Composites: Part A 185 (2024) 108355

Contents lists available at ScienceDirect

Composites Part A

journal homepage: www.elsevier.com/locate/compositesa

Real-time Bayesian inversion in resin transfer moulding using neural surrogates

M.E. Causon [a,*], M.A. Iglesias [a], M.Y. Matveev [b], A. Endruweit [b], M.V. Tretyakov [a]

[a] School of Mathematical Sciences, University of Nottingham, UK
[b] Composites Group, Faculty of Engineering, University of Nottingham, UK

ARTICLE INFO

Dataset link: https://doi.org/10.17639/nott.7437

Keywords:
B: Permeability
C: Computational modelling
C: Statistical properties/methods
E: Resin transfer moulding

ABSTRACT

In Resin Transfer Moulding (RTM), local variations in reinforcement properties (porosity and permeability) and the formation of gaps along the reinforcement edges result in non-uniform resin flow patterns, which may cause defects in the produced composite component. The ensemble Kalman inversion (EKI) algorithm has previously been used to invert in-process data to estimate local reinforcement properties. However, implementation of this algorithm in some applications is limited by the requirement to run thousands of computationally expensive resin flow simulations. In this study, a machine learning approach is used to train a surrogate model which can emulate resin flow simulations near-instantaneously. A partition of the flow domain into a low-dimensional representation enables an artificial neural network (ANN) surrogate to make accurate predictions, with a simple architecture. When the ANN is integrated within the EKI algorithm, estimates for local reinforcement permeability and porosity can be achieved in real time, as was verified by virtual and lab experiments. Since EKI utilises the Bayesian framework, estimates are given within confidence intervals and statements can be made on-line regarding the probability of defects within sections of the reinforcement. The proposed framework has shown good predictive capabilities for the set of laboratory experiments and estimates for reinforcement properties were always computed within 1 s.

1. Introduction

Resin transfer moulding (RTM) is a type of liquid composites moulding technique, commonly used for the production of high performance fibre-reinforced composite materials. Desirable due to their high specific stiffness and strength and the ability to form complex shapes, composites underpin the modern aerospace, automotive and marine industries [1]. The RTM process begins by placing a dry fibrous reinforcement (typically consisting of carbon, glass or aramid fibre) within a rigid mould. Liquid thermoset resin is then injected through an inlet, driven by a pressure gradient applied across the tool and it fills the pores of the reinforcement. Upon completion of the injection phase, the composite is left in the tool until curing is complete. The cured part is subsequently demoulded, finished and inspected as to whether its quality is acceptable for use according to its intended purpose, e.g. as an automotive or aircraft part.

Complete impregnation of the reinforcement during the resin injection phase is essential for the quality of the manufactured component. Consequently, there is considerable interest in predicting and controlling resin flow. Flow patterns during injection depend upon the material properties of the reinforcement, such as fibre volume fraction and fibre orientations. Real resin injections often depart from idealised resin flow simulations which do not account for uncertainties introduced throughout the RTM process. In particular, it has been widely observed [2–6] that the material properties of reinforcements are typically not uniform, exhibiting local variations in permeability and porosity. Discrepancies from the design may originate from inherent stochasticity of fibre arrangements in reinforcements, and deformation of reinforcements (mainly in shear) during the forming of three-dimensional shapes.

Due to local variations in reinforcement properties, the resin flow velocity is higher in some regions of the reinforcement than in others, leading to irregular flow front shapes. In extreme cases, resin may flow around a low permeability region and enclose a pocket of air, potentially leading to the formation of resin-free dry spots in the final part. Another undesirable effect in RTM is race-tracking (RT). Gaps formed between the reinforcement and mould cavity edges may occur, which provide low resistance to resin flow (compared to the permeability of the reinforcement) around the edges of the reinforcement. The resulting difference in local resin flow velocities will cause significant distortions

* Corresponding author.
E-mail address: pmymc12@nottingham.ac.uk (M.E. Causon).

https://doi.org/10.1016/j.compositesa.2024.108355
Received 19 April 2024; Received in revised form 21 June 2024; Accepted 6 July 2024
Available online 15 July 2024

in flow front shapes, which makes the flow front propagation difficult to control. RT can vary in strength and may only partially affect edges. Non-uniform resin flow front propagation may result in partially impregnated, i.e. defective parts, and lead to high scrap rates [1,6].

There have been various attempts to either predict or avoid defect formation. Early approaches to detect distorted resin flow patterns used decision trees [7] and genetic algorithms [8,9] to classify different RT phenomena amongst a defined set of scenarios. They successfully detected RT, but relied heavily upon experiments being comparable to the set of simulated scenarios. In cases where experiments deviate, such as the presence of central defects or partial RT, the estimates are likely to be less accurate. This inflexibility was alleviated later when partial RT was introduced [10], however only a small collection of permeability values were considered in the RT region, and deviations from this face the same issue. Another study used a gradient-boosting classifier to detect defects [11]. This work assumed that local permeability variations could be captured by segmenting the domain into 32 rectangular zones, each with its own permeability value. The idea was successful, identifying over 80% of test cases correctly. However, the classifier relied upon a large number of sensors, with a pressure sensor at the centre of each zone. This sensor density is expensive and likely impractical when considering industrial applications. Other approaches view RTM as an inverse problem. In order to understand this perspective, it is first necessary to formally introduce the forward problem.

1.1. Resin flow model — formulation of the forward model

Resin injection is frequently modelled as a single-phase Darcy flow problem [1,12,13], describing flow of a viscous liquid (the resin) through a porous medium with homogenised properties (the reinforcement). The reinforcement occupies the domain D, whose boundary ∂D consists of an inlet ∂D_I, an outlet ∂D_0, and perfectly sealed boundaries ∂D_N (where the normal component of the flow velocity is zero). The reinforcement properties, permeability $\tilde{\mathbf{K}}(\mathbf{x})$ and porosity $\phi(\mathbf{x})$, are spatially varying functions for $\mathbf{x} \in D$.

From the inlet boundary, resin with viscosity μ is injected into the domain D at a constant pressure p_I. The constant pressure on the moving boundary is given by p_0. It is assumed that the process is isothermal and that the resin does not cure during injection, thus μ is constant. During injection, the resin occupies a time-dependent domain $D(t)$, bounded by the inlet ∂D_I, a moving boundary $Y(t)$, and the relevant parts of ∂D_N.

In the fully saturated region of the domain $D(t)$, the phase-averaged flow velocity is given by Darcy's law,

$$\mathbf{v}(\mathbf{x},t) = -\frac{\tilde{\mathbf{K}}(\mathbf{x})}{\mu} \cdot \nabla p(\mathbf{x},t), \tag{1}$$

where p denotes the resin pressure. The resin is assumed incompressible, and thus the continuity equation applies:

$$\nabla \cdot \mathbf{v} = 0. \tag{2}$$

Combining these equations yields

$$-\nabla \cdot \left[\frac{\tilde{\mathbf{K}}(\mathbf{x})}{\mu} \cdot \nabla p(\mathbf{x},t) \right] = 0, \quad \mathbf{x} \in D(t), \tag{3}$$

which is augmented with the following initial conditions and boundary conditions:

$$V(\mathbf{x},t) = \frac{\mathbf{v}(\mathbf{x},t)}{\phi(\mathbf{x})} \cdot \mathbf{n}_Y(\mathbf{x},t), \quad \mathbf{x} \in Y(t), \quad t \geq 0, \tag{4}$$

$$p(\mathbf{x},t) = p_I, \quad \mathbf{x} \in \partial D_I, \quad t \geq 0, \tag{5}$$

$$\nabla p(\mathbf{x},t) \cdot \mathbf{n}(\mathbf{x}) = 0, \quad \mathbf{x} \in \partial D_N, \quad t \geq 0, \tag{6}$$

$$p(\mathbf{x},t) = p_0, \quad \mathbf{x} \in Y(t), \quad p(\mathbf{x},t) = p_0, \quad \mathbf{x} \in \partial D_0, \quad t > 0, \tag{7}$$

$$p(\mathbf{x},0) = p_0, \quad \mathbf{x} \in D, \quad Y(0) = \partial D_I, \tag{8}$$

where $\mathbf{n}_Y(\mathbf{x},t)$ and $\mathbf{n}(\mathbf{x})$ denote the unit outer normals to the moving boundary and the sealed boundaries, respectively, and $V(\mathbf{x},t)$ is the velocity of the moving boundary $Y(t)$ in the direction of $\mathbf{n}_Y(\mathbf{x},t)$. Here, $\tilde{\mathbf{K}}(\mathbf{x})$ and $\phi(\mathbf{x})$ will be treated as independent input parameters. Permeability and porosity are typically assumed to be connected through a geometry factor, which encodes information regarding fibre arrangement, orientation and cross-sectional area. All of these features are to some extent stochastic in nature [14], meaning that the relationship between porosity and permeability is not unique.

In this work, permeability is assumed to be isotropic, so the second-rank permeability tensor, $\tilde{\mathbf{K}}(\mathbf{x})$, can be replaced by the scalar function $K(\mathbf{x})$. This approach can be generalised to the anisotropic case [1,13]. It is also assumed, without loss of generality, that $p_0 = 0$ Pa.

Given a specific tool geometry, the Darcy flow formulation can be viewed as mapping from the known inputs to the solution of the moving boundary problem:

$$\big(K(\mathbf{x}), \phi(\mathbf{x}), \mu, p_I\big) \mapsto \big(p(\mathbf{x},t), Y(t)\big). \tag{9}$$

This is often called the *forward problem* and amounts to solving Eqs. (3) to (8) given the set of inputs, $(K(\mathbf{x}), \phi(\mathbf{x}), \mu, p_I)$. Conversely, the natural *inverse problem* to consider is to estimate $(K(\mathbf{x}), \phi(\mathbf{x}), \mu, p_I)$, given space- and time-discrete observations of $(p(\mathbf{x},t), Y(t))$ which can be collected during the resin injection process (e.g. pressure measurements using sensors and/or flow front tracking). Since μ and p_I are experimentally controlled (and thus known), the inverse problem is posed only in terms of estimating reinforcement properties $K(\mathbf{x})$ and $\phi(\mathbf{x})$. Solving the inverse problem enables the identification and characterisation of potential material defects (i.e. local variations in properties).

1.2. Inferring properties from RTM measurements

It is of particular interest to develop efficient algorithms to solve the inverse problem rapidly [15,16], preferably as resin flows through the reinforcement, i.e. on-line. This would allow to: (i) incorporate active control measures to control the flow front propagation, e.g. varying pressure gradients, with the aim of reducing the risk of defects in the final product, and (ii) pair each RTM injection with a digital twin that can be used for non-destructive evaluation (NDE) or structural health monitoring purposes concurrently with the manufacturing process.

Machine learning approaches have previously been used to rapidly estimate reinforcement permeability during injection. A convolutional neural network (CNN) was trained to address the inverse problem, using pressure measurements recorded during injection to estimate rectangular defects in the reinforcement structure [17]. Though this model recovered rectangular defects almost instantaneously, it struggled to generalise to non-rectangular shapes and was not applicable to cases with multiple defects. RT was addressed separately using a similar routine [18]. Finally, this work was unified by developing a single encoder–decoder model to address both central and RT defects [19]. The model, however, dealt with each defect type separately and could not detect both simultaneously. More recently, the model was used in conjunction with a so-called disturbance detector [20] to form a digital twin which detected disturbances in unidirectional flow fronts, due to RT, in real time.

Elsewhere, disturbances in radial flow fronts have been rapidly characterised by using physics-informed neural networks (PINNs) to invert images taken during injection, before propagating the estimated permeability field through a CNN surrogate forward model to predict the resin front location [21]. A simulation-to-real transfer learning approach has also been employed [22] to enhance CNNs, transformers and convolutional long short-term memory neural networks, trained to predict textile properties using images of the moving front, by enriching models trained on simulations with real data. Other attempts used optimisation procedures to find a permeability function that minimised the difference between observed and predicted (i.e. solving the forward problem) radial resin flow fronts [23] or control volume

arrival times [24]. Whilst successful in many cases, all of the aforementioned techniques produce deterministic estimates and, therefore, fail to address the non-uniqueness property that characterise inverse problems [25,26]: an array of different parameter sets can often feasibly produce the observed data.

The issue of non-uniqueness was addressed successfully [14,27] by presenting the problem within the standard inverse problem framework [25,26]. Using ideas from Bayesian inference, reinforcement porosity and permeability were estimated probabilistically, using data from pressure transducers and images of the resin front. Crucially, the ensemble Kalman inversion (EKI) algorithm [14] used was able to accurately detect defects in the structure of flat or preformed (in 3D) reinforcements without making assumptions about the shape, position or number of defects, and demonstrated success on both virtual and lab experiments. However, estimating reinforcement properties using EKI requires solving the forward problem, which can take minutes to compute, hundreds or even thousands of times with different input parameters. This prohibits real-time estimation of the algorithm, in spite of efforts to exploit its parallelisable nature.

1.3. Contributions

So far, solving the inverse problem has been attempted using machine learning models [17–19], which are fast but lack the mathematical rigour required for such ill-posed problems, and which can lead to unstable estimates when used outside of the setting in which those models are trained. On the other hand, posing the problem within a more suitable Bayesian inversion framework [14,27] addresses the ill-posedness of inverse problems, but the framework itself has so far prohibited on-line parameter estimation. In this study, the problem is viewed from the Bayesian perspective, but the aim is to solve it in real time. As such, the use of simple artificial neural network (ANN) surrogate models will be explored to accelerate the most time-consuming aspect of Bayesian inversion: simulating the forward problem. Unlike previous work [14,27], where the functions $K(\mathbf{x})$ and $\phi(\mathbf{x})$ are inferred on every point $\mathbf{x}$ of a fine computational mesh (thus leading to a high-dimensional inference problem), reinforcement properties are instead parameterised within a low-dimensional input space. More specifically, the tool domain is partitioned into rectangular sub-regions in order to capture local heterogeneities in permeability and porosity, reminiscent of the work by Mendikute et al. [11]. Here, however, a higher density of sub-regions is used with fewer pressure sensors. Moreover, sub-regions are allocated specifically in RT locations which enables the discovery of central and RT defects, simultaneously. To avoid the surrogate leading to less accurate estimates for permeability and porosity, modelling errors incurred by the surrogate are incorporated within the inversion process. The proposed surrogate-based Bayesian inversion approach is successfully tested using both synthetic and lab experiment data, demonstrating the power of the carefully designed surrogate in allowing the real-time estimation of reinforcement permeability and porosity.

