University of Stuttgart
Institute for Modelling Hydraulic and Environmental Systems
Department of Stochastic Simulation and Safety Research for Hydrosystems

Maxime Mustermann

On the Optimization and Uncertainty Quantification of Groundwater Models

Master's Thesis

On the Optimization and Uncertainty Quantification of Groundwater Models

Submitted by

Maxime Mustermann

Matriculation Number: 1234567

Examiners:	Prof. Dr.-Ing. Wolfgang Nowak
Supervisors:	M.Sc. Be Treuer

Institute for Modeling Hydraulic and Environmental Systems
Department of Stochastic Simulation and Safety Research for Hydrosystems
Stuttgart, XX MONTH 20XX

Author declaration

I declare that I have developed and written the enclosed thesis completely by myself and that I have not used sources or means without declaration in the text. Any thoughts from others or literal quotations are clearly marked.

The thesis was not used in the same or in a similar version to achieve an academic grading or is being published elsewhere.

The enclosed electronic version is identical to the printed versions.

Date   Signature

Abstract

This study investigates uncertainty quantification (UQ) for learning constitutive relations in contaminant transport modeling using the Finite Volume Neural Network (FINN). In the considered application case accurate retardation factor estimation is crucial for predicting contaminant fate and transport, but traditional methods rely on simplified isotherms that are inaccurate. FINN offers a data-driven alternative by learning complex retardation behavior from concentration data. However, existing UQ methods for FINN, such as Bayesian approaches, can be computationally demanding. We propose two novel, more efficient UQ methods: Stochastic Perturbation Analysis of Networks (SPAN) and Data-SPAN. SPAN explores uncertainty by training FINN with randomly sampled hyperparameters, while Data-SPAN utilizes randomly generated datasets using PI3NN. We empirically demonstrate the near-uniqueness of the retardation factor inverse problem, justifying our focus on solver and data uncertainties. Applying our methods to synthetic and experimental datasets, we construct prediction intervals for the retardation factor and analyze the sensitivity of FINN to variations of a selected set of hyperparameters. We compare our methods with a Markov Chain Monte Carlo (MCMC) baseline, finding that while SPAN and Data-SPAN are computationally more expensive per sample, they offer comparable prediction interval quality and provide a more comprehensive exploration of the uncertainty landscape. Data-SPAN, in particular, shows promising results in terms of reliability. Our findings contribute to more efficient and robust UQ for physics-aware machine learned geophysical processes.

Kurzfassung

This study investigates uncertainty quantification (UQ) for learning constitutive relations in contaminant transport modeling using the Finite Volume Neural Network (FINN). In the considered application case accurate retardation factor estimation is crucial for predicting contaminant fate and transport, but traditional methods rely on simplified isotherms that are inaccurate. FINN offers a data-driven alternative by learning complex retardation behavior from concentration data. However, existing UQ methods for FINN, such as Bayesian approaches, can be computationally demanding. We propose two novel, more efficient UQ methods: Stochastic Perturbation Analysis of Networks (SPAN) and Data-SPAN. SPAN explores uncertainty by training FINN with randomly sampled hyperparameters, while Data-SPAN utilizes randomly generated datasets using PI3NN. We empirically demonstrate the near-uniqueness of the retardation factor inverse problem, justifying our focus on solver and data uncertainties. Applying our methods to synthetic and experimental datasets, we construct prediction intervals for the retardation factor and analyze the sensitivity of FINN to variations of a selected set of hyperparameters. We compare our methods with a Markov Chain Monte Carlo (MCMC) baseline, finding that while SPAN and Data-SPAN are computationally more expensive per sample, they offer comparable prediction interval quality and provide a more comprehensive exploration of the uncertainty landscape. Data-SPAN, in particular, shows promising results in terms of reliability. Our findings contribute to more efficient and robust UQ for physics-aware machine learned geophysical processes.

Notation

This section provides an overview of the notations and symbols used in this thesis.

a Scalar variable

$\mathbf{v}$ Vector variable

$\mathbf{A}$ Matrix variable

$f(x)$ A function of x

∇ Gradient operator

α, β, γ Greek letters representing coefficients

Note: Bold symbols represent vectors, and capital bold symbols represent matrices.

Abbreviations

This section lists the abbreviations and acronyms used throughout this thesis.

AI	Artificial Intelligence
ML	Machine Learning
IoT	Internet of Things
GPU	Graphics Processing Unit
HTTP	HyperText Transfer Protocol
RAM	Random Access Memory
DNA	Deoxyribonucleic Acid

Contents

0.1 Introduction

0.1.1 Motivation

Understanding and predicting the transport of contaminants in groundwater is paramount for effective environmental remediation and protection. Traditional modeling approaches have been shown to be more inaccurate and less generalizable [1] than machine learning based approaches.

The Finite Volume Neural Network (FINN) [1] offers a promising alternative, leveraging a physics-aware, data-driven approach to learn various parts of the physical process directly from experimental data. This allows for greater flexibility and interpretability compared to traditional methods. However, with the additional complexity, the propagation of uncertainty in the data through the entire process up to the output becomes more challenging. But efficient and reliable uncertainty quantification (UQ) is crucial for determining the confidence in predicted contaminant transport. Existing UQ methods for FINN, such as Bayesian Neural Networks with Markov Chain Monte Carlo (MCMC) sampling [2], can be computationally demanding, limiting their practical applicability. This work addresses these limitations by proposing computationally efficient UQ methods for FINN, enabling faster and more reliable uncertainty estimation for contaminant transport predictions.

0.1.2 Background

The transport of contaminants in groundwater is a complex process that involves various physical and chemical interactions. One crucial aspect is diffusion-sorption, where contaminants dissolved in groundwater can interact with the solid phase of the porous medium, such as clay, leading to sorption and desorption processes. This interaction significantly affects the migration and fate of contaminants.

Traditional approaches for modeling contaminant transport rely on numerical solutions of partial differential equations (PDEs) that describe the physical processes. These PDEs often involve parameters that are difficult to measure directly, necessitating the development of parametric models based on physical principles and specific assumptions. One key parameter in these models is the retardation factor, $R(c)$, which quantifies the effect of sorption on contaminant transport. Sorption, the process by which contaminants attach to the solid phase of the porous medium, can significantly influence the rate and extent of contaminant migration. The retardation factor is typically a function of the contaminant concentration, c, and depends on the specific interaction between the contaminant and the soil properties. A common parametric model for defining $R(c)$ are sorption isotherms. Among the most frequently used are the linear, Freundlich, and Langmuir isotherms [1] (see Figure 1 on the facing page).

Determining the retardation factor from concentration data is an inverse problem. Instead of directly solving the PDE with known parameters, the goal is to infer the unknown retardation factor function ($R(c)$) from observed concentration data. This inverse problem is challenging due to the limited availability and potential noise in the data, as well as the complexity of the underlying sorption processes. Even without noise, uniqueness is not guaranteed for implicit equations. To date, we are not aware of any studies that have mathematically or empirically investigated the uniqueness of the retardation factor for the diffusion-sorption PDE given concentration data and boundary conditions. Understanding the uniqueness of the retardation factor is crucial because it significantly impacts the assessment of uncertainty. If multiple mathematically exact solutions exist, then differences between two solutions obtained by the inverse solver cannot be attributed to uncertainties in the data alone, but may reflect inherent non-uniqueness in the problem itself.

Recent advances like physics-informed neural networks in ML have opened up new possibilities for modeling complex physical systems, including contaminant transport. ML-based approaches can learn intricate relationships from data, offering potential advantages over traditional methods in terms of accuracy, flexibility, and generalizability. FINN [1] is a physics-informed ML approach that combines the strengths of traditional numerical methods with the adaptability of neural networks. FINN leverages the finite volume method to discretize the governing PDE and employs neural networks to learn the unknown or uncertain components, such as the retardation factor. This allows FINN to incorporate physical constraints and conservation laws while simultaneously learning from data.

While FINN has shown promising results in modeling contaminant transport, quantifying the uncertainty associated with its predictions remains a challenge. Existing UQ methods for FINN, such as BNNs with MCMC sampling, can be computationally demanding, limiting their practical applicability. This motivates the development of more efficient UQ methods for FINN, which is the focus of this study. Our goal is to develop computationally efficient UQ methods for estimating the uncertainty in the retardation factor predicted by FINN, enabling faster and more reliable assessment of contaminant transport predictions.

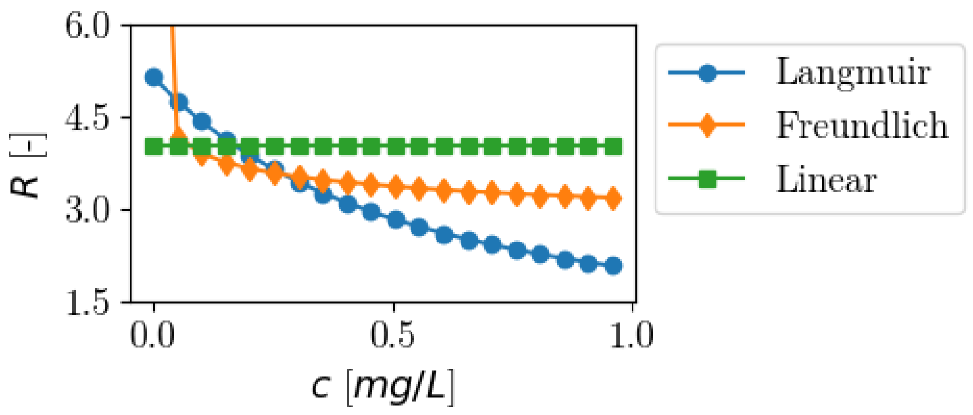

Figure 1: Parametric sorption isotherms for the set of parameters used in Section **??** on page ??.

0.1.3 Contributions of this Work

This study addresses UQ in the FINN framework when applied to a diffusion-sorption process. We begin with an empirical analysis of the uniqueness of the retardation factor, which is a crucial step for interpreting the FINN output. We summarize FINN and PI3NN (Prediction Intervals from Three Neural Networks), the foundational methods used in this work. A novel framework tailored for FINN called SPAN (Stochastic Perturbation Analysis of Networks) is introduced to assess both aleatoric and epistemic uncertainty, allowing us to construct prediction intervals for the retardation factor. We apply this framework to synthetic and experimental data to quantify the uncertainty of the retardation factor and compare the performance of our proposed methods against a BNN baseline employing MCMC sampling. Our results demonstrate that SPAN provides a computationally efficient alternative to MCMC for obtaining prediction intervals for the retardation factor, offering valuable insights into the reliability of the estimated parameter.

0.1.4 Related Work

Existing UQ methods for PDEs often rely on computationally expensive Monte Carlo simulations, limiting their applicability to complex problems. Bayesian Neural Networks, while a popular choice for UQ in deep learning, can be challenging to train and scale, particularly for high-dimensional parameter spaces. Methods like PI3NN offer a more efficient alternative for prediction interval estimation but have not yet been explored within the context of FINN. While considerable research has been dedicated to UQ in deep learning and inverse problems more broadly, efficient and scalable approaches specifically tailored for the unique challenges posed by FINN, such as those proposed in this work, remain limited.

0.1.5 Outline of this Work

This work is structured as follows: Section 2 empirically investigates the uniqueness of the retardation factor inverse problem. Section 3 provides background information on the employed methods FINN and PI3NN. Section 4 details our methodology, introducing the Stochastic Perturbation Analysis of Networks (SPAN) and Data-SPAN approaches for UQ. Section 5 describes the experimental data and setup. Section 6 presents the experimental results, including runtime comparisons and evaluations of likelihood and reliability. Section 7 discusses the findings and limitations of the proposed methods. Finally, Section 8 summarizes the work and outlines potential future research directions.

