


Article

# Greedy Kernel Methods for Approximating Breakthrough Curves for Reactive Flow from 3D Porous Geometry Data

Robin Herkert <sup>1,\*</sup> , Patrick Buchfink <sup>2</sup>, Tizian Wenzel <sup>3</sup>, Bernard Haasdonk <sup>1</sup>, Pavel Toktaliev <sup>4</sup> and Oleg Iliev <sup>4</sup>

<sup>1</sup> Institute for Applied Analysis and Numerical Simulation, University of Stuttgart, 70569 Stuttgart, Germany; haasdonk@mathematik.uni-stuttgart.de

<sup>2</sup> Department of Applied Mathematics, University of Twente, 7522 NH Enschede, The Netherlands; p.buchfink@utwente.nl

<sup>3</sup> Department of Mathematics, Universität Hamburg, 20146 Hamburg, Germany; tizian.wenzel@uni-hamburg.de

<sup>4</sup> Fraunhofer ITWM, Technical University Kaiserslautern, 67663 Kaiserslautern, Germany; pavel.toktaliev@itwm.fraunhofer.de (P.T.); oleg.iliev@itwm.fraunhofer.de (O.I.)

\* Correspondence: robin.herkert@mathematik.uni-stuttgart.de

**Abstract:** We address the challenging application of 3D pore scale reactive flow under varying geometry parameters. The task is to predict time-dependent integral quantities, i.e., breakthrough curves, from the given geometries. As the 3D reactive flow simulation is highly complex and computationally expensive, we are interested in data-based surrogates that can give a rapid prediction of the target quantities of interest. This setting is an example of an application with scarce data, i.e., only having a few available data samples, while the input and output dimensions are high. In this scarce data setting, standard machine learning methods are likely to fail. Therefore, we resort to greedy kernel approximation schemes that have shown to be efficient meshless approximation techniques for multivariate functions. We demonstrate that such methods can efficiently be used in the high-dimensional input/output case under scarce data. Especially, we show that the vectorial kernel orthogonal greedy approximation (VKOGA) procedure with a data-adapted two-layer kernel yields excellent predictors for learning from 3D geometry voxel data via both morphological descriptors or principal component analysis.

**Keywords:** machine learning; kernel methods; two-layered kernels; porous media; breakthrough curves

**MSC:** 68T05; 65D15; 46E22; 76S05



**Citation:** Herkert, R.; Buchfink, P.; Wenzel, T.; Haasdonk, B.; Toktaliev, P.; Iliev, O. Greedy Kernel Methods for Approximating Breakthrough Curves for Reactive Flow from 3D Porous Geometry Data. *Mathematics* **2024**, *12*, 2111. <https://doi.org/10.3390/math12132111>

Academic Editors: Lihua Wang, Benny Y. C. Hon and Sheng-Wei Chi

Received: 29 May 2024

Revised: 25 June 2024

Accepted: 3 July 2024

Published: 5 July 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The reactive flow in porous media plays an important role for many industrial, environmental and biomedical applications. Since the reactions occur at the pore scale, 3D pore scale simulations are very important. At the same time, pore scale measurements are difficult or impossible, and usually, some averaged quantities are measured. One such quantity of interest that can be measured is the breakthrough curve, i.e., the time-dependent integral of the species over the outlet. These breakthrough curves can be computed from a reaction–advection–diffusion equation on a porous medium, which is numerically solved. However, this usually leads to very high computational costs, which might be prohibitively high in a multiquery application, e.g., optimization of the geometry of the porous medium. In that case, a surrogate model for the full order simulation model (FOM) is required. A promising way of obtaining an adequate surrogate model is the use of machine learning (ML) techniques. ML techniques are nowadays widely spread and employed for a variety of tasks, including the estimation of damage for buried pipelines [1] or automatic vision-based sewer inspection [2]. ML algorithms are developed and tested for porous media flows with

different complexity. The literature in this area is very rich, and we cite some exemplary papers simply to point to the place of our research. Numerous papers discuss predicting permeability from microscale geometry. For example, see [3–5] and the references therein. At the next level of complexity, e.g., in [6], the capability of using deep learning techniques has been shown in the case where the velocity field is predicted from the morphology of a porous medium. On the same topic, improvement is achieved by incorporating coarse velocities in the learning process, see [7]. In the last decade, the application of ML techniques for the simulation of passive and reactive transport has rapidly grown. One of the directions in this case is using ML to predict reaction rates when those are very expensive in the case of complex reactions. This can be implemented without taking into account the geometry, see e.g., [8–10], or predicting the rate from structure features, see, e.g., [4,11,12], to name just a few. Recently, the potential of using machine learning models as surrogates for predicting breakthrough curves for varying physical parameters, i.e., Damköhler and Peclet numbers, on a fixed porous medium geometry [13] has been reported.

In the current paper, we address the task of predicting the breakthrough curves for varying geometries of the porous medium with fixed Damköhler and Peclet numbers. Our work is in the same direction as that of [4,11], but there are essential differences. In [11], ML addresses the impact of the structural features on the effective reaction rate to overcome the limitation of the well-mixing assumption. Pore scale reactive flow in an inert skeleton is considered there. In [4], pore scale colloid transport is considered as a part of a filtration problem. Steady-state problems are solved. In our case, our pore scale geometry is in fact a two scale media, as the active washcoat particles are nanoporous. We consider the diffusion of the species within the washcoat, where the reaction occurs. Furthermore, while the other papers discuss different neural network algorithms, we consider a kernel-based method, namely, VKOGA. To the best of our knowledge, such studies have not yet been discussed in the literature.

Because of the computationally demanding simulation of the FOM, we are limited to a scarce data regime as we can only afford to compute a few samples. Furthermore, the input dimension (the number of elements of the discretization of the porous medium) and the output dimension (the number of time steps used during the FOM simulation) are very high. This yields a challenging task for machine learning techniques [14]. For this purpose, greedy kernel approximation schemes [15,16] have been shown to be efficient. These meshless approximation techniques can be combined with deep learning techniques to yield two-layered kernels [17,18], which have also already been successfully applied for certified and adaptive surrogate modeling for heat conduction [18], as well as surrogate modeling of chemical kinetics [8].

Our work is organized as follows: In Section 2, we introduce the underlying equations of the 3D porous medium reaction–advection–diffusion model. In Section 3, we give an overview on greedy kernel methods and two-layered kernels. The numerical experiments are provided in Section 4, and we conclude our work in Section 5.

## 2. Problem Formulation

### 2.1. Geometry

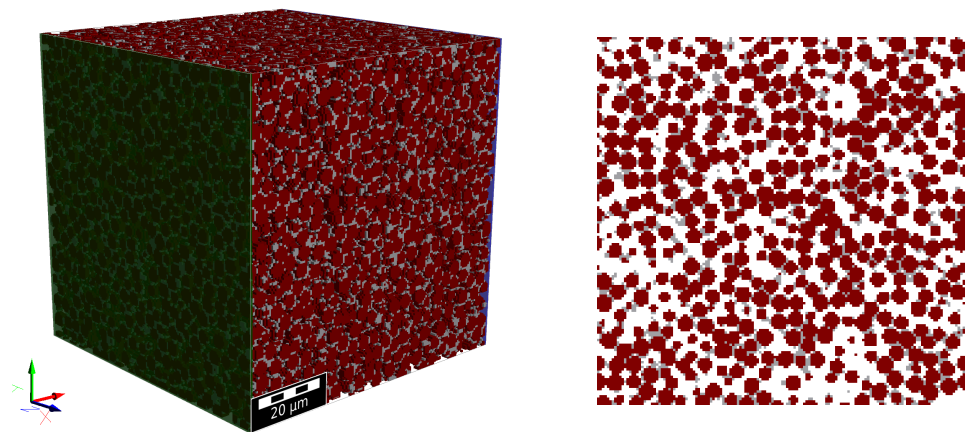
In the current paper, we consider artificially generated voxel-based geometries on the unit cube  $\Omega = [0, 1]^3$  as geometries for porous filter fragments. These consist of the solid skeleton,  $\Omega_s$ , free pores,  $\Omega_f$ , and effective porous space (washcoat or unresolved porosity region)  $\Omega_w$ . Accordingly, we assume that the porous geometry sample can be represented as a union of non-overlapping domains  $\Omega = \Omega_s \cup \Omega_f \cup \Omega_w$ . Each of the domains  $\Omega_w, \Omega_f$  is represented as a union of non-overlapping volume voxels:

$$V_{i,j,k} = [(i-1)h; ih] \times [(j-1)h; jh] \times [(k-1)h; kh], i, j, k = 1, \dots, N_h, h = 1/N_h,$$

where  $N_h$  is the number of voxels in each dimension. This leads to a uniform grid with  $N_V = 150^3$  elements (voxels). The assignment of the voxels to the three subdomains is

stored within two boolean arrays (a third one is not necessary as it can be calculated from the other two). That means each porous sample is represented by a vector  $\mathbf{z} \in \{0, 1\}^{2N_v}$ .