2. Bayesian inversion methodology

2.1. Partitioning the domain

A low-dimensional parameterisation of the unknown properties $K(\mathbf{x})$ and $\phi(\mathbf{x})$ is employed by defining a partition, in terms of a finite number of N_r sub-regions $[R_1, \dots, R_{N_r}]$. It is assumed that on each sub-region, R_i, $K(\mathbf{x})$ and $\phi(\mathbf{x})$ take constant values K_i and ϕ_i, respectively. In other words,

$$K(\mathbf{x}) = \sum_{i=1}^{N_r} K_i \mathbb{I}_{R_i}(\mathbf{x}), \qquad \phi(\mathbf{x}) = \sum_{i=1}^{N_r} \phi_i \mathbb{I}_{R_i}(\mathbf{x}), \tag{10}$$

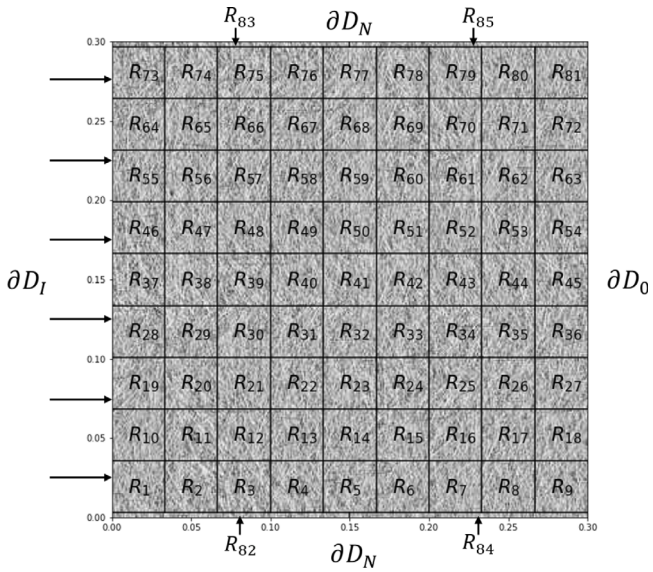

Fig. 1. Partition of the tool domain into 85 sub-regions: 81 central and 4 RT. Also labelled are the inlet ∂D_I, outlet ∂D_0 and no-flow boundaries ∂D_N.

where $\mathbb{I}_{R_i}(\mathbf{x})$ denotes the indicator function of R_i (i.e., $\mathbb{I}_{R_i}(\mathbf{x}) = 1$ if $\mathbf{x} \in R_i$ and $\mathbb{I}_{R_i}(\mathbf{x}) = 0$ otherwise). Eq. (10) delivers a parameterisation of $K(\mathbf{x})$ and $\phi(\mathbf{x})$ in terms of the following vectors:

$$\mathbf{K} := [K_1, \dots, K_{N_r}], \qquad \boldsymbol{\phi} := [\phi_1, \dots, \phi_{N_r}]. \tag{11}$$

This approach can easily be extended to the anisotropic case by replacing the scalar permeability value assigned to each region by elements of a permeability tensor, defined in an analogous manner. However, estimating more unknowns may require more data, either through more pressure sensors or images of the resin front.

In each of the following examples, D is a square domain of dimensions 300 mm × 300 mm, where a linear resin inlet ∂D_I and a linear vent ∂D_0 are placed on opposite edges of the domain. The domain is split into $N_r = 85$ sub-regions, which are labelled in Fig. 1. Two regions are assumed to run along the top and bottom edges of the reinforcement, protruding 2 mm into the domain. In order to identify partial RT, each region is bisected vertically, creating 4 total RT sub-regions of dimension 150 mm × 2 mm. The purpose of these sub-regions is to identify *effective* porosity and permeability values, which average over properties near the edge of the reinforcement and properties within the RT-causing gaps (should they exist). The remainder of the reinforcement, framed above and below by the RT sub-regions, are split equally into a 9 × 9 grid consisting of 33.33 mm × 32.89 mm sub-regions.

2.2. The forward map

The inputs of the forward problem in Eq. (9) can be written using the parameterisation introduced in Eq. (11):

$$\mathbf{u} := (\mathbf{K}, \boldsymbol{\phi}, \mu, p_I). \tag{12}$$

As stated in Section 1.1, the forward problem involves finding the solution, $(p(\mathbf{x}, t), Y(t))$, to the moving boundary problem (3)–(8) for a given set of inputs. Here, the forward problem is solved numerically using an implementation [28] of the Control Volume Finite Element Method (CVFEM) [1,13,29]. The forward map $\mathcal{F}(\mathbf{u})$ between the input $\mathbf{u}$ and the output – the resin pressure evaluated at specific points in space (sensor locations) and time – is defined as:

$$\mathcal{F}(\mathbf{u}) := \Big\{ \{p(\mathbf{x}_1, t_1) \dots, p(\mathbf{x}_M, t_1)\}, \dots, \{p(\mathbf{x}_1, t_N) \dots, p(\mathbf{x}_M, t_N)\} \Big\}, \tag{13}$$

where M is the total number of pressure sensors with locations $\{\mathbf{x}_m\}_{m=1}^{M} \subset D$, and N is the number of observation times denoted by $\{t_n\}_{n=1}^{N}$.

2.3. The Bayesian approach

The Bayesian approach to inverse problems consists of three elements. Firstly, an initial guess - or, in Bayesian Statistics terminology, a *prior* distribution - is provided. In the context of RTM, this guess encompasses prior knowledge of reinforcement porosity and permeability, for example the expected target values and the natural variability according to the design. A lack of confidence in the design may be reflected by choosing a prior with large variance. The probability density of the prior distribution is denoted by $\mathbb{P}(\mathbf{K}, \boldsymbol{\phi})$.

Secondly, an observational model that relates the in-process pressure measurements D and the unknown inputs $(\mathbf{K}, \boldsymbol{\phi})$ must be adopted. Following standard assumptions [14,26,27,30], the observed measurements D are assumed to be a realisation of the random variable:

$$D = \mathcal{F}(\mathbf{u}) + \eta, \tag{14}$$

where η is measurement noise, reflecting the sensors' accuracy and uncertainty. It is assumed that η comes from a Gaussian distribution with zero mean and, for simplicity, known diagonal covariance matrix Γ, i.e. $\eta \sim N(0, \Gamma)$. It is possible to modify this approach to incorporate dependencies within Γ, if known. From Eq. (14), it follows that the *likelihood* function, $D|\mathbf{u}$, is Gaussian with mean $\mathcal{F}(\mathbf{u})$ and covariance Γ, with probability density

$$\mathbb{P}(D|\mathbf{u}) \propto \exp\left(-\frac{1}{2}\left\|\Gamma^{-1/2}(D - \mathcal{F}(\mathbf{u}))\right\|^2\right). \tag{15}$$

The third and final element of the Bayesian approach is the *posterior* distribution which comprises of the probabilistic knowledge for reinforcement porosity and permeability given the measurements, balancing both prior and data uncertainty. From Bayes' rule and Eq. (15), the posterior probability density, denoted by $\mathbb{P}(\mathbf{K}, \boldsymbol{\phi}|D, \mu, p_I)$, is given by

$$\mathbb{P}(\mathbf{K}, \boldsymbol{\phi}|D, \mu, p_I) = \frac{1}{Z}\mathbb{P}(\mathbf{K}, \boldsymbol{\phi}) \exp\left(-\frac{1}{2}\left\|\Gamma^{-1/2}(D - \mathcal{F}(\mathbf{u}))\right\|^2\right), \tag{16}$$

where Z is the normalising constant.

Since the forward map $\mathcal{F}(\mathbf{u})$ is nonlinear, no analytical expression is available for the normalisation constant Z. On the other hand, numerical integration becomes futile for high-dimensional problems (here $N_r = 85$). Therefore, the posterior can only be characterised using sampling algorithms such as Markov chain Monte Carlo (MCMC) [31] or Sequential Monte Carlo (SMC) methods which avoid the computation of Z altogether. Once a large number of samples from the posterior are computed, quantities such as the mean, variance and credible intervals can be approximated via Monte Carlo estimation.

The main computational bottleneck of MCMC and, in general, any sampling methodology lies in the evaluation of the likelihood given by Eq. (15) and thus the evaluation of the forward map $\mathcal{F}(\mathbf{u})$ which is required for the computation of each sample along an MCMC chain. Each of these evaluations involves solving the moving boundary problem in Eqs. (3)–(8) via CVFEM, which takes minutes. This makes MCMC algorithms infeasible for inference problems of such dimension, which often require hundreds of thousands or even millions of samples to achieve significant decorrelation to enable accurate estimates of posterior quantities. While more sophisticated implementations of MCMC can reduce the computational burden via parallel computing, these are only substantially beneficial when vast computational resources from High Performance Computing (HPC) facilities are available. Furthermore, assessing the convergence of these methods is not straightforward and often requires a substantial level of expertise to determine optimal tuning parameters.

The above limitations for approximating the posterior motivate the need for a surrogate of the forward map that, once trained offline, can deliver sub-second approximations of the forward map, thus expediting the overall computations of the chosen sampling algorithm. However, even when using a fast surrogate, sampling algorithms such as MCMC cannot be expected to produce accurate approximations of the posterior in real time. Instead, the present work combines the ensemble Kalman inversion (EKI) algorithm [30] with a parameterisation of the unknown variable $(\mathbf{K}, \boldsymbol{\phi})$ that enables one to enforce constraints on the admissible (i.e. physically realistic) values that the unknown variables can take. The proposed EKI algorithm can provide an approximation of the posterior with a considerably smaller number of samples (e.g. in the order of hundreds). Moreover, the EKI algorithm is simple to implement and requires no tuning to produce stable and accurate estimates of the posterior.

2.4. Ensemble Kalman inversion

The employed EKI algorithm, shown in Appendix A, starts with an ensemble of J samples (called particles), $\{(\mathbf{K}_0^{(j)}, \boldsymbol{\phi}_0^{(j)})\}_{j=1}^J$, from the prior $\mathbb{P}(\mathbf{K}, \boldsymbol{\phi})$. The goal is to iteratively update each particle in a smooth manner controlled by a regularisation parameter α_n. Upon converging after q iterations, the final ensemble provides an approximation of the posterior.

The prior ensemble is generated so that the permeability and porosity values in central sub-regions lie within $[K_{\min}, K_{\max}]$ and $[\phi_{\min}, \phi_{\max}]$, respectively. The upper bounds are extended in RT sub-regions to $K_{\max}^{\mathrm{RT}}$ and $\phi_{\max}^{\mathrm{RT}}$, respectively. To preserve these bounds, a parameterisation $\mathcal{P}(\boldsymbol{\theta}) = (\mathbf{K}, \boldsymbol{\phi})$ is introduced as described in Appendix A.

The EKI algorithm [30] is applied to the unconstrained variable $\boldsymbol{\theta}$ with the initial ensemble $\boldsymbol{\theta}_0^{(j)} = \mathcal{P}^{-1}(\mathbf{K}_0^{(j)}, \boldsymbol{\phi}_0^{(j)})$. At the nth iteration of the EKI algorithm, the ensemble $\{\boldsymbol{\theta}_n^{(j)}\}_{j=1}^J$, consists of samples from a Gaussian $\mathbb{P}_n(\boldsymbol{\theta})$. The ensemble of transformed particles $(\mathbf{K}_n^{(j)}, \boldsymbol{\phi}_n^{(j)}) = \mathcal{P}(\boldsymbol{\theta}_n^{(j)})$ approximate an intermediate distribution on $(\mathbf{K}, \boldsymbol{\phi})$ from a sequence of distributions that gradually transitions from prior to posterior $\mathbb{P}(\mathbf{K}, \boldsymbol{\phi}|D, \mu, p_I)$ over q iterations.

In the EKI algorithm [30], the number of iterations q is controlled by the selection of α_n which is done automatically without the need for user defined inputs. Although the optimal choice of particles J is an open problem, with only a few hundred particles, the EKI algorithm can achieve estimates of the posterior (and therefore its mean and variance) of the reinforcement permeability with accuracy comparable to SMC but with two orders of magnitude reduction in the computational cost [27]. Despite the computational advantages of EKI with respect to other methods, its computational cost is owed to requiring J evaluations of the forward map $\{\mathcal{F}(\mathbf{u}_n^{(j)})\}_{j=1}^J$ (see Algorithm 1 in Appendix A) at each iteration of the algorithm, which cannot feasibly be achieved for on-line applications when conventional solvers are used for evaluating the forward map.

3. Surrogate modelling

One of the underpinning principles of supervised machine learning is to learn an input to output map using data. Recently, the simulation of partial differential equations (PDEs) – which in its most primitive form is an input to output map – has been posed as a machine learning problem [32]. The data are generated by running the computationally expensive PDE solver for a large number of parameter sets and collecting the outputs. This data set is receptive to a vast array of machine learning techniques, most prominently artificial neural networks (ANNs) and Gaussian processes (GPs), which can emulate the PDE solver; the model that emulates the solver is called the *surrogate*. Compared to the computationally expensive PDE solver, the data driven surrogate model interpolates outputs for unseen parameter sets near-instantaneously. Model order reduction techniques, such as Proper Generalised Decomposition [33], are not considered here but also offer fast simulations of expensive, high-dimensional forward models.

GPs are a popular choice of surrogate model [34,35], which use ideas from Bayesian inference to provide probabilistic estimates of the output. Probabilistic estimates are beneficial, particularly when interpolating over a sparse data set; either when there are few data

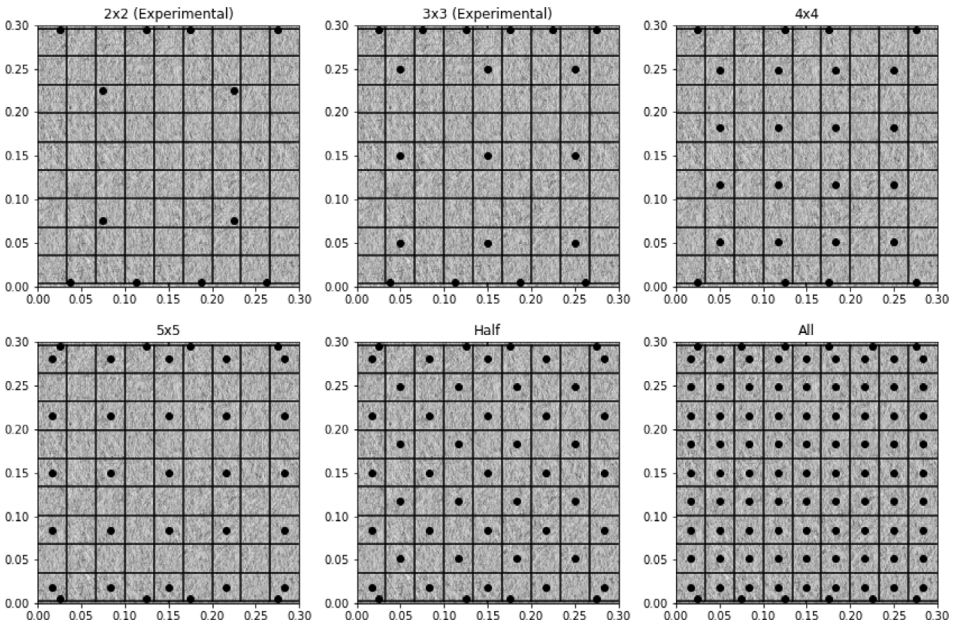

Fig. 2. Various sensor configurations of increasing density, along with the partition described in Section 2.1. The total sensors used by each configuration are as follows: $M = 12$ (2×2), $M = 19$ (3×3), $M = 24$ (4×4), $M = 33$ (5×5), $M = 49$ (Half), $M = 93$ (All).

available, or when the parameter space is high-dimensional. However, very high-dimensional problems like the one considered in this paper suffer from the curse of dimensionality [36] and thus require large data sets for accurate predictions. Coupled with the non-parametric nature of GPs (the size of the model increases with the size of the data), computer memory management soon becomes impractical even on powerful GPUs, where training and inference are usually performed, since the burden of the computational complexity is so high that CPUs (due to their architecture) are impractical for the task. Approximations, such as the use of inducing points [37,38], can, to some extent, relieve memory issues but often to the detriment of accuracy and uncertainty calibration on validation sets.

ANNs are another common approach in surrogate modelling [17–21]. Since ANNs are parametric, they are far more efficient than GPs for large and high-dimensional data sets. Recently, PINNs have demonstrated significant predictive power [21,39], however the design of the cost function and model architecture often poses a substantial challenge. CNNs [21,40] and encoder–decoders [19,20] have successfully been applied to create surrogate models but, like PINNs, can be highly complex. The model architecture can require significant experimentation in order to choose the number and combination of layers (e.g. convolutional, pooling and fully connected), along with the size/type of kernels and stride length used. Here, standard feedforward ANNs trained on simulated data are considered, which are straightforward to implement and can therefore easily be replicated in many experimental settings. Compared to more complex models, they are also more efficient to evaluate, and require far less computational overhead to train.