0.2 Basics

0.2.1 Problem Statement

The diffusion-sorption process, governed by a PDE, describes contaminant transport in groundwater. Experimentally, data can be obtained from a soil cylinder, including breakthrough curves at a fixed location ($x = L$) over time ($t \in [0, T_{\text{end}}]$), and a concentration profile at a specific time ($t = T_{\text{end}}$) across the cylinder's length ($x \in [0, L]$). While a complete concentration field $c(x, t)$ (see Figure 2 on the next page) would be ideal, it's practically unobtainable as measurements at all x values require destructive sampling. Solving the PDE requires knowing the retardation factor, $R(c)$, which quantifies sorption and is dependent on soil properties. Since $R(c)$ is typically unknown, FINN is employed to learn it from the available data. However, uncertainties arise from measurement errors in the data and non-uniqueness of the solver solution, leading to uncertainty in the learned retardation factor. Therefore, the goal is to estimate a prediction interval around the learned $R(c)$ to quantify this uncertainty.

The diffusion-sorption Equation (0.1) reads:

$$\frac{\partial c}{\partial t} = \frac{D}{R(c)} \frac{\partial^2 c}{\partial x^2}, \tag{0.1}$$

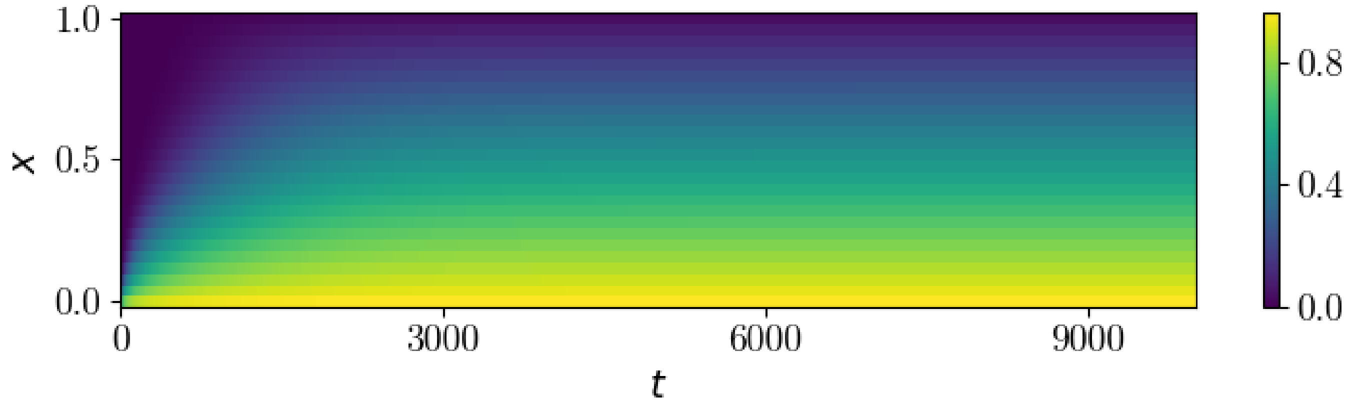

Figure 2: Synthetic concentration field $c(x,t)$ generated using the Langmuir isotherm for a spatial domain of $x \in [0,1]$ meters and a temporal domain of $t \in [0,10000]$ days.

where c is the issolved contaminant concentration, D represents the effective diffusion coefficient, t is time, and x is the distance along the flow path.

The following boundary conditions were considered here:

- Top Boundary: At the top end of the sample where pure-phase TCE is injected, a Dirichlet boundary condition is used:

$$c|_{x=0} = c_{sol} \quad \forall t : 0 \leq t \leq T, \tag{0.2}$$

  where c_{sol} is the solubility limit of TCE in water and T is the experiment time.

- Bottom Boundary: At the bottom end of the sample, which is flushed with clean water, a Cauchy boundary condition is applied:

$$c|_{x=L} = \frac{D}{Q}\frac{\partial c}{\partial x} \quad \forall t : 0 \leq t \leq T, \tag{0.3}$$

  where Q is the flow rate of the clean water and L is the sample length.

The goal is to learn $\hat{R}(x,t;\theta)$ (represented by a neural network) given the data $\mathcal{D} = \{(x_i, t_i, c_i)\}_{i=1}^{N}$ which consists of concentration measurements c_i at spatial positions x_i and temporal points t_i. The network needs to be trained such that the PDE solution using $\hat{R}$ minimizes the error with respect to the data.

0.2.2 Finite Volume Neural Network (FINN)

FINN is a specialized machine learning approach that combines the strengths of traditional numerical methods with the adaptability of neural networks to solve PDEs with unknown or uncertain components. It leverages the Finite Volume Method (FVM), dividing the problem domain into discrete control volumes. The core of FINN lies in its use of "flux kernels" F_i – neural networks that learn how quantities flow between neighboring volumes. Specifically, each volume has a flux kernel composed of two subkernels, f_{i-} and f_{i+}, one for each direction, which model these fluxes based on the concentrations in the volume and its neighbors. These subkernels consist of two parts: a "stencil" component Φ_N that approximates the FVM spatial scheme and a "diffusivity" component Φ_D that learns how the flow rate depends on the concentration itself:

$$
\begin{aligned}
f_{i-} &= \Phi_D(c_i) \cdot \Phi_N(c_i, c_{i-1}) \\
f_{i+} &= \Phi_D(c_i) \cdot \Phi_N(c_i, c_{i+1}).
\end{aligned}
$$

Furthermore, FINN employs "state kernels" S_i that take the local concentration and calculated fluxes as input to update the concentration over time. This time evolution is governed by a Neural Ordinary Differential Equation (NODE) solver [3], a differentiable method that integrates seamlessly into the neural network training. By integrating FVM discretization, specialized neural network modules, and a NODE solver, FINN is capable of learning complex time-dependent spatial patterns while adhering to physical conservation laws and boundary conditions.

0.2.3 Uniqueness of Inverse Problem

As a prerequisite for uncertainty quantification, it's crucial to investigate whether the inverse problem of determining the retardation factor, $R(c)$, from concentration data is unique. Non-uniqueness, meaning multiple possible $R(c)$ functions could produce the same concentration field, would complicate the interpretation of any UQ results. Instead of a purely theoretical analysis, we conduct an empirical investigation into the uniqueness of $R(c)$ for our specific problem.

Our approach leverages the available synthetic concentration data generated using known $R(c)$ functions (e.g., Langmuir, Freundlich, and linear isotherms). We rearrange the diffusion-sorption PDE (Equation (0.1)) to explicitly solve for $R(c)$:

$$
R(c) = D \frac{\partial^2 c}{\partial x^2} \bigg/ \frac{\partial c}{\partial t} \tag{0.4}
$$

We then numerically approximate the first-order time derivative ($\partial c/\partial t$) and the second-order spatial derivative ($\partial^2 c/\partial x^2$) of the concentration field $c(x, t)$ using a B-spline surrogate model. This spline allows us to estimate the derivatives at any point within the spatial and temporal domain of the data.

Figure 3 shows the results of this empirical uniqueness analysis for three different synthetic datasets generated using the Langmuir, Freundlich, and linear isotherms. For each case, we plot the raw $R(c)$ estimates and the binned and averaged $R(c)$ values as a function of c. We also include a comparison with the analytical isotherms used to generate the synthetic datasets.

The results suggest that, for these synthetic datasets, the inverse problem of determining $R(c)$ from concentration data is largely unique, apart from minor numerical inaccuracies. The binned and averaged $R(c)$ values closely follow the trend of the analytical isotherms, indicating that our approach can recover the underlying retardation behavior with reasonable accuracy.

This empirical evidence of uniqueness provides a foundation for subsequent uncertainty quantification. It suggests that discrepancies between different $R(c)$ solutions obtained using FINN can be attributed primarily to aleatoric uncertainty (e.g., measurement noise) and epistemic uncertainty (e.g., limitations in model knowledge) rather than inherent non-uniqueness in the inverse problem itself.

Uncertainty Quantification

There are two main aspects of UQ in this context:

1. UQ for FINN output: $p(\hat{c}(x, t)|\mathcal{D})$. This represents the probability distribution of the predicted contaminant concentration $\hat{c}(x, t)$ obtained from the FINN model, given the data $\mathcal{D}$.
2. UQ for the predicted retardation factor: $p(\hat{R}(c)|\mathcal{D})$. This is arguably more important because it represents the probability distribution of the predicted retardation factor $\hat{R}(c)$ given the data $\mathcal{D}$. This distribution helps in understanding the confidence in the estimated retardation factor.

But there are also two types of uncertainty that have to be differentiated [4, 5]:

1. Aleatoric uncertainty: This arises from inherent randomness in the data and the physical system. It is irreducible and is caused by factors such as measurement noise and variability in the system parameters.
2. Epistemic uncertainty: This stems from limited knowledge of the model (e.g. uncertainty over its parameters). It is associated with model structure, neural network architecture, and limited training data. It quantifies the model's predictive uncertainty.

While a clear decomposition into aleatoric and epistemic uncertainty is often challenging, as seen with methods like BNNs using MCMC, our primary focus is on quantifying the overall predictive uncertainty, which includes both components. However, our method has the added advantage of decomposing the uncertainty by design, providing a more precise and interpretable quantification of both aleatoric and epistemic uncertainties.

0.2.4 Bayes Neural Networks

This section describes the baseline methodology used for comparison, which employs Bayesian Neural Networks and MCMC to estimate the posterior distribution of the retardation factor, $\hat{R}(c(x,t))$, parameterized by a neural network (NN). The retardation factor is a function of concentration, c, which itself is a function of spatial position, x, and time, t. This approach updates prior beliefs about the retardation factor based on observed data, $\mathcal{D}$.

Prior Distribution A Gaussian prior distribution, $p(\theta)$, is placed over the NN parameters θ, reflecting prior knowledge or assumptions. This prior is centered on the parameters θ_{PT} of a pre-trained NN, with covariance matrix $\Sigma_p = 0.05\,I$:

$$p(\theta) = \mathcal{N}(\theta|\theta_{PT}, \Sigma_p)$$

Likelihood Function The likelihood function, $p(\mathcal{D}|\theta)$, quantifies the probability of observing the data $\mathcal{D}$ given a specific set of NN parameters θ. We assume additive Gaussian noise with standard deviation σ corrupts the observed concentrations. Let $c_{obs}(x,t)$ represent the observed concentration at position x and time t, and $\hat{c}(x,t;\theta)$ the concentration predicted by the model (using a FINN solver with $\hat{R}(c(x,t);\theta)$). The likelihood is:

$$p(\mathcal{D}|\theta) = \prod_{(x_i,t_i,c_i)\in\mathcal{D}} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(c_i - \hat{c}(x_i,t_i;\theta))^2}{2\sigma^2}\right) \tag{0.5}$$

Posterior Distribution Bayes' theorem provides the posterior distribution of the NN parameters given the data:

$$p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta)$$

[1] compare several methods to obtain samples from this distribution, including the Metropolis-Hastings algorithm, the MALA algorithm, and the Barker method. The accepted samples $\{\theta_i\}$ from the used algorithm are then treated as draws from the posterior distribution $p(\theta|\mathcal{D})$.