Depending on the application and the production process, the structure of the porous space in real filters could be strongly irregular and highly anisotropic. Nevertheless, for certain types of filters, the micro structure of porous media could be represented as a combination of regular shaped nanoporous granules and solid binder material [19]. We use this representation to reconstruct the porous domain  $\Omega_f$  and the washcoat domain  $\Omega_w$  and later, using existing experimental data on porosity, to generate a series of artificial porous geometries that could be used as models for real filter fragments. The size of each sample in voxels,  $N_h = 150$ , is chosen according to a representative volume element (RVE) study on the one hand and the amount of computational work to generate enough data for training on the other hand. In order to generate each porous sample, we use an analytical spheres piling algorithm for the washcoat region, then voxelize analytical spheres and distribute voxelized solid material (binder) uniformly and randomly, covering the washcoat skeleton surface. The main varying parameter during the generation of the samples was the washcoat volume fraction  $\epsilon_w$ ; thus, in order to generate each sample, the target value of  $\epsilon_w$  was set, but depending on geometry realization, the resulting value of  $\epsilon_w$  for a generated sample could differ by more than 5%. A typical porous sample generated with this two-step procedure with porosity  $\epsilon = 0.553$  is depicted in Figure 1.



**Figure 1.** Isometric (left) and middle plane (right) view of a typical porous sample; colors: brown—washcoat ( $\Omega_w$ ), grey—solid (binder,  $\Omega_s$ ), white (transparent)—free pores ( $\Omega_f$ ), green—inlet boundary section, opposite of the (non-visible)—outlet section. GeoDict visualization [20].

In addition to the typical porous geometry characterization parameters, such as porosity  $\epsilon$ , phase volume fraction (for washcoat)  $\epsilon_w$ , and phase specific surface area (for free pores)  $A_S$ , we also compute the integral Minkowski parameters as morphological features. A complete list of parameters with their definitions for each porous sample can be found in the supplementary Appendix A. In total, 59 artificial porous samples were generated for further processing. The geometry generation time per sample varied greatly depending on the porosity, from the minimum of 6 wall-clock seconds (wcs) to the maximum of 222 wcs, with the average of 111 wcs for a system with 40 cores Intel Xeon CPU E5-2600 v2, 2.8 GHz. On the contrary, the time for evaluation of geometric parameters was almost the same for all samples and did not exceed 2 wcs.

Both the generation and parameter evaluation phases for each geometry were performed with the GeoDict software (Release 2022) [20] and Python scripts.

## 2.2. Governing Equations for FOM

Different approaches can be used to describe the flow and transport of chemical species through the porous medium at the pore-scale. Many of them are based on solving convection–diffusion–reaction (CDR) equations for the species transport and (Navier–)

Stokes equations for the bulk flow. Thus, we consider, as the FOM solution in the current paper, the solution of the CDR equation for a species of interest and additionally assume, based on typical filtration application conditions, that (i) flow Reynolds numbers are sufficiently small  $Re \ll 1$  and Stokes equations are valid for any sub-region of the pore space  $\Omega_f$ , and (ii) the concentration of the species of interest,  $c$ , is much smaller than the bulk mixture concentration. Although these are rather strict assumptions, they work well for catalytic filtration applications and are in fact the standard in this field. This justifies a one-way coupling approach when the solution of the flow equations can be decoupled from the species transport equation. We also restrict ourselves to scalar CDR equations with a linear source term. This type of source term can be used to describe a first order sorption–desorption process in porous media (Henry isotherm). According to the aforementioned assumptions, the CDR equation for species concentration,  $c \geq 0$ , can be written in dimensionless form as

$$\partial_t c - \Delta c + Pe_L \nabla \cdot (\mathbf{v}c) + Da_L c = 0, \quad x \in (\Omega_f \cup \Omega_w), \quad t > 0, \quad (1)$$

where  $Da_L, Pe_L$  are the parameters:  $Da_L = \frac{k_R L}{D}$  is the piece-wise-constant Damköhler number,  $Pe_L = \frac{u_{in} L}{D}$  is the Peclet number,  $L, D > 0$  are the characteristic length and diffusion coefficient, respectively;  $u_{in}$  is the inlet velocity,  $k_R(x) \geq 0$  is the reaction rate constant and reaction occurs only in the washcoat region:

$$k_R(x) = \begin{cases} 0, & \mathbf{x} \in \Omega_f, \\ k_r > 0, & \mathbf{x} \in \Omega_w. \end{cases}$$

In Equation (1), the characteristic diffusion time was used as a time scale. Moreover, we fix the parameter values,  $(Pe_L, Da_L) = (5, 0.1)$ , and thus, consider only a convection-dominated regime for each geometry. For the numerical solution, Equation (1) was complemented with zero initial conditions, Dirichlet conditions for the inlet boundary section and zero Neumann conditions for the outlet boundary section and all other boundaries, see Figure 1. Based on the solution of Equation (1), a quantity of interest, the breakthrough curve, can be evaluated as an integral concentration over the outlet section of the geometry,  $\Gamma_{outlet}$ :

$$a(t) = \int_{\Gamma_{outlet}} c(x, t) \, d\sigma. \quad (2)$$

For the velocity  $\mathbf{v}$  within Equation (1), the stationary Stokes equations in  $\Omega_f$  were considered:

$$\begin{aligned} \mu \Delta \mathbf{v} &= \nabla p, \\ \nabla \cdot \mathbf{v} &= 0, \end{aligned} \quad (3)$$

where  $p : \Omega_f \rightarrow \mathbb{R}$  is the pressure and  $\mu \in \mathbb{R}_+$  is the dynamic viscosity of the gas mixture. Velocity inlet–pressure outlet boundary conditions were used for system Equation (3).

Thus, at the first step of the overall solution procedure, the velocity field  $\mathbf{v}$  was determined as a solution of system Equation (3). In the second step, due to (ii), using the predetermined velocity field, the concentration field  $c$  and integral Equation (2) were computed. Both steps were performed with the PoreChem software (Release 1.0 beta) [21].

### 3. Kernel Methods

Kernel methods [22] comprise versatile tools for scattered data approximation, revolving around the notion of a symmetric kernel  $k : \Omega_d \times \Omega_d \rightarrow \mathbb{R}$ . An important type of kernel is given by strictly positive definite kernels, i.e., kernels such that the so-called kernel matrix  $k(X_N, X_N) = (k(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1}^N$  is positive definite for any choice of pairwise distinct points  $X_N = \{\mathbf{x}_i\}_{i=1}^N \subset \Omega_d$ ,  $N \in \mathbb{N}$ . In the context of machine learning, the set  $X_N$

is often referred to as the training set  $X_{\text{train}}$ , i.e.,  $X_N = X_{\text{train}}$ . In the following, we focus on the popular class of radial basis function kernels on  $\Omega_d \subseteq \mathbb{R}^d$ , which can be expressed as  $k(\mathbf{x}, \mathbf{x}') = \phi(\|\mathbf{x} - \mathbf{x}'\|)$  using a radial basis function  $\phi : \mathbb{R} \rightarrow \mathbb{R}$ . Popular instances of such kernels are given by the following:

$$\begin{aligned}
 k(\mathbf{x}, \mathbf{x}') &= e^{-\epsilon^2 \|\mathbf{x} - \mathbf{x}'\|^2} && \text{Gaussian kernel,} \\
 k(\mathbf{x}, \mathbf{x}') &= (1 + \epsilon \|\mathbf{x} - \mathbf{x}'\|) e^{-\epsilon \|\mathbf{x} - \mathbf{x}'\|} && \text{Matérn 1,} \\
 k(\mathbf{x}, \mathbf{x}') &= (3 + 3\epsilon \|\mathbf{x} - \mathbf{x}'\| + \epsilon^2 \|\mathbf{x} - \mathbf{x}'\|^2) e^{-\epsilon \|\mathbf{x} - \mathbf{x}'\|} && \text{Matérn 2,}
 \end{aligned}$$

which are also later on used in Section 4. As we do not know about the regularity of the target function, we apply kernels that are infinitely smooth (Gaussian kernels) or finitely smooth (Matérn kernels) in order to cover various possible regularities of the target function. The parameter  $\epsilon$  is the so-called shape parameter, which allows us to tune the width of these kernels.