3.1. Building the ANN surrogate

The general premise of feedforward ANNs is to connect the input nodes to the output nodes via a series of node-containing hidden layers. Connections between nodes in adjacent layers are weighted randomly initially, before being learnt using training data with optimisation procedures (e.g., stochastic gradient descent or the Adam algorithm [36,40]). Weights are adjusted to minimise a selected loss function, using gradients, in a process called backpropagation. There is a limited literature regarding the optimal architecture for ANNs, along with the choice of activation functions, so the final model is often selected through trial and error by assessing performance on a validation data set (or by using cross-validation).

Table 1
Bounds with which the surrogate is trained on.

Parameter	Range
$[K_{\min}, K_{\max}]$	$[2\times10^{-10}, 7\times10^{-10}]$ mm^2
$[\phi_{\min}, \phi_{\max}]$	$[0.40, 0.85]$
$K^{\mathrm{RT}}_{\max}$	500×10^{-10} mm^2
$\phi^{\mathrm{RT}}_{\max}$	0.9882
μ	$[0.099, 0.11]$ Pa s
p_I	$[90, 110]$ kPa

The first step in building the ANN surrogate of the forward map $\mathcal{F}(\mathbf{u})$ is to compute the input and output sets, X and Y, defined by

$$X = \{\mathbf{u}_1, \dots, \mathbf{u}_{N_s}\}, \quad Y = \{\mathcal{F}(\mathbf{u}_1), \dots, \mathcal{F}(\mathbf{u}_{N_s})\}. \tag{17}$$

From the definition in Eq. (12), each of the N_s inputs takes the form $\mathbf{u}_i = (\mathbf{K}_i, \boldsymbol{\phi}_i, \mu_i, (p_I)_i)$. The parameterisation introduced in Appendix A ensures that the ensemble members always produce permeability and porosity estimates in central sub-regions within $[K_{\min}, K_{\max}]$ and $[\phi_{\min}, \phi_{\max}]$, respectively, with extended upper bounds in RT sub-regions, denoted by $K^{\mathrm{RT}}_{\max}$ and $\phi^{\mathrm{RT}}_{\min}$. Hence, one need only train the surrogate to interpolate accurately within these bounds for $\mathbf{K}$ and $\boldsymbol{\phi}$. Even though the pair (μ, p_I) is known from the perspective of the inverse problem, the surrogate for the forward map is built to account for small variations from their target values, since they vary between the lab experiments described in Section 5; it is essential to build a flexible surrogate which is robust to this variability in order to avoid retraining the surrogate every time the experimental conditions change.

In order to efficiently explore the high-dimensional space of inputs, X, a Sobol sequence [41] is first used to generate $N_s = 50{,}000$ points. According to the bounds in Table 1, these points undergo an affine transformation to generate the corresponding elements of the input set X. The bounds for permeability and porosity are selected in accordance with the lab experiments, and ensure that the surrogate interpolates accurately for both target values and values corresponding to defects for the reinforcement properties. These bounds can be adapted according to a particular experimental setting, but it must be ensured that

the model is presented with the most extreme (yet possible) defects during the training process. The bounds for the inlet pressure and resin viscosity are selected so that the ANN is able to accommodate a ±10% deviation from the intended experimental conditions of $p_I = 100$ kPa and $\mu = 0.1$ Pa s. More general surrogates which interpolate over wider intervals can be used, but require more data to populate the enlarged parameter space.

For each input $\mathbf{u}_i$, the output $\mathcal{F}(\mathbf{u}_i)$ is computed via Eq. (13), where $p(\mathbf{x}, t)$ is the solution to Eqs. (3)–(8) discretised via CVFEM on a highly refined computational mesh that consists of 4096 cells. In order to resolve the narrow RT channels more precisely, the mesh has greater refinement in the RT regions. The numerical solution is then evaluated at $M = 116$ locations, corresponding to all of the sensors used for the various sensor configurations shown in Fig. 2. These configurations will be used for both virtual and laboratory experiments. Only the 2×2 and 3×3 configurations are available in the lab experiments outlined in Section 5. At each sensor location, $\mathbf{x}_m$, the numerical approximation is evaluated at the following $N = 14$ observation times:

$$(t_1, \ldots, t_{14}) = (1, 3, 5, 7, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55) \text{ s}. \tag{18}$$

The final observation time was selected by conducting several lab experiments on a nominal reinforcement, described later in Section 5.1. Prior knowledge regarding the approximate filling time is beneficial in characterising the properties of the entire reinforcement; estimates will be provided only within the resin impregnated region at the final observation time, which is particularly problematic if the true filling time is far longer. The intermediate observation times are chosen such that increments in the resin front are approximately uniform. The workload for the computations of the set of outputs Y was distributed over 35 cores of a Supermicro 620U Linux RHEL8.8 server with 48 Intel Xeon (Ice Lake class) CPUs and 1.5Tb RAM and took approximately 67 CPU-hours to run.

3.2. Training and performance

The ANN architecture consists of 172 input neurons: 85 permeability-porosity pairs, inlet pressure and resin viscosity. For the purpose of training, each component of the input is normalised into $[0, 1]$. The unpacking and normalisation processes are shown in Fig. 3. Since each of the $M = 116$ sensors measure pressure at $N = 14$ observation times, there is a total of 1624 output neurons. To ensure positivity in pressure predictions, each output neuron is parameterised before training with $\tilde{p}(\mathbf{x}_m, t_n) = \log\big(p(\mathbf{x}_m, t_n) + 1000\big)$, where $p(\mathbf{x}_m, t_n)$ is the pressure at a generic output neuron. Once trained, the surrogate predicts each $\tilde{p}(\mathbf{x}_m, t_n)$, before being converted to positive $p(\mathbf{x}_m, t_n)$ by reversing the parameterisation.

One thousand samples from X and Y are assigned to testing sets, with input and output denoted by X_{test} and Y_{test}, respectively. Upon training the surrogate, the testing set will be used to provide Monte Carlo estimates for the accuracy of the solution to the inverse problem. The remaining data, used to generate the surrogate model, are split 70:30 between training $(X_{\text{train}}, Y_{\text{train}})$ and validation $(X_{\text{val}}, Y_{\text{val}})$ sets.

In order to demonstrate the simplicity and flexibility of this approach, a minimalist architecture is considered with only one hidden layer, accompanied by a Sigmoid activation function as shown in Fig. 3. Using the mean-squared error (MSE) loss function, ANNs are trained with various hidden layer sizes using PyTorch [42]. In each case, the Adam algorithm is used with a learning rate of 0.001 and a mini-batch size of 128. In order to avoid overfitting, the training process is terminated when validation MSE fails to improve upon the minimum for 50 consecutive epochs. The selected model parameters correspond to those at the epoch which achieved the best validation MSE. The output of the trained ANN surrogate for a given input $\mathbf{u}$ is denoted by $\mathcal{F}_s(\mathbf{u})$.

The optimal hidden layer size was determined by evaluating the average relative error over the validation set, defined by

$$E_{\text{val}} = \frac{1}{|S_{\text{val}}|} \sum_{i \in S_{\text{val}}} \frac{\|\mathcal{F}(\mathbf{u}_i) - \mathcal{F}_s(\mathbf{u}_i)\|}{\|\mathcal{F}(\mathbf{u}_i)\|}, \tag{19}$$

where S_{val} is the set of indices of X and Y that belong to the validation set. Fig. 4 indicates that the optimal hidden layer size is approximately 1000, and any additional complexity is detrimental to the surrogate's accuracy. This model took less than eight minutes to train, using an NVIDIA RTX A5000 24 GB GPU card on a Windows 10 Workstation with an AMD Threadripper Pro 3945WX Processor (12-cores) 128 GB RAM. The surrogate predictions yield only 1.47% average relative error. More complex models with two and three hidden layers were trained using the same routine but did not significantly improve upon this relative error. Compared to the CVFEM solver which typically takes 109 s to simulate a single injection, a single surrogate model call on the aforementioned GPU takes (430 ± 24) μs. In contrast to the CVFEM solver, however, the trained surrogate can only provide approximations to pressure at the discrete sensor locations.

3.3. Using the surrogate for Bayesian inversion

With the aid of the surrogate $\mathcal{F}_s(\mathbf{u})$, the aim is to apply EKI to compute the posterior of the reinforcement porosity and permeability. In the context of inverse problems, simply replacing the forward map with a surrogate model has been shown to be substantially detrimental to the accuracy of the estimates of the unknown that one wishes to infer. Instead, modelling errors incurred by the surrogate must be incorporated within the inversion. Here, an enhanced modelling error approach [43] is employed.

When the surrogate model is used for the inversion, the observational model defined in Eq. (14) can be re-written as

$$\mathcal{D} = \mathcal{F}_s(\mathbf{u}) + \varepsilon + \eta, \quad \eta \sim N(0, \Gamma), \tag{20}$$

where $\varepsilon = \mathcal{F}(\mathbf{u}) - \mathcal{F}_s(\mathbf{u})$ is the error incurred by using the surrogate. The enhanced modelling error approach assumes that the surrogate error does not depend on the inputs $(\mathbf{u})$ and that it is normally distributed with $\varepsilon \sim N(\bar{\varepsilon}, \Sigma)$. The surrogate error covariance Σ encompasses the error uncertainty inherent in the surrogate predictions at each of the pressure sensor locations. By setting $\gamma = \varepsilon - \bar{\varepsilon} + \eta$, the formulation can be written as

$$\mathcal{D} - \bar{\varepsilon} = \mathcal{F}_s(\mathbf{u}) + \gamma, \quad \gamma \sim N(0, \Gamma + \Sigma). \tag{21}$$

The standard practice is to approximate $\bar{\varepsilon}$ and Σ empirically by computing the mean and covariance of a large number of sample errors. When building the surrogate, this comes at no extra cost since such samples are available from the validation set, $\{\mathcal{F}(\mathbf{u}_i) - \mathcal{F}_s(\mathbf{u}_i)\}_{i \in S_{\text{val}}}$. Fig. 17 in Appendix B shows estimates for $\bar{\varepsilon}$ and $\sqrt{\text{diag}(\Sigma)}$. The surrogate-augmented inverse formulation in Eq. (21) is solved in precisely the same way as Eq. (14) by using Algorithm 1 from Appendix A, but with surrogate evaluations $\mathcal{F}_s$, adjusted data $\mathcal{D} - \bar{\varepsilon}$, and inflated covariance $\Gamma + \Sigma$. The inflation due to surrogate error covariance works by assigning greater importance to data arriving from sensor locations that the surrogate predicts with greater accuracy.

4. Virtual experiments

4.1. Inversion of the test set

Unlike the training and validation data which have been used to produce and evaluate the surrogate for the forward map, the remaining test data will be used in conjunction with the surrogate to rapidly solve the inverse problem. Consider a generic element of X_{test}, $\mathbf{u} = (\mathbf{K}^\dagger, \boldsymbol{\phi}^\dagger, \mu, p_I)$, with corresponding output $\mathcal{F}(\mathbf{u})$ in Y_{test}. The aim is to recover the true reinforcement properties, $(\mathbf{K}^\dagger, \boldsymbol{\phi}^\dagger)$, using synthetic data which are generated by adding noise to $\mathcal{F}(\mathbf{u})$. The noise perturbation

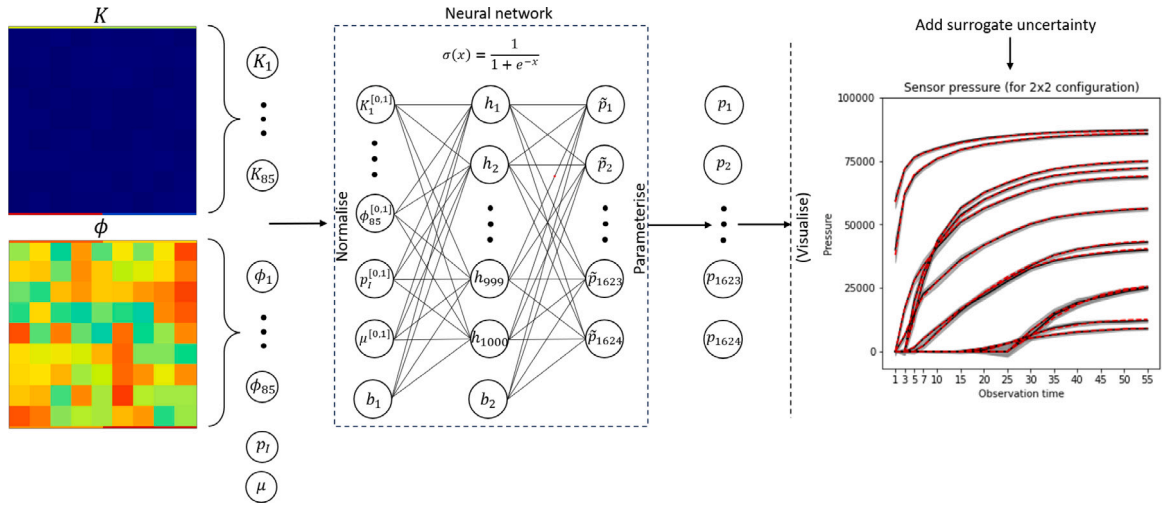

Fig. 3. Process of making predictions with a trained neural network. Also included is a visualisation of surrogate predictions for the 2 × 2 configuration: the true pressures (generated by the expensive solver) are shown in red, whilst the surrogate predictions are shown in black. The 95% uncertainty bands are calculated using the surrogate error uncertainty matrix, Σ.

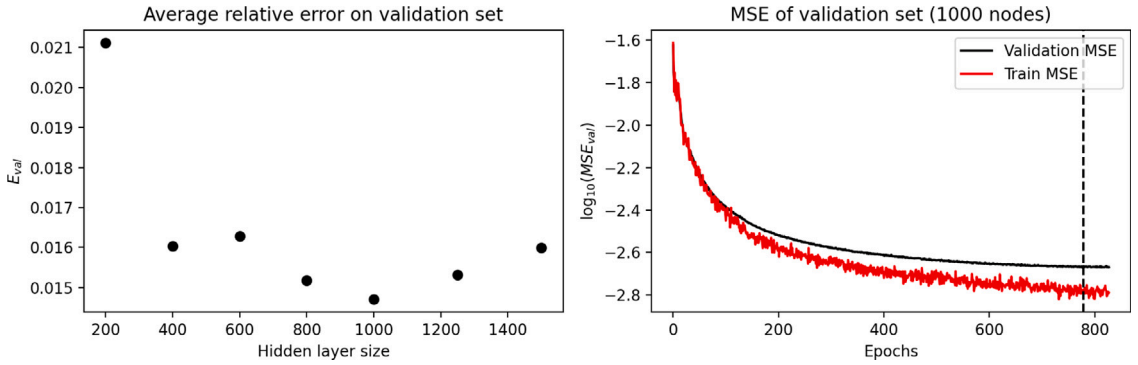

Fig. 4. (Left) average relative error over validation set with respect to the hidden layer size, (right) training and validation MSE of best performing model over the training period, along with the epoch that minimised validation MSE.

mimics sensor inaccuracy by producing data of the form $\mathcal{D} = \mathcal{F}(\mathbf{u}) + \eta$ where $\eta \sim N(0, \Gamma)$. It is assumed that Γ is a diagonal matrix with kth entry

$$\Gamma_{kk} = \sigma_0^2 \Big(\max \mathcal{F}(\mathbf{u}) - \min \mathcal{F}(\mathbf{u}) \Big)^2, \tag{22}$$

where σ_0 scales the size of the noise added. In practice, σ_0 may be informed by the manufacturer-specified precision of the sensors. In cases where the approach in Eq. (22) is not general enough (e.g. non-constant noise), one can otherwise estimate elements of Γ using Gaussian process regression [14].