Posterior of Retardation Function Finally, the posterior distribution of the retardation factor $\hat{R}(c)$ is obtained by evaluating the NN with the sampled parameters $\{\theta_i\}$:

$$\{\hat{R}(c|\theta_i)\} \sim p(\hat{R}(c)|\mathcal{D})$$

This set of retardation factor samples provides a probabilistic description of the retardation behavior, incorporating the uncertainty arising from the limited and noisy data.

Since [1] obtain the best results with the Barker method and no burn-in period by starting from the pre-trained model, we follow their approach to have a fair comparison.

0.2.5 PI3NN (Prediction Intervals from Three Neural Networks)

PI3NN [6] is a method for constructing prediction intervals (PIs) that uses three independently trained neural networks. It aims to provide tight, non-crossing PIs across various confidence levels without requiring retraining for each level.

The method's theoretical foundation lies in approximating the ground-truth PI bounds with a family of neural networks. Specifically, PI3NN approximates the median of the target variable, $M[y]$, and the expected deviations above and below the median, $E[(y - M[y])\mathbb{1}_{\{y-M[y]>0\}}]$ and $E[(M[y] - y)\mathbb{1}_{\{M[y]-y>0\}}]$, respectively, using three independently trained neural networks ($f_w(x)$, $u_\theta(x)$, and $l_\epsilon(x)$). These networks are trained using the standard mean squared error loss, simplifying the training process and avoiding the need for specialized loss functions and sensitive hyperparameters that often require fine-tuning in other PI methods.

Once trained, PI3NN constructs the PI bounds as linear combinations of the three networks' outputs. The coefficients of these linear combinations (α and β) are determined through a root-finding algorithm that ensures the desired PICP (Prediction Interval Coverage Probability) for a given confidence level q (γ in [6]). This process allows for calculating PIs for multiple confidence levels without retraining the networks, offering significant computational advantages.

Figure 4 on page 24 illustrates the 6-step breakdown of the PI3NN process:

1. **Learn the mean:** Train a neural network to approximate the mean of the target data.
2. **Estimate the median:** Calculate the median by shifting the learned mean.
3. **Calculate and split residuals:** Compute the residuals between the actual data and the estimated median, then separate these into positive and negative residuals.
4. **Learn residuals:** Train two more neural networks, one to approximate the positive residuals and the other to approximate the negative residuals.
5. **Construct PI:** Calculate a PI using the learned median and the learned positive and negative residuals.
6. **Generate multiple PIs:** Use a root-finding algorithm to compute PIs for different confidence levels q without retraining the networks.

0.3 Methodology

0.3.1 SPAN (Stochastic Perturbation Analysis of Networks)

Our objective is to determine the posterior predictive distribution $p(\hat{R}(c;\theta) = R(c)|\mathcal{D})$, which represents the probability of computing a specific retardation factor value $R(c)$ for a given concentration c, based on the observed breakthrough curve data $\mathcal{D}$. We present a method that computes this distribution by marginalizing over uncertain factors, denoted by $\mathcal{X}$, of the model and employing an approximate likelihood function $p(\mathcal{D}|\mathcal{X})$.

Approximating the Posterior Predictive Distribution via Marginalization over Uncertain Factors

Model Formulation We represent our neural network as $\hat{R}(c;\theta)$, where c is the input concentration and θ denotes the network's parameters (weights and biases). The parameters θ are determined by a deterministic solver $S(h,\mathcal{D})$, which takes hyperparameters h and training data $\mathcal{D}$ as input. A prior distribution $p(\mathcal{X})$ is assigned over $\mathcal{X}$; the exact form of $p(\mathcal{X})$ is not crucial for the derivation, as it suffices to generate samples from it. When we choose to treat h or the training data $\mathcal{D}$ to be a random variable, so to be our uncertain factor $\mathcal{X}$, the network parameters $\theta = S(h,\mathcal{D})$ become a random variable and thus uncertain. We denote this by $\theta_{\mathcal{X}}$ This allows us to calculate the posterior predictive distribution as follows:

$$\begin{aligned}
p(\hat{R}(c;\theta_{\mathcal{X}}) = R(c)|\mathcal{D}) &= \int p(\hat{R}(c;\theta_{\mathcal{X}}) = R(c)|\mathcal{X},\mathcal{D})\, p(\mathcal{X}|\mathcal{D})\, d\mathcal{X} \\
&= \int p(\hat{R}(c;\theta_{\mathcal{X}}) = R(c)|\mathcal{X},\mathcal{D})\, p(\mathcal{X})\, \underbrace{\frac{p(\mathcal{D}|\mathcal{X})}{p(\mathcal{D})}}_{=w(\mathcal{X})}\, d\mathcal{X} \\
&= \int p(\hat{R}(c;\theta_{\mathcal{X}}) = R(c)|\mathcal{X},\mathcal{D})\, p(\mathcal{X})\, w(\mathcal{X})\, d\mathcal{X} \\
&= \int \delta(\hat{R}(c;\theta_{\mathcal{X}}) - R(c))\, p(\mathcal{X})\, w(\mathcal{X})\, d\mathcal{X}
\end{aligned}$$

The first step is the marginalization over $\mathcal{X}$, the second the application of Bayes' Theorem, and the last stems from the fact that the solver is deterministic given $\mathcal{X}$ and $\mathcal{D}$ and thus the distribution becomes a delta distribution.

Importance Sampling Sampling from $p(\hat{R}(c;\theta_{\mathcal{X}}) = R|\mathcal{D})$ is equivalent to sampling from $p(\mathcal{X})$, evaluating the solver to obtain the model parameters $\theta_{\mathcal{X}}$, evaluating the model $\hat{R}(c;\theta_{\mathcal{X}})$, and reweighting the samples with $w(\mathcal{X})$. This is the concept of importance sampling. To approximate the distribution, we draw M samples $\{\mathcal{X}_m\}_{m=1}^{M}$ from the prior distribution $p(\mathcal{X})$ and calculate the importance weight for each:

$$w_m = \frac{p(\mathcal{D}|\mathcal{X}_m)}{p(\mathcal{D})} \propto p(\mathcal{D}|\mathcal{X}_m)$$

As $p(\mathcal{D})$ is a constant, it can be omitted, and the weights can be normalized subsequently.

Likelihood Computation We use the same assumption about our data that the Bayesian baseline uses, which results in an easy computation of the likelihood by substituting θ with $\theta_{\mathcal{X}}$ in Equation (0.5) on page 8:

$$p(\mathcal{D}|\mathcal{X}) = \prod_{x_i,t_i,c_i \in \mathcal{D}} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(c_i - \hat{c}(x_i,t_i;\theta_{\mathcal{X}}))^2}{2\sigma^2}\right)$$

SPAN Variants: Hyperparameter and Data Marginalization

The generalized framework above can be applied to different sources of uncertainty. Here, we describe two variants:

SPAN (Hyperparameter Marginalization) In this case, we consider the uncertain factor $\mathcal{X}$ to be the hyperparameters h of the solver. This variant, which we refer to as SPAN, captures epistemic uncertainty arising from the choice of hyperparameters. The equations above are applied directly with $\mathcal{X} = h$.

Hyperparameter Sampling and Rationale The following hyperparameters were selected for perturbation in the SPAN framework to assess their influence on model output uncertainty. Each hyperparameter was chosen based on its inherent uncertainty or significant impact on the training process:

- **Weight Initialization Seed:** This hyperparameter dictates the initial conditions for the neural network training process. It influences various stochastic elements, including random number generation within algorithms and the initialization of network weights. We employed the Kaiming Uniform initialization scheme [7], which mitigates the vanishing/exploding gradient problem in deep networks by ensuring consistent variance of activations and gradients throughout the network. *Rationale:* The initialization of a neural network is a critical step, and the seed provides the starting point for optimization. Since a zero initialization is often detrimental, a random value is typically chosen. The impact of this fundamental choice on model output is important to quantify. Seed values were sampled uniformly from the integer range $[0, 10^9]$.
- **Number of Epochs:** This parameter determines the number of complete passes the model makes over the training dataset during optimization. *Rationale:* Determining the optimal number of training epochs for a neural network is often challenging. While monitoring loss is a common practice, loss can sometimes decrease unexpectedly after long plateaus. This implies that the ideal stopping point is inherently uncertain. Additionally, selecting an inadequate number of epochs can mimic the effect of a poorly chosen learning rate. A learning rate that is too high may prevent convergence to the optimal solution, similar to stopping the training process prematurely. By varying the number of epochs, we can capture this uncertainty and its effect on the model's final predictions. The number of epochs was sampled uniformly from the integer range $[10, 30]$.

- **MSE Loss Factor (a_{MSE}):** This factor weighs the contribution of the mean squared error (MSE) loss between the predicted concentration ($\hat{c}$) and the true concentration (c) in the overall loss function. The loss function is defined as: $a_{MSE} \cdot MSE(c, \hat{c}) + a_{Phys} \cdot Loss_{Phys}(\hat{R}(c))$. *Rationale:* The optimal balance between the data-driven MSE loss and the physics-informed loss (described below) is often unknown a priori. There is no established theoretical framework to guide the selection of this factor, making it a source of uncertainty. Thus, by perturbing a_{MSE}, we investigate how the model's sensitivity to data fidelity influences its predictions. The values for a_{MSE} were sampled from a log-uniform distribution within the range $[10^2, 10^8]$.
- **Physical Loss Factor (a_{Phys}):** This parameter, analogous to the MSE loss factor, governs the weight of the physical loss term, $Loss_{Phys}(\hat{R}(c))$, in the overall loss function. *Rationale:* Similar to the MSE loss factor, the optimal weighting for the physics-informed loss is generally unknown and lacks a strong theoretical basis. By varying a_{Phys}, we explore how strongly the model relies on the underlying physical constraints and how this affects its predictions. The values for a_{Phys} were also sampled from a log-uniform distribution within the range $[10^2, 10^8]$.

Data-SPAN (Data Marginalization) Alternatively, we can consider the uncertain factor $\mathcal{X}$ to be the dataset $\tilde{\mathcal{D}}$ itself. This variant, which we call Data-SPAN, captures aleatoric uncertainty stemming from the inherent randomness in the data. The equations are applied with $\mathcal{X} = \tilde{\mathcal{D}}$.

Mask-based Random Dataset Sampling To create variations in the training data, we employed a random masking technique. For each sample, a unique seed was used to randomly select 50% of the data points for masking to generate a random dataset. "Masking" in this context means that these data points were excluded when computing the loss during training. Each data point had an equal probability (0.5) of being masked or retained. The seed values were drawn uniformly from the integer range $[0, 10^9]$. Figure 5 illustrates an example of the full synthetic concentration training data with 50% of the data points randomly masked.

Noise-based Random Dataset Sampling Another approach to generating dataset variations involved the introduction of additive Gaussian noise. Following [8], which suggests measurement error can be modeled as Gaussian with zero mean and a standard deviation of 5% of the measured value, we used this noise level to simulate data uncertainty comparable to experimental data. A unique seed, uniformly sampled from the integer range $[0, 10^9]$, was used to generate the noise for each dataset.

PI3NN-based Random Dataset Sampling To generate random datasets for Data-SPAN, we leverage PI3NN, applying it to the original breakthrough curve (BTC) dataset of core 2 $\mathcal{D}_{\text{BTC-2}}$. For more detailed information about the data, see Section 0.4.2 on the next page. Let $\hat{c}(x = L, t; \theta_{\mathcal{D}_{\text{BTC-2}}})$ be the mean concentration curve predicted by FINN trained on $\mathcal{D}_{\text{BTC-2}}$, with $\theta_{\mathcal{D}_{\text{BTC-2}}}$ representing the learned FINN parameters. This FINN-predicted mean curve is used as input to the PI3NN algorithm, as it provides a more accurate representation of the underlying process compared to the mean curve learned by a standard MLP within the PI3NN framework.

