



# Inferring material properties from FRP processes via sim-to-real learning

Simon Stieber<sup>1</sup> · Niklas Schröter<sup>1</sup> · Ewald Fauster<sup>2</sup> · Marcel Bender<sup>2</sup> · Alexander Schiendorfer<sup>3</sup> · Wolfgang Reif<sup>1</sup>

Received: 20 October 2022 / Accepted: 30 April 2023 / Published online: 27 July 2023  
© The Author(s) 2023

## Abstract

Fiber reinforced polymers (FRP) provide favorable properties such as weight-specific strength and stiffness that are central for certain industries, such as aerospace or automotive manufacturing. Liquid composite molding (LCM) is a family of often employed, inexpensive, out-of-autoclave manufacturing techniques. Among them, resin transfer molding (RTM), offers a high degree of automation. Herein, textile preforms are saturated by a fluid polymer matrix in a closed mold. Both impregnation quality and level of fiber volume content are of crucial importance for the final part quality. We propose to simultaneously learn three major textile properties (fiber volume content and permeability in X and Y direction) presented as a three-dimensional map based on a sequence of camera images acquired in flow experiments and compare CNNs, ConvLSTMs, and Transformers. Moreover, we show how simulation-to-real transfer learning can improve a digital twin in FRP manufacturing, compared to simulation-only models and models based on sparse real data. The overall best metrics are: IOU 0.5031 and Accuracy 95.929 %, obtained by pretrained transformer models.

**Keywords** Sequence-to-image learning · Architecture comparison · FRP · LCM · Transfer learning · Industry 4.0 · Digital twin

## 1 RTM and permeability measurements

When it comes to weight-specific mechanical characteristics like strength and stiffness, fiber reinforced polymers (FRP) are superior to other engineering materials. FRP, espe-

cially when using carbon fibers, can replace conventional steel or aluminum, which lowers fuel use and CO<sub>2</sub> emissions throughout the product lifecycle. These composite materials are built up from a polymer matrix that has fiber reinforcement. Liquid composite molding (LCM) techniques are some of the most well-known and cost-effective out-of-autoclave manufacturing methods for FRP components production. For medium volumes (1,000 to 10,000 components per year), resin transfer molding (RTM, [1]) is a regularly used manufacturing method. Under pressure, a liquid thermoset polymer (a resin) is injected into a mold cavity containing the reinforcement material, such as a fabric made of carbon or glass fibers (called the *preform*). During the injection procedure, this results in a “flow front” that separates saturated from dry material (cf. Figure 3).

The *permeability* properties of the preform, which explain how well a porous medium transmits fluids, have a major impact on the dynamics of the flow front. In FRP, (i) fiber volume content ( $V_f$ ) and (ii) preform layout primarily determine the preform permeability. Localized changes in preform permeability can take place at locations where the wall thickness varies or at curved sections of the component (compression or even folding of the preform on the inner surface of the curvature). Additionally, they could result from manual treatment of the fibrous structure or from the material’s flawed qualities

✉ Simon Stieber  
stieber@isse.de

Niklas Schröter  
schroeter@isse.de

Ewald Fauster  
ewald.fauster@unileoben.ac.at

Marcel Bender  
marcel.bender@unileoben.ac.at

Alexander Schiendorfer  
alexander.schiendorfer@thi.de

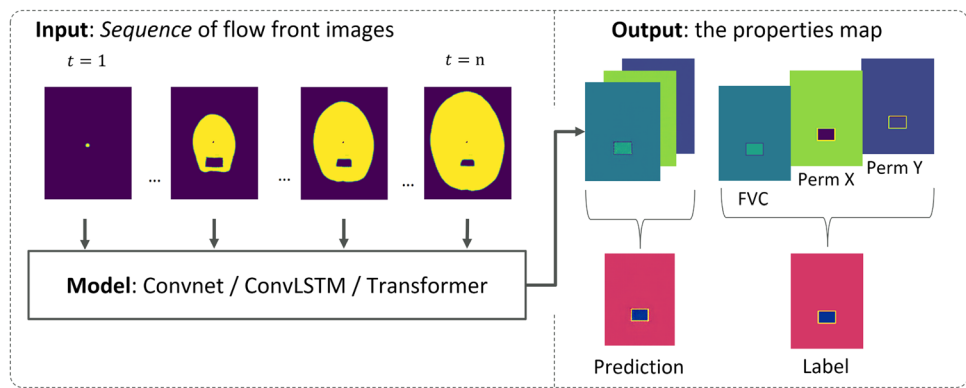
Wolfgang Reif  
reif@isse.de

<sup>1</sup> Institute for Software & Systems Engineering, University of Augsburg, Augsburg, Germany

<sup>2</sup> Processing of Composites Group, Montanuniversität Leoben, Leoben, Austria

<sup>3</sup> Institute AImotion Bavaria, Technische Hochschule Ingolstadt, Ingolstadt, Germany

**Fig. 1** Overview of the sequence-to-instance learning task: a sequence of flow front images is mapped to an image that contains the permeability values of the material, i.e. the permeability map



such as missing or misplaced fiber bundles (e.g., fringing of fiber bundles, breakage of filaments). For one type of textile, the permeability might vary by up to 20 %, according to Tifkitsis [2]. Local differences in permeability might result in subpar impregnation quality or even dry areas, which will reduce the mechanical performance of the final composite part. These variations also affect the temporal progression of the flow front. Before or during an industrial RTM process, these changes are often unknown for a specific preform; therefore, it would be ideal to have a *map of the permeability and  $V_f$  values in order to identify problems as soon as possible*.

We suggest using machine learning (ML) models that, given an observed *sequence of flow front images* during an injection procedure in a permeameter, predict *changes* in the permeability and  $V_f$ , resulting in a *sequence-to-instance task* (cf. Figure 1). The training data is collected from two sources: (1) flow experiment data from a permeameter, and (2) flow simulation data from PAM RTM, a commercially available FEM simulation tool for RTM processes (see Sect. 2). This assignment is a component of a broader pipeline that ultimately results in the prediction of a textile property map from in-mold sensory data. For such, an intermediate step that converts *sensors* to *flow front images* is required, such as employing transposed convolutions [3], generative models, or analytical techniques [4]. To sum up, our *contributions* are as follows:

- For this sequence-to-instance task in an engineering context, we compare several ML models (see Tables 4 and 5).
- We demonstrate the effectiveness of sim-to-real transfer learning in low data regimes by using transfer learning to adapt a model trained in the simulation domain to real data (see Table 6).

This method can be viewed as a component of the process' digital twin [5]. The component can be regarded as acceptable if it does not exhibit excessive fluctuations in permeability and  $V_f$ .

This paper substantially extends preliminary results discussed in [6]: In order to collect real injection data, we conducted additional physical experiments. We used real data not only for evaluation but also for training purposes and measured the quality effects. Finally, we extended the models to not only predict deviations of  $V_f$  but also of the permeability in X and Y direction. We also take into account textiles with anisotropic characteristics which increases both the difficulty and the realism of the proposed scenario.

## 1.1 Related Work

The composites processing literature has taken an interest in online measurements of permeability or its variation. The issue of in-situ monitoring the filling status of RTM processes using lineal sensors is taken into consideration by Tifkitsis et al. [2]. They use Kriging models to forecast the arrival times of the flow front at specific sensors. The same simulation program we used, PAM-RTM<sup>®</sup>, is utilized to generate the RTM simulations that are needed to fit these substitute models. However, we take into account the entire spatial expansion of the flow front (i.e., a high-dimensional prediction) while they just take into account two measurement lines across a plate. By using Darcy's law [7] to calculate it, Wei et al. [8] estimate global and local permeability during the injection with pressure and flow front sensors:

$$\mathbf{v} = -\frac{1}{\eta} \mathbf{K} \nabla p, \quad (1)$$

with volume-average flow velocity  $\mathbf{v}$ , fluid viscosity  $\eta$ , permeability tensor  $\mathbf{K}$ , and pressure gradient  $\nabla p$ . Simulating liquid composite molding relies heavily on Darcy's law. The planar, anisotropic permeability tensor is modeled as follows for 2-dimensional flow (which is covered in the work at hand):

$$\mathbf{K}_{2D, \text{aniso}} = \begin{bmatrix} k_x & k_{xy} \\ k_{xy} & k_y \end{bmatrix} \quad (2)$$

$k_x$  and  $k_y$  represent the permeability in the x and y directions, respectively, whereas  $k_{xy}$  describes how the flow in one major direction is dependent on a pressure gradient in the other major direction. A Regularising Ensemble Kalman filter Algorithm (RENKA) was demonstrated by Matveev et al. [9] as a means of detecting permeability changes throughout an RTM process. To reduce the computational complexity of pure Bayesian inversion, they introduce this filtering technique. However, the algorithm is less usable than a fully data-driven ML approach because it requires nontrivial mathematical steps and must be modified for new component designs. CNNs were utilized by Gonzales et al. [10] to monitor changes in the flow front caused by pressure sensors to detect permeability changes. However, they are only able to identify discrete, rectangular changes in permeability by using the data of all recorded runs. Our method is more flexible since it creates the full permeability map of the preform rather than just identifying specific elements of the patch, like its length, width, or center point. Moreover, the flow fronts in each of the aforementioned works differ because they all employ a lineal injection gate rather than a central one. CNNs are used by Caglar et al. [11] to measure the permeability of textiles at the microscopic level. Other papers integrate RTM analysis and ML in addition to online permeability estimation: Stieber et al. [3] describe a learning-based dry spot classifier that solely operates on simulated data. Transfer learning from simulated to actual data is a suggested research area, particularly for low-data regimes. Weiss et al., Stieber and Hofer et al. [12–14] All the above works employ rectilinear flow, whereas our work employs radial flow. From the manufacturing perspective, this a more realistic case since it is two dimensional instead of one dimensional and thus is applicable to a broader range of processes. Besides online permeability estimation, other papers combine RTM analysis and ML: Stieber et al. [3] present a learning-based dry spot classifier, working only on simulated data. Especially for low-data regimes, transfer learning from simulated to real data is a suggested research direction [12–14].