A prior ensemble is generated at $t_0 = 0$ by sampling permeability and porosity values uniformly and independently from the bounds provided in Table 1, denoted by $\{(\mathbf{K}^{(j)}(t_0), \boldsymbol{\phi}^{(j)}(t_0))\}_{j=1}^{J}$. By sampling from such an uninformed prior, the posterior will depend largely on the information provided by the sensor data which, in turn, will demonstrate the effect of sensor density and precision on the solution to the inverse problem. Later, more complex priors will be considered that integrate a natural relationship between porosity and permeability.

At any arbitrary time during the injection, one can collect all of the newly acquired measurements and perform an inversion using Algorithm 1 (see Appendix A). As mentioned in Section 3.3, the surrogate-accelerated approach to EKI requires that the forward map $\mathcal{F}$ is replaced with the surrogate $\mathcal{F}_s$, and that any instance of the observational noise covariance matrix Γ is inflated with surrogate uncertainty to become $\Gamma + \Sigma$. Resin viscosity μ and inlet pressure p_I, required for every evaluation of $\mathcal{F}_s$, are assumed given in the final two elements of $\mathbf{u}$. Following a successful inversion, the resulting posterior serves as a prior for any subsequent inversions as new data becomes available. With this in mind, it is important to make a distinction between observation times and inversion times — though it is feasible to perform the inversion at each observation time in Eq. (18) (or indeed at any time during injection), for clarity, only 7 inversion times are considered: $(t_1, t_3, t_5, t_7, t_9, t_{11}, t_{14})$.

Denote by $\big(\mathbf{K}^{(j)}(t_n), \boldsymbol{\phi}^{(j)}(t_n)\big)$ the posterior approximation of $(\mathbf{K}^\dagger, \boldsymbol{\phi}^\dagger)$ by the jth ensemble member at time $t = t_n$. The time-dependent ensemble mean and standard deviation estimates for $\mathbf{K}^\dagger$ are computed simply via

$$\mu_{\mathbf{K}}(t_n) = \frac{1}{J}\sum_{j=1}^{J} \mathbf{K}^{(j)}(t_n) \quad \text{and} \quad \sigma_{\mathbf{K}}(t_n) = \sqrt{\frac{1}{J-1}\sum_{j=1}^{J} \big(\mathbf{K}^{(j)}(t_n) - \mu_{\mathbf{K}}(t_n)\big)}, \tag{23}$$

and analogously for $\mu_{\boldsymbol{\phi}}(t_n)$ and $\sigma_{\boldsymbol{\phi}}(t_n)$, the mean and standard deviation estimates for $\boldsymbol{\phi}^\dagger$. For ease of visualisation, the natural logarithm of permeability samples will often be considered, with mean and standard deviation $\mu_{\log \mathbf{K}}$ and $\sigma_{\log \mathbf{K}}$, respectively.

Fig. 5 shows the reconstruction of one example $(\mathbf{K}^\dagger, \boldsymbol{\phi}^\dagger)$ from X_{test} at each inversion time. The mean and standard deviation vectors are each plotted in accordance with the partition described in Eq. (10), though the width of the RT regions are exaggerated for ease of viewing. All of the available sensors were used (see "All" configuration in Fig. 2) with $\sigma_0 = 0.005$ (i.e. 0.5% noise perturbation). A conservative ensemble

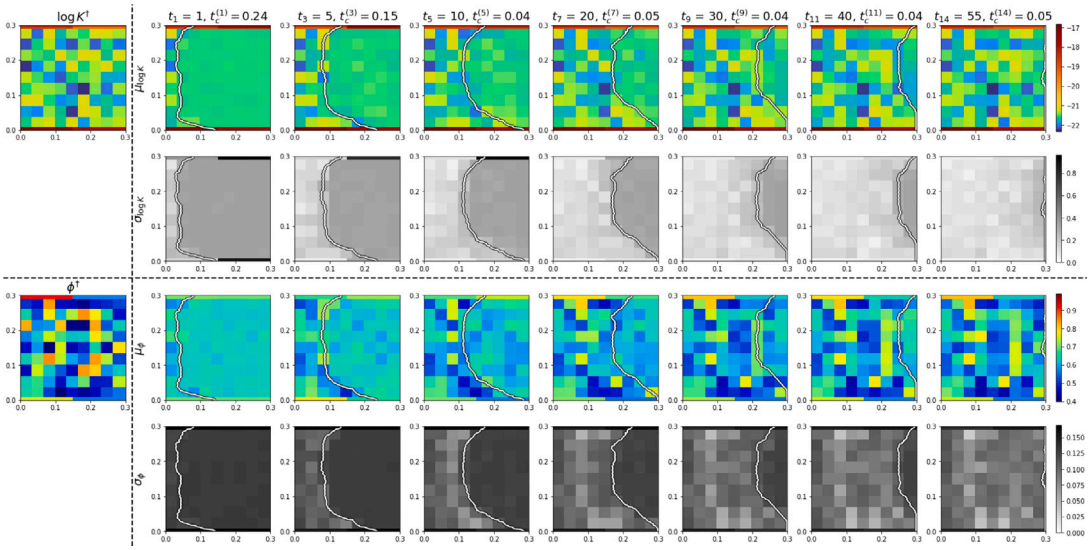

Fig. 5. (Left) true but unknown log-permeability $\log K^{\dagger}(\mathbf{x})$ and porosity $\phi^{\dagger}(\mathbf{x})$ to be recovered, (right) mean and standard deviation of the posterior ensemble for log-permeability and porosity at each inversion time. Also included is the true resin front location (according to the CVFEM solver) and the computation time taken, in seconds, to carry out each inversion, $t_c^{(i)}$.

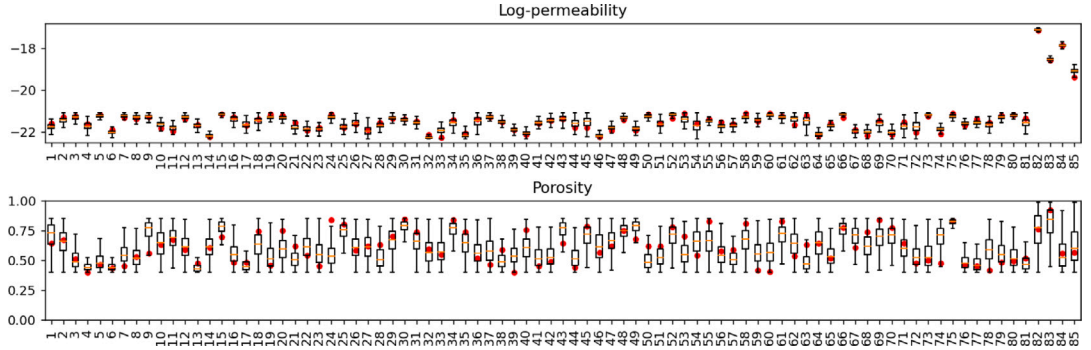

Fig. 6. Boxplots of the final-time posterior ensemble, with the true log-permeability and porosity values shown in red. The first 81 boxes of each plot correspond to the (unpacked) central sub-regions, whilst the final 4 correspond to RT sub-regions.

of size $J = 10{,}000$ is used to ensure that the algorithm is robust under the selection of the (random) initial ensemble. The ensemble mean estimate for log-permeability is accurate in both central and RT sub-regions. Porosity is recovered with less accuracy, a result which is discussed later in this section, though one can still recover many of the broader patterns. The ensemble standard deviation estimates behave as expected, since uncertainty is high in areas which are yet to be impregnated with resin; with such a high sensor density, one can even approximately locate the resin front at each inversion time. Crucially, the surrogate-accelerated EKI algorithm is able to compute the approximate posterior in real time. Boxplots of the ensemble estimates are shown in Fig. 6, providing greater clarity regarding the accuracy of the final-time posterior distribution.

However, this setting is idealistic; such a high density of accurate sensors is not realistic when considering industrial applications. In order to assess the effect of using inferior sensors and of using only a limited number of them, the (final-time) relative error of the solution to the inverse problem is introduced, defined by

$$E_{\mathbf{K}} = \frac{\|\mathbf{K}^{\dagger} - \mu_{\mathbf{K}}(t_{14})\|}{\|\mathbf{K}^{\dagger}\|}, \quad \text{and} \quad E_{\phi} = \frac{\|\boldsymbol{\phi}^{\dagger} - \mu_{\boldsymbol{\phi}}(t_{14})\|}{\|\boldsymbol{\phi}^{\dagger}\|}. \tag{24}$$

Relative error indicates the fidelity with which the true permeability and porosity values are recovered. In order to investigate the dependence of relative error on the number of sensors and noise levels, the procedure described above is carried out for all 1000 elements of the test set, using each sensor configuration and various noise levels. Fig. 7 shows the expected result that by increasing sensor density or by improving the quality of sensors, the true parameter set can be recovered with greater accuracy, on average. Also included is the average error achieved by using the prior ensemble means, $\mu_{\mathbf{K}}(t_0)$ and $\mu_{\boldsymbol{\phi}}(t_0)$, which represents the estimate for reinforcement properties that one might suggest without access to in-process sensors. The average computation time in every case is negligible compared to the filling time in this setting.

Fig. 7 supports the result shown in Fig. 5 that porosity is more difficult to recover than permeability; the posterior performs only marginally better than the prior. Porosity has previously been recovered accurately [14,27] but required both pressure measurements and tracking of the propagating resin flow front. Though the trigger times of many pressure sensors in unison provide some information about the flow front location, it appears that higher resolution images of the front via cameras or multiple linear sensors for front detection are required to characterise the porosity of reinforcements more accurately.

4.2. Synthetic defects

Though the previous section demonstrates the effect of in-process measurements on the solution to the inverse problem, there are no guarantees that the true permeability and porosity functions lie conveniently within the partition defined in Section 2.1. In cases which

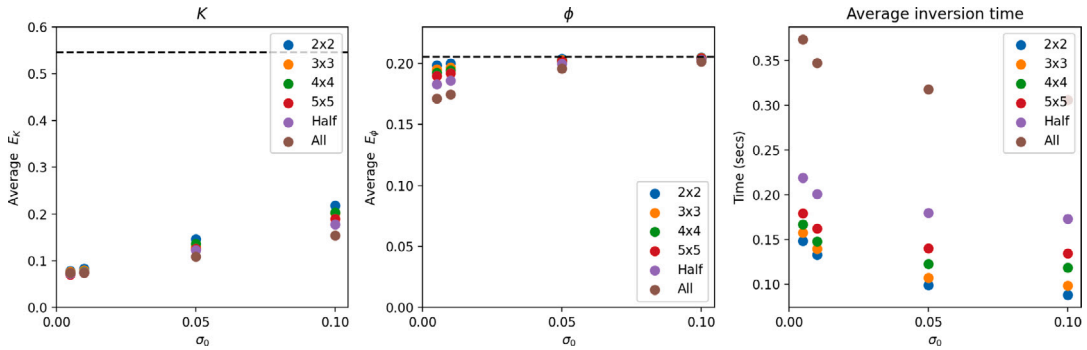

Fig. 7. Average final-time relative error in permeability and porosity estimates over test set for varied noise levels and sensor densities, along with the average computation time. The dashed black line represents the average relative error using the prior ensemble mean.

Table 2
Details for the synthetic cases. Permeability in the background, defect regions and RT channels are denoted by K_{bg}, K_{defect} and K_{RT}, respectively.

Exp.	p_I (kPa)	μ (Pa s)	K_{bg} (mm^2)	K_{defect} (mm^2)	K_{RT} (mm^2)
1	110.0	0.091	4.50×10^{-10}	7.00×10^{-10}	–
2	105.0	0.105	4.50×10^{-10}	–	2.00×10^{-8}
3	102.5	0.095	4.50×10^{-10}	2.04×10^{-10}	2.50×10^{-8}
4	100.0	0.100	4.50×10^{-10}	2.04×10^{-10}	1.00×10^{-8}

deviate from this, the aim is to recover the 'closest' low-dimensional representation of reinforcement properties.

In order to demonstrate the efficacy of the proposed approach in defect discovery, 4 synthetic configurations with defects are considered: (1) a high permeability circular defect of radius 40 mm centred about (75, 150) mm, (2) a one-sided RT channel of width 2 mm, (3) two low permeability circular defects of radius 40 mm centred about (100, 100) mm and (200, 200) mm, along with 2 mm RT channels on opposite corners, (4) a low permeability circular defect of radius 35 mm centred about (200, 100) mm and rectangle of size 23.4 mm × 206.3 mm along with 2 mm RT channels in the first half of the domain. In each case, regions of the reinforcement which are not considered as containing defects have a fixed permeability value. Further details regarding each case are listed in Table 2. Once the true permeability fields are generated, porosity fields are resolved by assuming a power-law relationship

$$\phi = 1 - \left(\frac{K}{a}\right)^{\frac{1}{b}} := f(K), \tag{25}$$

where $a = 10^{-10}$ m^2 and $b = -1.4$.

Synthetic data for each scenario are generated by running the accurate (but computationally expensive) CVFEM solver for the true permeability and porosity fields, before perturbing the simulated pressure values in the same manner as before, i.e. with $\sigma_0 = 0.005$.

It became evident from Figs. 5 to 7 that generating a prior ensemble which disregards the relationship between permeability and porosity yields less accurate estimates of porosity fields. To this end, the Bayesian approach is exploited by proposing a prior ensemble which is consistent with the relationship in Eq. (25). A prior ensemble, $\{(\mathbf{K}^{(j)}(t_0), \boldsymbol{\phi}^{(j)}(t_0))\}_{j=1}^{J}$, of size $J = 10{,}000$ is constructed by first sampling $\{\mathbf{K}^{(j)}(t_0)\}_{j=1}^{J}$ uniformly from the bounds in Table 1, before calculating each of the corresponding porosity particles via $\boldsymbol{\phi}^{(j)}(t_0) = f(\mathbf{K}^{(j)}(t_0))$ using Eq. (25). The image of these permeability samples under f is not contained within the bounds for porosity in Table 1, so any element of $\boldsymbol{\phi}^{(j)}(t_0)$ that exceeds these bounds is assigned the maximum or minimum possible values, respectively. In the interest of presentation, rather than performing sequential inversions as described in Section 4.1, a single inversion is performed at the final observation time, using all of the data collected.

The final-time ensemble means, $\mu_{\log \mathbf{K}}(t_{14})$ and $\mu_{\boldsymbol{\phi}}(t_{14})$, obtained using various sensor densities are shown in Figs. 8 and 9 with the ensemble standard deviations, $\sigma_{\log \mathbf{K}}(t_{14})$ and $\sigma_{\boldsymbol{\phi}}(t_{14})$, deferred to Appendix C. For each experiment, the surrogate-accelerated EKI algorithm converges in 3 to 5 iterations. Estimates using low sensor densities appear less accurate, as expected, but often allude to the broad defect shapes and locations well. Predictably, the size, accuracy and uncertainty of each estimate improves with increase of sensor density. For the most dense sensor configurations, mean estimates give the best low-dimensional representation of the defect.

