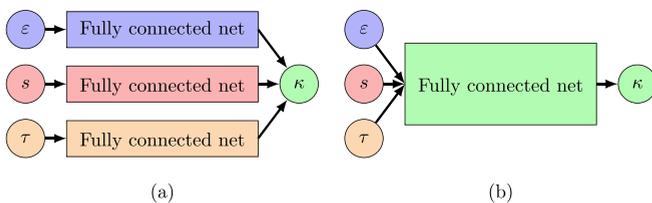


FIG. 2. Schematic illustration of the two network topologies used in this paper, either the separated topology (a) or the nonseparable, standard, ANN topology (b).

If  $\sigma_1$  is convex and lower bounded,  $\sigma_2$  is concave and upper bounded, and  $\sigma_3$  is both lower and upper bounded (and sigmoidal, i.e., neither convex nor concave). Further, if  $\sigma_1$  is monotonic, the same holds for  $\sigma_2$  and  $\sigma_3$ . As our  $\sigma_1$ , we use the ELU activation function in Eq. (4). Then, the other activation functions become

$$\sigma_2(x) = \begin{cases} x, & x < 0 \\ 1 - e^{-x}, & x \geq 0 \end{cases} \quad (8)$$

and

$$\sigma_3(x) = \begin{cases} e^{x+1} - 2, & x < -1 \\ x, & -1 \leq x < 1 \\ 2 - e^{-x+1}, & x \geq 1 \end{cases} \quad (9)$$

The activation functions are illustrated in Fig. 3. By splitting the nodes of each layer into three subsets, with each subset using one of the three activation functions, the ANN can represent functions that are both locally convex and locally concave. After constraining the weights appropriately, the desired monotonicity is obtained. The monotonic counterparts to the mappings above are denoted  $f^M(\varepsilon, s, \tau)$ ,  $f_\varepsilon^M(\varepsilon)$ ,  $f_s^M(s)$ , and  $f_\tau^M(\tau)$  (monotonically increasing for  $\varepsilon$ , and monotonically decreasing for  $s$  and  $\tau$ ).

It should be noted that the approach of Runje and Shankaranarayana [43] puts some restrictions on the original activation function  $\sigma_1$ . First, it should be lower or upper bounded, but not lower *and* upper bounded, which excludes sigmoidal functions such as logistic and tanh activations. Second, it should be monotonic, which excludes Swish and Gaussian error linear unit activations. Third, we additionally choose to use the smooth ELU activation over, e.g., ReLU because we are modeling a differentiable and smooth relationship.

Further, we consider dimensionally consistent models, meaning that we remove the contribution of the specific

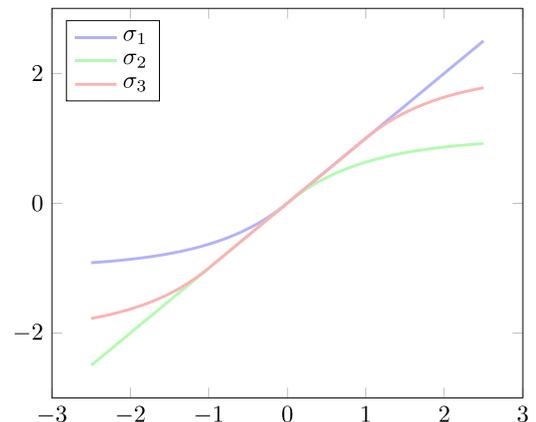


FIG. 3. Illustration of the activation functions used to ensure monotonicity.

TABLE II. Summary of the eight neural network models, covering monotonic, separable, and both monotonic and separable neural networks, in both dimensionally consistent and inconsistent versions.

Model	Type	Separable	Monotone	Dimensionally consistent	Equation
3	ANN	No	No	No	$\kappa = f(\varepsilon, s, \tau)$
4	ANN	No	Yes	No	$\kappa = f^M(\varepsilon, s, \tau)$
5	ANN	Yes	No	No	$\kappa = f_\varepsilon(\varepsilon)f_s(s)f_\tau(\tau)$
6	ANN	Yes	Yes	No	$\kappa = f_\varepsilon^M(\varepsilon)f_s^M(s)f_\tau^M(\tau)$
7	ANN	No	No	Yes	$\kappa = f(\varepsilon, \tau)s^{-2}$
8	ANN	No	Yes	Yes	$\kappa = f^M(\varepsilon, \tau)s^{-2}$
9	ANN	Yes	No	Yes	$\kappa = f_\varepsilon(\varepsilon)f_\tau(\tau)s^{-2}$
10	ANN	Yes	Yes	Yes	$\kappa = f_\varepsilon^M(\varepsilon)f_\tau^M(\tau)s^{-2}$

surface area  $s$  from the ANNs and predict  $\log(\kappa s^2)$  by ANNs that use only  $\varepsilon$  and  $\tau$  as inputs. Here, and in the following,  $\log$  indicates the natural logarithm. The specific surface area  $s$  is moved back to the right-hand side after training.