Given not only the input data  $\{\mathbf{x}_i\}_{i=1}^N$  but also the corresponding (possibly vector-valued) target values  $\{\mathbf{f}_i\}_{i=1}^N \subset \mathbb{R}^b$ ,  $b \geq 1$ , a well known representer theorem states that the optimal least squares kernel approximant is given by

$$s_N(\mathbf{x}) = \sum_{i=1}^N \alpha_i k(\mathbf{x}, \mathbf{x}_i), \tag{4}$$

with coefficients  $\{\alpha_j\}_{j=1}^N \subset \mathbb{R}^b$ . These coefficients can be computed directly by solving the regularized linear equation system  $(k(X_N, X_N) + \lambda I)\alpha = y$ , where  $\alpha \in \mathbb{R}^{N \times b}$  and  $y \in \mathbb{R}^{N \times b}$  refer to a collection of the coefficients and target values in arrays. The regularization parameter  $\lambda \geq 0$  allows us to steer the robustness to outliers/noise versus the approximation accuracy in the training points. The value  $\lambda = 0$  corresponds to kernel interpolation. For this case  $\lambda = 0$ , the assumption on the strict positive definiteness of the kernel still ensures the solvability of the system.

**Greedy kernel approximation:** In order to obtain a sparse kernel model, one strives to have a smaller expansion size  $n \ll N$  within Equation (4). For this, greedy algorithms can be leveraged, which select a suitable subset of centers  $X_n$  from  $X_N$ . For this, they start with  $X_0 = \{\}$  and iteratively add points from  $X_N$  to  $X_n$  as  $X_{n+1} := X_n \cup \{\mathbf{x}_{n+1}\}$ , according to some selection criterion. While there are several selection criteria in use in the literature, we focus on the  $f$ -greedy selection criterion, which reads

$$\mathbf{x}_{n+1} := \operatorname{argmax}_{\mathbf{x}_i \in X_N \setminus X_n} |\mathbf{f}_i - s_n(\mathbf{x}_i)|.$$

This residual-based selection criterion incorporates the data point of the largest error, i.e., directly aims at minimizing the maximal absolute error of the kernel model. An efficient implementation of such greedy kernel algorithms is provided by the VKOGA package [15], and a comprehensive analysis of the convergence of such greedy kernel algorithms was provided in [16]. Because of the greedy strategy, VKOGA is often a superior method compared to other kernel-based strategies. Other methods for obtaining sparse, kernel-based models are sparse Gaussian processes [23,24], support vector regression (SVR) [25] or the reduced support vector set approach [25]. The computational efficiency of VKOGA kernel models is shown, for example, in [26], where a comparison of VKOGA with support vector machines was performed, which revealed that VKOGA results in sparser models and, hence, less computational time for training. Similarly, less computational time for prediction was observed in [13], where Gaussian processes and VKOGA were compared. In [27], a comparison of VKOGA and SVR over other ML techniques was performed in a turbulent flow prediction scenario. The study revealed that those kernel methods gave the best performance regarding the quality of the approximated discrete-time dynamics.

**Two-layered kernels:** In order to incorporate feature learning into kernel models, we make use of two-layered kernels [17,18]: These make use of a base kernel  $k$ , as given above, and combine it with a matrix  $\mathbf{A} \in \mathbb{R}^{d \times d}$ , such that the two-layered kernel is given by

$$k_{\mathbf{A}}(\mathbf{x}, \mathbf{x}') = k(\mathbf{A}\mathbf{x}, \mathbf{A}\mathbf{x}') = \phi(\|\mathbf{A}(\mathbf{x} - \mathbf{x}')\|). \tag{5}$$

With this, the two-layered kernel can be optimized to the given data  $(\{\mathbf{x}_i\}_{i=1}^N, \{\mathbf{f}_i\}_{i=1}^N)$  by optimizing the first layer matrix  $\mathbf{A} \in \mathbb{R}^{d \times d}$ . Thus, this two-layered structure may considerably improve the performance of the kernel model, especially for medium- to high-dimensional input data, where an equal importance of all features is highly unlikely. The strength of two-layered kernel models and their superior performance over shallow kernel models has been observed, for example, when applied to heat conduction [18] or when used as a surrogate model for chemical kinetics [8]. We investigate whether this behavior can also be observed for our current problem. For the optimization of the matrix  $\mathbf{A}$ , we employ the fast-gradient-descent-based mini-batch optimization proposed in [17] and extended it to vector valued target values  $b > 1$  in [18]. The overall idea is to leverage an efficiently computable leave-one-out cross validation (LOOCV) error loss, which, thus, makes the kernel generalize well to the unseen data. In particular, this cross validation error loss makes use of both input and target values, i.e., it is a supervised optimization.

The matrix  $\mathbf{A}$  makes the two-layered kernel  $k_{\mathbf{A}}$  even more interpretable: The large singular values with corresponding right singular vectors within the singular value decomposition of the matrix  $\mathbf{A}$  correspond to the more important features within the data set, while the smaller singular values with the corresponding right singular vectors correspond to the less important features within the data set.

In the following Section 4, we make use of the notion “single layered kernel” to refer to the standard kernel  $k$ , while we make use of the notion “two-layered kernel” to refer to (optimized) kernels  $k_{\mathbf{A}}$ .

#### 4. Numerical Experiments

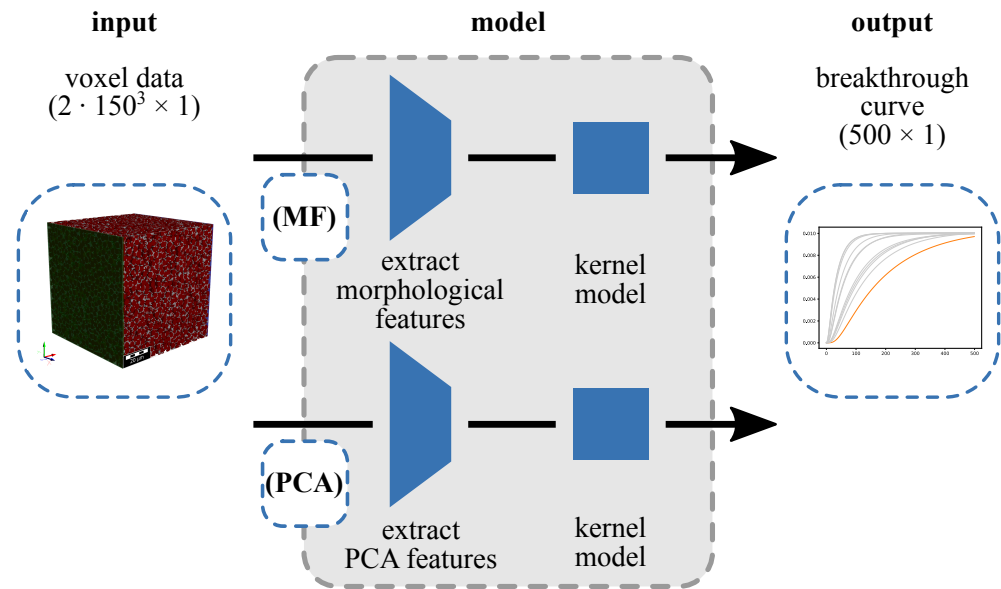
In this section, we present the numerical experiments. The goal is to approximate breakthrough curves Equation (2) from voxel data, which characterizes the geometry of the porous medium. As explained in Section 2, the voxel data are described by a vector  $\mathbf{z} \in \{0, 1\}^{2N_V}$ . In our numerical experiment, we choose  $N_V = 150^3$ . We discretize a breakthrough curve  $a(t)$  on an equidistant temporal grid  $t_i, i = 1, \dots, n_t$ , as the vector  $\mathbf{a} := (a(t_i))_{i=1}^{n_t} \in \mathbb{R}^{n_t}$  with  $n_t = 500$  time steps.