## 2 Data regime

Training neural networks for online prediction of preform textile properties usually requires a large amount of data paired with knowledge about material properties and experimental parameters. Such data sets are rarely available, as (i) RTM runs are elaborate in terms of experimental time as well as cost, and (ii) RTM molds, that are industrially employed and not prototypically or in landmark research projects, typically show a very limited number of sensors, that are mostly limited to only temperature and pressure signals [15–17].

In this work, we propose to use data sets from both, real experiments and simulation runs, in order to outperform

models based on simulation data only. In particular, the data sets used for this work stem either from the optical permeameter testrig depicted in Fig. 3 or from an FEM-based flow simulation, run in PAM-RTM<sup>®</sup>.

## 2.1 Materials

### 2.1.1 Test fluid

Comparative rheometric tests revealed that the color pigment does not affect the viscosity of the test fluid.

For the flow experiments on the optical permeameter, standard plant oil was used as a test fluid. This is common practice in experimental permeability characterization [18] to avoid handling of chemically reactive thermosets. The viscosity of the test fluid was experimentally characterized in a temperature range between 15 and 30 °C, which covers the range typically seen in non-conditioned research labs. In this range, a nearly linear decreasing trend was found between about 90 and 48 mPas. Thus, the viscosity of the test fluid is well comparable to that of uncured (this is the relevant condition during impregnation of the fibrous preform) epoxy resin commonly used in RTM. In our flow experiments, the fluid temperature was measured in the feeding line and then interpolated into the above-mentioned characteristics in order to obtain the fluid viscosity relevant for the respective experiment. In order to enhance the contrast of saturated against unsaturated preform regions in the digital images acquired during the flow experiments, the test fluid was colored with a red color pigment (Sudanred IV, Sigma Aldrich). Comparative rheometric tests revealed that the color pigment does not affect the viscosity of the test fluid.

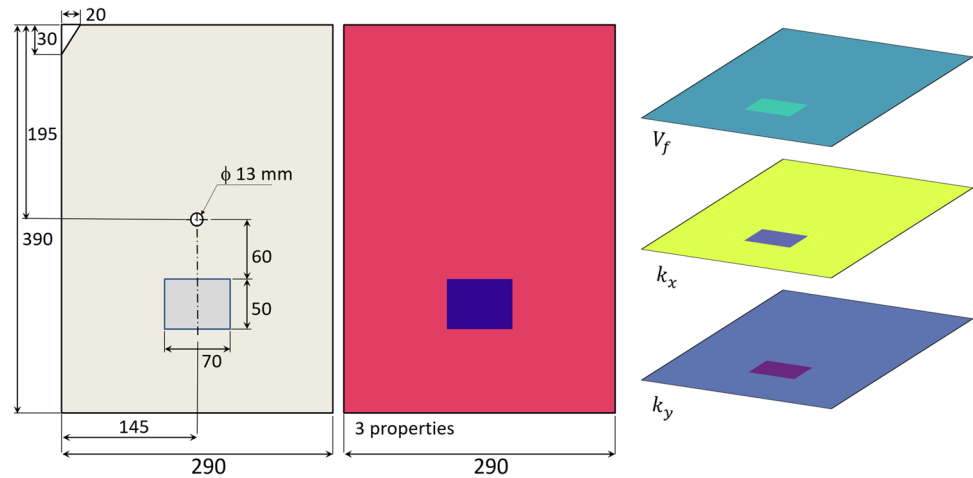
### 2.1.2 Fabric

The experiments were run with a glass fiber, 2/2 twill woven fabric, type Hexion 1202 of Hexcel, with a nominal areal weight of 290 g/m<sup>2</sup> and in-plane permeability characteristics as listed in Table 1. The material is commercially available and well known from recent international benchmark exercises (IBE) on in-plane permeability characterization [18]. The research group co-authoring this paper has participated in these IBE with their optical permeameter and the in-plane permeability data used for the simulation runs were found

**Table 1** Preform setup for real-world trials: it is created by homogeneous stacking the fabric layers in uniform orientation

Region	# of layers	$V_f$ [%]	$k_x$ [mm <sup>2</sup> ]	$k_y$ [mm <sup>2</sup> ]
Preform mat	11	41.7	72.47e−6	17.21e−6
Patch region	16	60.7	1.19e−6	15.32e−6

**Fig. 2** Left: Geometric measures of an exemplary chosen test configuration. Middle: Three properties combined as channels of the output image. Right:  $V_f$ ,  $k_x$  and  $k_y$



by interpolating into the corresponding results of principal in-plane permeability  $k_x$  and  $k_y$  vs. volume-averaged fiber volume content  $V_f$ . This  $V_f$  was estimated according to

$$V_f = \frac{nm_A}{h\rho_f} \quad (3)$$

with the number of layers  $n$ , areal weight (grammage)  $m_A$ , cavity height  $h$  and fiber mass density  $\rho_f$ . The areal weight was determined by weighing each preform prior to the flow experiment. The mass density of the glass fibers was taken from the technical data sheet with  $2550 \text{ kg/m}^3$ . The material shows principal flow directions well aligned with the orientation of the woven fiber bundles and thus:  $k_{xy} \approx 0$ , cf. Equation 2.

### 2.1.3 Preform and Patches

The flow experiments and resulting data presented in Sect. 2.2 are based on specifically manipulated preforms of the woven fabric. The preform is created by homogeneous stacking of the fabric layers in uniform orientation. A rectangular patch of the fabric was introduced in a particular location of the

preform, varying in terms of: (i) number of layers, (ii) orientation of the fabric, (iii)  $V_f$ , and thus, (iv) in-plane permeability. Table 1 lists the most relevant properties of preform and patch region.

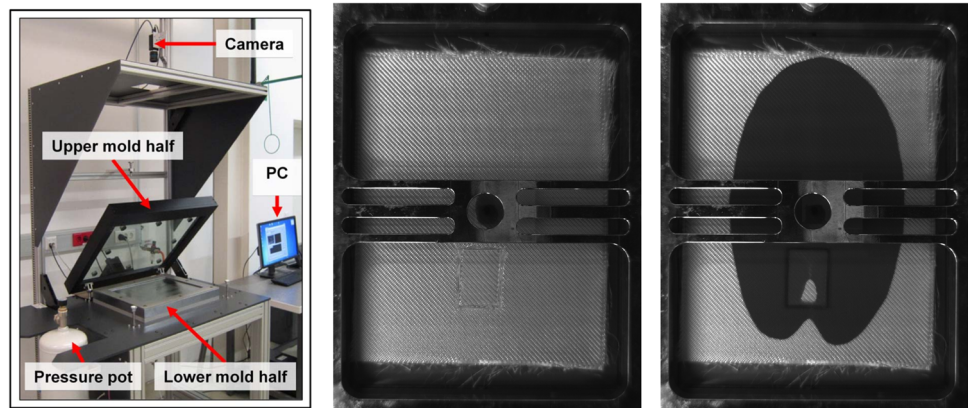
Several different patch locations were chosen for the flow experiments to provide a meaningful data set for the subsequent training of the neural network. Figure 2 shows the geometric details of such a configuration.

## 2.2 Data from flow experiments

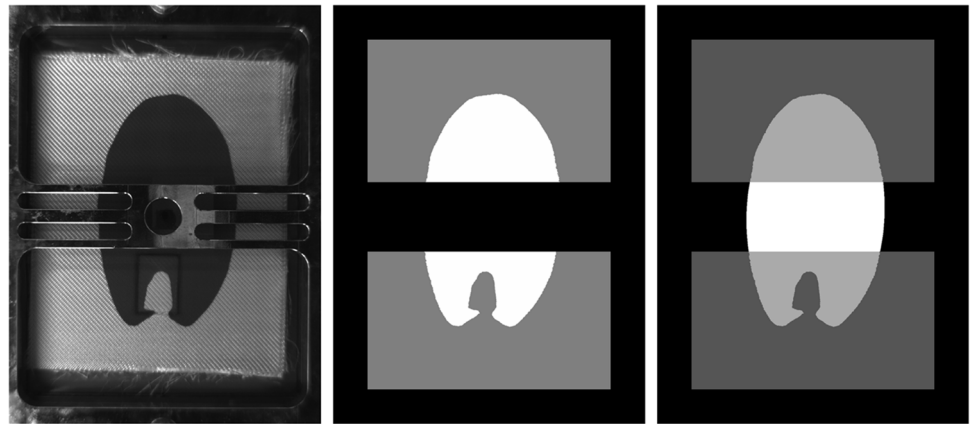
### 2.2.1 Optical permeameter

For the study at hand, an optical permeability characterization cell (briefly termed *permeameter* for the remainder of the work) was used, which follows the radial flow technique combined with optical flow front tracking. Using an optical permeameter offers the advantage that almost the entire flow front can be tracked in the form of planar *images* instead of sensory time series at certain locations only or, in current industrial practice, even no in-mold sensors at all. Moreover, the flow front images acquired (cf. Figure 3) align closely

**Fig. 3** Permeameter experiments: Left: In-plane permeameter testrig with an optically transparent upper mold half. Middle: Beginning of the injection trial in permeameter, a dark spot is visible between the frame. Right: the injection is in progress, with a visible deviation of the flow front caused by the patch with changed permeability to provide a test ground. This is part of the the experimental test plan described in Sect. 2.2.3



**Fig. 4** Evaluation procedure on an example image: raw camera image (left), label image (center) and locally completed label image (right)



with those from an FEM-based process simulation (cf. Figure 1) which enables a sim-to-real transfer approach.