We consider monotonic, separable, and both monotonic and separable neural networks, in both the dimensionally consistent and inconsistent cases, as summarized in Table II.

### 3. Fitting and training

The data are split into training (70%, 63 000), validation (15%, 13 500), and test (15%, 13 500) data to include an equal number of samples from each model. The dataset sizes and the split are identical to Prifling *et al.* [32].

All models are optimized with respect to mean squared error (MSE) loss in logarithmic scale:

$$\text{MSE} = \frac{1}{N} \sum_{n=1}^N (\log \hat{\kappa}_n - \log \kappa_n)^2, \quad (10)$$

where  $\kappa_n$  are the target values and  $\hat{\kappa}_n$  are the predicted values. The rationale for using a logarithmic scale loss in this manner is that it penalizes prediction errors independent of scale; specifically, if the prediction error is expressed as a multiplicative factor,  $\hat{\kappa} = \alpha \kappa$ , then

$$\log \hat{\kappa} - \log \kappa = \log(\alpha \kappa) - \log \kappa = \log \alpha \quad (11)$$

which is independent of  $\kappa$ . Hence, the loss function penalizes relative (multiplicative) errors, which is crucial because the  $\kappa$  values cover several orders of magnitude. It should be noted that performing predictions in logarithmic scale and transforming the outputs back to linear scale prevents nonphysical negative predictions of  $\kappa$ . In the case of the ANNs, the original functions that are learned can be understood as composite mappings involving both scale transformations and predictions. In contrast, we use the more intuitive mean absolute percentage error (MAPE) loss in linear scale,

$$\text{MAPE} = 100 \times \frac{1}{N} \sum_{n=1}^N \left| \frac{\hat{\kappa}_n - \kappa_n}{\kappa_n} \right| \%, \quad (12)$$

for final assessment of prediction performance.

For the analytical equations, we use standard nonlinear least squares fitting. For example, to fit model 1, we minimize

$$\sum_{n=1}^N (\log c + c_\varepsilon \log \varepsilon_n + c_s \log s_n + c_\tau \log \tau_n - \log \kappa_n)^2, \quad (13)$$

evaluated over the training set. The fitting is implemented in MATLAB (Mathworks, MA, US).

In all network architectures (models 3–10), we use four hidden layers, each with 18 nodes. A hyperparameter search on the dimensionally inconsistent monotone separated model led to each of the three subnetworks having four layers, each with six nodes. Thus, the total number of nodes in each layer in the total network is 18. To facilitate comparison, the same number of nodes and layers was chosen for all other models. Separability and nonseparability thus only influence the weight sparsity of the network, and not the total number of nodes.

The input data are normalized to improve training performance [50]. Xavier uniform initialization is used to initialize the weights [51]. The biases are initialized as zero. To train the model, stochastic gradient descent with momentum was used [52,53]. The batch size, learning rate scheme, and value of the momentum were chosen as in Prifling *et al.* [32], i.e., a batch size of 128, momentum of 0.9, and a learning rate scheme with a stepwise increasing and then stepwise decreasing learning rate.

All ANN models except model 4 and 8 were implemented in PYTORCH 2.0.1 [54]. Model 4 and 8 were implemented in TENSORFLOW 2.11 [55] using code from Runje and Shankaranarayana [43].

### 4. Prediction results

Prediction results of all models are shown in Table III, including MSE and MAPE for the training, validation, and test sets. Note that the usual trend of the test error being larger than the validation error being larger than training error does not always hold; for the analytical models, this might be taken as an indication of underfitting. We use the test data MAPE as the basis for evaluating performance, which is why these values are bold in Table III. First, we note the best model is the full and dimensionally inconsistent ANN and that the analytical models are, unsurprisingly, outperformed by all ANN models.

In Figs. 4–6 the predicted  $\log \kappa$  are plotted against the true data. It is clear in Fig. 4 that the analytical model fails to accurately predict  $\kappa$ , indicated by the bent shape of the point cloud. Further, for all ANN models except model 3, the scatter plots in Figs. 5 and 6 all have clear structures for low values of  $\kappa$ . This indicates that all models except model 3, the full ANN, fail to capture some dependence between the microstructural descriptors and  $\kappa$ .