We compare two different kernel-based models for learning the breakthrough curves (see Figure 2), in which either morphological features (MF) prescribed by expert knowledge (see Appendix A) are extracted or principal component analysis (PCA) features are learned from the data using a PCA. The kernel function  $k$  is chosen to be either a shallow one-layered or a two-layered kernel. We refer to these models in the scope of our paper as MF-1L-kernel and MF-2L-kernel or PCA-1L-kernel and PCA-2L-kernel, depending on the depth of the kernel (one-layered/two-layered) and whether we extract morphological features or PCA features.

Both types of models (MF) and (PCA) are based on a feature map  $\Phi : \mathbb{R}^{2N_V} \rightarrow \mathbb{R}^{n_f}$  with  $n_f$  features and a kernel function  $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$  with  $d = n_f$ , such that the resulting kernel models using the single-layer or the two-layered kernels are given as

$$s_n(\mathbf{z}) = \sum_{i=1}^n \alpha_i k(\Phi(\mathbf{z}), \Phi(\mathbf{z}_i)), \quad \text{respectively,} \quad s_n(\mathbf{z}) = \sum_{i=1}^n \alpha_i k(\mathbf{A}\Phi(\mathbf{z}), \mathbf{A}\Phi(\mathbf{z}_i)), \tag{6}$$

with coefficients  $\alpha_i \in \mathbb{R}^b, b = n_t$  and centers  $\mathbf{z}_i \in \{0, 1\}^{2N_V}$  for  $i = 1, \dots, n$ . The expansion size  $n$  is fixed to  $n = 10$ . Choosing smaller values for  $n$  would worsen the results presented in the next subsections considerably, while increasing  $n$  would only influence the approximation quality on the test set slightly.



**Figure 2.** Two feature extraction-strategy-based models to approximate breakthrough curves from voxel data: Model with morphological features (MF), and model with PCA features (PCA).

All of these models involve hyper-parameters, which are listed in Table 1. Suitable values for the hyper-parameters are determined via an LOOCV on the training data set. Note that in the cases where two-layered kernels are applied, no LOOCV for the shape parameter is performed, and instead, the matrix  $\mathbf{A}$  is optimized using the procedure described in the previous section. The kernel function and the regularization parameter are determined via LOOCV in both cases.

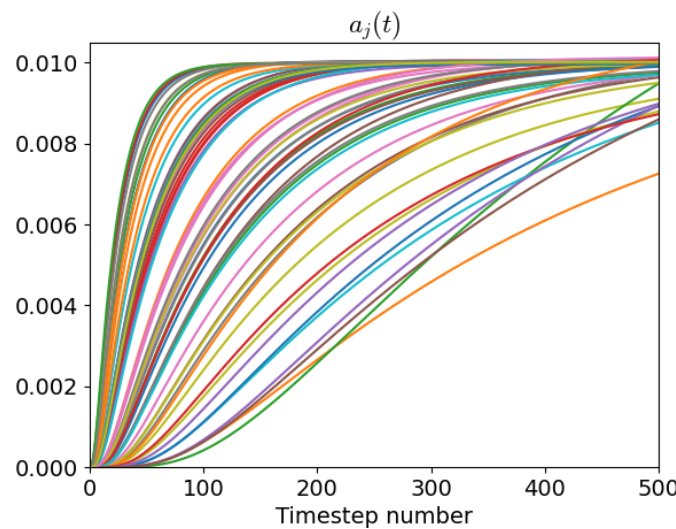
**Table 1.** Hyper -parameters of the kernel function  $k$ , shape parameter  $\epsilon$  and regularization parameter  $\lambda$  used during LOOCV of the MF-based models and PCA-based models.

Hyper-Parameter	Values
kernel fun.	Matérn 1, Matérn 2, Gaussian kernel
Shape parameters	1, 1/2, 1/4, 1/8, 1/16, 1/32
Regularization parameters	0, $10^{-2}$ , $10^{-3}$ , $10^{-4}$ , $10^{-5}$ , $10^{-6}$

In the following, we discuss the generation of the training and test data (Section 4.1), the training and results of the (MF) models based on morphological features (Section 4.2) and of the (PCA) models based on PCA features (Section 4.3).

#### 4.1. Training and Test Data

In total, we consider  $n_s = 59$  samples  $X := \{\mathbf{z}_i\}_{i=1}^{n_s}$  of voxel data  $\mathbf{z}_i$  and the corresponding time-discretized breakthrough curves  $\mathbf{a}_i$  (see Figure 3). The breakthrough curves  $\mathbf{a}_i$  are obtained by solving Equations (1) and (3) based on the geometry described by the corresponding voxel data  $\mathbf{z}_i$ . The average computational cost of solving Equations (1) and (3) for one sample is 7514 wcs with a standard deviation of 286 wcs on a workstation with two Intel Xeon Gold 6240R (48 cores in total).



**Figure 3.** All breakthrough curves  $a_i(t)$  computed from the voxel data  $\mathbf{z}_i$  for  $i \in \{1, \dots, 59\}$ . Different colours correspond to breakthrough curves computed from different geometries.

We use an approximate 80–20% training–test split, which results in  $n_{s,\text{train}} = N = 47$  training samples  $X_{\text{train}} = X_N$  and  $n_{s,\text{test}} = 12$  test samples  $X_{\text{test}}$  with  $X = X_{\text{train}} \cup X_{\text{test}}$  and  $X_{\text{train}} \cap X_{\text{test}} = \emptyset$ . To demonstrate robustness with respect to the choice of the training–test split, we consider three different random splits, where we use the same three random splits to measure the performance of each model. As an error measure, we use a relative error on the test set

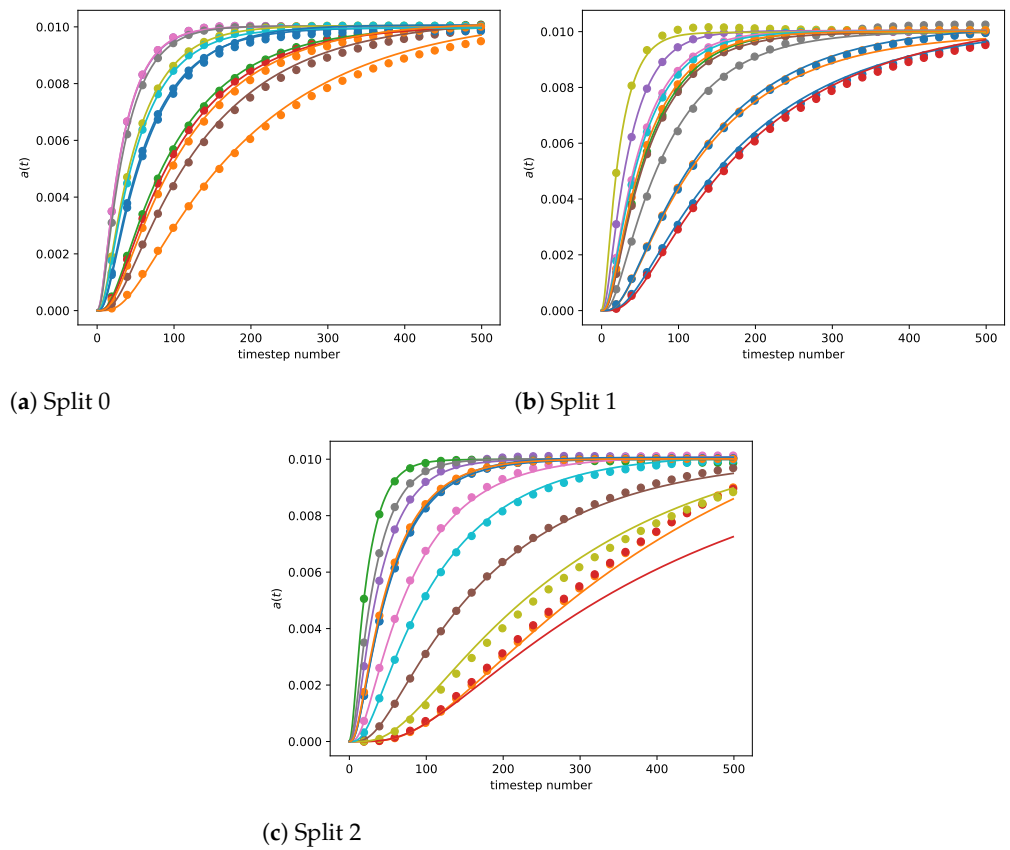
$$e_{\text{rel}} = \frac{1}{|X_{\text{test}}|} \sqrt{\sum_{\mathbf{z}_i \in X_{\text{test}}} \frac{\|\mathbf{a}_i - s_N(\mathbf{z}_i)\|^2}{\|\mathbf{a}_i\|^2}}. \tag{7}$$

#### 4.2. Kernel Models on Morphological Features

Because of the high dimensionality of the input space, it is necessary to apply some reduction technique before training the machine learning model. In this section, we extract  $n_f = 6$  morphological geometry features that describe the porous medium. These are the porosity, the washcoat volume fraction, the volume of free pores, the surface area for the free pores, the integral of mean curvature of the free pores and the integral of total curvature of the free pores. For further information on how they are computed, see supplementary Appendix A. Thus, the machine learning model can be represented by setting  $\Phi = \Phi_{\text{MF}}$  in Equation (6)  $\Phi_{\text{MF}} : \mathbb{R}^{2N_V} \rightarrow \mathbb{R}^{n_f}$  the mapping that computes the morphological features from a given geometry.