PI3NN then learns the residuals, effectively modeling the distribution of c given t. Let $F^{-1}(q|t)$ denote the quantile function predicted by PI3NN, where $q \in [0, 1]$ is the quantile level and t is the time point. This function provides the concentration value at time t corresponding to the q-th quantile of the learned distribution (see Figure 6 on page 26).

To create a random dataset sample $\tilde{\mathcal{D}}(q)$, we sample a quantile level q from a uniform distribution $\mathcal{U}(0, 1)$. Then, the random dataset sample is constructed as:

$$\tilde{\mathcal{D}}(q) = \{(t_i, F^{-1}(q|t_i))\}_{i=1}^{N}$$

This process is repeated M times to obtain a collection of random datasets $\{\tilde{\mathcal{D}}_j\}_{j=1}^{M}$. Each $\tilde{\mathcal{D}}_j$ is then used to train a separate instance of FINN, effectively providing a set of retardation factor samples, analogous to the samples obtained from the hyperparameter perturbation approach.

To enhance the accuracy of the likelihood computation in Section 0.5.2 on page 19, we intentionally include the quantile levels 0 and 1 in the random samples. This improves the fit of the distribution to the histogram of the samples.

Full-SPAN We can even treat both the hyperparameters and the training dataset as random variables at the same time and obtain a combined result. So $\mathcal{X} = (h, \mathcal{D})$ and both are sampled independently.

For both hyperparameter and dataset sampling, a few samples resulted in FINN not converging during training. These samples were discarded from further analysis. Convergence was determined by monitoring the loss value, requiring it to fall below a threshold of 1e-5 and remain unchanged for a certain number of iterations.

0.4 Data and Setup

0.4.1 Experimental Setup

The experiments were conducted using Python 3.11.3. Modified versions of the FINN code from [1] and the PI3NN code from [6] were used. The code and specific library versions are available on GitHub. GNU parallel [9] was used for parallel execution of experiments.

Simulations and timing measurements were conducted on a M1 Pro Macbook. This setup provided the necessary computational resources for efficient model training and evaluation.

0.4.2 Data

Synthetic Data For the synthetic datasets, we used the exact parameter values from [1] (Table 1), including porosity, effective diffusion coefficient, and others. The concentration data was generated using the same solver employed by FINN, ensuring consistency in our experimental setup.

Experimental Data This study utilizes experimental data originally presented by [8] and subsequently employed by [1]. The data consist of breakthrough curves (BTCs) and a concentration profile, obtained from laboratory column experiments. These are detailed below and summarized in Table 1 on the following page.

- **Core 1:** BTC data representing the concentration, c, at the outlet ($x = L$) of the column over time, t. Measurements were taken at $L = 0.0254\,\text{m}$, with time points spanning $t \in \{0.792\,\text{days} \cdot i \mid i = 0, \ldots, 49\}$ (see Figure 7 on page 27, left).
- **Core 2:** BTC data similar to Core 1, but with $L = 0.026\,\text{m}$ and time points $t \in \{0.737\,\text{days} \cdot i \mid i = 0, \ldots, 54\}$ (see Figure 7 on page 27, middle).
- **Core 2B:** Concentration profile data representing c as a function of distance, x, along the column at a fixed time $T = 48.88\,\text{days}$. Measurements were taken at intervals of $0.00362\,\text{m}$, spanning $x \in \{0.00362\,\text{m} \cdot i \mid i = 0, \ldots, 29\}$ (see Figure 7 on page 27, right).

Following the approach of [1], Core 2 was used for model training, while Core 1 and Core 2B served as independent test datasets. The parameter values provided alongside the concentration data in [8] were also used.

Table 1: Summary of Experimental Data

Core	Type	Variables	Parameter Values
Core 1	BTC	$c(x = L, t)$	$L = 0.0254\,\text{m}$, $t \in \{0.792\,\text{days} \cdot i \mid i = 0, \ldots, 49\}$
Core 2	BTC	$c(x = L, t)$	$L = 0.026\,\text{m}$, $t \in \{0.737\,\text{days} \cdot i \mid i = 0, \ldots, 54\}$
Core 2B	Profile	$c(x, t = T)$	$T = 48.88\,\text{days}$, $x \in \{0.00362\,\text{m} \cdot i \mid i = 0, \ldots, 29\}$

0.4.3 Training Details

The architecture and training is consistent with [1]. For the dissolved concentration c, the module $\phi_D(c)$ is defined as a feedforward neural network with a 5-layer configuration of [1, 10, 20, 10, 1] neurons. For the total concentration c_t, $\phi_D(c_t)$ is defined as a scalar parameter to learn the unknown diffusion coefficient D. Although the FINN framework allows for learning the stencil, we did not utilize this feature here.

The L-BFGS optimizer [10] with a learning rate of 0.1 was used.

During training, runs were discarded if the normalized mean squared error (NMSE) on the training set exceeded 10^{-5}. This threshold ensures a minimum level of accuracy in the learned retardation factor.

$$\text{NMSE}(y, \hat{y}) = \text{MSE}(y, \hat{y})/\text{mean}(y)$$

0.5 Results and Discussion

0.5.1 Synthetic Data Case

SPAN

We developed and validated our method using synthetic data prior to its application on experimental data, where actual comparisons and quantifications could be performed. As a result, our initial evaluation focused on a simplified parameter space, with only a single hyperparameter, namely the weight initialization seed. The number of samples varied and is indicated in the respective figure captions for clarity and comparison within those specific contexts.

The influence of the seed is relatively small as can be seen in Figure 8 on page 27.

Data-SPAN

When applying gaussian noise of the same strength as the experimental data has, the variation in learned retardation factors becomes much larger compared to SPAN as can be seen in Figure 9 on page 27.

Masking the training data has a smaller effect, as seen in Figure 10 on page 28, although there is one outlier retardation factor, indicating, that there is the potential for high variations similarly to applying gaussian noise. More samples would be needed to confirm this.

0.5.2 Experimental Data Case

SPAN

For the experimental data, the full set of hyperparameters was used: **Weight Initialization Seed**, **Number of Epochs**, **MSE Loss Factor**, **Physical Loss Factor**.

To introduce greater variability in the retardation factor output, we simultaneously sampled all hyperparameters, as opposed to the individual approach used in the synthetic data case. To still assess the impact of each hyperparameter, we conducted a sensitivity analysis, which is presented in Section 0.5.2. This experiment generated a dataset of 780 samples, based on the defined hyperparameter ranges, and the results are illustrated in Figure 11 on page 28.

Sensitivity Analysis Sensitivity analysis investigates how variations in input parameters affect the model output. Unweighted SPAN samples inherently perform sensitivity analysis because each sample represents a different set of hyperparameters, and the resulting retardation factor and its impact on the concentration field demonstrate the influence of those hyperparameters.

One sensitivity analysis method is based on standardized regression coefficients, sometimes referred to as beta coefficients or beta weights. They represent the relative importance of each predictor variable (hyperparameter h in this case) in explaining the variance of the outcome variable ($R(c)$). Importantly, the sum of these normalized coefficients can exceed 1 as parameters may explain common variances. The value shows how important the respective parameter is in relation to the rest. Since these parameters are based on linear regression, they should be interpreted with caution, especially when non-linear relationships exist between the hyperparameters and the retardation factor.

We average the importance for $R(c)$ over a set of c values and obtain 0.033 for **Physical Loss Factor**, 0.051 for **Weight Initialization**, 0.159 for **MSE Loss Factor**, and 0.570 for **Number of Epochs**.

This indicates that **Number of Epochs** and **MSE Loss Factor** have the largest impact on $R(c)$ overall. The seed used for weight initialization and the physical loss factor play only negligible roles.

Figure 12 on page 29 shows the relative importance for values of c over which the average was taken.

Data-SPAN

This approach was applied only to the experimental data because the synthetic training data lacks noise, leading to extremely small residuals and a very narrow distribution that does not impact sampling.

For the experimental data, we sampled 70 quantiles as detailed in Section 0.3.1 on page 14. The results are shown in Figure 13 on page 29.

Full-SPAN

This approach is equivalent to Data-SPAN but additionally, for every sampled quantile, the hyperparameters are also randomly sampled. The results are shown in Figure 14 on page 30.

Full-SPAN vs. MCMC

Baseline (Bayes NN via MCMC) We computed 10k samples using the same parameters as [1]; the samples are visualized in Figure 15 on page 31.

Retardation Uncertainty Comparison Figure 16 on page 32 illustrates that the 90% PIs for both the MCMC and full SPAN methods encompass similar concentration values. This agreement arises from their comparable coverage of $R(c)$ values when $c \leq 1$. While the methods diverge for larger values of c, this discrepancy has minimal impact on the concentration fields, as these are less sensitive to $R(c)$ in that regime. Given that full SPAN produces similar c PIs but incorporates larger uncertainty in R, it provides a more robust uncertainty estimate. A more quantitative exploration of these findings is presented in the following sections.

Runtime Comparison We measure and compare the runtime of generating a single sample of our method and the baseline. Since SPAN as well as Data-SPAN require training of the NN from scratch for each sample, the runtime approximately equals the training time. Averaging over 10 trials, the runtime is 170 seconds.

The MCMC method uses thinning, saving only every 10th sample. The average runtime for this is about 11.4 seconds, almost 15 times faster than our method as also seen in Figure 17 on page 33. However, when considering the total number of samples used for both methods, our method is about 10 times faster.

Likelihood Comparison To evaluate the predictive quality of our method, we use its samples to estimate a probability distribution. This is done by computing a histogram. We can then estimate the likelihood of the training data $\mathcal{D}$ given this distribution. We transform the likelihood into a negative log-likelihood (NLL) to obtain more accurate results. A lower NLL is better. Applying this procedure on SPAN yields a NLL of -6.41, -7.11 for MCMC, and -7.14 for Data-SPAN. A good baseline to compare this against, is a normal distribution around the FINN BTC prediction mean and standard deviation equal to the sample standard deviation computed from the residuals:

$$p(c;t) = \frac{1}{\sqrt{2\pi \int^2}} \exp(-\frac{1}{2\int^2}(\hat{c}(x = L, t; \theta_{\mathcal{D}}) - c)^2)$$

This yields a NLL of -7.50, which is the lowest value.

PI Calibration Comparison Another quantification done by [1] is the average calibration of the PI. This can be visualized using the reliability curve.

The reliability curve assesses the calibration of predicted confidence intervals. It plots the observed frequency of the true value falling within a given confidence interval against the predicted confidence level. To compute it, the prediction range is divided into bins (e.g., by confidence levels). For each bin, the proportion of predictions where the true value falls within the corresponding PI is calculated. Ideally, the curve should follow the diagonal line (perfect calibration). A curve above the diagonal indicates underconfidence (the true value falls within the PI more often than predicted), while a curve below the diagonal indicates overconfidence.

The results, shown in Figure 18 on page 34, suggest that Data-SPAN provides the best calibrated PI. Overall, all methods are generally overconfident, SPAN the strongest. Only Data-SPAN is a little underconfident for very small c values.

0.5.3 Summary

The analysis of our results reveals several key insights regarding the performance and characteristics of the different uncertainty estimation methods, particularly focusing on the comparison between Full-SPAN and MCMC.