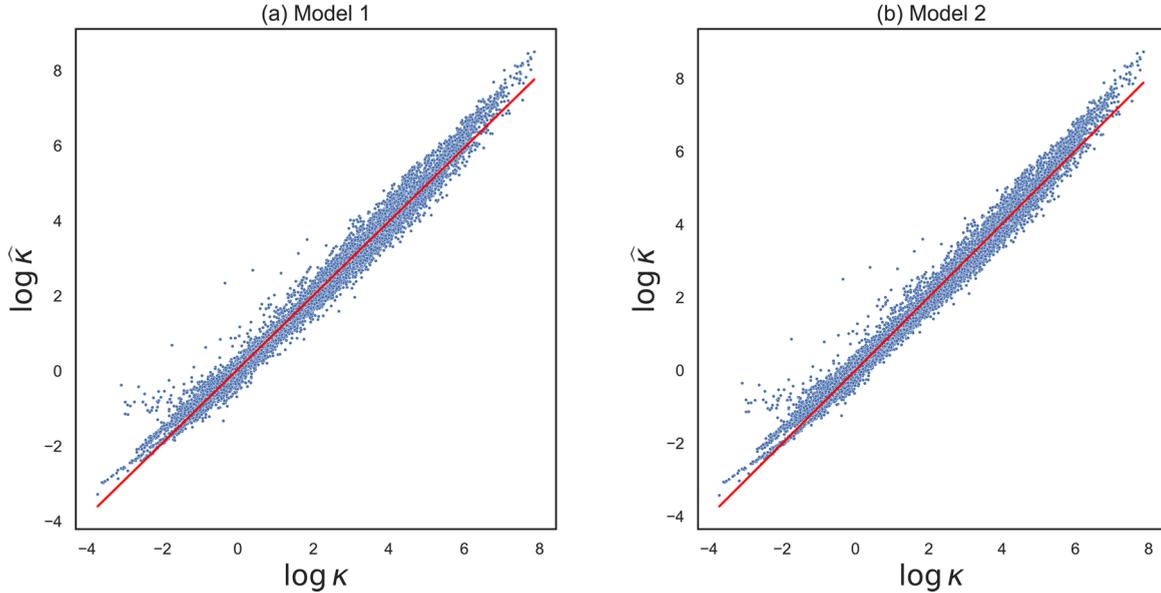


FIG. 4. Prediction of permeability of the analytical models (models 1 and 2) on the test data set.

Regarding interpretability, there are several aspects to consider. First, we note that the dimensionally consistent models always give larger losses than their dimensionally inconsistent counterparts. This is not a surprise given that the degrees of freedom are reduced by dimensionality constraints, making the ANN less expressive.

Secondly, enforcing monotonicity, but not separability (e.g., model 4), does not seem to lead to a considerable decrease in performance. Using autodifferentiation to compute

TABLE III. Results for all prediction models, including MSE and MAPE for the training, validation, and test sets. Note that MSE is evaluated in logarithmic scale and MAPE in linear scale. The bold values indicate the test data MAPE, which are used as the basis for evaluating performance.

Model	Loss	Training	Validation	Test
1	MSE	0.0635	0.0667	0.0653
	MAPE (%)	25.1214	24.1486	<b>20.1083</b>
2	MSE	0.0679	0.0718	0.0702
	MAPE (%)	27.3467	26.3429	<b>21.3367</b>
3	MSE	0.0248	0.0322	0.0324
	MAPE (%)	10.6111	14.7955	<b>11.3386</b>
4	MSE	0.0327	0.0328	0.0341
	MAPE (%)	14.6172	16.8317	<b>13.2692</b>
5	MSE	0.03373	0.03687	0.03512
	MAPE (%)	18.7880	18.1096	<b>13.5888</b>
6	MSE	0.0349	0.0379	0.0362
	MAPE (%)	18.9864	13.4779	<b>13.8764</b>
7	MSE	0.0310	0.0404	0.0365
	MAPE (%)	12.4945	18.3617	<b>13.1494</b>
8	MSE	0.03551	0.0388	0.0371
	MAPE (%)	19.0884	18.8436	<b>14.0196</b>
9	MSE	0.0339	0.0372	0.0355
	MAPE (%)	19.1678	18.2421	<b>13.6384</b>
10	MSE	0.0361	0.0393	0.0374
	MAPE (%)	19.9276	18.8896	<b>14.1969</b>

partial derivatives, we confirm that the monotone networks indeed are monotone. On the other hand, the nonmonotone networks are not fully monotone. They learn, however, to behave essentially as monotone functions, but with small amplitude oscillations. Therefore, the practical purpose of introducing monotonicity would be to have monotonicity guarantees in our setting rather than substantially changing the behavior.