For the first experiment, we use shallow kernels and present the approximated breakthrough curves in Figure 4. We observe that except for the red outlier curve in the bottom diagram, all curves are well approximated. This corresponds to the relative test error presented in Table 2, where we see that for the first two data splits, a relative error of approximately 0.01% can be achieved; whereas, for the third split, we only achieve an error of approximately 0.42%. We further observe from Table 2 that for the first two data splits, exactly the same hyper-parameters (kernel function, shape parameter and regularization parameter) are chosen. However, they are chosen differently for the third data set, which may be due to the red outlier curve being part of the test set for the third data split.



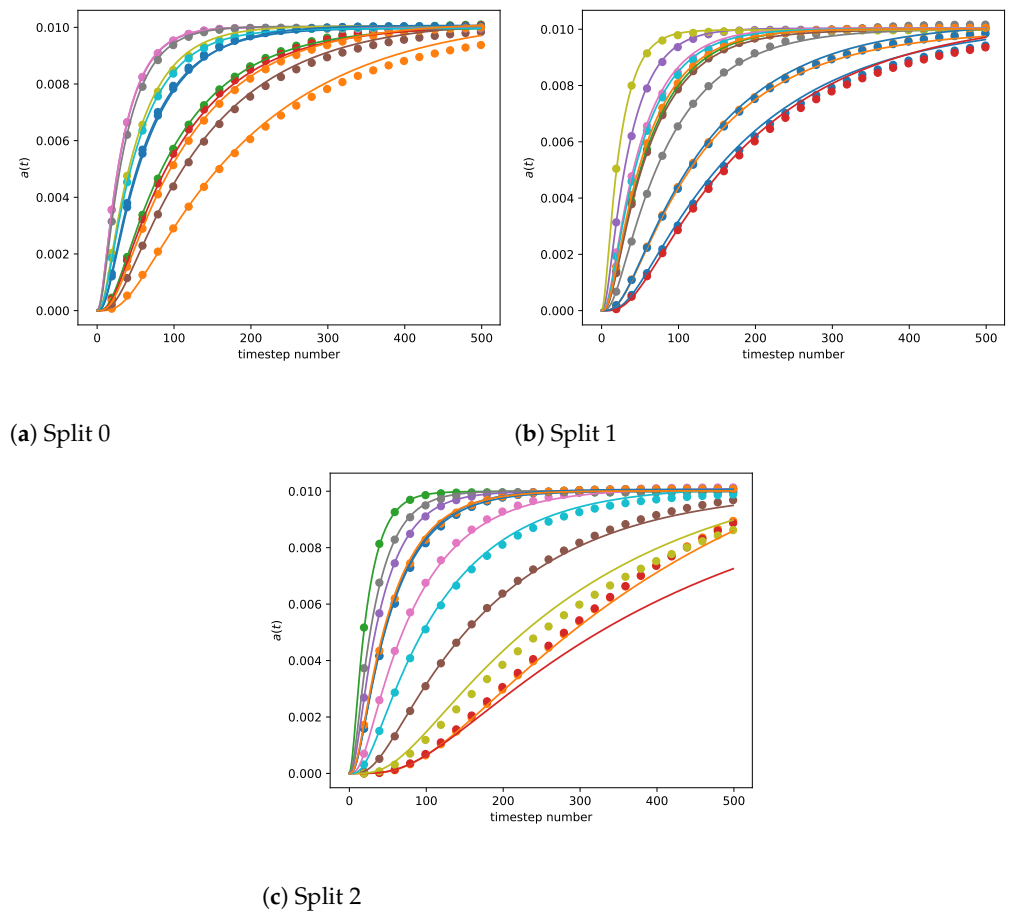


**Figure 4.** MF-1L-kernel: Predicted breakthrough curves on the test set  $X_{\text{test}}$  for three different training–test splits. Solid: Breakthrough curves, dotted: predictions. The different colors correspond to the different breakthrough curves.

**Table 2.** MF-1L-kernel: Relative error (7) and the selected hyper-parameters from Table 1 for three different training–test splits (0–2).

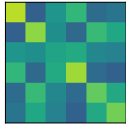
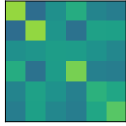
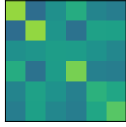
Split	Rel. Err.	Kernel Fun.	Shape Parameter	Reg. Par.
0	$1.49 \cdot 10^{-3}$	Matérn 1	1.00	$1.00 \cdot 10^{-2}$
1	$1.60 \cdot 10^{-4}$	Matérn 1	1.00	$1.00 \cdot 10^{-2}$
2	$4.23 \cdot 10^{-3}$	Matérn 1	$2.50 \cdot 10^{-1}$	$1.00 \cdot 10^{-4}$
average	$1.51 \cdot 10^{-3}$			


For the next experiment, we included two-layer optimization of the kernel model and present the approximated breakthrough curves in Figure 5. We observe that a similar approximation quality can be achieved using two-layered kernels. All breakthrough curves except for the red outlier curve in the bottom diagram are well approximated. The relative test error (see Table 3) is slightly improved for the third data split. However, for the first and second split, it becomes slightly worse. We observe that similar to the experiment with shallow kernel models, for each data split, the Matérn 1 kernel is selected and that for the third split, a smaller regularization parameter is selected than for the first and second one. To compare the shape transformations by the first layer, we present the first layer matrix  $\mathbf{A}$  in Table 3. For an easier comparison, the matrix  $\mathbf{A}$  is visualized using the color-map at the bottom of the table. We observe that the matrices look very similar for each data split.



**Figure 5.** MF-2L-kernel: Predicted breakthrough curves on the test set  $X_{\text{test}}$  for three different training–test splits. Solid: Breakthrough curves, dotted: predictions. The different colors correspond to the different breakthrough curves.

**Table 3.** MF-2L-kernel: Relative error (7) and the selected hyper-parameters from Table 1 for three different training–test splits (0–2).

Split	Rel. Err.	Kernel Fun.	A	Reg. Par.
0	$1.54 \cdot 10^{-4}$	Matérn 2		$1.00 \cdot 10^{-3}$
1	$2.37 \cdot 10^{-4}$	Matérn 2		$1.00 \cdot 10^{-3}$
2	$3.99 \cdot 10^{-3}$	Matérn 1		$1.00 \cdot 10^{-4}$
average	$1.46 \cdot 10^{-3}$			