The probabilistic nature of the Bayesian approach allows one to make statements about the probability of defect occurrence, by using the entire posterior ensemble (from which the mean and standard deviation are computed). Recall that the true but unknown vectors are given by $\mathbf{K}^\dagger = [K_1^\dagger, \dots, K_{85}^\dagger]$ and $\boldsymbol{\phi}^\dagger = [\phi_1^\dagger, \dots, \phi_{85}^\dagger]$. For each sub-region R_i, with the true permeability and porosity value $(K_i^\dagger, \phi_i^\dagger)$, denote its posterior ensemble approximation by $\{(K_i^{(j)}, \phi_i^{(j)})\}_{j=1}^{J}$. The probability that sub-region R_i, $i = 1, \dots, 85$, is defective is given by

$$\mathbb{P}(R_i \text{ is defective}) = \frac{1}{J}\sum_{j=1}^{J} \mathbb{I}_{D_i}\left((K_i^{(j)}, \phi_i^{(j)})\right), \tag{26}$$

where $\mathbb{I}_{D_i}$ is the indicator function of the set D_i which is the range of permeability and porosity values corresponding to a defect associated with the sub-region R_i; D_i will differ based on the context of the problem. For example, if one were interested in RT, then the sets D_i, $i \in \{82, \dots, 85\}$, are defined as $D_i = D^{\text{RT}} \equiv \{(K, \phi) \in \mathbb{R}^2 | K > K_{\text{RT}},\ \phi > \phi_{\text{RT}}\}$ for pre-defined thresholds, K_{RT} and ϕ_{RT}. The defect probability is the proportion of the posterior ensemble which satisfies the defect condition.

Here, the RT threshold for permeability is defined to be an order of magnitude larger than the average value in the central sub-regions according to Table 1, $K_{\text{RT}} = 45\times10^{-10}$ mm^2. The corresponding RT threshold for porosity is set using Eq. (25) to $\phi_{\text{RT}} = f(K_{\text{RT}}) = 0.934$.

The sets D_i, $i \in \{1, \dots, 81\}$, corresponding to the central sub-regions may consist of either one-sided or two-sided conditions, depending on the expected nature of a synthetic defect. For virtual experiments, the sets will be one of the following:

$$D^{\text{low}} = \{(K, \phi) \in \mathbb{R}^2 | K < 3\times10^{-10},\ \phi < 0.544\}, \tag{27}$$

$$D^{\text{high}} = \{(K, \phi) \in \mathbb{R}^2 | K > 6\times10^{-10},\ \phi > 0.722\}, \tag{28}$$

$$D^{\text{both}} = D^{\text{low}} \cup D^{\text{high}}. \tag{29}$$

For $i \in \{1, \dots, 81\}$, experiment 1 has $D_i = D^{\text{high}}$, experiment 2 has $D_i = D^{\text{both}}$, and experiments 3 and 4 have $D_i = D^{\text{low}}$. The thresholds in Eqs. (27) to (29) can be chosen easily for virtual experiments since the true permeability and porosity fields are known; in Section 5, a general method of threshold selection is described which does not require knowledge of the true underlying material properties.

Fig. 10 shows the defect probability map for each of the virtual experiments. Probability maps are more informative, in general, than

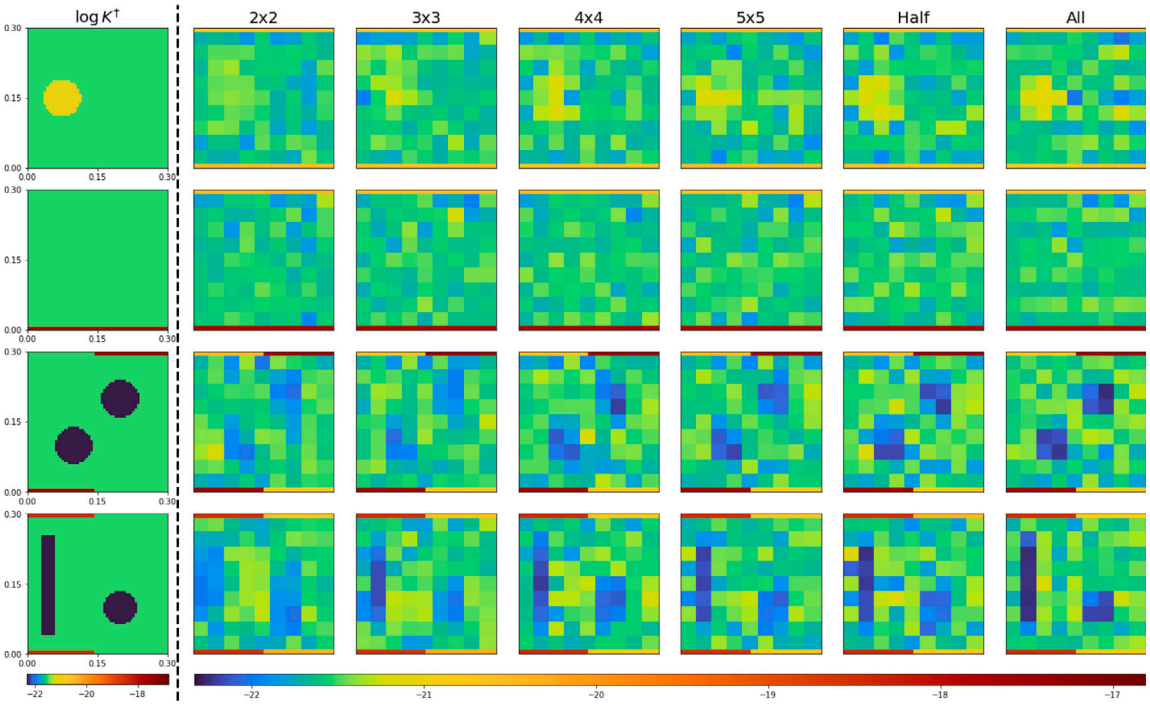

Fig. 8. (Left) true log-permeability $\log K^\dagger(\mathbf{x})$ for each experiment, (right) ensemble mean estimates using different sensor configurations.

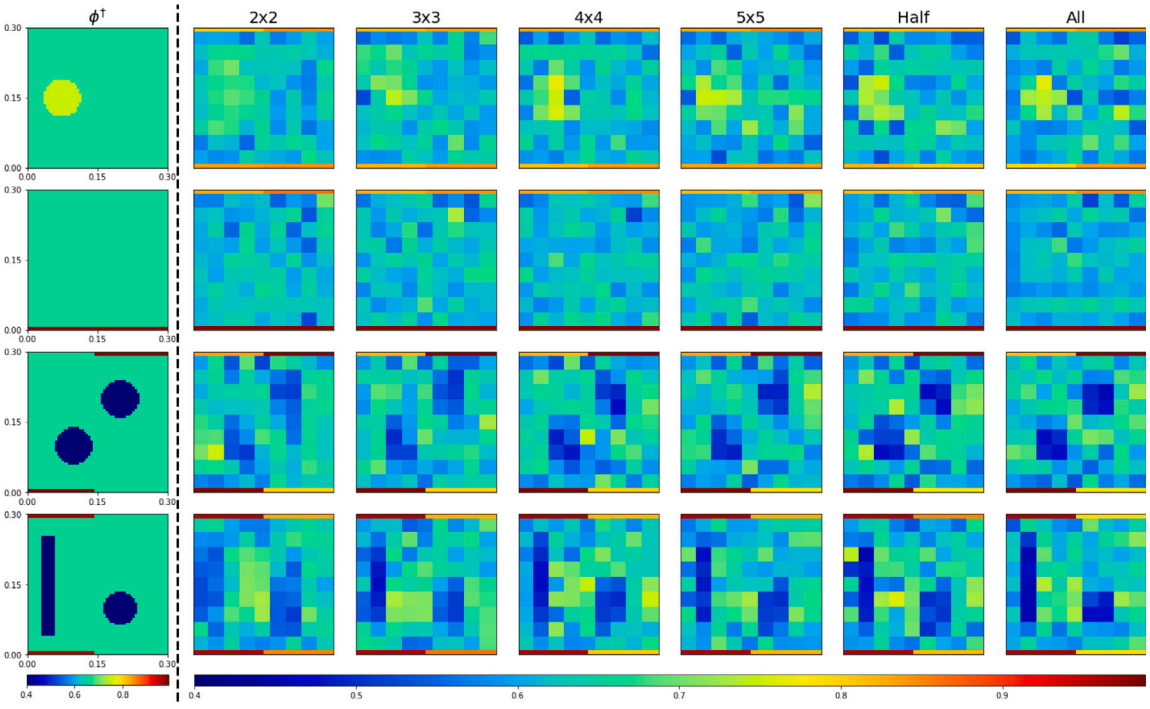

Fig. 9. (Left) true porosity $\phi^\dagger(\mathbf{x})$ for each experiment, (right) ensemble mean estimates using different sensor configurations.

the ensemble mean estimates shown in Figs. 8 and 9 since they utilise predictive uncertainty which is central to the Bayesian methodology used here. Regions in close proximity of defects are assigned elevated defect probability, even when using lower sensor densities. The use of more sensors increases the confidence with which defects are predicted and reduces the false positive rate of detecting defects. Moreover, RT defects (or lack thereof) are identified successfully for each experiment, regardless of the sensor configuration.

4.3. Comparing surrogate-accelerated EKI with full-model EKI

While the predictions of the surrogate model, $\mathcal{F}_s(\mathbf{u})$, are highly accurate as shown in Section 3.2, there are unavoidable modelling

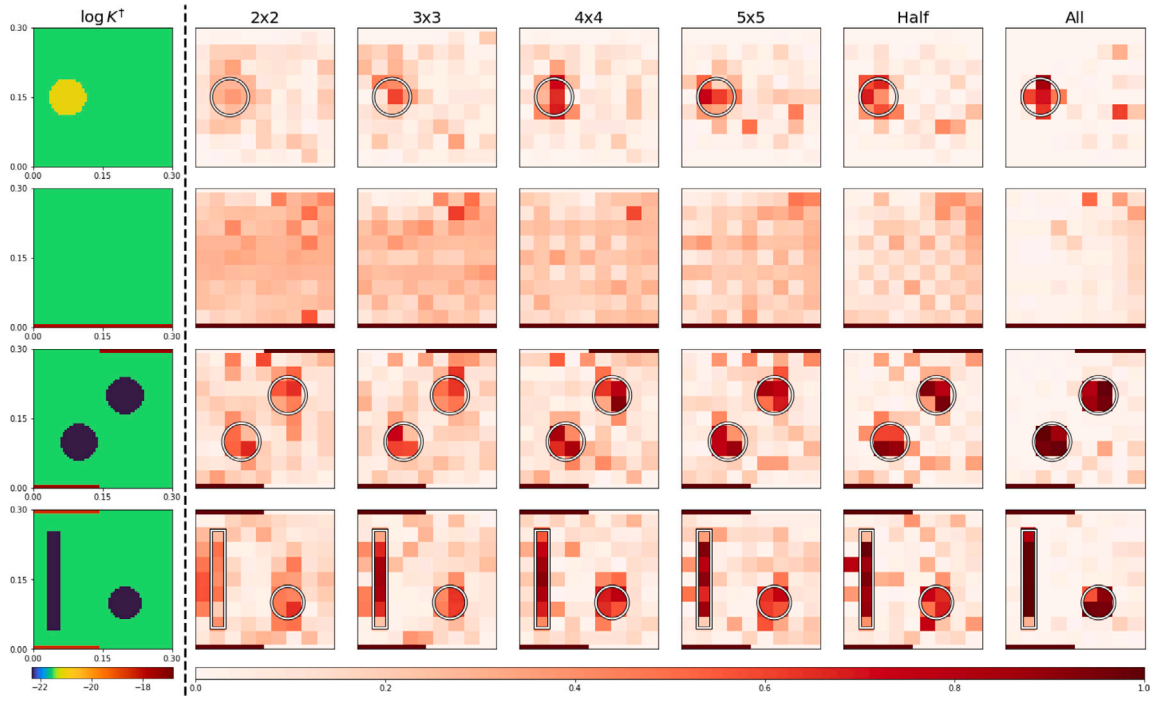

Fig. 10. (Left) true log-permeability $\log K^{\dagger}(\mathbf{x})$ for each experiment, (right) defect probability map using different sensor configurations.

errors that arise from using the surrogate rather than CVFEM simulations, $\mathcal{F}(\mathbf{u})$, referred to in this section as the full-model. These errors could potentially be detrimental to the accuracy of the inference of the unknown reinforcement properties. Since surrogate-accelerated EKI involves inflating the error covariance matrix (see Section 3.3), this approach to inversion effectively amounts to having more uncertain measurements. However, it is shown in this section that those detrimental effects are not substantial, while the computational advantages of the surrogate-accelerated inversion are significant, enabling real-time inference of reinforcement properties.

In order to compare surrogate-accelerated and full-model EKI, synthetic experiment 1 is considered for the set of measurement configurations in Fig. 2. For full-model EKI, the algorithm introduced in Section 2.4 is applied using the same prior on the unknown $(\mathbf{K}, \boldsymbol{\phi})$ as discussed in Section 4.2. It is worth reiterating that every iteration of full-model EKI (see step (a) in Algorithm 1) involves the evaluation of $\mathcal{F}(\mathbf{u}^{(j)})$, i.e. a run of the CVFEM simulator, for each particle $\mathbf{u}^{(j)}$ which is computationally costly. To amortise this cost, a reduced number of $J = 10^3$ particles is employed (compared to $J = 10^4$ for the surrogate-accelerated EKI). Furthermore, the EKI algorithm is run on the UK Tier 2 Midlands Plus HPC platform and, at every iteration, step (a) of EKI is distributed over 100 cores. The total number of iterations required to achieve convergence for each measurement configuration varied from 5 to 6.

Table 3 shows the cost, in terms of wall-time, of full-model EKI on the HPC versus surrogate-accelerated EKI on the GPU. The probability maps for experiment 1 are computed as described in Section 4.2 (i.e. using Eq. (26) with $D_i = D^{\text{high}}$ for $i \in \{1, \ldots, 81\}$ and $D_i = D^{\text{RT}}$ for $i \in \{82, \ldots, 85\}$) and are shown in Fig. 11. There is little difference between the probability maps generated by each approach, indicating that the inherent error incurred by the surrogate does not considerably change estimates for reinforcement properties.

Table 3
Computation times using full-model EKI versus surrogate-accelerated EKI.

	2 × 2	3 × 3	4 × 4	5 × 5	Half	All
Full-model EKI time (h)	2.40	2.40	2.40	1.96	1.96	1.96
Surrogate-accelerated EKI time (s)	0.339	0.113	0.147	0.190	0.381	0.775

5. Lab experiments

5.1. Experimental setup

An injection tool, shown in Fig. 12, with a steel bottom and a transparent PMMA top (with a thickness of 90 mm) was manufactured for the laboratory experiments. The tool was designed to hold specimens with 300 mm × 300 mm edge length. Galleries along two opposite edges act as linear inlet and linear vent to promote unidirectional flow. The tool was equipped with 23 pressure sensors, arrangements of which are shown in the 2 × 2 and 3 × 3 configurations in Fig. 2. One additional sensor is placed at the inlet to record inlet pressure. The pressure sensors are connected to a data acquisition system that collects readings of the fluid pressure at a rate of 10 s^{-1}. A video camera is used to record the flow front propagation.

A continuous random glass fibre reinforcement with an areal weight of (210 ± 15) g/m^2 was used in all experiments. The baseline specimen consisted of 7 layers of the reinforcement. The fibre volume fraction of the specimen was controlled by a spacer frame between the top and bottom parts of the mould. The frame, together with the seals, had a thickness of approximately 2.1 mm. The mould deflection with the reinforcement placed in the mould was measured and found to be < 0.1 mm for the baseline specimen of 7 layers. Therefore, the porosity of the baseline specimen is equal to approximately 0.73. The reinforcement permeability can be approximately described by Eq. (25).

Circles of 80 mm diameter were cut from the same reinforcement. For experiments with inclusions, several layers of these circles were placed between the layers of a specimen, distributed evenly through the thickness. Three additional layers of the reinforcement resulted in a local porosity of approximately 0.62. Race-tracking defects were created by cutting off a narrow strip in the selected location in a specimen. Several configurations were considered.