Finally, in the dimensionally inconsistent case, introducing sparsity is what causes the most substantial decrease in performance. It is not inconceivable that this is due to the sparse network not being able to capture complex and higher order dependencies between the input microstructural descriptors and the output. The execution time for a single permeability prediction varies from model to model but is  $\ll 1$  ms.

### 5. Comment on generalizability

We emphasize that the three microstructural descriptors and the single mass transport property studied herein are only examples of a general principle. Also, even for the rather low-dimensional problem at hand, we have not covered every possible partially separable model. Specifically, comparing the models  $\kappa = f(\varepsilon, s)f(\tau)$ ,  $\kappa = f(\varepsilon, \tau)f(s)$ , and  $\kappa = f(\varepsilon)f(s, \tau)$  could provide insights into dependencies between different microstructural descriptors. Of course it would be possible to consider a larger number of microstructural descriptors, but doing so would lead to a combinatorial explosion in the number of models to be considered. Given that porosity, specific surface area, and (geodesic) tortuosity are probably the three most common descriptors, we have explored separability, monotonicity, and dimensional consistency using those.

It is also possible to accommodate other functional forms. For example, considering Eq. (13) in this work, and Eq. (16) in Prifling *et al.* [32], a prediction involving a fourth descriptor  $\beta$  called constrictivity (a measure of the "tightness" of the pore space; we refer to Prifling *et al.* [32] for details) can be done