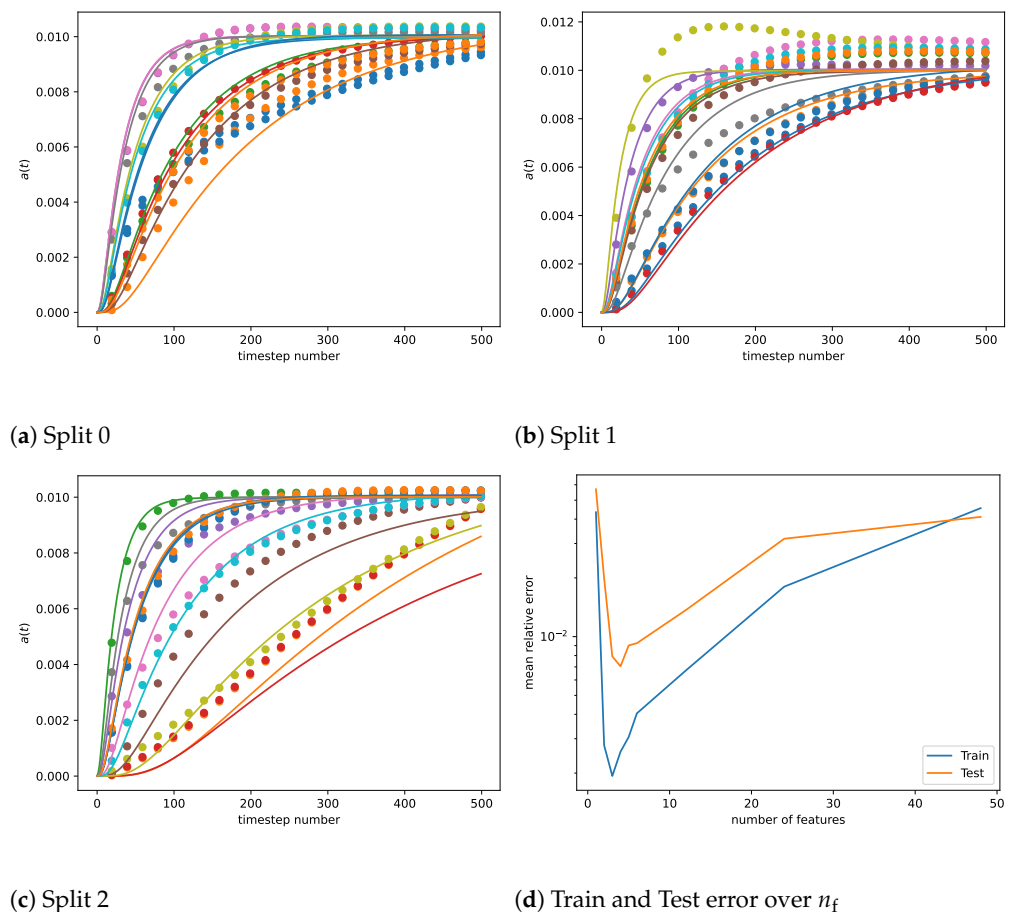
### 4.3. Kernel Models on PCA Features

Next, we use the PCA to define a feature map  $\Phi$ . In order to be comparable to the previous experiment, we choose the number of features as  $n_f = 6$ . Following the idea of the PCA [28], we choose the PCA feature map  $\Phi_{\text{PCA}}(\mathbf{z}) = \mathbf{U}_{n_f}^T \mathbf{z}$  based on the first  $n_f$  left-singular vectors  $\mathbf{U}_{n_f} \in \mathbb{R}^{2N_V \times n_f}$  of the matrix  $\mathbf{Z} := (\mathbf{z})_{\mathbf{z} \in X_{\text{train}}} \in \mathbb{R}^{2N_V \times n_{s,\text{train}}}$  via the SVD

$$\mathbf{Z} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad \text{with} \quad \mathbf{U}_{n_f} := \mathbf{U}(:, : n_f) \quad (8)$$

and set  $\Phi = \Phi_{\text{PCA}}$  in Equation (6) (technical note: The voxel data are saved as a boolean array  $\mathbf{z} \in \{0, 1\}^{2N_V}$ . To compute the SVD, we convert these data to a floating point number, which is why  $\mathbf{Z} \in \mathbb{R}^{2N_V \times n_{s,\text{train}}}$ ).

In Figure 6, we present the prediction of the breakthrough curves for the PCA-1L-kernel model on the test set  $X_{\text{test}}$  for the three different randomized training–test splits. We observe that many of the breakthrough curves are well approximated and a mean relative error of approximately 0.5–1.3% (see Table 4) can be achieved. However, some breakthrough curves are badly approximated, especially the two blue curves for the first data split. We observe that three different kernels, three different shape parameters and three different regularization parameters are chosen for each data split. We further observe from Section 4.3 that  $n_f = 6$  is also a suitable choice in the sense that a small test error is achieved for that setting compared to higher and smaller values of  $n_f$ .

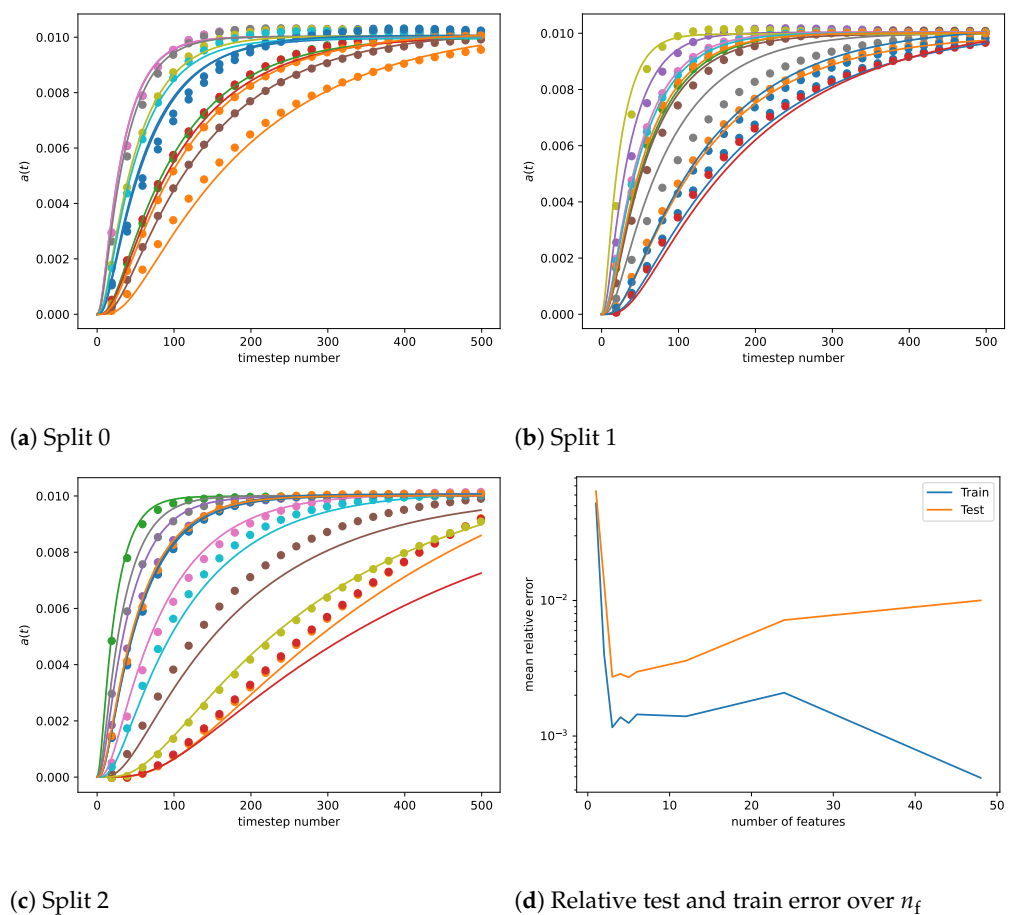


**Figure 6.** PCA-1L-kernel: (a–c): Predicted breakthrough curves on the test set  $X_{\text{test}}$  for  $n_f = 6$  and three different training–test splits. Solid: Breakthrough curves, dotted: predictions. The different colors correspond to the different breakthrough curves. (d): Relative test and train error (Equation (7)) over  $n_f$ .

**Table 4.** PCA-1L-kernel: Relative error (7) and the selected hyper-parameters from Table 1 for three different training–test splits (0–2).

Split	Rel. Err.	Kernel Fun.	Shape Parameter	Reg. Par.
0	$9.62 \cdot 10^{-3}$	Matérn 1	$6.25 \cdot 10^{-2}$	$1.00 \cdot 10^{-3}$
1	$4.77 \cdot 10^{-3}$	Gaussian kernel	$6.25 \cdot 10^{-2}$	$1.00 \cdot 10^{-4}$
2	$1.34 \cdot 10^{-2}$	Matérn 2	$2.5 \cdot 10^{-1}$	$1.00 \cdot 10^{-2}$
average	$9.26 \cdot 10^{-3}$			

For the next experiment, we consider the PCA-2L-kernel model. The results for the analogous experiments to the previous paragraph are presented in Figure 7. We chose again  $n_f = 6$  to be consistent with the previous experiments. However, as we observe in Section 4.3, for  $n_f = 3$ , slightly better results could have been achieved.