The experimental data used for this work was acquired with the permeameter depicted in (Fig. 3). The mold of this cell can be seen as a flat plate RTM mold with a central injection gate and optically transparent top mold half, which allows for tracking of the flow front using an industrial camera system [19].

### 2.2.2 Image and data processing

To use the experimental data for passing them through a model trained on simulation-only data, the acquired sequence images had to be pre-processed (cf. Figure 4): (i) The images were rotated to align them with the images created by the simulation, (ii) empty zones outside the fabric were removed, (iii) image areas showing the stiffening frame of the mechanical setup were removed and (iv) parts of the flow front, occluded by the stiffening frame, were supplemented by a specifically developed, automated mechanism. The latter involves fitting an elliptical geometry model to selected data points along the fluid flow front [19] and extrapolating parabolic models for the major and minor ellipse axis length [20], respectively.

### 2.2.3 Experimental test plan and data augmentation

Two series of flow experiments were run with specific dedication for the work at hand which are briefly termed REALPERMSMALL (from 2020) and REALPERMLARGE (from

2021). Table 2 provides an overview of the entirety of data sets.

To benefit the most from this limited number of real-world samples, the injection experiments were designed such that the patches were located in one quadrant of the preform only. Data augmentation through flipping the resulting image sequences along the  $x$ - and  $y$ -direction, respectively, yielded additional image sequences

### 2.3 Data from flow simulation runs

To obtain data from simulation runs, briefly termed the *simulated* data, a strategy comparable to previous works of the authors, in particular, [3] and [6], was followed involving an automated pipeline: Starting with a two-dimensional representation of a preform with anisotropic material properties, briefly termed *preform*, small patches (rectangular or circular) with varying in-plane permeability and  $V_f$  were randomly inserted. Subsequently, fluid flow through the preform was numerically predicted through a commercial RTM simulation software, resulting in a sequence of “label images”, or briefly *labels* (see Fig. 1), over the percentage of preform filling.

Simulation data were generated in several evolutionary stages, Table 3 gives an overview of the evolution of data sets generated for this work. The difference in  $V_f$  of the 2021 and 2022 data, respectively, and the setup specified in Table 1 arises from slight deviations between nominal and real values of the areal weight of the fabric. The latter were measured directly before starting the flow experiments and

**Table 2** Datasets - Reality: Textile: HexForce1202 by Hexcel

Dataset	Year	# of Runs	$V_f$ Preform / Patches (in %)	Properties
REALPERMSMALL	2020	6	40 / 56..61	Evaluation of simulation-trained model on real data (no transfer learning) [6]
REALPERMLARGE	2021	92	40 / 56..61	Study specifically designed for this work

**Table 3** Datasets - Simulation

Dataset	Year	# of Runs (ca.)	$V_f$ Preform; Patches (in %)	Textile	Properties
SIMISO	2019	40k	26.8; 20..80	Natural Fiber (iso.)	First dataset, for [3]
SIMISOCUSTOM	2020	10k	26.8; 20..80	Natural Fiber (iso.)	Custom uniform subsampling, for [6]
SIMANISO	2021	10k	40; 56..61	HexForce1202 (aniso.)	Closer to reality through use of preform data from flow experiments (Section 2.2)
SIMANISOCHANNEL	2022	10k	40; 56..61	HexForce1202 (aniso.)	Introduction of flow channels to represent “race tracking” effects

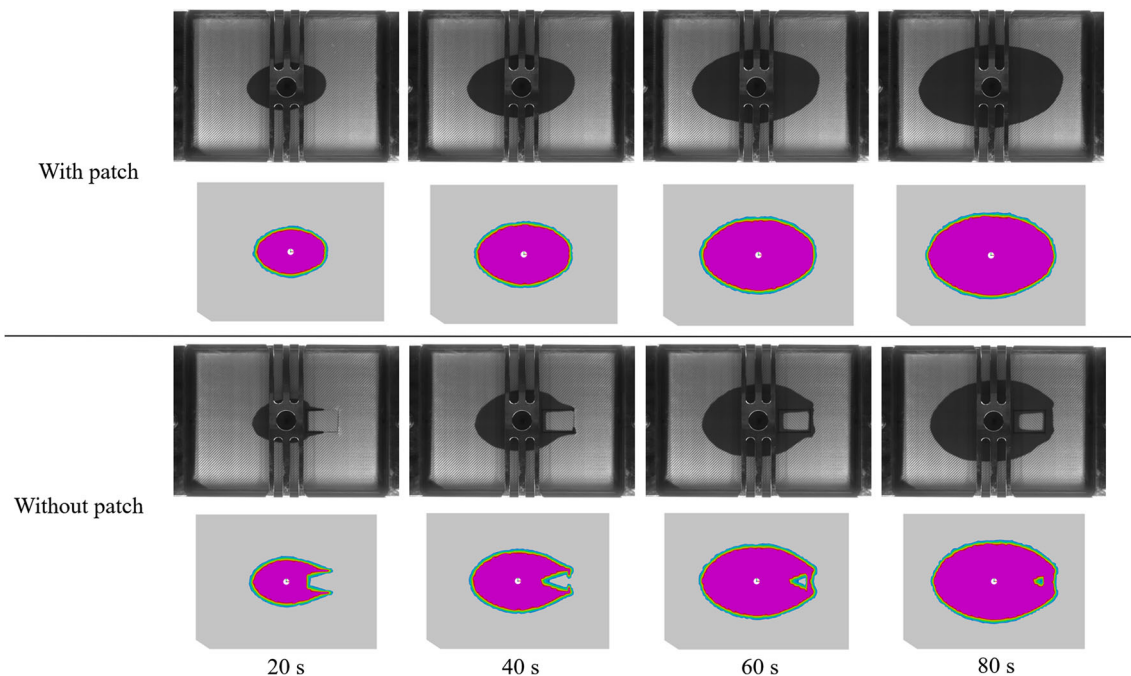
were found within the tolerances specified in the technical data sheet of the material.

The most important aspects of the data sets are specified in the “properties” column of Table 3. It is an evolution towards more realistic flow simulations. The first campaign, SIMISO in 2019, represents a very first try targeting a single image classification task, where the sampling over time did not matter. The SIMISOCUSTOM data set from 2020 introduced regular sampling but was still based on isotropic material with (unrealistically) strong changes in  $V_f$ . With the creation of the SIMANISO data set in 2021, the simulation was driven further towards reality: preform properties were chosen according to real flow experiments and anisotropic in-plane permeability was introduced as an extra layer of complexity. Furthermore,  $V_f$  for both, preform and patch

region, was sampled according to the probability found in the flow experiments (cf. Figure 12 in the Appendix). Finally, SIMANISOCHANNEL represents a data set covering geometrically small (i.e. 1..2 mm) flow channels along the patch edges in order to reflect “race tracking”, an effect well known for LCM [21].

In Fig. 5 we show the timing of simulation and reality in two different cases: one without a patch of different  $V_f$ , thus homogenous preform and one with a patch of higher  $V_f$ . It shows great correspondence of the simulation carried out with PAM RTM and the real experiment, the flow in y direction in the particular example with patch is a little slower.

In the following subsections, particularities of flow simulations in preforms with (i) anisotropic in-plane permeability and (ii) inhomogeneities introduced by patches with



**Fig. 5** Comparison of two cases in simulation and reality: one without a patch of changed  $V_f$  and one with a patch of changed  $V_f$ . Comparisons are made at four different points in time

properties deviating from the preform are described. These are particularly important for the application of machine learning techniques in the FRP domain as the simulation-reality gap is widened, if not taken care of.

### 2.3.1 Flow in isotropic preforms

In flow simulations on isotropic material, the material properties can directly be derived from each other, i.e. increasing  $V_f$  results in a lower in-plane permeability, and the changes are equivalent in  $x$ - and  $y$ - direction. In other words, an ML model predicting  $V_f$  would automatically allow for an analytic computation of in-plane permeability, assuming that a corresponding equation is known for a certain material. In addition, the orientation of the patch is irrelevant as it does not affect directional permeability values.