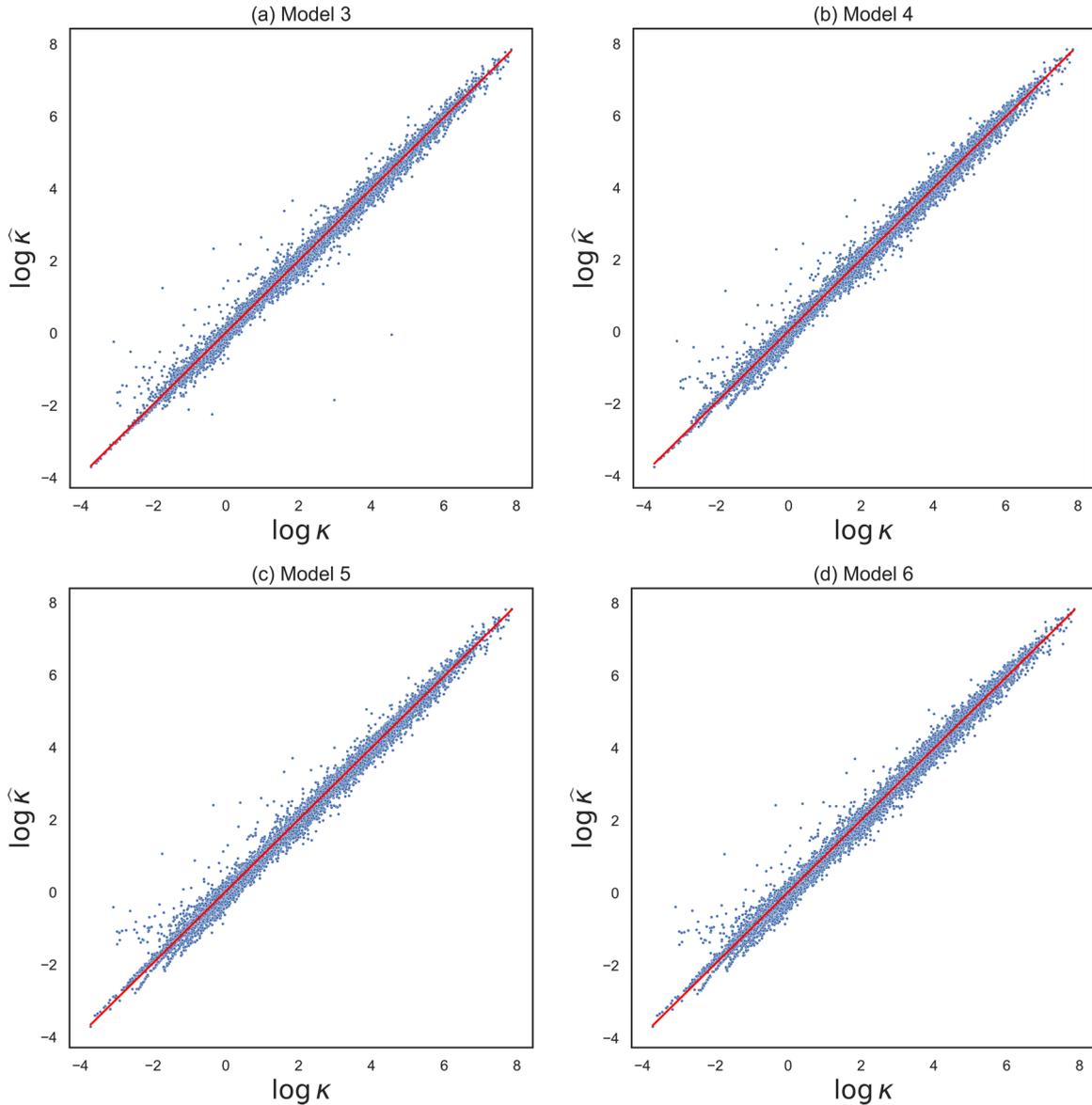


FIG. 5. Prediction of permeability of the dimensionally inconsistent models (models 3 to 6) on the test data set.

in more than one way, either including  $\beta$  as a separate factor or as an exponent for  $\varepsilon$ , i.e.,

$$\kappa = f_\varepsilon(\varepsilon)f_\beta(\beta)f_\tau(\tau)f_s(s) \quad (14)$$

or

$$\kappa = f_\varepsilon(\varepsilon)^{f_\beta(\beta)}f_\tau(\tau)f_s(s). \quad (15)$$

Finally, relationships for predicting other properties (mass transport and others), such as effective diffusivity covered in Prifling *et al.* [32], can be studied using the same methodologies.

### III. CONCLUSION

In this paper, we investigate a range of analytical and neural-network-based models for predicting permeability in porous materials using the microstructural descriptors porosity, specific surface area, and geodesic tortuosity. These models cover a spectrum in terms of the tradeoff between, on the

one hand, performance and, on the other hand, interpretability and physical plausibility. This tradeoff is explored along three "axes" by constraining the models in different ways by imposing monotonicity, separability, and dimensional consistency. Compared to simulating permeability using the lattice Boltzmann method, the entire workflow of computing descriptors and predicting permeability using the analytical or neural network models provides a speedup in the order of a factor of 50. Whether this is a meaningful improvement depends on the purpose. It is worth pointing out that the primary goal of this paper is not to improve prediction speed but rather to investigate prediction performance for different models.

Not surprisingly, all neural networks perform better than the analytical models. Furthermore, the fully connected and unconstrained neural network gives the highest predictive performance.

Imposing separability is performed by representing the output as the product of fully connected subnetworks, and this has a major impact on performance. This suggests that