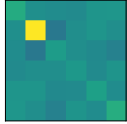
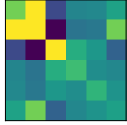




**Figure 7.** (a–c): Top and left: Predicted breakthrough curves on the test set  $X_{\text{test}}$  for  $n_f = 6$  and three different training–test splits. Solid: Breakthrough curves, dotted: predictions. The different colors correspond to the different breakthrough curves. (d): Relative test and train error (Equation (7)) over  $n_f$ .

We observe that the results from the shallow kernel models are considerably improved by using two-layered kernels. The relative errors from Table 5 can be reduced to 0.08–0.7%, and we do not observe badly approximated curves for the first data split anymore. Compared to the the MF-2L-kernel experiments from the previous section, we also observe that the approximation qualities are quite similar. This means that the PCA features transformed by the first layer of the two-layered kernel are able to describe the geometry almost as well

as the morphological features. This shows the strength of two-layered kernels, as the PCA feature extraction is purely data-based, and no expert knowledge had to be used.

**Table 5.** PCA-2L-kernel: Relative error (7) and the selected hyper-parameters from Table 1 for three different training–test splits (0–2).

Split	Rel. Err.	Kernel Fun.	A	Reg. Par.
0	$7.94 \cdot 10^{-4}$	Matérn 1		$1.00 \cdot 10^{-3}$
1	$1.10 \cdot 10^{-3}$	Matérn 2		$1.00 \cdot 10^{-2}$
2	$7.05 \cdot 10^{-3}$	Matérn 1		$1.00 \cdot 10^{-4}$
average	$2.98 \cdot 10^{-3}$			



We further observe from Table 5 that, again, similar hyper-parameters are selected for each data split. For example, the same kernel (Matérn 1) was selected for the first and third data split. Moreover, we see similarities in the first layer matrices **A** of the kernels. For all three matrices, the entries in the left upper  $3 \times 3$  block are considerably larger than the other entries, which means that the first three modes are the important ones. Compared to Table 3, there are larger differences between the first layers. This can be explained by the fact that in contrast to the MF-feature map, the data-based PCA-feature map is different for all three data splits. This is due to the linear mappings defined by the PCA modes being different in each split.

Lastly, we compare the mean errors averaged over the three different training–test splits in Table 6. We observe that the models based on morphological feature extraction work very well in combination with both one-layered and two-layered kernels. In contrast, combining one-layered kernels with PCA feature extraction yields an average error that is almost one order of magnitude larger. This error is considerably improved by applying two-layered kernels, and almost the same accuracy as with the morphological feature extraction is achieved.

**Table 6.** Mean relative test errors, averaged over the three training–test splits.

	1L	2L
MF	$1.51 \cdot 10^{-3}$	$1.46 \cdot 10^{-3}$
PCA	$9.26 \cdot 10^{-3}$	$2.98 \cdot 10^{-3}$

#### 4.4. Runtime Discussion

In this subsection, we compare the runtimes of the different methods in Table 7. We compare the time needed for the computation of the feature map added to the feature extraction time on the test set (first column), the time needed for LOOCV, the training time of the final model and the evaluation time on the test set. Note that the full order model solves and the morphological feature extraction are performed on a different machine (workstation with two Intel Xeon Gold 6240R (48 cores in total)) than the PCA feature extraction and kernel model training (performed on a computer with 64 GB RAM and a 13th Gen Intel i7-13700K processor) due to licensing limitations. We observe that the LOOCV for the two-layered models is way more expensive than for the shallow models, which is due to the optimization of the first layer. However, it is still considerably lower (approximately 30 times for the MF-2L-kernel and 7 times for the PCA-2L-kernel) than the generation of a single training sample (7514 wcs). Furthermore, the LOOCV time and final model training of the MF-kernel models is less expensive than the training on the morphological features. However, the computation of the PCA feature map is less expensive than the computation of the morphological features. Moreover, the evaluation time of the PCA-kernel models is larger than the evaluation of the MF kernel models. Since the computation of the morphological features i.e., the evaluation of  $\Phi_{MF}$  takes approximately 2 wcs per sample, the MF approaches have a larger overall evaluation time (summing evaluation time of MF extraction and MF-1L-kernel/MF-2L-kernel). Compared to the FOM, this results in a speed-up of almost 6 orders of magnitude for the PCA kernel models and 3 orders of magnitude for the MF kernel models. Nevertheless, in both cases, the evaluation times are considerably lower than the computation of the training samples.

**Table 7.** Runtime comparison of the machine learning methods in wcs averaged over the three training–test splits.

Model	Feat. Map	LOOCV Time	Final Model Train Time	Eval. Time
FOM	-	-	-	$9.02 \cdot 10^4$
MF extraction	$9.40 \cdot 10^1$	-	-	$2.40 \cdot 10^1$
MF-1L-kernel	-	$5.90 \cdot 10^0$	$9.95 \cdot 10^{-4}$	$2.91 \cdot 10^{-4}$
MF-2L-kernel	-	$2.01 \cdot 10^2$	$2.12 \cdot 10^{-1}$	$2.67 \cdot 10^{-4}$
PCA feat. map comp.	$1.28 \cdot 10^1$	-	-	-
PCA-1L-kernel	-	$2.29 \cdot 10^1$	$1.36 \cdot 10^{-3}$	$1.83 \cdot 10^{-1}$
PCA-2L-kernel	-	$9.18 \cdot 10^3$	$8.63 \cdot 10^{-1}$	$2.13 \cdot 10^{-1}$

## 5. Conclusions and Outlook

In this work, we demonstrated how breakthrough curves can be predicted from the geometry of a 3D porous medium. We presented two approaches on how to treat the high dimension of the input data: For the first approach, we computed morphological features of the geometries and learned a mapping from these morphological features to the breakthrough curves. For the second approach, we computed PCA features of the geometry data set and learned a mapping from these PCA features to the breakthrough curve.

We observed that the MF approach worked well in combination with one-layered kernels and that both approaches worked well in combination with two-layered kernels. This is compelling, as the morphological features used in the study are well-known informative and predictive quantities for porous media, and in contrast, the PCA feature extraction approach is purely data-based. Thus, the results indicate the strength of using two-layered kernels, as they automatically adapt to the relevant features. We further observed that the scarce-data situation did not prevent these approaches from easily predicting the high-dimensional outputs.

The future work will focus on how ideas from convolutional neural networks can be combined with the framework of deep multi-layered kernels. Moreover, we will investigate whether data augmentation (rotating/reflecting geometry samples without changing the breakthrough curves) can further improve the PCA-feature approach while avoiding the computation of expensive FOM solutions.

**Author Contributions:** Conceptualization, R.H., P.B., T.W., B.H., P.T. and O.I.; methodology, R.H., P.B., T.W. and B.H.; software, R.H., P.B., T.W. and P.T.; writing—original draft preparation, R.H., P.B., T.W. and P.T.; writing—review and editing, R.H., P.B., T.W., B.H., P.T. and O.I.; visualization, R.H., P.B., T.W. and P.T.; supervision, B.H. and O.I.; funding acquisition, B.H. and O.I. All authors have read and agreed to the published version of the manuscript.

**Funding:** Funded by BMBF under the contracts 05M20VSA and 05M20AMD (ML-MORE). The authors acknowledge the funding of the project by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2075 - 390740016.

**Data Availability Statement:** The code to reproduce this study got published on 6 June 2024 and is openly available at <https://doi.org/10.18419/darus-4227>, accessed on 1 May 2024.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Appendix A

In the following, morphological features, i.e., geometry parameters are listed that were computed based on the voxel representation of the porous samples. We assume that we already have a voxel representation of the domain  $\Omega$  and its subsets  $\Omega_f, \Omega_w, \Omega_s$ , where  $\Omega_f, \Omega_w, \Omega_s$  are the free pores, washcoat and solid domains, respectively.