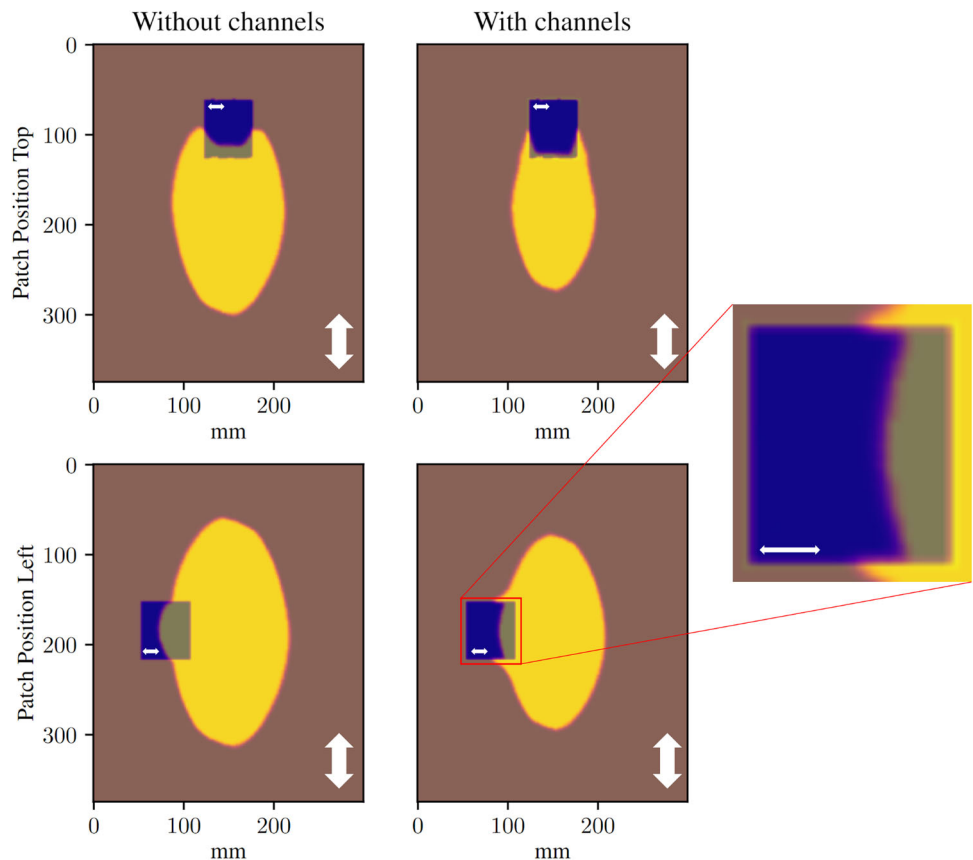
### 2.3.2 Flow in anisotropic preforms

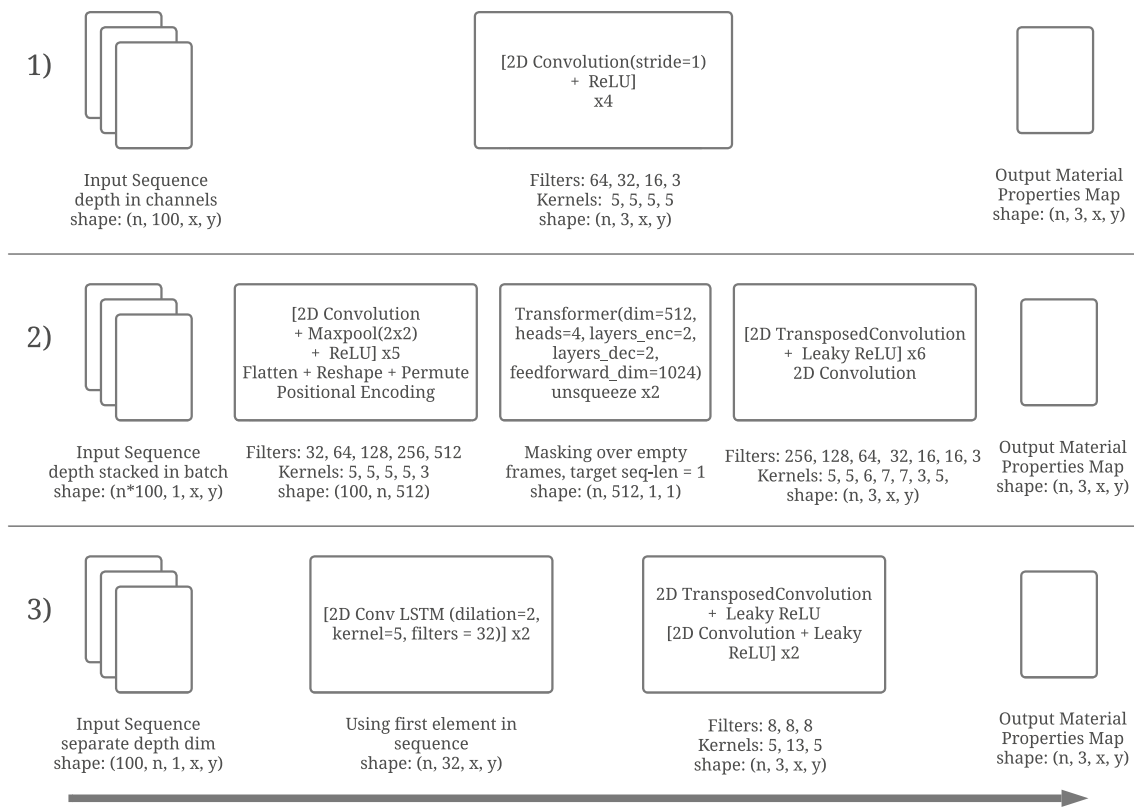
Restricting an ML model to predict only  $V_f$  in an anisotropic material, information about in-plane permeability and their principal directions cannot be deduced. However, it is crucial to know the principal flow directions of the fabric to be

able to reason from fluid flow, in particular when introducing patches with unknown fabric orientation. Hence, consideration of in-plane anisotropy was the major driver towards a model that predicts all three properties, i.e.  $V_f$ ,  $k_x$ , and  $k_y$ , from sequences of flow front images.

In Fig. 6 (left), the effect of anisotropic in-plane permeability in a preform with a rectangular patch is highlighted with two different patch locations. The patch exhibits a higher number of fabric layers and thus, a higher level of  $V_f$ . Moreover, the orientation of the fabric layers in the patch is rotated by  $90^\circ$  with respect to the remaining preform, which is indicated by the arrows pointing in the direction of major in-plane permeability. The top left image shows the expected impact: the flow front is trailing inside the patch region as a result of reduced permeability in the vertical flow direction. However, the bottom left image shows an opposite situation: the flow front inside the patch region is leading compared to the surrounding preform. This effect is caused by (i) the particular relation of in-plane permeability values (see Table 1) and (ii) the level of pressure gradient driving the fluid flow in this particular region of preform and patch, respectively. Although numerically correct, this flow pattern is not observed in the flow experiments. There, the flow front is trailing inside the patch region, which is explained by an additional effect, known as “race tracking”.

**Fig. 6** Filling factors from flow simulation for two different patch locations (left column) revealing the effect of anisotropic in-plane permeability. Impact of considering “race tracking” channels on the flow pattern in the flow simulation for two different patch locations (right column). Arrows indicate the orientation of major in-plane permeability in preform and patch regions, respectively





**Fig. 7** An overview of the three proposed models: 1) A simple fully 2D convolutional network, 2) a Transformer based approach with a 2D convolutional encoder and decoder and 3) a 2D convolutional LSTM network

### 2.3.3 Race tracking

In the simulation environment, the edges along the patch region can be implemented as “ideal”, i.e. without any gap between preform and patch. The mesh properties change instantaneously from a cell of the preform to a cell in the patch region. However, this is not reflecting reality. There, despite the highest possible efforts in manual handling, a gap remains which forms a flow channel around the patch. This channel, typically in the size of 1..2 mm, shows a level of axial permeability which is typically one to two orders of magnitude higher compared with the surrounding fibrous and thus, causes the fluid flow to advance along the edges of the patch. This effect is well-known for LCM and referred to as “race tracking” [21].

To reflect this effect in the flow simulation, 2 mm wide flow channels were added around the rectangular patches of the data set SIMANISOCHANNEL. The permeability of the corresponding mesh cells was specified with an *equivalent isotropic permeability*, also employed in this context by Bickerton et al. [21] which is one order of magnitude higher than the major in-plane permeability of the preform. The results of these changes are shown in Fig. 6 (right) and are considered to be well in line with observations in the flow experiments.

### 2.4 Final remarks

Although a fairly “basic” situation with strictly two-dimensional flow in a plate-shaped mold and homogeneous fabric material was considered in this work, flow simulations meeting the real-world situation require a decent amount of expert knowledge. However, such flow simulations scale very well. Flow experiments by contrast are very costly in a lab or industrial environment and they do scale only linearly. As a result, limited data is available.

To combine data from flow experiments and flow simulations, image and data pre-processing are required as shown in Fig. 4. These pre-processing steps also include the sampling of the image sequences over the filling percentage of the visible flow front, while adding padding to create sequences of the same length.

## 3 Approach

The modified textile property maps provoke altered flow fronts and need to be rediscovered by the ML models. These models get a sequence of flow front images as input, as shown in Fig. 1. In previous work [6], we predicted the  $k_x$  permeability as a first feasibility check of this approach. In this



**Table 4** Results: Simulation Data Test Set - Model Comparison for 3 properties goal:  $k_x$ ,  $k_y$  and  $V_f$ ; including Training Time for one Epoch of 10k samples on 4 GPUs

Model	Acc. Tot	IOU (Mean)	Training Time (One Epoch)	Parameters
CNN	85.802	0.1406	<b>0:24 m</b>	225,315
Transformer	92.191	0.4068	1:45 m	17,293,691
ConvLSTM	<b>96.893</b>	<b>0.6943</b>	27:30 m	328,491

work, we predict all three major properties of the fluid flow through the textile:

- $V_f$  - volume-averaged fiber volume content
- $k_x$  - permeability in x direction
- $k_y$  - permeability in y direction

### 3.1 Data split

The training/test and validation set were fixed over all experiments. All real data samples flipped both in  $x$  and  $y$  direction made up the test set, while the training and validation sets were made up of the remaining three flip configurations: not flipped, flipped in  $x$ , and flipped in  $y$  direction.