Firstly, the synthetic data case exhibits significantly lower uncertainty compared to the experimental data scenario. This can be attributed to several factors: a larger number of datapoints, which reduces epistemic uncertainty; the reduced effectiveness of Data-SPAN due to the absence of real noise and presence of artificial noise in the synthetic data, which, although introducing some level of uncertainty, does not represent a real distribution, rendering the application of PI3NN, our strongest method to generate datasets, unreasonable since it is designed to learn the underlying real distribution; and the utilization of less effective SPAN hyperparameters (such as excluding the strongest options like **Number of Epochs** or **MSE loss factor**, as shown in Section 0.5.2 on page 17).

Given that the hyperparameters of SPAN have a negligible impact on output variation relative to Data-SPAN, and considering Data-SPAN's ability to efficiently sample the entire space due to its one-dimensional nature, Full-SPAN requires substantially fewer samples than MCMC. This is a favorable characteristic, as it implies that Full-SPAN can achieve reliable uncertainty estimates even with a limited number of samples and thus computational effort. Furthermore, Full-SPAN benefits from trivial parallelizability, unlike MCMC, which requires sequential sampling or more advanced methods. This, coupled with the ability to operate effectively with fewer samples, establishes Full-SPAN as a more computationally efficient method for UQ.

0.5.4 Limitations

One limitation of our approach lies in its "brute-force" nature, due not only to the computational effort but also to the process itself. While we gain insights into uncertainty, the obtained result is highly dependent on the used solver and selected hyperparameters in the case of SPAN and highly dependent on the given dataset in the case of Data-SPAN. Ideally, uncertainty is a byproduct of the method and the uncertainty in the data.

Furthermore, the weighting process raises concerns. The BTC is constrained to a small subset of concentration values compared to the entire field (only 0 to 0.04 instead of 0 to 1.0). This raises questions about the validity of weighting, e.g. $R(c = 0.0)$ samples the same way as $R(c = 1.0)$, as the latter do not directly influence the predicted BTC and thus their likelihood contribution is questionable.

0.6 Conclusion

Although our sensitivity analysis indicated that certain hyperparameters exerted minimal influence on the results, their inclusion does not negatively affect performance. This is because Monte Carlo sampling, the core technique employed here, is inherently independent of dimensionality.

Finally, a comparison of the PIs generated by Full-SPAN and MCMC, specifically the $R(c)$ PI, and their respective likelihoods, reveals comparable performance. Qualitatively, both methods appear to achieve a similar level of accuracy in capturing uncertainty.

0.6.1 Summary

This work investigated UQ for the retardation factor in a diffusion-sorption process using the FINN framework. We proposed two novel UQ methods, SPAN and Data-SPAN, based on perturbing hyperparameters and training data, respectively. Empirical analysis demonstrated the near uniqueness of the inverse problem, enabling interpretation of the UQ results. Although computationally more expensive than MCMC, our methods offer a different perspective on uncertainty by directly exploring the effects of various uncertainties within the training data and solver process.

0.6.2 Future Work

Appendix

0.6.1 Retardation Range

For some c $R(c)$ does not affect the whole c field equally. This can be seen by looking at Figure 20 on page 35 which shows constant retardation factors that are pertubed additively by triangle functions with centers at different c values. With increasing c center, the error caused by the hat functions goes quickly to zero. For $c > 1$, the error becomes zero. This is because the concentration field $c(x, t)$ only contains values between 0 and 1. But there is also a noticeable difference between the maximum error measured on the full field vs. the BTC. The BTC error is consistently much lower. The full field error can not be detected in the case of experimental data as there is no access to the full field solution. For this reason, the range of c values for which significant statements about the uncertainty of

$R(c)$ can be made, is restricted to a narrow range. The exact range is difficult to estimate as the stength of influence for different retardation factors is unclear.

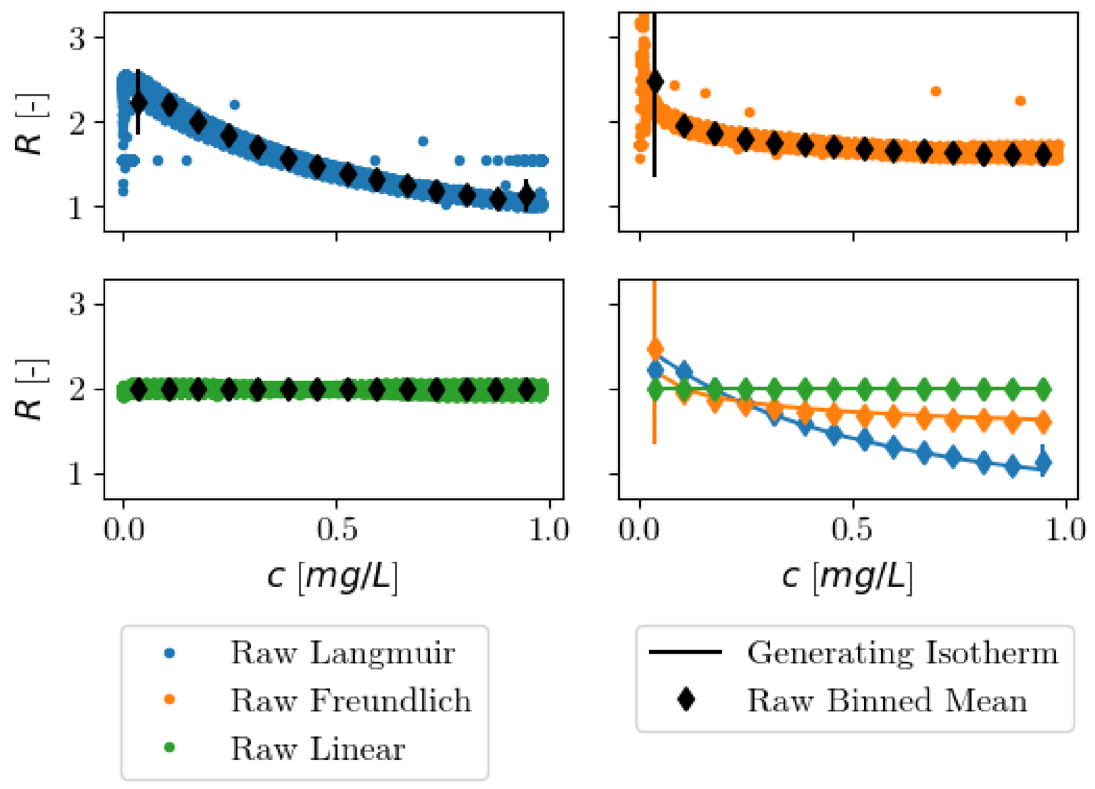

Figure 3: Empirical investigation of the uniqueness of the retardation factor $R(c)$ for three different synthetic datasets generated using the Langmuir, Freundlich, and linear isotherms. The left and top panels show the raw $R(c)$ estimates (colored circles) and the binned and averaged $R(c)$ values (black line with circle markers) as a function of c. The bottom-right panel compares the binned and averaged $R(c)$ values (colored circle markers) with the analytical isotherms (colored line) used to generate the synthetic datasets. Error bars indicate the standard deviation inside the bin.

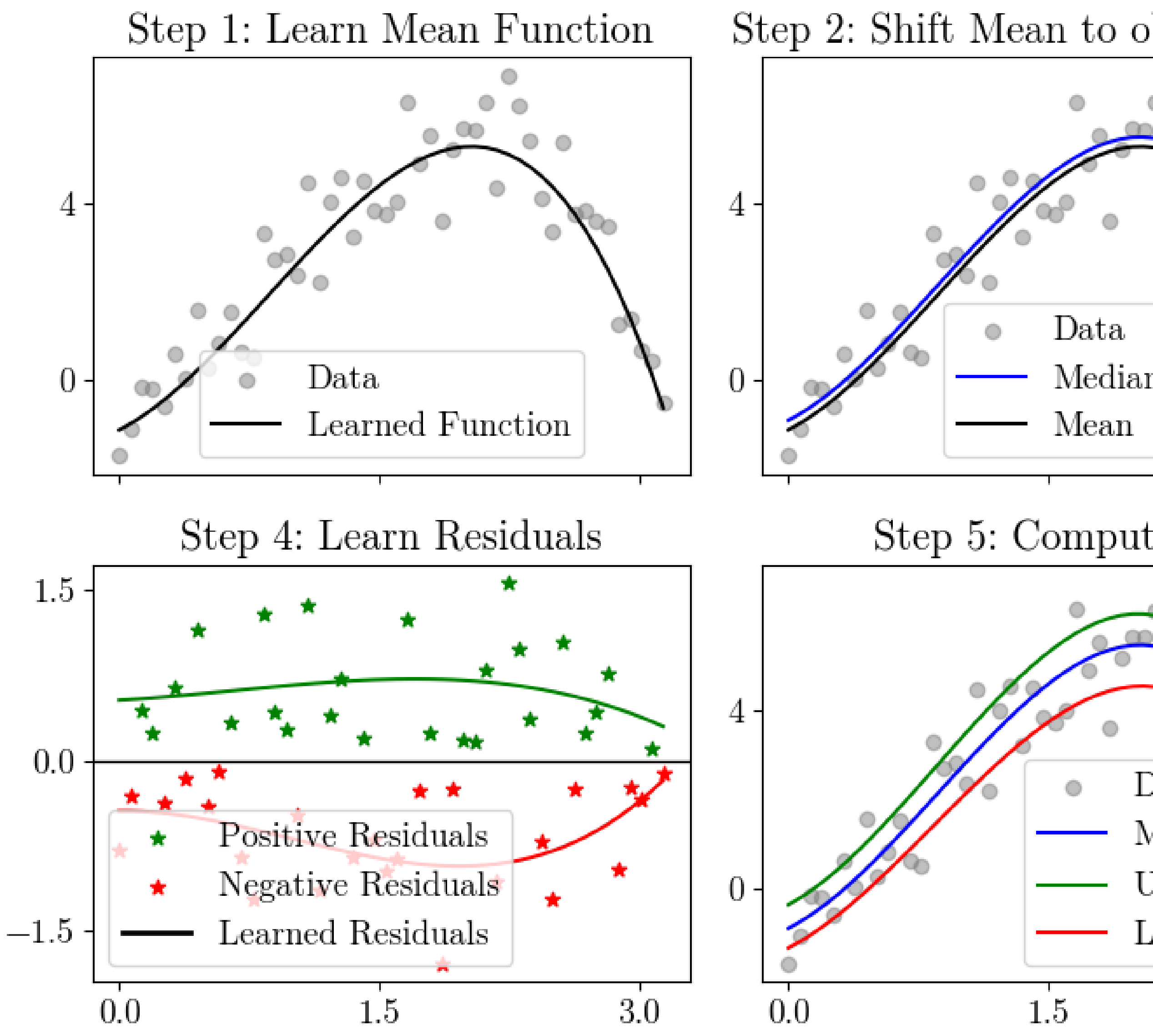

Figure 4: Illustration of the PI3NN method.