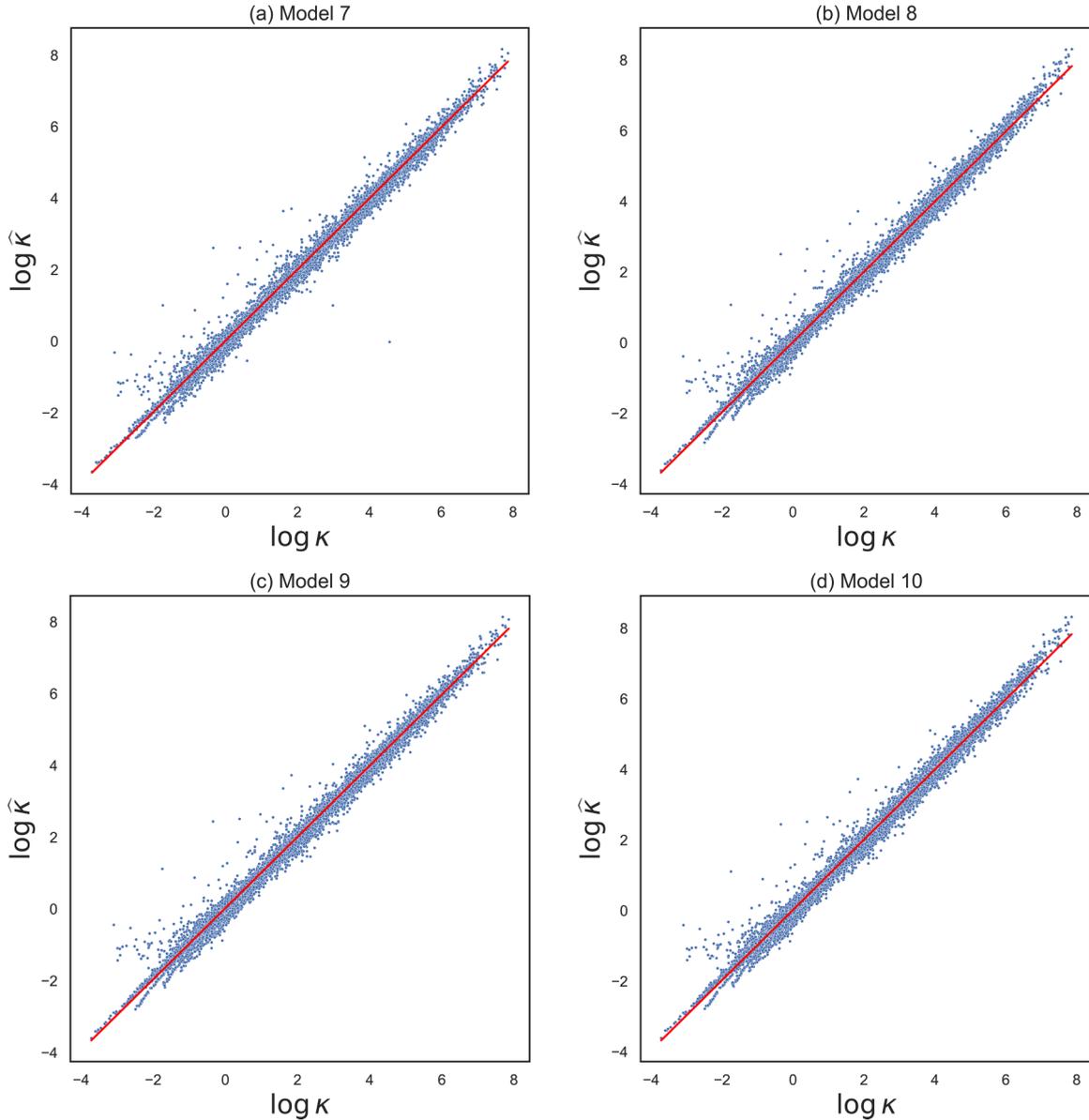


FIG. 6. Prediction of permeability of the dimensionally consistent models (models 7 to 10) on the test data set.

separability imposes a strong prior on the structure of the neural network. This can be understood as forcing numerous weights of the corresponding fully connected network to be zero, reducing the number of degrees of freedom. Further, that separability has a major impact on predictive performance indicates that "classical" analytical structure-property relationships are too simplified; indeed, the "real" relationship is likely nonseparable, with a complex dependence structure between the microstructural descriptors.

Imposing dimensional consistency in this case is the same as imposing a partial separability with respect to one of the descriptors (specific surface area). The notable impact of imposing separability hence explains the impact of imposing dimensional consistency as well. Indeed, the relative change in performance of imposing dimensional consistency is largest for the fully connected monotonic and nonmonotonic neural networks, which to begin with are the most flexible and expressive models with no separability constraints.

Imposing monotonicity has, by comparison, a minor impact on performance. This does not suggest that monotonicity is irrelevant. In contrast, it does generally impose a strong prior on a neural network. Further, if the model is used as a surrogate model in combination with gradient-based methods to solve inverse microstructure design problems, violations of monotonicity may become problematic. In this case, however, the data strongly support monotonicity anyway, so the impact becomes smaller. Most likely, monotonicity would have a larger impact on performance for a much smaller dataset, but would on the other hand also reduce the need for data.

In conclusion, the results provide insight into some possible tradeoffs in terms of prediction performance versus interpretability, and the intended use of the model (i.e., straightforward prediction model, differentiable surrogate model, or study of the information content of the descriptors and the ability of the prediction model to store that

information) ultimately has to determine the suitable choice of tradeoff.

Interesting future work would include investigating all cases of partial separability, i.e., all combinations of separable and nonseparable parts of the neural networks, and investigating robustness, generalization performance, and dataset size requirements for the different models by training on much smaller datasets. Because of the increased variability for much smaller datasets, however, that study would likely have to be quite extensive and performed for numerous random subsets of the data and for different random seeds of the neural networks.

## ACKNOWLEDGMENTS

M.R. acknowledges the financial support of the Swedish Research Council for Sustainable Development (Grant No. 2019-01295) and the Swedish Research Council (Grant No. 2023-04248). The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council (Grant No. 2022-06725).

E.J. declares no conflict of interest. M.R. is an employee of AstraZeneca and has stock ownership and/or stock options or interests in the company.