This is a reasonable split of the data since all our models use convolutions for feature extractions. Convolutions are invariant to translation but, in general, not to rotation [22], so the model must generalize well, to be able to predict dry spots in places with a different flow front direction.

### 3.2 Sim-to-real transfer learning

We use Sim-to-Real Transfer Learning for our models. First, the models are trained on a larger data set from simulation only. Then, the trained networks are trained again on a smaller data set containing data from real experiments with a smaller learning rate. Decreasing the learning rate in the second training step helps to not alter the weights of the network too much, which can lead to decreasing performance. In general, this approach enables the usage of much larger and more capable models for data sets that are normally too small, as described in Sect. 1.1. In this Transfer Learning approach, the model trained on simulated data is used as an initializa-

**Table 5** Results: Simulation-pretrained networks with real data as test set and pretrained networks with retraining with 10 real samples - Model comparison for 3 properties, goal:  $k_x$ ,  $k_y$ ,  $V_f$

Model	Acc. Tot	IOU (Mean)
CNN (Real Data Test Set)	64.901	0.1985
CNN (Transfer)	81.964	0.2262
Transformer (Real Data Test Set)	93.404	0.3321
Transformer (Transfer)	<b>95.331</b>	<b>0.3826</b>
ConvLSTM (Real Data Test Set)	91.783	0.3375
ConvLSTM (Transfer)	94.148	0.3333

tion point or warm start for the model trained on real-world data. This helps the model trained on real-world data to learn faster and achieve better performance.

### 3.3 Models

All models are trained end-to-end, from the input sequence of flow front images to the property map (cf. Figure 7).

Several neural networks are suitable to address this 3-dimensional sequence-to-instance task. As a baseline model, we employed a convolutional neural network (CNN) with four conv2D layers. From a subsampling step, we get 100 single-channel images of the time steps of the injection process. Since 2D convolutions work over any number of channels, we use the 100 time steps as independent channels for the first convolutional layer.

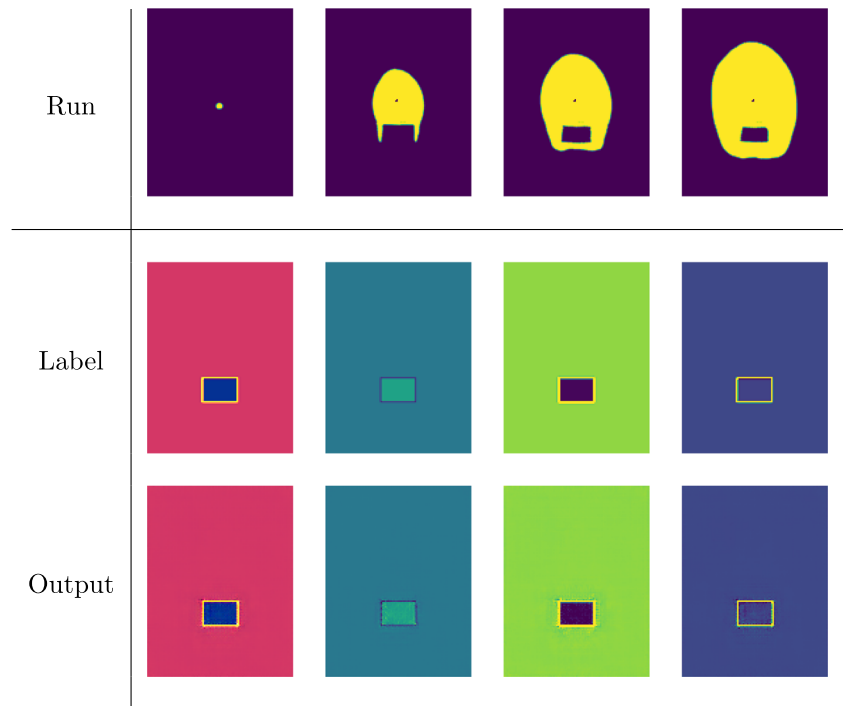
The second proposed model relies on the Transformer mechanism [23] which works on one-dimensional embeddings. Hence, the input image sequence needs to be converted into an embedding sequence. To do so, a fully convolutional encoder creates feature vectors for the individual images and is trained end-to-end in the sequence-to-instance pipeline. For the encoder part, the sequence length is fused with the batch dimension, so the convolutions work over every single picture separately. After the encoder, the dimensions are restored to utilize the sequence capabilities of a Transformer (cf. Fig. 7). The Transformer output vector is used to create the property map using 2D transposed convolutions. Summarizing, the model has an encoder -- Transformer -- decoder structure.

The last approach emphasizes the temporal aspect by using a Convolutional Long Short Term Memory (ConvLSTM) architecture [24]. In contrast to regular LSTMs, ConvLSTMs work over sequences of two-dimensional matrices, instead of one-dimensional vectors, which makes them suitable for our task.

**Table 6** Overview: Effects of Sim-to-real pre-training for the Transformer

Model	Acc. Tot	IOU (Mean)
Real data only: 240 samples	80.323	0.2143
Sim data only + No re-training	93.404	0.3321
Transfer: 10 real Samples	95.331	0.3826
Transfer: 240 real samples	<b>95.929</b>	<b>0.5031</b>

**Fig. 8** Excellent example: Input data (top row) Label and output of best model: Transformer, pretrained on SIMANISOCHANNEL, retrained on real data. The labels and outputs are in the following order: Second row: 3 properties together,  $V_f, k_x, k_y$ ; Third row: Outputs in the same order. This specimen shows very well-defined patches with sharp borders

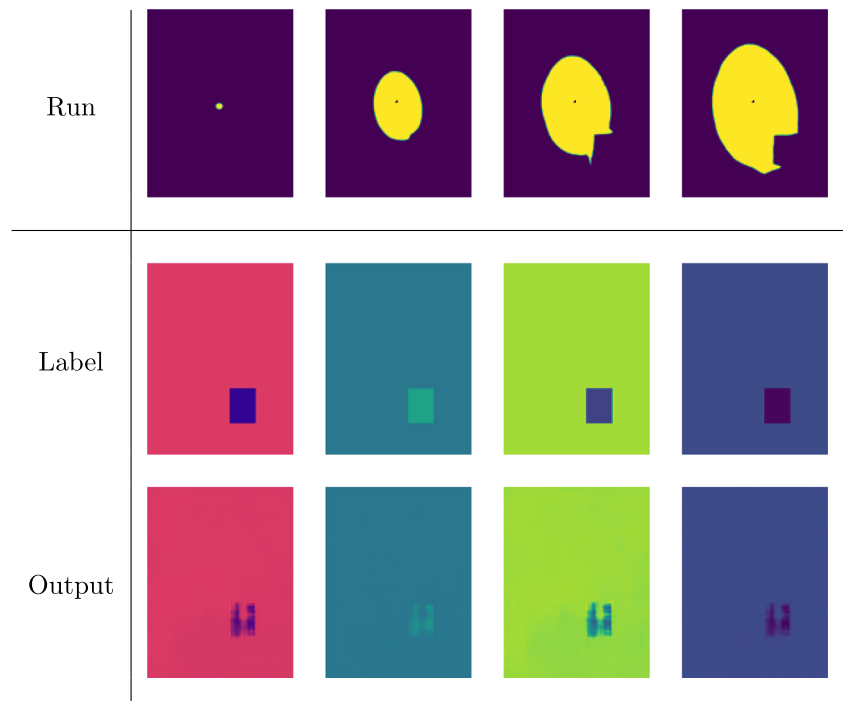


We improved all models compared to [6] by using leaky ReLU instead of ordinary ReLU activation functions in the image reconstruction part and by adding batch normalization to the network. These changes resulted in faster and more stable training.

#### 4 Evaluation

Based on Stieber et al.'s work [6], which involved using networks trained on simulated data for inference on real data, we wanted to investigate the following research questions:

**Fig. 9** Underwhelming example in comparison to Fig. 8, here the borders of the patches are not as well defined



**Table 7** Transformer model pre-trained on two different data sets of different levels of simulation fidelity: SIMANISO (SA) and SIMANISOCHANNEL (SAC)

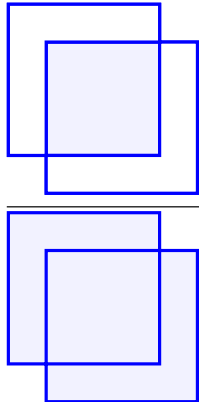
Model	Data set	Acc. Tot	IOU (Mean)
Transformer	SA	92.079	0.1648
Transformer (Transfer on 10 samples)	SA	93.974	0.3036
Transformer	SAC	93.404	0.3321
Transformer (Transfer on 10 samples)	SAC	95.331	0.3826

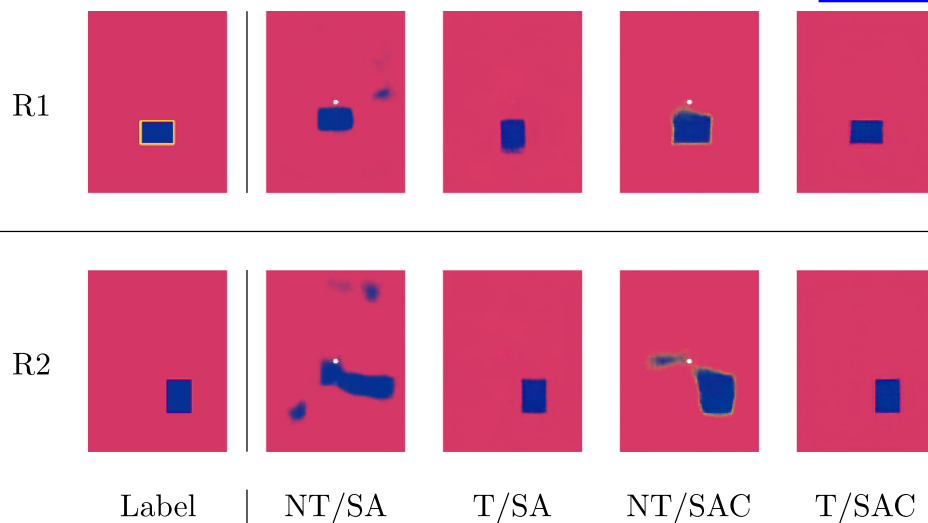
1. How do the network architectures proposed in Sect. 3.3 compare against each other?
2. Are the real-world samples enough to learn the task by themselves and how can simulation data help?
3. How much performance can be added by retraining neural networks with real data that have been pre-trained on simulated data?
4. How does the simulation fidelity affect the output of the trained models?