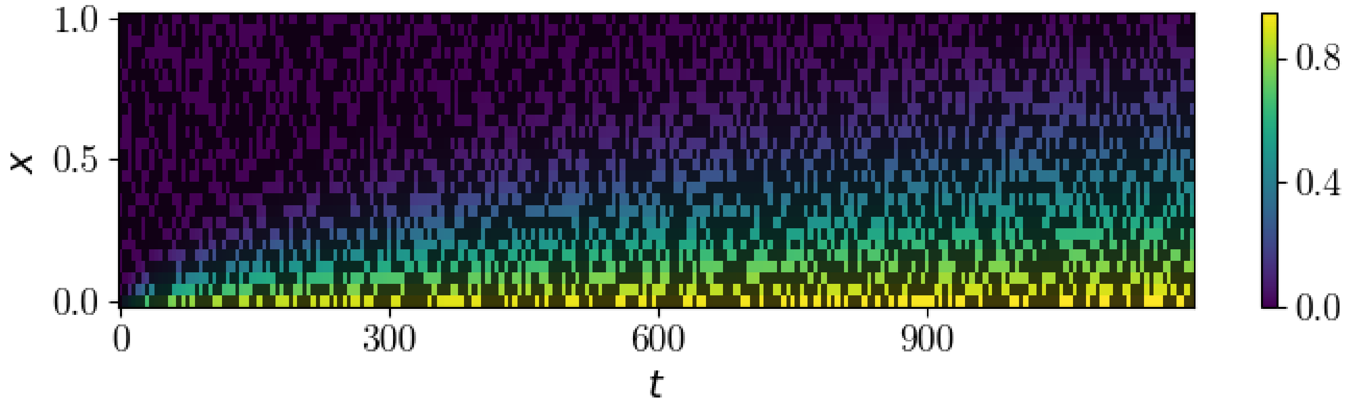

Figure 5: Full synthetic concentration training data where 50% of datapoints are randomly masked (dark patches).

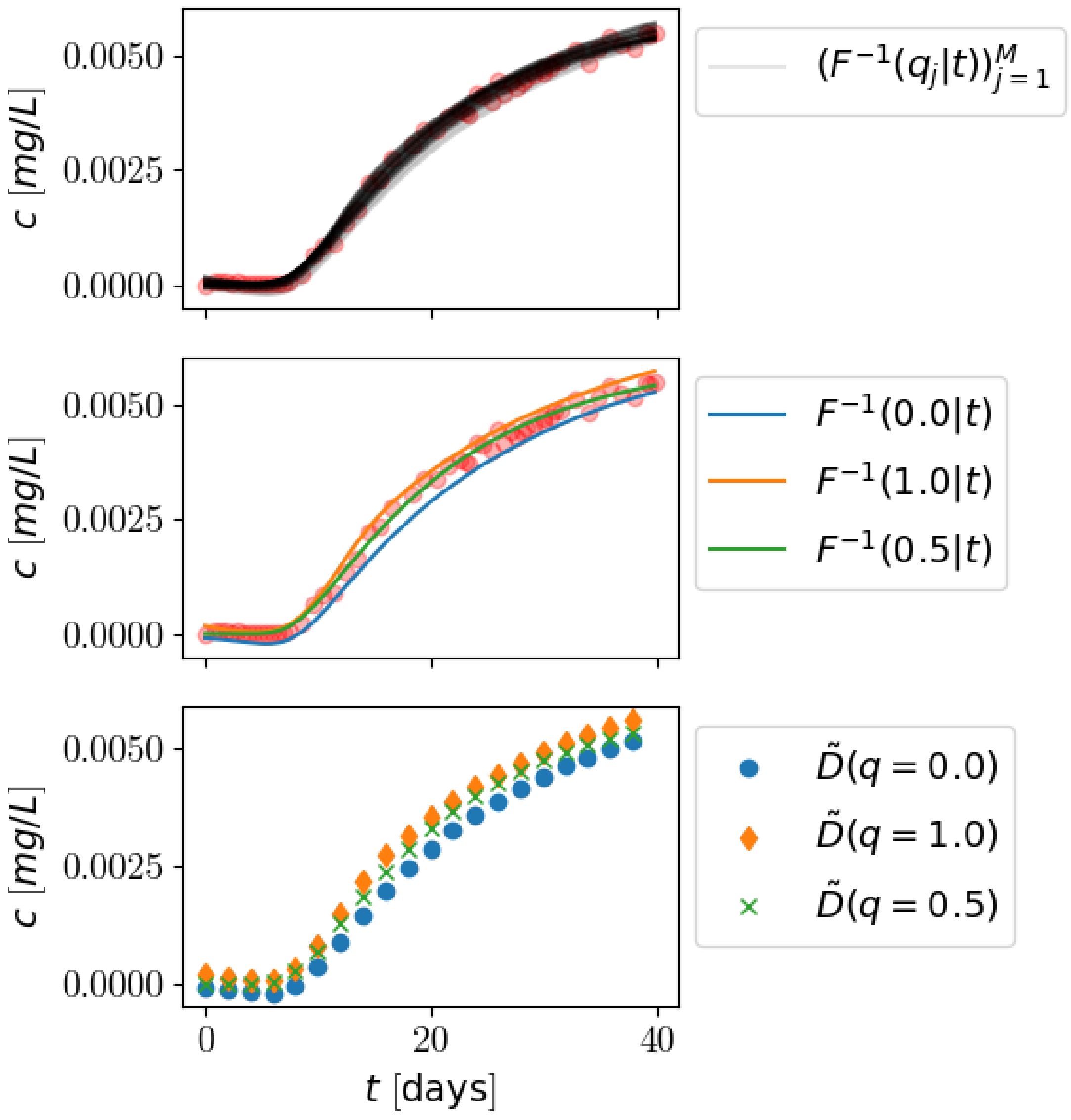

Figure 6: Left: BTC quantile functions for all quantile samples q_j. Middle: BTC quantile functions for quantiles 0, 0.5, and 1. Right: BTC datasets for quantiles 0, 0.5, and 1.

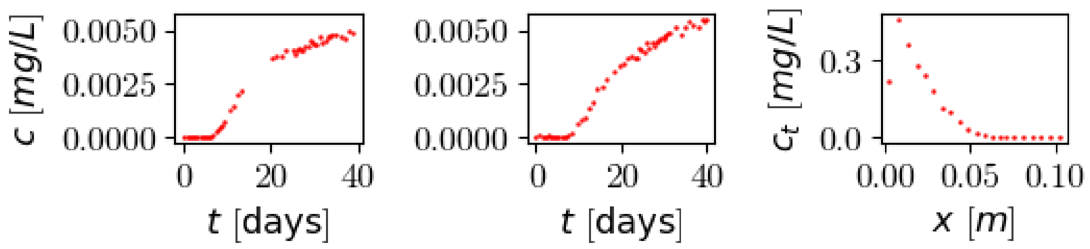

Figure 7: Experimental concentration data obtained from the “Core 1”, “Core 2”, and “Core 2B” sample (left to right) by [8].

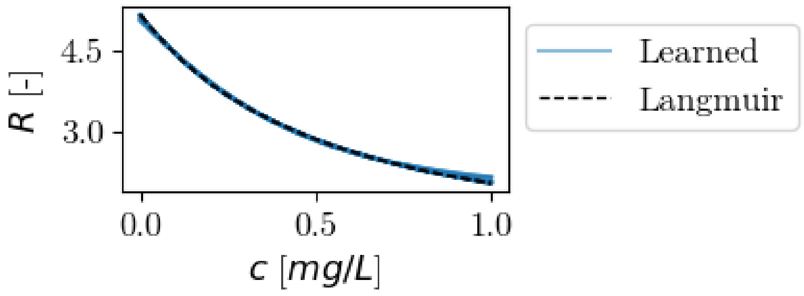

Figure 8: Retardation factors learned by FINN with random weight initialization seed samples and dataset generated by langmuir isotherm.

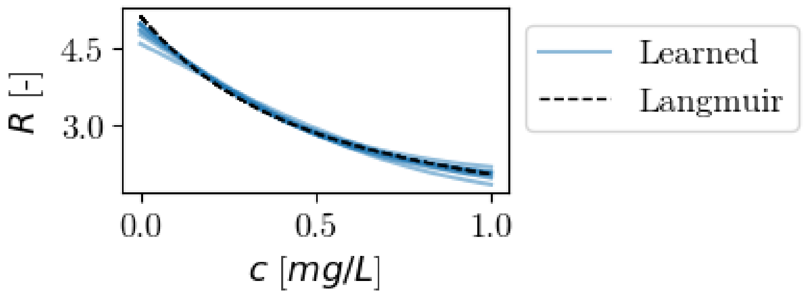

Figure 9: Retardation factors learned by FINN on synthetic dataset (generated by langmuir isotherm) pertubed by gaussian noise.

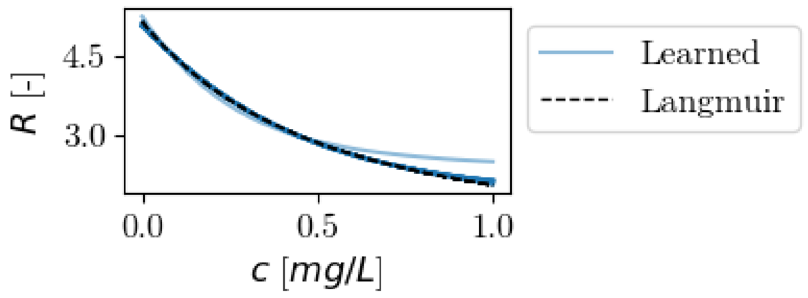

Figure 10: Retardation factors learned by FINN on synthetic dataset (generated by langmuir isotherm) with 50% of training data randomly masked.

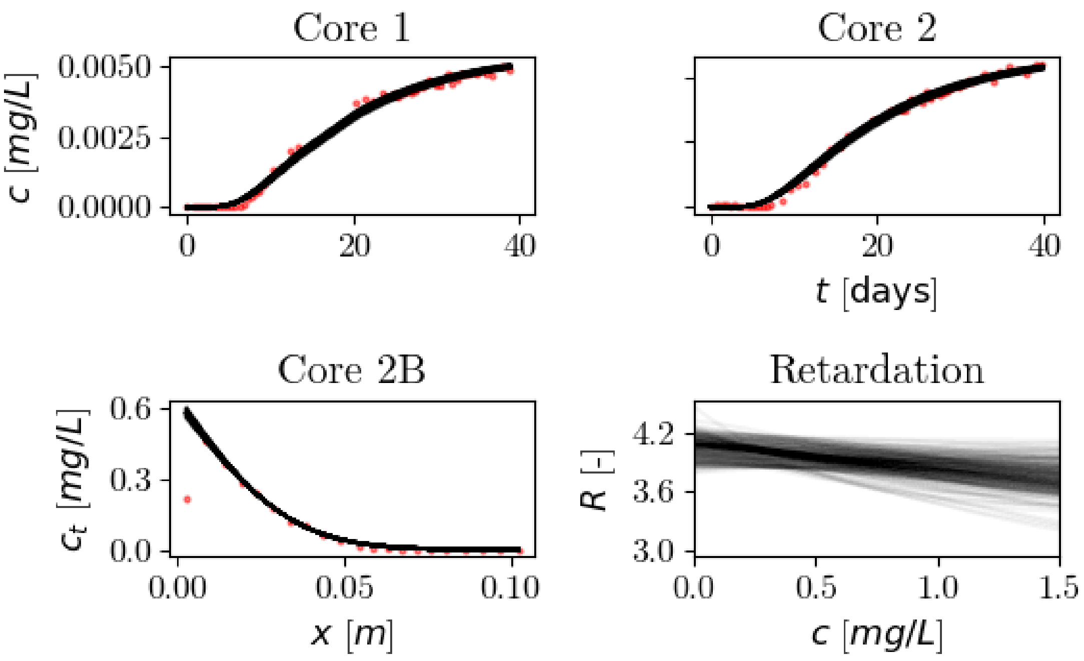

Figure 11: Retardation samples and correspoding concentration curves obtained via FINN by training with random hyperparameters.