- 
- [1] S. Torquato, *Random Heterogeneous Materials: Microstructure and Macroscopic Properties* (Springer, New York, 2002).
- [2] J. Kozeny, *Sitzungsber. Akad. Wiss., Wien* **136**, 271 (1927).
- [3] P. C. Carman, *AIChE J.* **15**, 150 (1937).
- [4] S. N. Chiu, D. Stoyan, W. Kendall, and J. Mecke, *Stochastic Geometry and its Applications* (Wiley, New York, 2013).
- [5] D. Jeulin, *Morphological Models of Random Structures* (Springer, New York, 2021).
- [6] L. Petrich, O. Furat, M. Wang, C. E. Krill, III, and V. Schmidt, *Front. Mater.* **8**, 760602 (2021).
- [7] B. Prifling, M. Ademmer, F. Single, O. Benevolenski, A. Hilger, M. Osenberg, I. Manke, and V. Schmidt, *Comput. Mater. Sci.* **192**, 110354 (2021).
- [8] O. Furat, L. Petrich, D. Finegan, D. Diercks, F. Usseglio-Viretta, K. Smith, and V. Schmidt, *npj Comput. Mater.* **7**, 105 (2021).
- [9] M. Neumann, J. Staněk, O. M. Pecho, L. Holzer, V. Beneš, and V. Schmidt, *Comput. Mater. Sci.* **118**, 353 (2016).
- [10] H. Moussaoui, J. Laurencin, Y. Gavet, G. Delette, M. Hubert, P. Cloetens, T. L. Bihan, and J. Debayle, *Comput. Mater. Sci.* **143**, 262 (2018).
- [11] M. Neumann, M. Osenberg, A. Hilger, D. Franzen, T. Turek, I. Manke, and V. Schmidt, *Comput. Mater. Sci.* **156**, 325 (2019).
- [12] P. Townsend, E. Larsson, T. Karlson, S. A. Hall, M. Lundman, P. Bergström, C. Hanson, N. Lorén, T. Gebäck, A. Särkkä, and M. Röding, *Comput. Mater. Sci.* **194**, 110433 (2021).
- [13] M. Weber, A. Grießer, D. Mosbach, E. Glatt, A. Wiegmann, and V. Schmidt, *Sci. Rep.* **13**, 19359 (2023).
- [14] S. Barman, H. Rootzén, and D. Bolin, *AIChE J.* **65**, 446 (2019).
- [15] B. Prifling, M. Neumann, D. Hlushkou, K. Kübel, U. Tallarek, and V. Schmidt, *Comput. Mater. Sci.* **187**, 109934 (2021).
- [16] D. Westhoff, J. Skibinski, O. Šedivý, B. Wysocki, T. Wejrzanowski, and V. Schmidt, *Mater. Des.* **147**, 1 (2018).
- [17] J. Rao, A. Püttmann, S. Khirevich, U. Tallarek, C. Geuzaine, M. Behr, and E. von Lieres, *Comput. Chem. Eng.* **178**, 108355 (2023).
- [18] L. Riley, P. Cheng, and T. Segura, *Nat. Comput. Sci.* **3**, 975 (2023).
- [19] A. Gayon-Lombardo, L. Mosser, N. P. Brandon, and S. J. Cooper, *npj Comput. Mater.* **6**, 82 (2020).
- [20] L. Xu, N. Hoffman, Z. Wang, and H. Xu, *Mater. Des.* **223**, 111223 (2022).
- [21] L. Holzer, P. Marmet, M. Fingerle, A. Wiegmann, M. Neumann, and V. Schmidt, *Tortuosity and Microstructure Effects in Porous Media: Classical Theories, Empirical Data and Modern Methods* (Springer, New York, 2023).
- [22] M. Neumann, O. Stenzel, F. Willot, L. Holzer, and V. Schmidt, *Int. J. Solids Struct.* **184**, 211 (2020).
- [23] D. Hlushkou, H. Liasneuski, U. Tallarek, and S. Torquato, *J. Appl. Phys.* **118** (2015).
- [24] Z. Ma and S. Torquato, *Phys. Rev. E* **98**, 013307 (2018).
- [25] L. Masaro and X. Zhu, *Prog. Polym. Sci.* **24**, 731 (1999).
- [26] J. H. van der Linden, G. A. Narsilio, and A. Tordesillas, *Phys. Rev. E* **94**, 022904 (2016).
- [27] O. Stenzel, O. Pecho, L. Holzer, M. Neumann, and V. Schmidt, *AIChE J.* **63**, 4224 (2017).
- [28] M. Röding, Z. Ma, and S. Torquato, *Sci. Rep.* **10**, 15239 (2020).
- [29] K. Graczyk and M. Matyka, *Sci. Rep.* **10**, 21488 (2020).
- [30] H. Wang, Y. Yin, X. Hui, J. Bai, and Z. Qu, *Energy AI 2*, 100035 (2020).
- [31] S. Gärttner, F. Alpak, A. Meier, N. Ray, and F. Frank, *Comput. Geosci.* **27**, 245 (2023).
- [32] B. Prifling, M. Röding, P. Townsend, M. Neumann, and V. Schmidt, *Front. Mater.* **8**, 786502 (2021).
- [33] N. Archer and S. Wang, *Decis. Sci.* **24**, 60 (1993).
- [34] J. Sill and Y. Abu-Mostafa, *Adv. Neural Inf. Process. Syst.* **9**, 634 (1996).
- [35] J. Sill, *Adv. Neural Inf. Process. Syst.* **10**, 661 (1997).
- [36] H. Daniels and M. Velikova, *IEEE Trans. Neural Netw.* **21**, 906 (2010).
- [37] M. M. Fard, K. Canini, A. Cotter, J. Pfeifer, and M. Gupta, *Adv. Neural Inf. Process. Syst.* **29**, 2919 (2016).
- [38] S. You, D. Ding, K. Canini, J. Pfeifer, and M. Gupta, *Adv. Neural Inf. Process. Syst.* **30**, 2981 (2017).
- [39] A. Wehenkel and G. Louppe, *Adv. Neural Inf. Process. Syst.* **32**, 1545 (2019).
- [40] A. Gupta, N. Shukla, L. Marla, A. Kolbeinsson, and K. Yellepeddi, [arXiv:1909.10662](https://arxiv.org/abs/1909.10662).
- [41] X. Liu, X. Han, N. Zhang, and Q. Liu, *Adv. Neural Inf. Process. Syst.* **33**, 15427 (2020).
- [42] N. Nolte, O. Kitouni, and M. Williams, Expressive monotonic neural networks, [arXiv:2307.07512](https://arxiv.org/abs/2307.07512).
- [43] D. Runje and S. Shankaranarayana, Constrained monotonic neural networks, [arXiv:2205.11775](https://arxiv.org/abs/2205.11775).
- [44] K. Schladitz, J. Ohser, and W. Nagel, in *13th International Conference Discrete Geometry for Computer Imagery*, edited by