### 4.1 Metrics

But first, we had to select a set of suitable metrics apart from the “expert eye” that were used on the output images. Finding a statistic that accurately reflects observed performance proved to be more challenging than anticipated because, for example, *pixel wise accuracy* frequently emphasizes “blurry” forecasts. Since a significant section of the property maps had nearly similar base properties, accuracy alone was not enough. Therefore, we permitted  $\epsilon$ -tolerances to continue classifying specific pixels as correct. This  $\epsilon$  between prediction and label was manually determined to be 0.03. Values above 3 % reduce the metric’s sensibility, whereas values below 3 % increase it. Aside from that, intersection over union (IOU) was our method. This metric is concerned with the size and position of the introduced patch, which is the extent of the observed variation in the attributes. To calculate it, we introduced certain thresholds for every

property, based on the label, for instance: for  $V_f$ , within the REALPERMLARGE dataset (from 2021), the lower bound would be 0.38 whereas the upper bound is 0.41. As listed in Table 2,  $V_f$  was designed to be 0.4, but manual handling of the textile leads to this (measured) spread between 0.38 and 0.41. These values stem from inspecting the overall dataset for one property (here  $V_f$ ), binning it into 100 between 0 and 1, and taking the bins with the most entries as the bins that represent the most of the textile. Thus, for  $V_f$  in this case, the border values are 0.38 and 0.41. Afterward, a binary image is made with pixels falling into the desired window of values and values out of that window, i.e. black pixels would be all values  $> 0.41$  and  $< 0.38$  and white pixels would be the rest. Then the IOU is calculated according to Eq. 4, with this binary image and a binary version of the label.

$$IOU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{\text{area of overlap}}{\text{area of union}} \tag{4}$$




**Fig. 10** Label and output of aggregated channels from different simulation datasets: SIMANISO (SA) has no channels for race tracking, SIMANISOCHANNEL (SAC) has channels. Matches Table 7

The IOU metric is calculated for all three different properties independently and we also produce the mean of the three values. For accuracy, we also calculate both the three independent values and one of all channels together. In most tables in the following, we show the aggregated metrics of the values but we also added more extensive evaluations with all independent metrics to Sect. 1 (appendix).

## 4.2 Results

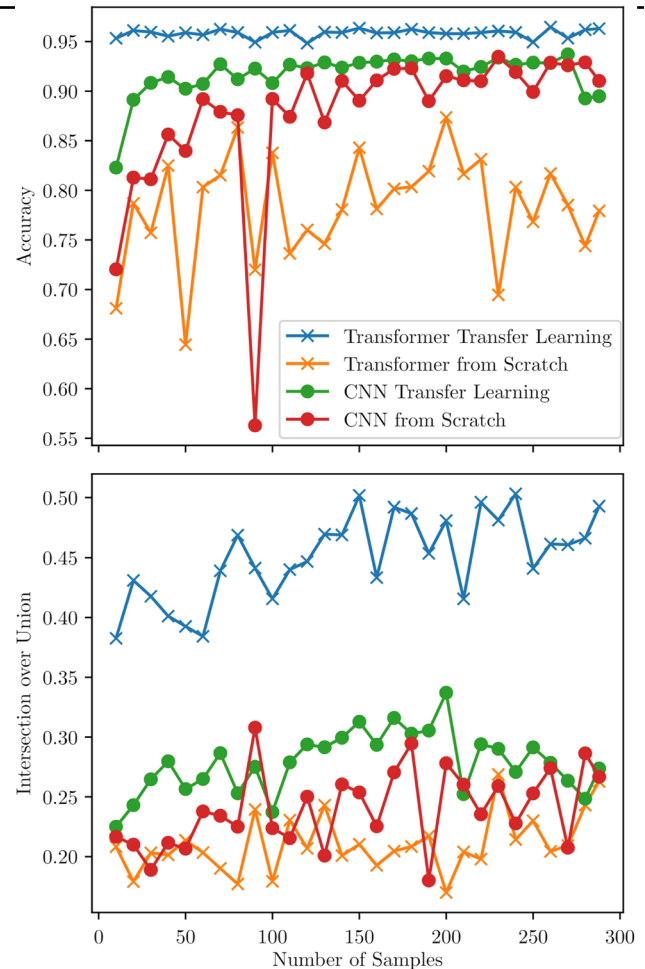
### *How do the network architectures proposed in Section 3.3 compare against each other?*

To start with the first research question, we show the resulting metrics from all three proposed architectures on the SIMAN-ISOCHANNEL data set in Table 4. It shows that the ConvLSTM yields the best results on the simulation test set in terms of both accuracy and IOU, albeit at the longest training duration. *Are the real-world samples enough to learn the task by themselves and how can simulation data help?*

Table 5 shows that the Transformer outperforms the ConvLSTM when only using real data as a test set and when using only 10 real samples for re-training, which answers research question 2. These were sampled from the REALPERMSMALL and REALPERMLARGE datasets (cf. Section 2.2) to have a good starting point of what only few samples can do. For the retraining step, no weights were frozen and the learning rate was kept constant at  $1e-4$ . It is also important to state that the Transformer trains much faster than the ConvLSTM and thus yields results more promptly during development and evaluation (cf. Table 4).

### *How much performance can be added by retraining neural networks with real data that have been pre-trained on simulated data?*

From this point, we investigated, how much the best model – the retrained Transformer benefits from additional real data for training. This is a particularly interesting question since the question before implementing data-driven models for specialized use cases often is: “How much real data are you going to need?” We address this question by using our augmented training data set of 288 injections and training the Transformer model both from scratch and from a checkpoint stemming from the training on the simulated data. An overview of how the models benefit from pre-training is given in Table 6. The results can also be seen in Fig. 11. A possible explanation is that when the Transformer is trained from scratch using only the real data, it is only able to predict dry spot presence, ignoring the offset to left or right and the orientation of the patch. However, it can differentiate between dry spot locations in the Y direction. The number of samples has seemingly no impact on the overall performance; the metrics are erratic and stay on a certain level. When the training is started with a sim-data-only checkpoint, the performance is much better: predictions of orientation, location,



**Fig. 11** Accuracy and IOU over different data set sizes (from REALPERMLARGE and REALPERMLARGE) on with CNN and Transformer (Transfer from models pre-trained on SAC)

and extent are much more accurate which is also reflected by the metrics. Additionally, the performance increases almost linearly with the number of real data samples used.

When using different networks on different amounts of data, we take into account that the Transformer has many parameters and thus may show underwhelming performance on a data set as small as the real-world data set we use. Therefore we utilize the CNN with its much smaller parameter number as a second baseline for real-data-only training. In Fig. 11 we compare the CNN and the Transformer regarding real-data-only learning and transfer learning capabilities. The figure shows that while the parameter-wise smaller CNN outperforms the bigger Transformer accuracy-wise, they are on par in terms of IOU. But when leveraging the simulation data via pre-training, the Transformer outperforms the CNN by far.

The overall best metrics are hit at a data set size of 240: IOU 0.5031 (up 0.12 from 10 samples) and Accuracy 95.929 % (up 0.6 % from 10 samples). This answers research ques-

tion 3, one example of very well defined property maps is shown in Fig. 8 and another in Fig. 9, with less perfect outcomes. The latter occurs much more seldom than the former, but we wanted to include imperfect outcomes in the evaluation as well.

#### ***How does the simulation fidelity affect the output of the trained models (SA vs SAC)?***

To investigate research question 4, on how the quality of the simulation influences the overall performance of the Sim-to-Real Model, we took the best-performing model, the Transformer, and tested it on a lower-quality simulation data set. In Table 3, and the paragraph on anisotropy (2.3.2), we explain why using anisotropic textile features changes the phenomenology of flow fronts drastically and why we introduced channels to simulate race-tracking. To test our model on a dataset that is farther away from showing realistic phenomena, we tested the Transformer on a dataset without channels for race-tracking. The comparison can be made in Table 7. The accuracies are all over 90 %, with better values for the transfer-learning models each, and the models pre-trained on the more elaborate SIMANISOCHANNEL data are each better than their SIMANISO counterpart. When inspecting the IOU values, it becomes obvious, that the SIMANISOCHANNEL data models perform better than the SIMANISO models. This difference can also be observed in Fig. 10. While both transfer learning models show better results than the models trained on simulation data only, there are obvious differences in performance when observing “R1”: The Transfer SA (T/SA) model produces a misaligned patch compared to the Transfer SAC (T/SAC) model. When inspecting the models that were only trained on simulation data and have never seen real data in training, the differences are even more apparent: In “R2”, the No Transfer SA (NT/SA) model produces strong artifacts whereas the No T/SAC model shows light artifacts and produces the patch in the almost correct size and position. These excerpts from the dataset reinforce the claim taken from the metrics, that better simulation datasets ease the transfer learning process from simulation to reality (Fig. 11).