1. Porosity  $\epsilon$ :

$$\epsilon := \frac{|\Omega_f|}{|\Omega|}.$$

2. Washcoat volume fraction  $\epsilon_w$ :

$$\epsilon_w := \frac{|\Omega_p|}{|\Omega|}.$$

3. Volume of the free pores  $V$ :

$$V := |\Omega_f|.$$

4. Surface area for the free pores phase  $S$ :

$$S := |\partial\Omega_f|.$$

5. Integral of mean curvature for free pores phase,  $c_f$ . For smooth surfaces, this quantity is usually defined by the integral

$$c_f := \frac{1}{2} \int_{\partial\Omega_f} \left( \frac{1}{k_1} + \frac{1}{k_2} \right) ds,$$

here,  $ds$  is a surface element of  $\partial\Omega_f$ , and  $k_1$  and  $k_2$  are the two principal curvatures from the respective surface element. Since the boundaries of the voxelized geometry are piece-wise flat, the software package computes an approximation of this quantity for our phase boundaries.

6. Integral of total curvature for free pores phase,  $ct_f$ . For smooth surfaces, this quantity is usually defined by the integral

$$ct_f := \int_{\partial\Omega_f} \left( \frac{1}{k_1 k_2} \right) ds,$$

here,  $ds$  is a surface element of  $\partial\Omega_f$ , and  $k_1$  and  $k_2$  are the two principal curvatures from the respective surface element. The software package computes an approximation of this quantity for our piece-wise flat phase boundaries.

## References

- Zhao, N.; Li, D.Q.; Gu, S.X.; Du, W. Analytical fragility relation for buried cast iron pipelines with lead-caulked joints based on machine learning algorithms. *Earthq. Spectra* **2024**, *40*, 566–583. [\[CrossRef\]](#)
- Hu, C.; Dong, B.; Shao, H.; Zhang, J.; Wang, Y. Toward purifying defect feature for multilabel sewer defect classification. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–11. [\[CrossRef\]](#)
- Wu, J.; Yin, X.; Xiao, H. Seeing permeability from images: Fast prediction with convolutional neural networks. *Sci. Bull.* **2018**, *63*, 1215–1222. [\[CrossRef\]](#) [\[PubMed\]](#)
- Marcato, A.; Boccardo, G.; Marchisio, D. From computational fluid dynamics to structure interpretation via neural networks: An application to flow and transport in porous media. *Ind. Eng. Chem. Res.* **2022**, *61*, 8530–8541. [\[CrossRef\]](#)
- Gärttner, S.; Alpak, F.O.; Meier, A.; Ray, N.; Frank, F. Estimating permeability of 3D micro-CT images by physics-informed CNNs based on DNS. *Comput. Geosci.* **2023**, *27*, 245–262. [\[CrossRef\]](#)
- Santos, J.E.; Xu, D.; Jo, H.; Landry, C.J.; Prodanović, M.; Pyrcz, M.J. PoreFlow-Net: A 3D convolutional neural network to predict fluid flow through porous media. *Adv. Water Resour.* **2020**, *138*, 103539. [\[CrossRef\]](#)
- Zhou, X.H.; McClure, J.E.; Chen, C.; Xiao, H. Neural network-based pore flow field prediction in porous media using super resolution. *Phys. Rev. Fluids* **2022**, *7*, 074302. [\[CrossRef\]](#)
- Döppel, F.; Wenzel, T.; Herkert, R.; Haasdonk, B.; Votsmeier, M. Goal-Oriented Two-Layered Kernel Models as Automated Surrogates for Surface Kinetics in Reactor Simulations. *Chem. Ing. Tech.* **2023**, *96*, 759–768. [\[CrossRef\]](#)
- Laloy, E.; Jacques, D. Speeding up reactive transport simulations in cement systems by surrogate geochemical modeling: Deep neural networks and k-nearest neighbors. *Transp. Porous Media* **2022**, *143*, 433–462. [\[CrossRef\]](#)
- Silva, V.; Regnier, G.; Salinas, P.; Heaney, C.; Jackson, M.; Pain, C. Rapid Modelling of Reactive Transport in Porous Media using Machine Learning. In Proceedings of the ECMOR 2022, Hague, The Netherlands, 5–7 September 2022; European Association of Geoscientists & Engineers: Hague, The Netherlands, 2022; Volume 1, pp. 1–9. [\[CrossRef\]](#)
- Liu, M.; Kwon, B.; Kang, P.K. Machine learning to predict effective reaction rates in 3D porous media from pore structural features. *Sci. Rep.* **2022**, *12*, 5486. [\[CrossRef\]](#) [\[PubMed\]](#)
- Marcato, A. Deep Neural Networks as Scale-Bridging Tools for Flow and Transport Modelling in Porous Media. Ph.D Thesis, Politecnico di Torino, Torino, Italy, 2023.
- Fokina, D. Machine Learning Algorithms for Solution of Convection-Diffusion-Reaction Equation at Pore-Scale. Ph.D Thesis, Fachbereich Mathematik, RPTU Kaiserslautern, Germany, 2023.
- Grigo, C.; Koutsourelakis, P.S. A physics-aware, probabilistic machine learning framework for coarse-graining high-dimensional systems in the Small Data regime. *J. Comput. Phys.* **2019**, *397*, 108842. [\[CrossRef\]](#)
- Santin, G.; Haasdonk, B. Kernel Methods for Surrogate Modeling. In *Model Order Reduction*; De Gruyter: Berlin, Germany, 2021; Volume 2. [\[CrossRef\]](#)
- Wenzel, T.; Santin, G.; Haasdonk, B. Analysis of target data-dependent greedy kernel algorithms: Convergence rates for  $f$ -,  $f \cdot P$ - and  $f/P$ -greedy. *Constr. Approx.* **2023**, *57*, 45–74. [\[CrossRef\]](#)
- Wenzel, T.; Marchetti, F.; Perracchione, E. Data-Driven Kernel Designs for Optimized Greedy Schemes: A Machine Learning Perspective. *SIAM J. Sci. Comput.* **2024**, *46*, C101–C126. [\[CrossRef\]](#)
- Wenzel, T.; Haasdonk, B.; Kleikamp, H.; Ohlberger, M.; Schindler, F. Application of Deep Kernel Models for Certified and Adaptive RB-ML-ROM Surrogate Modeling. In *Proceedings of the Large-Scale Scientific Computations*; Lirkov, I., Margenov, S., Eds.; Springer: Cham, Switzerland, 2024; pp. 117–125. [\[CrossRef\]](#)
- Kato, S.; Yamaguchi, S.; Uyama, T.; Yamada, H.; Tagawa, T.; Nagai, Y.; Tanabe, T. Characterization of secondary pores in washcoat layers and their effect on effective gas transport properties. *Chem. Eng. J.* **2017**, *324*, 370–379. [\[CrossRef\]](#)
- Math2Market. *GeoDict simulation Software Release 2022*; Math2Market GmbH: Kaiserslautern, Germany, 2022. Available online: <https://www.geodict.com/Solutions/aboutGD.php> (accessed on 9 December 2022).
- Iliev, O.; Toktaliev, P. On pore scale numerical simulation of complex homogeneous reactions with application to filtration processes. In Proceedings of the FILTECH 2022, Cologne, Germany, 8–10 March 2022; ISBN 978-3-941655-19-5.
- Wendland, H. *Scattered Data Approximation*; Cambridge Monographs on Applied and Computational Mathematics; Cambridge University Press: Cambridge, UK, 2005; Volume 17. [\[CrossRef\]](#)
- Leibfried, F.; Dutordoir, V.; John, S.; Durrande, N. A tutorial on sparse Gaussian processes and variational inference. *arXiv* **2020**, arXiv:2012.13962.
- Snellson, E.; Ghahramani, Z. Sparse Gaussian processes using pseudo-inputs. *Adv. Neural Inf. Process. Syst.* **2005**, *18*, 1257–1264. [\[CrossRef\]](#)
- Schölkopf, B.; Smola, A.J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*; MIT Press: Cambridge, MA, USA, 2002. [\[CrossRef\]](#)
- Wirtz, D.; Karajan, N.; Haasdonk, B. Surrogate modeling of multiscale models using kernel methods. *Int. J. Numer. Methods Eng.* **2015**, *101*, 1–28. [\[CrossRef\]](#)



27. Carlberg, K.T.; Jameson, A.; Kochenderfer, M.J.; Morton, J.; Peng, L.; Witherden, F.D. Recovering missing CFD data for high-order discretizations using deep neural networks and dynamics learning. *J. Comput. Phys.* **2019**, *395*, 105–124. [[CrossRef](#)]
28. Jolliffe, I.T. *Principal Component Analysis*; Springer: Berlin/Heidelberg, Germany; New York, NY, USA, 2002. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.