- A. Kuba, L. Nyúl, and K. Palágyi (Springer, New York, 2007), pp. 247–258.
- [45] M. Neumann, C. Hirsch, J. Staněk, V. Beneš, and V. Schmidt, *Scand. J. Stat.* **46**, 848 (2019).
- [46] T. Gebäck and A. Heintz, *Commun. Comput. Phys.* **15**, 487 (2014).
- [47] T. Gebäck, M. Marucci, C. Boissier, J. Arnehed, and A. Heintz, *J. Phys. Chem. B* **119**, 5220 (2015).
- [48] Q. Zou and X. He, *Phys. Fluids* **9**, 1591 (1997).
- [49] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, in *4th International Conference on Learning Representations, ICLR 2016*, edited by Y. Bengio and Y. LeCun (ICLR, San Juan, Puerto Rico, 2016).
- [50] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, in *Neural Networks: Tricks of the Trade*, edited by G. Montavon, G. B. Orr, and K.-R. Müller, 2nd ed. (Springer-Verlag, Berlin, 2012), pp. 9–48.
- [51] X. Glorot and Y. Bengio, in *JMLR W&CP: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (Proceedings of Machine Learning Research, Sardinia, Italy, 2010), Vol. 9, pp. 249–256.
- [52] N. Qian, *Neural Netw.* **12**, 145 (1999).
- [53] L. Bottou, in *Proceedings of COMPSTAT'2010* (Springer, New York, 2010), pp. 177–186.
- [54] J. Ansel, E. Yang, H. He, N. Gimelshein, A. Jain, M. Voznesensky, B. Bao, P. Bell, D. Berard, E. Burovski, G. Chauhan, A. Chourdia, W. Constable, A. Desmaison, Z. DeVito, E. Ellison, W. Feng, J. Gong, M. Gschwind, B. Hirsh *et al.*, in *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS'24* (Association for Computing Machinery, New York, NY, 2024), Vol. 2, pp. 929–947.
- [55] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg *et al.*, TensorFlow: Large-scale machine learning on heterogeneous systems (2015), software available from <https://www.tensorflow.org/>.