## **5 Discussion and future work**

In-mold sensor data would be necessary to apply the results of this paper to an industrial setting [5]. A modular approach could first take sensor data and map them to flow front images and then feed those outputs directly into the models presented here. Alternatively, the mapping from sensor data to permeability deviation maps could be learned end-to-end, given a small dataset of respective real data.

## **6 Conclusion**

We set out to investigate how machine learning can be used to infer material properties during the RTM process. Generating huge amounts of real data for the RTM process is time-consuming, expensive, and therefore undesirable. Instead, we chose to leverage sim-to-real learning, by generating large amounts of simulated RTM data, and only a small amount of real data. We trained three different models, a simple Conv2D net, a ConvLSTM, and a novel Encoder-Transformer-Decoder and evaluated their performance. Our research shows that leveraging simulated data for pretraining can greatly improve performance compared to a model trained solely on a small amount of real data. Further, we investigated how little data is truly needed to have a real-world improvement. We found that even as little as 10 samples can be enough if the data is generated carefully with respect to expert knowledge. To add a little more detail, using 10 real samples with a network that was pre-trained with a well-crafted simulation, we gain 25 percent points in accuracy on a non-pretrained network. In terms of IOU, we gain 0.20 points. The difference between two data sets based on better or worse simulations is 1.4 percent points in accuracy, from 93.9 to 95.3 and 0.08 in IOU, from 0.30 to 0.38. That shows that only small amounts of data can be used when a simulation for pretraining is available and better simulations leverage the results even further. Additionally, we showed how the quality of the simulated data can impact model performance. Our results suggest that keeping the domain shift between real and simulated as small as possible can help to improve the quality of the predictions.

**Author Contributions** All authors contributed to the study conception and design, the main conceptual idea was formed by Simon Stieber advised by Wolfgang Reif. Material preparation, data collection were performed by Ewald Fauster and Marcel Bender. Preprocessing and analysis of experimental data was done by Simon Stieber and Ewald Fauster. Simulation design, execution and data collection was done by Simon Stieber. The main machine learning experiment design was executed by Simon Stieber and Alexander Schiendorfer. Machine learning model and metric design and code implementation was done by Niklas Schröter and Simon Stieber. The initial draft was mainly written by Simon Stieber and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

**Funding** Simon Stieber was funded by the Bavarian State Ministry of Economic Affairs and Media, Energy and Technology (StMWi) within the project MAI CC4 CosiMo (Composites for sustainable Mobility). The experiments conducted in the REALPERMLARGE experiment campaign at the Processing of Composites Group (Montanuniversität Leoben) in 2021 were also financed by this project. All other authors declare no conflicts of interest. Open Access funding enabled and organized by Projekt DEAL.

### **Competing Interests**

The authors have no relevant financial or non-financial interests to disclose

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## Appendix

### Detailed results

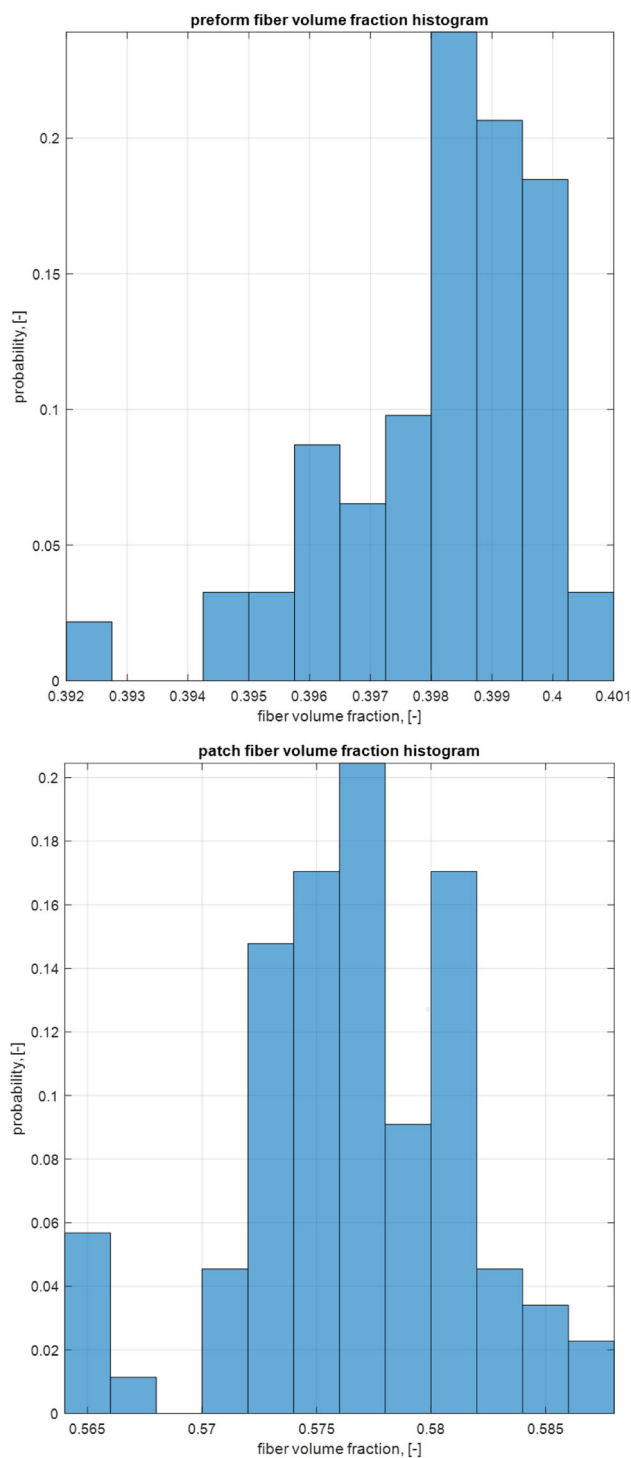
**Table 8** No pre-training Transformer - detailed results with different amounts of data; Aggregated values shown in Fig. 11

Samples	Loss	Acc. Tot	Acc. $V_f$	Acc. $k_x$	Acc. $k_y$	IOU $V_f$	IOU $k_x$	IOU $k_y$	IOU Mean
10	0.0165422	68.099%	72.523%	50.257%	81.516%	0.034013	0.344049	0.246932	0.208331
20	0.0158130	78.690%	82.139%	61.143%	92.788%	0.044389	0.339176	0.153751	0.179105
30	0.0159587	75.717%	82.690%	54.952%	89.510%	0.043476	0.354622	0.211363	0.203154
40	0.0161202	82.501%	89.477%	67.178%	90.847%	0.073880	0.340391	0.190077	0.201449
50	0.0164062	64.440%	71.970%	33.255%	88.094%	0.034064	0.37213	0.234532	0.213575
60	0.0161861	80.329%	88.660%	60.618%	91.708%	0.065708	0.357713	0.186732	0.203384
70	0.0161451	81.508%	88.427%	64.143%	91.955%	0.065586	0.336541	0.168408	0.190178
80	0.0161155	86.332%	90.629%	75.347%	93.021%	0.108479	0.297488	0.125445	0.177137
90	0.0166188	71.991%	86.392%	38.661%	90.921%	0.047686	0.38306	0.287587	0.239444
100	0.0163216	83.778%	90.016%	68.560%	92.756%	0.071704	0.32526	0.141037	0.179334
110	0.0166828	73.637%	81.038%	48.240%	91.631%	0.033620	0.369688	0.288263	0.230524
120	0.0187994	76.010%	86.828%	49.563%	91.640%	0.039363	0.366596	0.214816	0.206925
130	0.0160387	74.607%	83.895%	47.241%	92.685%	0.040746	0.378602	0.309703	0.243017
140	0.0177869	78.064%	88.949%	53.281%	91.963%	0.057633	0.370932	0.173969	0.200845
150	0.0170197	84.299%	91.089%	69.182%	92.626%	0.074312	0.353615	0.203177	0.210368
160	0.0168831	78.123%	88.521%	53.275%	92.574%	0.046457	0.369295	0.162155	0.192636
170	0.0169577	80.149%	84.661%	63.163%	92.623%	0.047978	0.375136	0.190836	0.204650
180	0.0170416	80.345%	89.339%	59.471%	92.223%	0.068956	0.367028	0.189216	0.208400
190	0.0172743	81.931%	92.167%	62.078%	91.547%	0.067358	0.364322	0.220082	0.217254
200	0.0159881	87.352%	91.832%	77.182%	93.042%	0.082572	0.316509	0.110834	0.169972
210	0.0164499	81.665%	89.767%	62.699%	92.531%	0.061473	0.375503	0.174768	0.203915
220	0.0147800	83.130%	93.862%	62.403%	93.124%	0.078310	0.352127	0.163489	0.197975
230	0.0137620	69.460%	90.057%	49.717%	68.607%	0.041242	0.381856	0.383208	0.268769
240	0.0149745	80.323%	93.218%	54.109%	93.642%	0.096225	0.355923	0.190733	0.214294
250	0.0160068	76.830%	94.091%	44.708%	91.690%	0.097430	0.378871	0.213739	0.230013
260	0.0188773	81.699%	91.560%	60.962%	92.577%	0.051598	0.371917	0.189565	0.204360
270	0.0141966	78.513%	91.595%	52.164%	91.779%	0.062836	0.367261	0.197622	0.209240
280	0.0150943	74.397%	94.389%	38.747%	90.056%	0.061794	0.376857	0.290734	0.243128
288	0.0182947	77.929%	93.754%	48.260%	91.772%	0.068721	0.384171	0.335697	0.262863