Synthetic oil was used as a test fluid. The viscosity of the oil had been characterised, at a range of temperatures, using a Brookfield viscometer. The temperature of the oil was measured before the experiments, which allowed the viscosity to be determined from the viscosity–temperature curve. Experiments were carried out with an

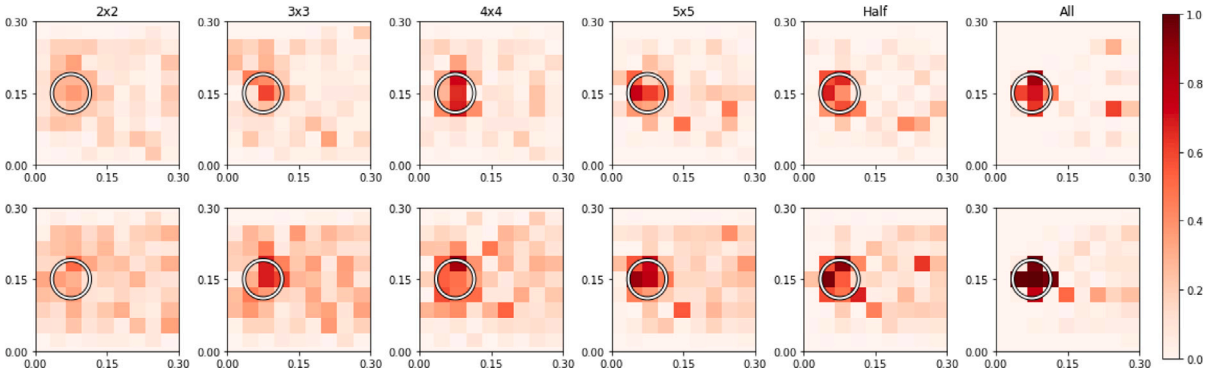

Fig. 11. Comparison of defect probability for experiment 1 using surrogate-accelerated EKI (top), full-model EKI (bottom).

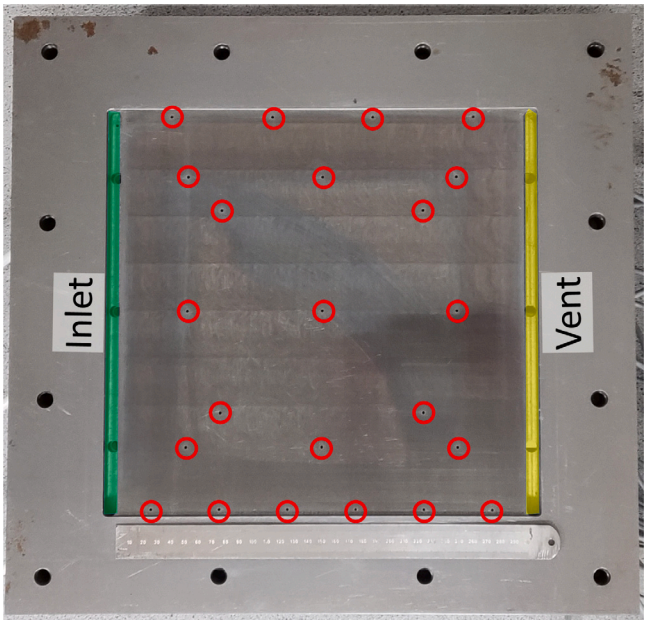

Fig. 12. The experimental tool. Highlighted are the sensor locations (red circles), the inlet (green) and the vent (yellow).

Table 4
Setup for each lab experiment.

Experiment	p_I (Pa)	μ (Pa s)
1	96 066	0.1088
2	106 050	0.0922
3	97 633	0.1088
4	104 410	0.0922

approximate inlet (gauge) pressure of 10^5 Pa. The inlet pressure and resin viscosity associated with each experiment are provided in Table 4.

Here, the prior ensemble is generated dependently, similarly to the virtual experiments in Section 4, by first uniformly sampling permeability values $\{\mathbf{K}^{(j)}(t_0)\}_{j=1}^{J}$ before using Eq. (25) to generate porosity samples. To replicate the worst-case scenario, knowledge of the approximate reinforcement properties are ignored. As stated in Section 4.1, an uniformed prior demonstrates purely the effect of pressure data on the solution to the inverse problem. In practice, more informed priors, such as the truncated Gaussian or beta distributions, are beneficial in estimating reinforcement properties, if designed appropriately. It is assumed that the data are corrupted by 1% noise (i.e. $\sigma_0 = 0.01$).

For the defect probability maps, the same defect thresholds are employed in RT regions, i.e. $D_i = D^{\text{RT}} \equiv \{(K,\phi) \in \mathbb{R}^2 | K > 45 \times 10^{-10},\ \phi > 0.934\}$ for $i \in \{82,\dots,85\}$. In central sub-regions, defect thresholds are selected on a case-by-case basis by pooling all central ensemble estimates for permeability and porosity, i.e. $\left\{\{K_i^{(j)}(t_n)\}_{j=1}^{J}\right\}_{i=1}^{81}$ and $\left\{\{\phi_i^{(j)}(t_n)\}_{j=1}^{J}\right\}_{i=1}^{81}$, and finding the $\alpha-$ and $(1-\alpha)-$quantiles, denoted respectively by q_α^K, $q_{1-\alpha}^K$, q_α^ϕ and $q_{1-\alpha}^\phi$. Defects are then defined as regions of the reinforcement which have permeability and porosity values which are consistently higher or lower than the extreme values in the central pool, according to the defect sets:

$$D_\alpha^{\text{low}} = \{(K,\phi) \in \mathbb{R}^2 | K < q_\alpha^K,\ \phi < q_\alpha^\phi\}, \tag{30}$$

$$D_\alpha^{\text{high}} = \{(K,\phi) \in \mathbb{R}^2 | K > q_{1-\alpha}^K,\ \phi > q_{1-\alpha}^\phi\}, \tag{31}$$

$$D_\alpha^{\text{both}} = D_{\alpha/2}^{\text{low}} \cup D_{\alpha/2}^{\text{high}}. \tag{32}$$

5.2. Example 1 — no defect

In order to demonstrate this approach in the absence of defects, the first experiment contains no intended variations in reinforcement properties. The defect sets in central sub-regions are each set to $D_{0.2}^{\text{both}}$. Fig. 13 shows the inversion of experiment 1. In spite of attempts to produce a specimen with homogeneous material properties, the resin front is not uniform. Hence, sub-regions with high defect probability are unlikely to be false positives, rather genuine variations in reinforcement properties. This is evident when cross-referencing local disturbances in the flow front with the location of sub-regions with elevated defect probability.

Since the prior pessimistically assumes the presence of RT (on average) unless proven otherwise using data, permeability values in RT regions remain slightly elevated compared to the permeability in the rest of the perform. However, uncertainty within these regions remains high, which is reflected within the probability of defect occurrence: surrogate-accelerated EKI detects an absence of RT which is consistent with images of the experiment. Computation times do not exceed 0.22 s.

5.3. Example 2 — circular defect

The second experiment consists of a specimen with a circular low porosity defect. The inversion of experiment 2 is shown in Fig. 14, using the central defect sets $D_{0.2}^{\text{low}}$. The approximate size and location of the central defect are found with high probability within the first 20 s of the experiment with minimal false positives. The false positives in the corners of the reinforcement at earlier times are likely due to the location of sensors providing information in this region. Fig. 2 indicates that the pressure sensors in the problematic corners are located at the border between the RT and central sub-regions. Therefore, although the purpose of these sensors is to provide information regarding RT behaviour, they simultaneously provide information regarding central sub-regions, which could lead to errors in estimates for reinforcement properties. These errors may be exacerbated by, for example, unforeseen deviations in experimental conditions or in the data acquisition

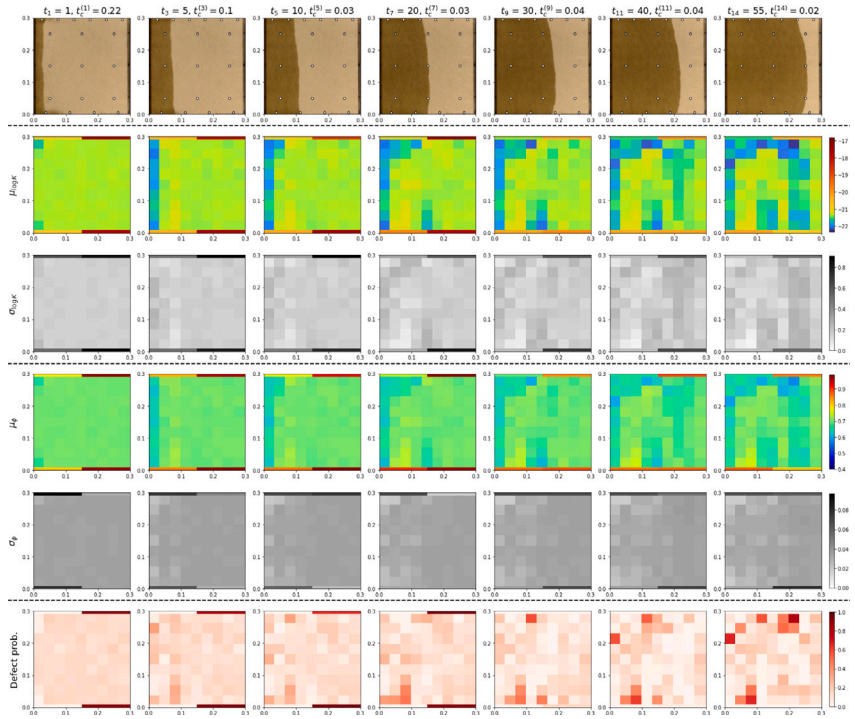

Fig. 13. Inversion of the control (experiment 1). From top to bottom row: images of the lab experiment at each inversion time overlayed with the 3 × 3 sensor configuration, the posterior mean for log-permeability, the posterior standard deviation for log-permeability, the posterior mean for porosity, the posterior standard deviation for porosity, the defect probability map. The computation time, in seconds, required for each inversion is denoted by $t_c^{(i)}$.

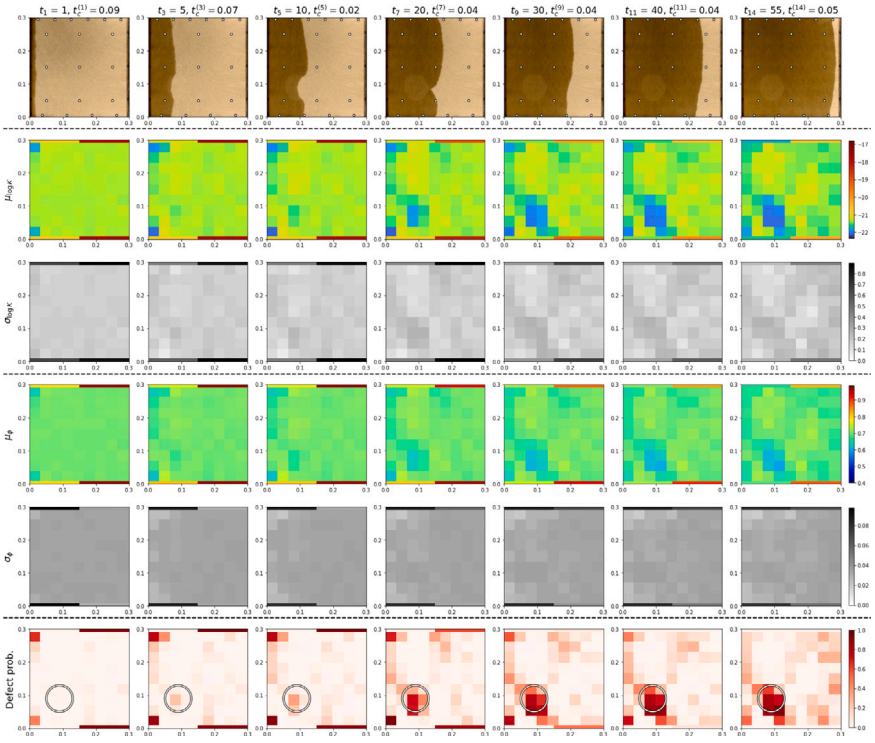

Fig. 14. Inversion of the circular defect (experiment 2). See Fig. 13 for description.

system. The risk of this could be mitigated by ensuring that RT sensors are fully contained within RT sub-regions. Nonetheless, the severity of these false positives is reduced as more data is collected. RT is successfully rejected and the posterior mean porosity in the background is approximately equal to the nominal value of 0.73. Computation times do not exceed 0.09 s.

5.4. Example 3 — RT

For the third experiment, 5 mm strips were cut from the second half of the specimen, along each of the walls. Fig. 15 shows the inversion for the data from experiment 3 using the central defect sets $D_{0.2}^{\text{both}}$. The RT channels are located successfully, with no obvious defects found in the

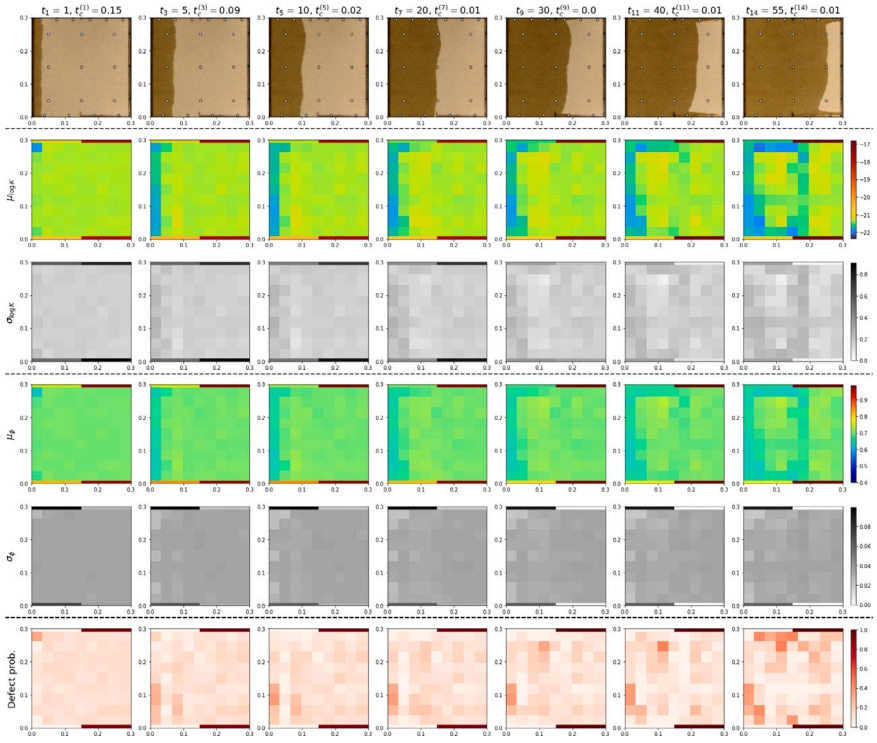

Fig. 15. Inversion of the RT defects (experiment 3). See Fig. 13 for description.

central sub-regions. Before the resin reaches the RT channels, the shape of the front is described well by the mean permeability and porosity estimates. Computation times do not exceed 0.15 s.

5.5. Example 4 — circular defect with RT

Finally, the most complex example is shown in Fig. 16, which includes both a central defect and a 3 mm RT channel along the top edge of the specimen. The central defects sets are each given by $D_{0.2}^{\text{low}}$. Although the central defect is located reasonably accurately, some false positives are also found. Sub-regions with elevated defect probability near the inlet are likely due to a small period of time required for inlet pressure to rise to its specified value, which is compensated for by reducing reinforcement permeability and porosity near the inlet. One could mitigate this artefact by introducing a time-dependent inlet pressure condition, e.g. $p_I(t) = p_I(1 - e^{-\gamma t})$, where $\gamma > 0$ becomes another input to the neural network surrogate model. RT along the top edge of the specimen is found, and is successfully rejected along the bottom edge. Computation times do not exceed 0.12 s.