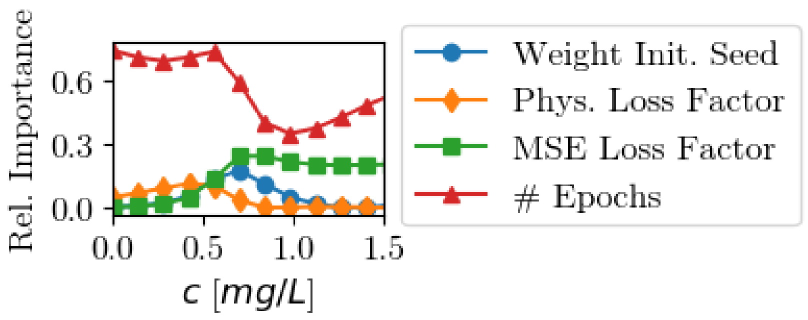

Figure 12: Relative importance on the variance of $R(c)$ of each hyperparameter for different values of c.

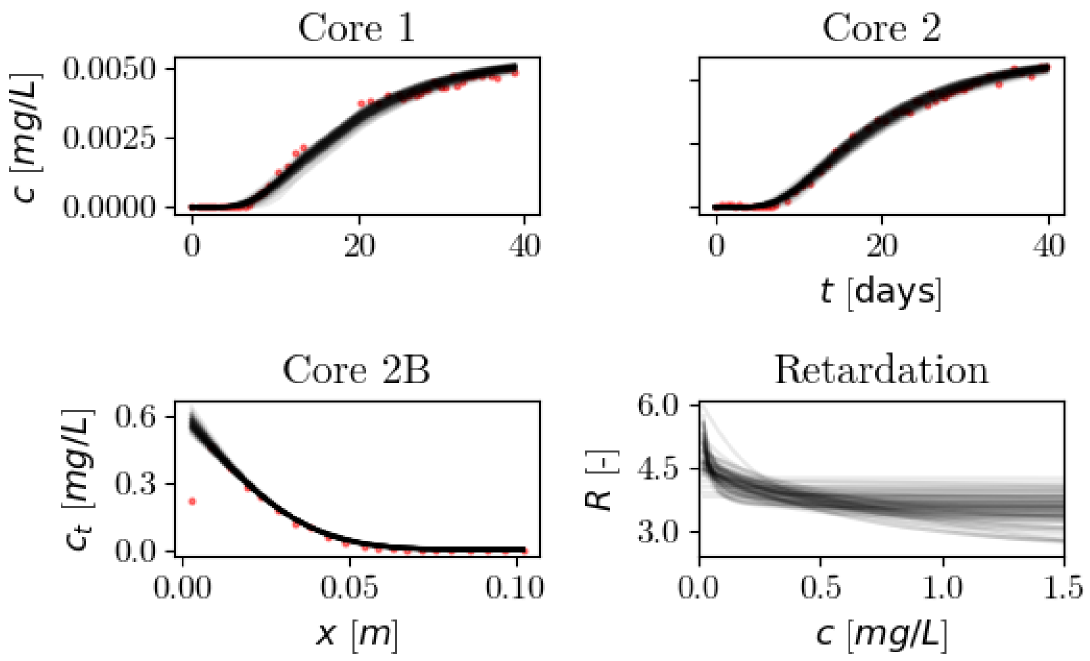

Figure 13: Retardation samples and correspoding concentration curves obtained from it by training FINN with random datasets using PI3NN.

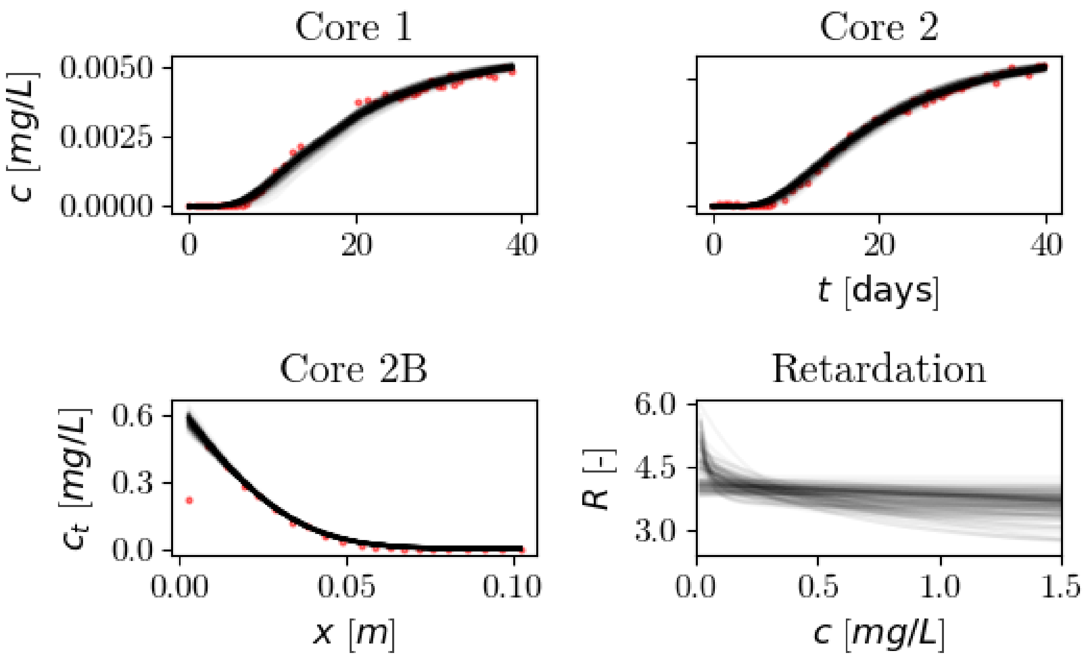

Figure 14: Retardation samples and correspoding concentration curves obtained from it by training FINN with random hyperparameters and random datasets using PI3NN.

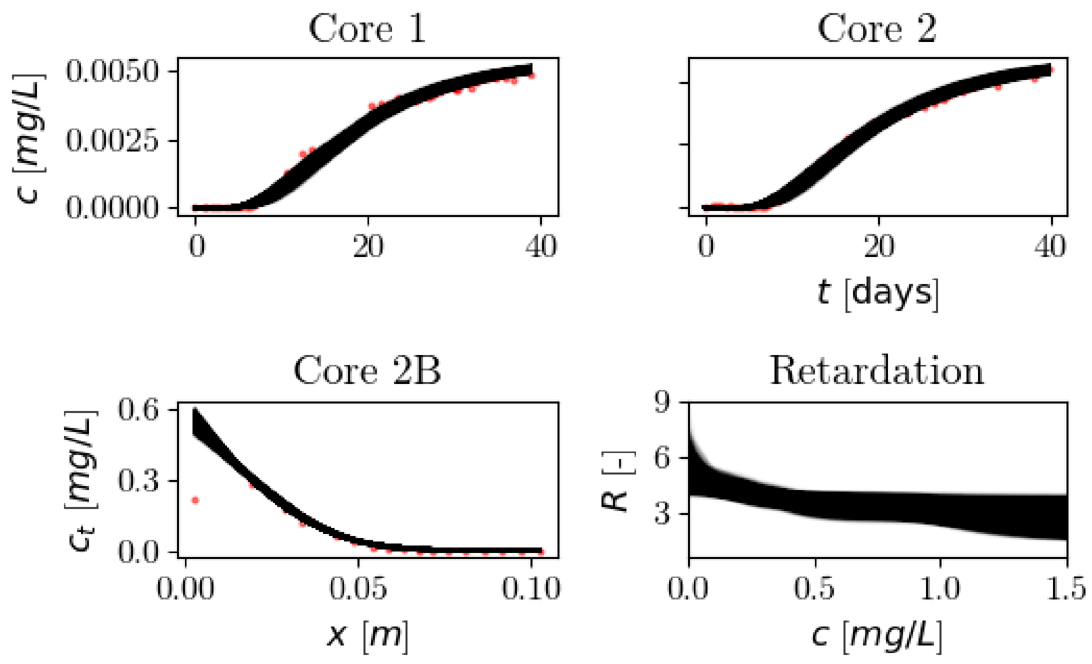

Figure 15: Retardation factors generated by MCMC sampling as detailed in Section 0.2.4 on page 8.

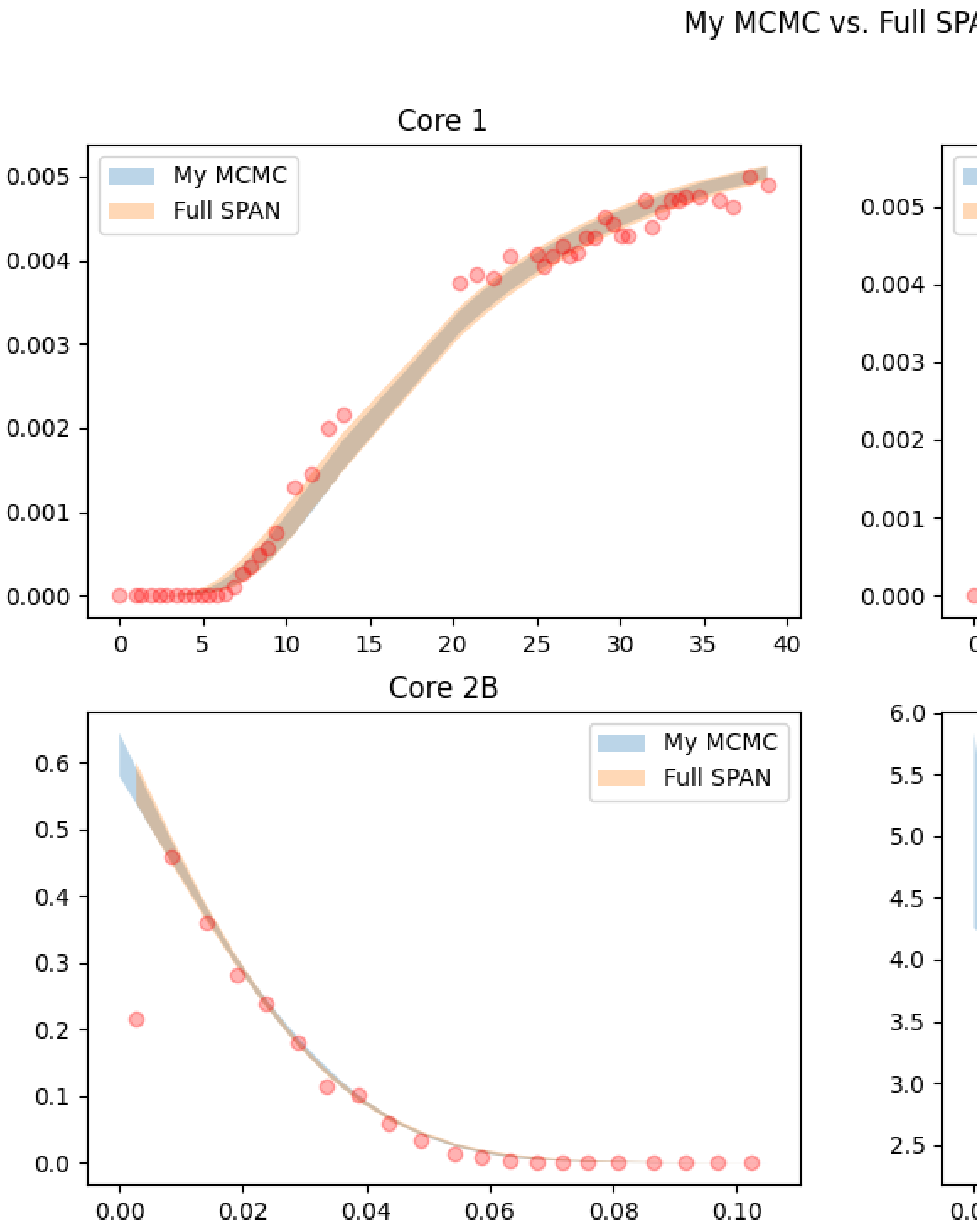

Figure 16: Retardation PIs and correspoding concentration PIs obtained from it by training FINN with random hyperparameters and random datasets using PI3NN (orange). Compared with MCMC approach (blue). (90% PIs are depicted.)