**Table 9** Pretrained Transformer - transfer learning from simulation data - detailed results with different amounts of data; Aggregated values shown in Fig. 11

Samples	Loss	Acc. Tot	Acc. $V_f$	Acc. $k_x$	Acc. $k_y$	IOU $V_f$	IOU $k_x$	IOU $k_y$	IOU Mean
10	0.0133460	95.331%	97.173%	92.650%	96.171%	0.612649	0.292418	0.242679	0.382582
20	0.0129446	96.118%	98.076%	93.549%	96.730%	0.621275	0.253385	0.418366	0.431009
30	0.0129888	95.955%	98.143%	92.732%	96.990%	0.651668	0.234002	0.367467	0.417712
40	0.0121212	95.546%	97.518%	92.551%	96.569%	0.600641	0.230532	0.372306	0.401160
50	0.0108170	95.888%	97.806%	93.110%	96.748%	0.623077	0.247487	0.307187	0.392584
60	0.0104289	95.682%	97.411%	93.076%	96.558%	0.550819	0.274899	0.327013	0.384244
70	0.0110687	96.243%	97.905%	93.896%	96.928%	0.646681	0.248117	0.422256	0.439018
80	0.0106635	95.918%	97.996%	92.885%	96.874%	0.684139	0.274285	0.448208	0.468877
90	0.0094973	94.933%	97.918%	90.084%	96.795%	0.648591	0.255770	0.419495	0.441285
100	0.0096477	95.921%	97.689%	93.384%	96.690%	0.603805	0.245180	0.397894	0.415626
110	0.0087622	96.131%	97.833%	93.824%	96.734%	0.662984	0.322421	0.334758	0.440054
120	0.0094640	94.832%	97.886%	89.805%	96.804%	0.660667	0.300703	0.379054	0.446808
130	0.0087991	95.962%	98.008%	92.982%	96.895%	0.666404	0.326243	0.416300	0.469649
140	0.0102821	95.913%	97.787%	93.136%	96.816%	0.622517	0.357582	0.426917	0.469005
150	0.0094783	96.354%	97.961%	94.064%	97.036%	0.680125	0.384144	0.441130	0.501800
160	0.0103803	95.868%	97.476%	93.478%	96.649%	0.578461	0.340827	0.380739	0.433342
170	0.0101718	95.896%	97.992%	92.654%	97.043%	0.684960	0.378962	0.412551	0.492158
180	0.0097338	96.248%	97.620%	94.343%	96.780%	0.611421	0.387581	0.461896	0.486966
190	0.0109057	95.905%	97.851%	92.958%	96.905%	0.630847	0.338971	0.391042	0.453620
200	0.0089327	95.791%	97.984%	92.462%	96.928%	0.677504	0.356018	0.409484	0.481002
210	0.0135155	95.793%	97.406%	93.160%	96.812%	0.561755	0.323209	0.361790	0.415585
220	0.0090755	95.904%	97.981%	92.615%	97.117%	0.676223	0.377183	0.435174	0.496193
230	0.0105710	96.066%	97.802%	93.543%	96.853%	0.653986	0.374656	0.415561	0.481401
240	0.0095079	95.929%	97.660%	93.163%	96.965%	0.672315	0.395464	0.441653	0.503144
250	0.0103522	94.946%	97.705%	90.302%	96.831%	0.604031	0.340273	0.378697	0.441000
260	0.0098892	96.459%	97.790%	94.708%	96.880%	0.591426	0.360057	0.433019	0.461501
270	0.0101265	95.339%	97.763%	91.322%	96.933%	0.622377	0.373093	0.387097	0.460856
280	0.0098410	96.187%	97.826%	93.760%	96.975%	0.611177	0.376601	0.411001	0.466260
288	0.0099005	96.312%	97.891%	93.961%	97.083%	0.656179	0.373017	0.449664	0.492953

## Further information on real data



**Fig. 12** Distribution of  $V_f$  in the preform (top) and the patch region (bottom) in the real-world permeameter experiments

## References

- Babb DA, Richey WF, Clement K, Peterson ER, Kennedy AP, Jezic Z et al inventors; Dow Chemical Co, assignee. Resin transfer molding process for composites. <https://patents.google.com/patent/US5730922A/en>
- Tifkitsis KI, Skordos AA (2020) Real time uncertainty estimation in filling stage of resin transfer molding process. *Polymer Composites* 41(12):5387–5402. <https://doi.org/10.1002/pc.25803>
- Stieber S, Schröter N, Schiendorfer A, Hoffmann A, Reif W (2021) FlowFrontNet: Improving carbon composite manufacturing with CNNs. In: ECML PKDD. Springer International Publishing pp 411–426. [https://link.springer.com/10.1007/978-3-030-67667-4\\_25](https://link.springer.com/10.1007/978-3-030-67667-4_25)
- Achzet M, Schlech T, Linscheid FF, Faber J, Sause MGR (2022) Ultraschallbasierte prozessüberwachung am beispiel eines T-RTM prozesses. In: DGZfP-Jahrestagung pp 1–11
- Stieber S, Hoffmann A, Schiendorfer A, Reif W, Beyrle M, Faber J et al (2020) Towards real-time process monitoring and machine learning for manufacturing composite structures. In: IEEE Symposium on Emerging Technologies and Factory Automation, ETFA. vol. 2020-Sept pp 1455–1458
- Stieber S, Schroter N, Fauster E, Schiendorfer A, Reif W (2021) PermeabilityNets: Comparing neural network architectures on a sequence-to-instance task in cfrp manufacturing. In: 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE pp 694–697. <https://ieeexplore.ieee.org/document/9680220/>
- Darcy HPG (1856) Les fontaines publiques de la ville de Dijon. V. Dalmont
- Wei BJ, Chang YS, Yao Y, Fang J (2016) Online estimation and monitoring of local permeability in resin transfer molding. *Polymer Composites* 4;37(4) pp 1249–1258. <https://doi.org/10.1002/pc.23290>
- Matveev M, Endruweit A, Long A, Iglesias M, Tretyakov M (2021) Part I: Bayesian inversion algorithm for estimation local variations in permeability and porosity of reinforcements using experimental data. *Composites Part A* 143(2021):1–22. <https://doi.org/10.1016/j.compositesa.2021.106323>
- González C, Fernández-León J (2020) A machine learning model to detect flow disturbances during manufacturing of composites by liquid moulding. *J Compos Sci* 4(2):71. <https://doi.org/10.3390/jcs4020071>
- Caglar B, Broggi G, Ali MA, Orgéas L, Michaud V (2022) Deep learning accelerated prediction of the permeability of fibrous microstructures. *Composites Part A: Applied Science and Manufacturing* 158 (Feb) <https://doi.org/10.1016/j.compositesa.2022.106973>
- Weiss K, Khoshgoftaar TM, Wang D (2016) A survey of transfer learning. *J Big Data* 12;3(1):9. <https://doi.org/10.1186/s40537-016-0043-6>
- Stieber S (2018) Transfer learning for optimization of carbon fiber reinforced polymer production. *Organic Computing: Doctoral Dissertation Colloquium 2018*:61–74
- Hofer S, Bekris K, Handa A, Gamboa JC, Mozifian M, Golemo F et al (2021) Sim2Real in robotics and automation: applications and challenges. *IEEE Transactions on Automation Science and Engineering* 4;18(2):398–400. <https://doi.org/10.1109/TASE.2021.3064065>
- Fauster E : personal communication
- Chaloupka A : personal communication
- Konstantopoulos S, Fauster E, Schledjewski R (2014) Monitoring the production of FRP composites: A review of in-line sensing methods. *Express Polym Lett* 8(11):823–840. <https://doi.org/10.3144/expresspolymlett.2014.84>



18. May D, Aktas A, Advani SG, Berg DC, Endruweit A, Fauster E et al (2019) In-plane permeability characterization of engineering textiles based on radial flow experiments: A benchmark exercise. *Composites Part A* 121:100–114. <https://doi.org/10.1016/j.compositesa.2019.03.006>
19. Fauster E, Berg DC, Abliz D, Grössing H, Meiners D, Ziegmann G et al (2019) Image processing and data evaluation algorithms for reproducible optical in-plane permeability characterization by radial flow experiments. *J Compos Mater* 53(1):45–63. <https://doi.org/10.1177/0021998318780209>
20. Fauster E, Berg DC, May D, Blöbl Y, Schledjewski R (2018) Robust evaluation of flow front data for in-plane permeability characterization by radial flow experiments. *Advanced Manufacturing: Polymer and Composites Science* 4(1):24–40. <https://doi.org/10.1080/20550340.2018.1439688>
21. Bickerton S, Advani SG (1999) Characterization and modeling of race-tracking in liquid composite molding processes. *Compos Sci Technol* 11;59(15) pp 2215–2229. [https://doi.org/10.1016/S0266-3538\(99\)00077-9](https://doi.org/10.1016/S0266-3538(99)00077-9)
22. Weiler M, Cesa G (2019) General E(2) - Equivariant steerable CNNs. *Advances in Neural Information Processing Systems* 32(2)
23. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN et al (2017) Attention is all you need. In: *Advances in Neural Information Processing Systems* pp 5999–6009
24. Shi X, Chen Z, Wang H, Yeung DY, Wong WK, Woo WC (2015) Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In: *Advances in Neural Information Processing Systems* pp 802–810 (Jan)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.