6. Discussion

The surrogate-accelerated approach to EKI presented in this study offers real-time probabilistic estimation of reinforcement properties by inverting pressure data obtained during the RTM process. The approach was validated with 4 lab experiments, using real data. Circular defects were successfully found, where present, using only 9 sensors within the central region. Sub-regions that were assigned elevated defect probability in locations which were not purposefully designed as defects were frequently caused by unintended variations in reinforcement properties, rather than false positives, which can be explained by cross-referencing images of the propagating resin. RT (or lack thereof) was correctly identified in each of the experiments. Over all of the tested cases (virtual and real), computation times did not exceed 1 s.

By considering a low-dimensional parameterisation of the permeability and porosity functions across the reinforcement, an effective emulator of the Darcy flow forward model can be achieved with only a simple single hidden layer ANN. The surrogate expedites CVFEM simulations of RTM injection, required by the EKI algorithm, reducing computation time from minutes to sub-milliseconds, with only 1.47 % relative error.

Surrogate-accelerated EKI performed well in cases when the defects align with the selected low-dimensional parameterisation. In cases where defects fall outside of the low-dimensional parameterisation, it was demonstrated through virtual experiments that estimates for reinforcement properties approach (with increasing sensor density) the best low-dimensional representation of the high-dimensional truth.

However, a limitation of the parameterisation employed is that, in more extreme cases, there is no guarantee that the size and shape of defects can be accurately represented by a collection of $N_r = 85$ sub-regions. In particular, more complex RT behaviour cannot be accurately expressed using only 4 RT sub-regions. A more descriptive representation might be achieved by using a more refined partition or using the entire mesh, though one must consider the increase in input dimensionality of the problem and/or the need to incorporate some degree of correlation between the values on these regions. In this case, a deeper architecture with convolutional layers is necessary. Surrogates of this nature may require more training data and computational resources, but have shown success [19,20,44] in image-to-image regression of both pressure fields and moving boundaries. Utilising a surrogate which predicts the moving resin front location allows for the integration of flow front information (e.g., through images of the front or arrival time sensors), which could significantly improve estimates for reinforcement properties.

An investigation into the effect of sensor density and sensor precision on the solution to the inverse problem concluded that uninformed prior estimates for reinforcement properties were indeed improved upon by utilising in-process data, particularly in the case of permeability. As expected, reinforcement properties were recovered with greater accuracy when using either a higher density of sensors or more precise sensors.

A method of calculating defect probability was described which exploits the probabilistic nature of the Bayesian approach used here.

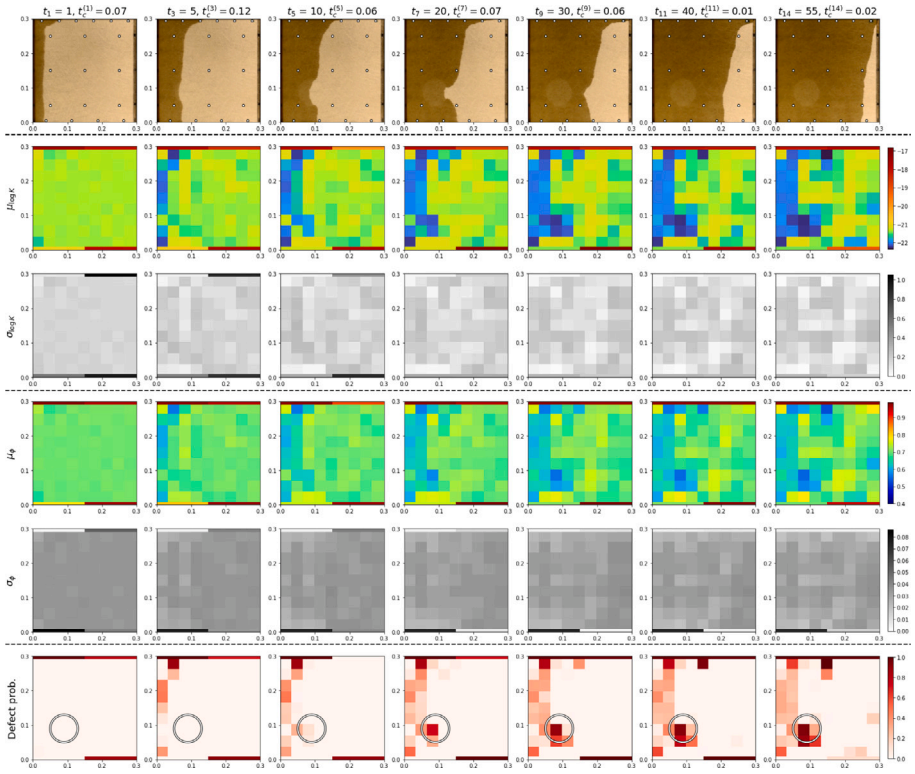

Fig. 16. Inversion of the circular defect with RT (experiment 4). See Fig. 13 for description.

In each of the virtual experiments, which included central and RT defects (both separately and simultaneously), defective sub-regions were successfully assigned elevated defect probability, which was naturally accentuated when using higher sensor densities.

A comparison was made between the full-model EKI algorithm and the surrogate-accelerated approach. It was shown that each method provided comparable estimates for reinforcement properties. It is worth pointing out that the computational resources required by each approach differ hugely — compared to HPCs, sufficiently powerful GPUs are more readily available and comparatively inexpensive. Even with the computational resources of an HPC, full-model EKI takes hours to perform each inversion. In comparison, surrogate-accelerated EKI yields similar results in sub-second times.

The significance of the real-time inversion is that it enables the implementation of active control measures to prevent formation of irregular flow front shapes and to minimise the probability of void formation in the composite. Aiming at uniform flow velocity fields, local pressure gradients in the injection tool can be optimised to compensate for areas of low porosity and permeability in the reinforcement as soon as they are detected (see e.g. [9,45–47] and references therein). For instance, resin inlets and/or vents can be opened or closed, and pressure values at inlets/vents can be adjusted to implement interventions while the resin injection process is running. In addition, the map of local reinforcement properties obtained for each individual component can stay with the component (as a digital twin) and inform the following processing steps as well as the assessment of the mechanical properties.

7. Conclusions

This study demonstrates that reinforcement properties can be estimated in real time using in-process data, even when sensors are distributed sparingly, a claim which is supported by both virtual and lab experiments. Since computation times were negligible in each case, surrogate-accelerated EKI enables rapid NDE. Likewise, integration of this algorithm within active control systems is feasible and warrants further investigation.

It was shown that with a relatively small number $N_r = 85$ of sub-regions, the surrogate, which is ultimately responsible for on-line estimation, need not be particularly elaborate in order to be effective. Whilst more sophisticated models can be powerful (particularly, in the context of more complex fluid flow models or tool geometries), the uncomplicated nature of this approach benefits from being computationally efficient, highly adaptable to other settings, and does not necessitate vast expertise in the field of machine learning. It is worth pointing out that the surrogate is used as a black box in this setting and can be replaced with a model of any desired complexity. Therefore, a natural extension of this work is towards estimation of properties of formed reinforcements in the 3D setting, where a more complex surrogate model is essential in enabling real-time inversion required for active control of RTM.

CRediT authorship contribution statement

M.E. Causon: Writing – original draft, Validation, Investigation, Data curation, Conceptualization. **M.A. Iglesias:** Writing – original draft, Supervision, Methodology, Conceptualization. **M.Y. Matveev:** Writing – review & editing, Supervision, Investigation. **A. Endruweit:** Writing – review & editing, Supervision, Investigation. **M.V. Tretyakov:** Writing – review & editing, Supervision, Project administration, Methodology, Conceptualization.

Declaration of competing interest

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

Data availability

The data used in this research are available from the University of Nottingham data repository: https://doi.org/10.17639/nott.7437.

Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council, UK [grant number EP/P006701/1]; through the EPSRC Future Composites Manufacturing Research Hub. Some of the simulations were performed using the Sulis Tier 2 HPC platform hosted by the Scientific Computing Research Technology Platform at the University of Warwick. Sulis is funded by EPSRC Grant EP/T022108/1 and the HPC Midlands+ consortium, UK.

Appendix A. The EKI algorithm

The ensemble Kalman inversion algorithm shown in Algorithm 1 is the version proposed by Iglesias and Yang [30] (see also [14,27]) combined with a parameterisation of the reinforcement permeability and porosity to incorporate some constraints. In more detail, each permeability-porosity pair must be strictly positive. According to the particular experimental setting, it is further imposed that permeability and porosity values in central sub-regions lie within $[K_{\min}, K_{\max}]$ and $[\phi_{\min}, \phi_{\max}]$, respectively. To account for RT effects, the upper bounds are extended in RT sub-regions to $K^{\mathrm{RT}}_{\max}$ and $\phi^{\mathrm{RT}}_{\max}$, respectively.

Even when the prior ensemble $\{(\mathbf{K}_0^{(j)}, \boldsymbol{\phi}_0^{(j)})\}_{j=1}^J$ is designed to satisfy the above constraints, there is no guarantee that sequence of intermediate distributions generated by EKI does. To address this, it is common practice to parameterise the unknown in order to enforce these constraints. Here, this parameterisation is represented by $\mathcal{P}$, which links $(\mathbf{K}, \boldsymbol{\phi})$ with an intermediate variable $\boldsymbol{\theta} = (\boldsymbol{\psi}, \boldsymbol{\xi})$ via $(\mathbf{K}, \boldsymbol{\phi}) = \mathcal{P}(\boldsymbol{\theta})$, which is defined as describe below.

First, the variable $\mathbf{K} = [K_1, \dots, K_{N_r}]$ is parameterised in terms of an intermediate (artificial) variable $\boldsymbol{\psi} = [\psi_1, \dots, \psi_{N_r}]$ as follows:

$$K_i = \mathcal{P}_K(\psi_i) = \begin{cases} \mathcal{Q}\left[\psi_i, K_{\max}, K_{\min}\right], & 1 \le i \le 81, \\ \mathcal{Q}\left[\psi_i, K^{\mathrm{RT}}_{\max}, K_{\min}\right], & i > 81, \end{cases} \tag{A.1}$$

where

$$\mathcal{Q}\left[\psi, A, B\right] = \frac{B + Ae^{\psi}}{1 + e^{\psi}}. \tag{A.2}$$

The machine learning notation $\mathcal{P}_K(\boldsymbol{\psi})$ is employed to denote the vector that results from the evaluation of $\mathcal{P}_K$ on each component of $\boldsymbol{\psi}$.

With an analogous parameterisation $\boldsymbol{\phi} = \mathcal{P}_\phi(\boldsymbol{\xi})$ for reinforcement porosity, the sought parameterisation is defined as

$$(\mathbf{K}, \boldsymbol{\phi}) = \mathcal{P}(\boldsymbol{\theta}) = \left(\mathcal{P}_K(\boldsymbol{\psi}), \mathcal{P}_\phi(\boldsymbol{\xi})\right) \tag{A.3}$$

Note that the intermediate variable $\boldsymbol{\theta} = (\boldsymbol{\psi}, \boldsymbol{\xi})$ need not have physical meaning so, unlike $(\mathbf{K}, \boldsymbol{\phi})$, can be unconstrained without consequence. The EKI algorithm in Algorithm 1 is therefore performed on $\boldsymbol{\theta}$, before being transformed to $(\mathbf{K}, \boldsymbol{\phi})$ via $\mathcal{P}$. Consequently, upon convergence, Algorithm 1 produces an ensemble $\{\boldsymbol{\theta}_n\}_{n=1}^J$ which are samples from a Gaussian that approximates the posterior on $\boldsymbol{\theta}$, i.e. $\mathbb{P}(\boldsymbol{\theta}|D, \mu, p_I, p_0)$. Thus, the corresponding transformed samples $\{(\mathbf{K}_n^{(j)}, \boldsymbol{\phi}_n^{(j)})\}_{n=1}^J$ are aimed at approximating the posterior $\mathbb{P}(\mathbf{K}, \boldsymbol{\psi}|D, \mu, p_I, p_0)$.

Algorithm 1: The EKI algorithm.

Input : Fixed/known parameters: μ, p_I, p_0
Forward map $\mathcal{F}(\mathbf{u})$
Measurements: D
Noise covariance matrix: Γ
Prior ensemble: $\{(\mathbf{K}_0^{(j)}, \boldsymbol{\phi}_0^{(j)})\}_{j=1}^J$

Output: Posterior ensemble: $\{(\mathbf{K}_n^{(j)}, \boldsymbol{\phi}_n^{(j)})\}_{j=1}^J$

Let M be the number of measurements. Set $n = 0$ and $s_n = 0$.

Transform the initial ensemble: $\boldsymbol{\theta}_0^{(j)} = \mathcal{P}^{-1}(\mathbf{K}_0^{(j)}, \boldsymbol{\phi}_0^{(j)})$ (for $j \in \{1, \dots, J\}$).

while $s_n < 1$ **do**

(a) Evaluate $\mathcal{F}_n^{(j)} = \mathcal{F}(\mathbf{u}_n^{(j)})$ for $j \in \{1, \dots, J\}$, where $\mathbf{u}_n^{(j)} = (\mathbf{K}_n^{(j)}, \boldsymbol{\phi}_n^{(j)}, \mu, p_I, p_0)$, and compute:

$$C_n^{\mathcal{F}\mathcal{F}} = \frac{1}{J-1}\sum_{j=1}^J \left(\mathcal{F}_n^{(j)} - \bar{\mathcal{F}}_n\right)\left(\mathcal{F}_n^{(j)} - \bar{\mathcal{F}}_n\right)^T,$$

where $\bar{\mathcal{F}}_n = \frac{1}{J}\sum_{j=1}^J \mathcal{F}_n^{(j)}$.

(b) Calculate $\alpha_n^* = \frac{1}{M}\frac{1}{J}\sum_{j=1}^J \|\Gamma^{-1/2}(D - \mathcal{F}_n^{(j)})\|^2$

(c) **if** $s_n + \frac{1}{\alpha_n^*} \ge 1$ **then**

Set $\alpha_n = \frac{1}{1-s_n}$ and $s_{n+1} = 1$

else

Set $\alpha_n = \alpha_n^*$ and $s_{n+1} = s_n + \frac{1}{\alpha_n}$

end

(d) Compute:

$$C_n^{\theta\mathcal{F}} = \frac{1}{J-1}\sum_{j=1}^J \left(\boldsymbol{\theta}_n - \bar{\boldsymbol{\theta}}_n\right)\left(\mathcal{F}_n^{(j)} - \bar{\mathcal{F}}_n\right)^T,$$

where $\bar{\boldsymbol{\theta}}_n = \frac{1}{J}\sum_{j=1}^J \boldsymbol{\theta}_n^{(j)}$.

(e) Sample $\eta^{(j)} \sim N(0, \Gamma)$ and update the ensemble via $\boldsymbol{\theta}_n^{(j)} \to \boldsymbol{\theta}_n^{(j)} + C_n^{\theta\mathcal{F}}(C_n^{\mathcal{F}\mathcal{F}} + \alpha_n\Gamma)^{-1}(D - \mathcal{F}_n^{(j)} + \eta^{(j)}), \quad j \in \{1, \dots, J\}$

(f) Transform back: $(\mathbf{K}_n^{(j)}, \boldsymbol{\phi}_n^{(j)}) = \mathcal{P}(\boldsymbol{\theta}_n^{(j)})$

(g) Update $n \to n + 1$

end

Appendix B. Surrogate error statistics

It is assumed that the surrogate error is Gaussian, with $\varepsilon \sim N(\bar{\varepsilon}, \Sigma)$. Estimates for $\bar{\varepsilon}$ and Σ, are computed by finding the mean and covariance of samples from the validation set, $\{F(u_i) - F_s(u_i)\}_{i \in S_{\mathrm{val}}}$. Fig. 17 shows estimates for $\bar{\varepsilon}$ and $\sqrt{\mathrm{diag}(\Sigma)}$.