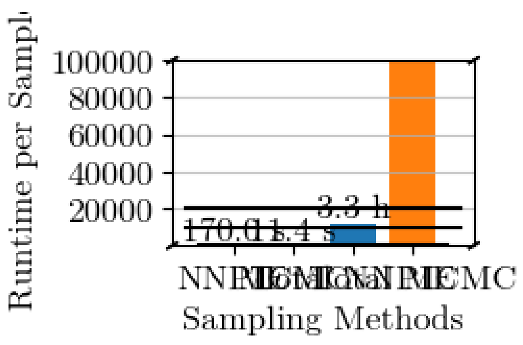

Figure 17: Runtime for generating a single sample (left panel) and all samples (right panel) using SPAN (left bar) and MCMC (right bar). Measured on the same machine and averaged over 10 trials.

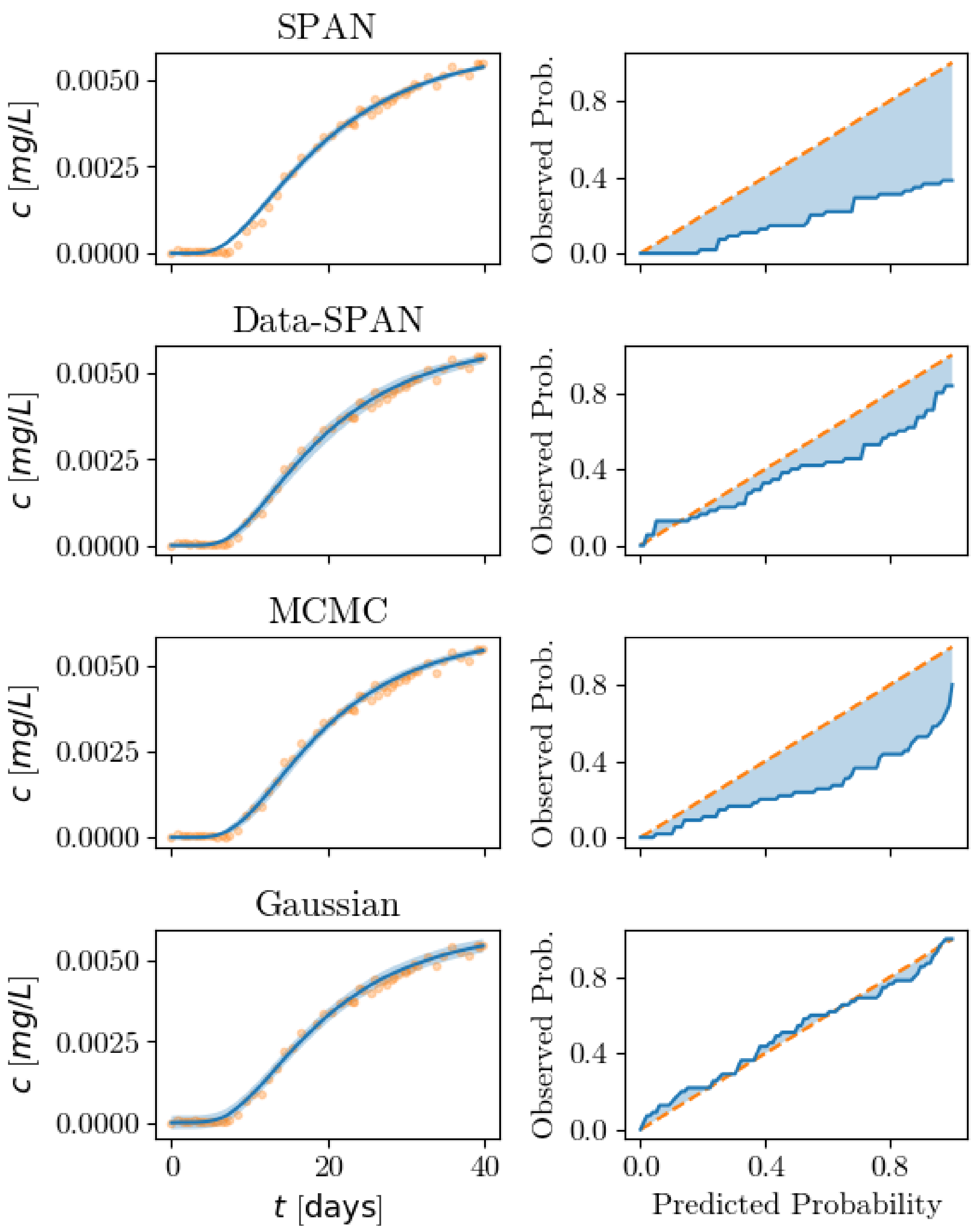

Figure 18: Top: Core 2 BTC 90% PIs for the different methods (SPAN, Data-SPAN, MCMC). Bottom: Reliability curve for each method.

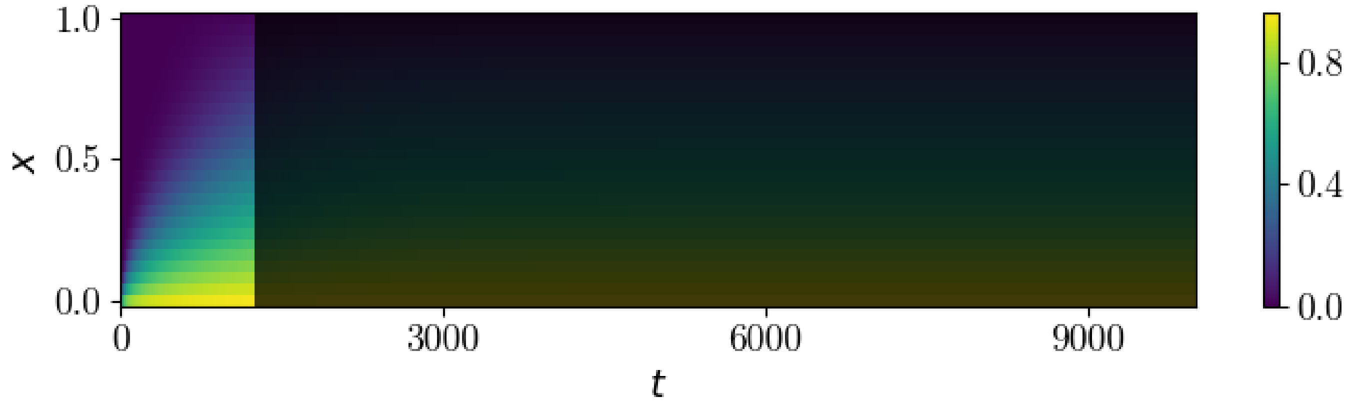

Figure 19: Training data for the model, represented by a pcolor plot of the synthetic concentration field $c(x, t)$ generated using the Langmuir isotherm. The spatial domain is $x \in [0, 1]$ meters, and the temporal domain is $t \in [0, 10000]$ days. The period from $t = 1255$ to $t = 10000$ days is masked, indicating that these values are not used for training but are reserved for testing.

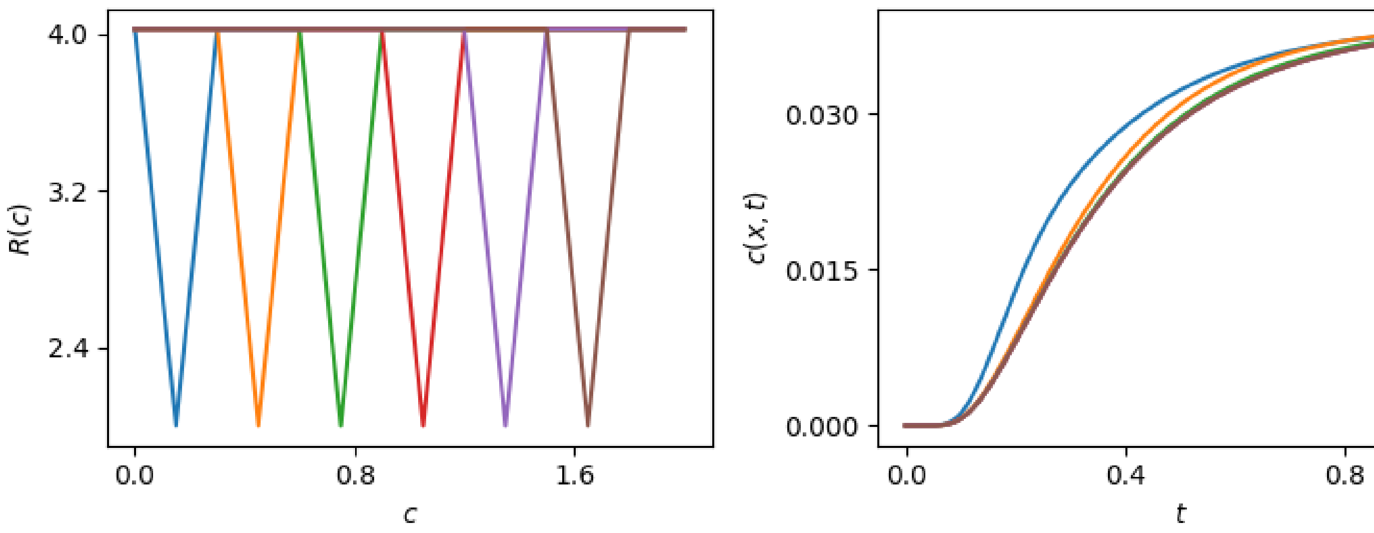

Figure 20: Left: Retardation factors with triangle peaks. Middle: Concentration BTC for these retardation factors. Right: MAE for each on the full field and BTC.

List of Figures

List of Tables

Listings

Bibliography

[1] Timothy Praditia, Matthias Karlbauer, Sebastian Otte, Sergey Oladyshkin, Martin V Butz, and Wolfgang Nowak. Learning groundwater contaminant diffusion-sorption processes with a finite volume neural network. *Water Resources Research*, 58(12):e2022WR033149, 2022.

[2] Rémi Bardenet, Arnaud Doucet, and Chris Holmes. On markov chain monte carlo methods for tall data. *Journal of Machine Learning Research*, 18(47):1–43, 2017.

[3] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations, 2019.

[4] Stefan Depeweg, Jose-Miguel Hernandez-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In *International conference on machine learning*, pages 1184–1193. PMLR, 2018.

[5] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, 56(Suppl 1):1513–1589, 2023.

[6] Siyan Liu, Pei Zhang, Dan Lu, and Guannan Zhang. Pi3nn: Out-of-distribution-aware prediction intervals from three neural networks. *arXiv preprint arXiv:2108.02327*, 2021.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[8] Wolfgang Nowak and Anneli Guthke. Entropy-based experimental design for optimal model discrimination in the geosciences. *Entropy*, 18(11):409, 2016.

[9] Ole Tange. Gnu parallel 20231122 ('grindavík'), November 2023. GNU Parallel is a general parallelizer to run multiple serial command line programs in parallel without changing them.

[10] Robert Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*, 2002.