Appendix C. Standard deviations for virtual experiments

The final-time approximations for standard deviation for each of the four virtual experiments considered in Section 4.2 are shown in Figs. 18 and 19.

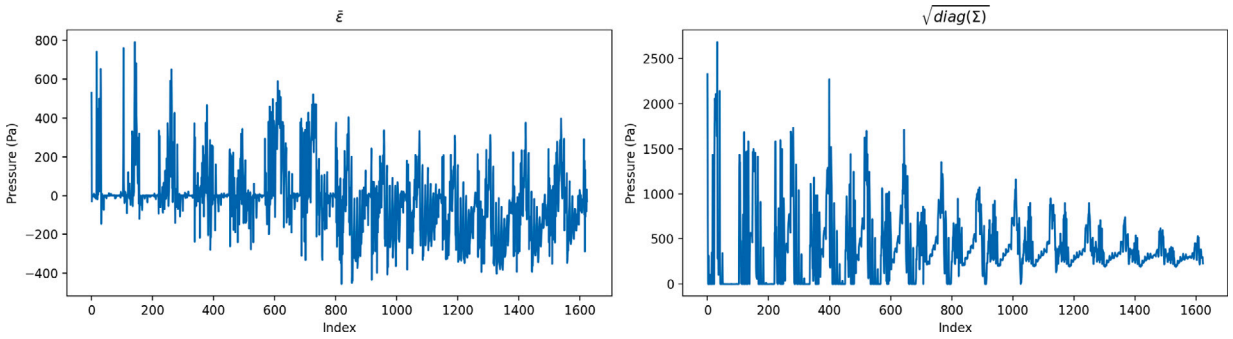

Fig. 17. Visualisation of the estimates for $\bar{\varepsilon}$ and Σ, using the validation set.

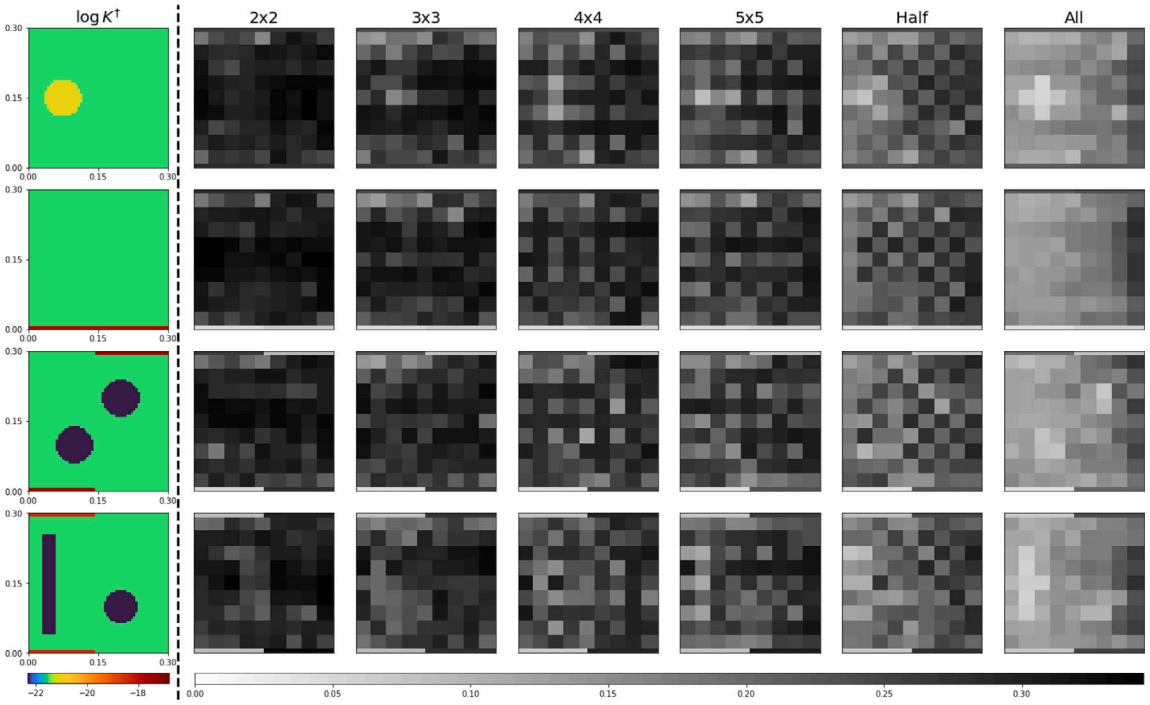

Fig. 18. (Left) true log-permeability $\log K^\dagger(\mathbf{x})$ for each experiment, (right) ensemble standard deviation estimates for log-permeability using different sensor configurations.

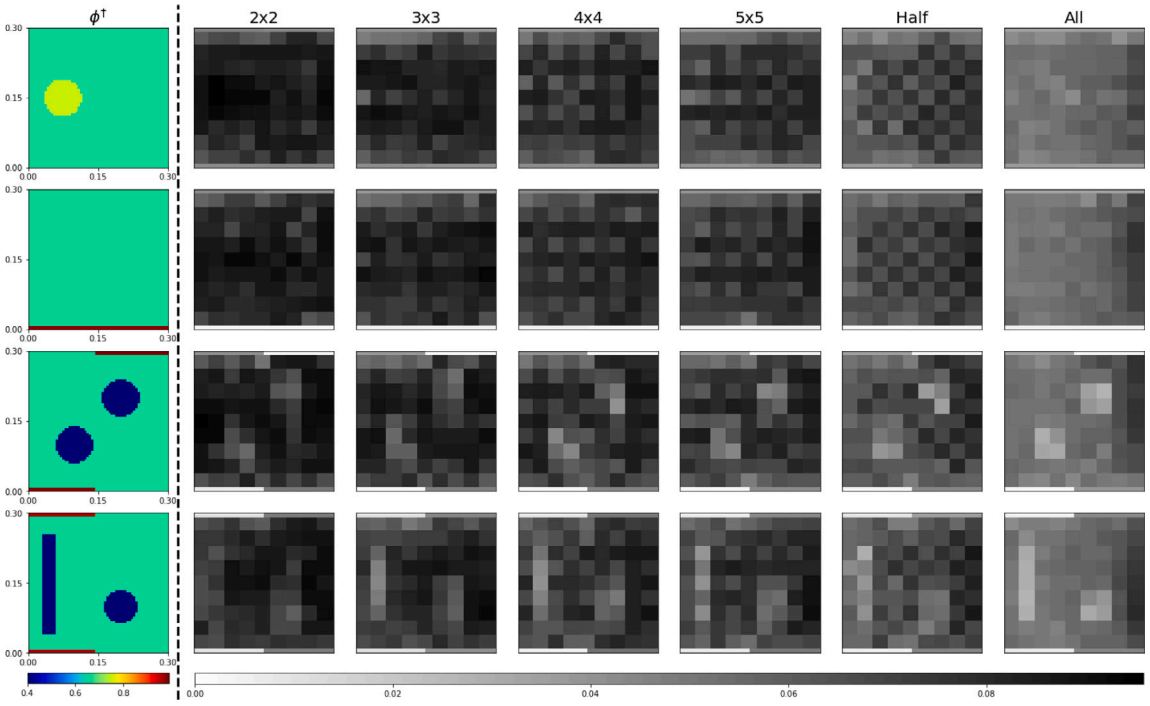

Fig. 19. (Left) true porosity $\phi^\dagger(\mathbf{x})$ for each experiment, (right) ensemble standard deviation estimates for porosity using different sensor configurations.

References

[1] Advani S, Sozer E. Process modeling in composite manufacturing. CRC Press; 2010.

[2] Bodaghi M, Lomov SV, Simacek P, Correia NC, Advani SG. On the variability of permeability induced by reinforcement distortions and dual scale flow in liquid composite moulding: a review. Composites A 2019;120:188–210.

[3] Endruweit A, Long AC. Influence of stochastic variations in the fibre spacing on the permeability of bi-directional textile fabrics. Composites A 2006;37(5):679–94.

[4] Endruweit A, Long AC, Robitaille F, Rudd C. Influence of stochastic fibre angle variations on the permeability of bi-directional textile fabrics. Composites A 2006;37:122–32.

[5] Matveev MY, Ball F, Jones I, Long AC, Schubel P, Tretyakov MV. Uncertainty in geometry of fibre preforms manufactured with automated dry fibre placement and its effects on permeability. J Compos Mater 2018;52(16):2255–69.

[6] Mesogitis T, Skordos AA, Long AC. Uncertainty in the manufacturing of fibrous thermosetting composites: A review. Composites A 2014;57:67–75.

[7] Sozer EM, Bickerton S, Advani SG. On-line strategic control of liquid composite mould filling process. Composites A 2000;31:1383–94.

[8] Devillard M, Hsiao KT, Gokce A, Advani SG. On-line characterization of bulk permeability and race-tracking during the filling stage in resin transfer molding process. J Compos Mater 2003;37:1525–41.
[9] Hsiao KT, Advani SG. Flow sensing and control strategies to address race-tracking disturbances in resin transfer molding. Part I: design and algorithm development. Composites A 2004;35:1149–59.
[10] Siddig NA, Binetruy C, Syerko E, Simacek P, Advani SG. A new methodology for race-tracking detection and criticality in resin transfer molding process using pressure sensors. J Compos Mater 2018;52:4087–103.
[11] Mendikute J, Plazaola J, Baskaran M, Zugasti E, Aretxabaleta L, Aurrekoetxea J. Impregnation quality diagnosis in resin transfer moulding by machine learning. Composites B 2021;221.
[12] Tartakovsky DM, Winter CL. Dynamics of free surfaces in random porous media. SIAM J Appl Math 2001;61:1857–76.
[13] Park M, Tretyakov MV. Stochastic resin transfer molding process. SIAM/ASA J Uncertainty Quantif 2017;5(1):1110–35.
[14] Matveev MY, Endruweit A, Long AC, Iglesias MA, Tretyakov MV. Bayesian inversion algorithm for estimating local variations in permeability and porosity of reinforcements using experimental data. Composites A 2021;143:106323.
[15] Nielsen D, Pitchumani R. Control of flow in resin transfer molding with real-time preform permeability estimation. Polym Compos 2002;23:1087–110.
[16] Wei B-J, Chang Y-S, Yao Y, Fang J. Online estimation and monitoring of local permeability in resin transfer molding. Polym Compos 2016;37:1249–58.
[17] González C, Fernández-León J. A machine learning model to detect flow disturbances during manufacturing of composites by liquid moulding. J Compos Sci 2020;4(2).
[18] Fernández-León J, Keramati K, Garoz D, Baumela L, Miguel C, González C. A machine learning strategy for race-tracking detection during manufacturing of composites by liquid moulding. Integr Mater Manuf Innov 2022;11(2):296–311.
[19] Fernández-León J, Keramati K, Miguel C, González C, Baumela L. A deep encoder-decoder for surrogate modelling of liquid moulding of composites. Eng Appl Artif Intell 2023;120.
[20] Fernández-León J, Keramati K, Baumela L, González C. A digital twin for smart manufacturing of structural composites by liquid moulding. Int J Adv Manuf Technol 2024;130(9):4679–97.
[21] Hanna JM, Aguado JV, Comas-Cardona S, Le Guennec Y, Borzacchiello D. A self-supervised learning framework based on physics-informed and convolutional neural networks to identify local anisotropic permeability tensor from textiles 2D images for filling pattern prediction. Composites A 2024;179:108019.
[22] Stieber S, Schröter N, Fauster E, Bender M, Schiendorfer A, Reif W. Inferring material properties from FRP processes via sim-to-real learning. Int J Adv Manuf Technol 2023;128(3–4):1517–33.
[23] Comas-Cardona S, Cosson B, Bickerton S, Binetruy C. An optically-based inverse method to measure in-plane permeability fields of fibrous reinforcements. Composites A 2014;57:41–8.
[24] Caglar B, Salvatori D, Sozer EM, Michaud V. In-plane permeability distribution mapping of isotropic mats using flow front detection. Composites A 2018;113:275–86.
[25] Kaipio JP, Somersalo E. Statistical and computational inverse problems. Springer; 2005.
[26] Stuart A. Inverse problems: a Bayesian perspective. Acta Numer 2010;19:451–559.
[27] Iglesias MA, Park M, Tretyakov MV. Bayesian inversion in resin transfer molding. Inverse Problems 2018;34(10):105002.
[28] Park M, Matveev MY, Tretyakov MV. Control volume FEM solver for 2D moving boundary problems: MATLAB toolbox. Zenodo; 2024, http://dx.doi.org/10.5281/zenodo.1091458.
[29] Bruschke MV, Advani SG. A finite element/control volume approach to mold filling in anisotropic porous media. Polym Compos 1990;11:398–405.
[30] Iglesias M, Yang Y. Adaptive regularisation for ensemble Kalman inversion. Inverse Problems 2020;37:025008.
[31] Liu JS, Liu JS. Monte Carlo strategies in scientific computing. Springer; 2001.
[32] Kudela J, Matousek R. Recent advances and applications of surrogate models for finite element method computations: a review. Soft Comput 2022;26(24):13709–33.
[33] Chinesta F, Ladeveze P, Cueto E. A short review on model order reduction based on proper generalized decomposition. Arch Comput Methods Eng 2011;18:395–404.
[34] Gramacy RB. Surrogates: Gaussian process modeling, design, and optimization for the applied sciences. Chapman and Hall/CRC; 2020.
[35] Tifkitsis KI, Skordos AA. Real time uncertainty estimation in filling stage of resin transfer molding process. Polym Compos 2020;41(12):5387–402.
[36] Hastie T, Tibshirani R, Friedman JH. The elements of statistical learning: data mining, inference, and prediction. Springer; 2009.
[37] Quinonero-Candela J, Rasmussen CE, Williams CK. Approximation methods for Gaussian process regression. In: Large-scale kernel machines. MIT Press; 2007, p. 203–23.
[38] Hensman J, Matthews A, Ghahramani Z. Scalable variational Gaussian process classification in artificial intelligence and statistics. PMLR; 2015, p. 351–60.
[39] Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 2019;378:686–707.
[40] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT Press; 2016.
[41] Sobol' IM. On the distribution of points in a cube and the approximate evaluation of integrals. Zh Vychisl Mat Mat Fiz 1967;7(4):784–802.
[42] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: an imperative style, high-performance deep learning library. In: Proceedings of the 33rd international conference on neural information processing systems. Red Hook, NY, USA: Curran Associates Inc.; 2019.
[43] Calvetti D, Dunlop M, Somersalo E, Stuart A. Iterative updating of model error for Bayesian inversion. Inverse Problems 2017;34(2):025008.
[44] Caglar B, Broggi G, Ali MA, Orgéas L, Michaud V. Deep learning accelerated prediction of the permeability of fibrous microstructures. Composites A 2022;158:106973.
[45] Liu B, Bickerton S, Advani SG. Modelling and simulation of resin transfer moulding (RTM) - gate control, venting and dry spot prediction. Composites A 1996;27:135–41.
[46] Modi D, Correia N, Johnson M, Long A, Rudd C, Robitaille F. Active control of the vacuum infusion process. Composites A 2007;38(5):1271–87.
[47] Endruweit A, Matveev MY, Tretyakov MV. Controlling resin flow in Liquid Composite Moulding processes through localized irradiation with ultraviolet light. Polym Compos 2022;43(11):8308–21.