

Warpage prediction for fiber reinforced injection molding via geometric feature learning and differentiable FEM

Julian Greif, Nils Meyer

Angaben zur Veröffentlichung / Publication details:

Greif, Julian, and Nils Meyer. 2026. "Warpage prediction for fiber reinforced injection molding via geometric feature learning and differentiable FEM." *Composites Part A: Applied Science and Manufacturing* 204: 109653.
<https://doi.org/10.1016/j.compositesa.2026.109653>.



Warpage prediction for fiber reinforced injection molding via geometric feature learning and differentiable FEM[☆]

Julian Greif^{a,*,} Nils Meyer^{a,b}

^a University of Augsburg, Institute of Materials Resource Management, Augsburg, Germany

^b University of Augsburg, Centre for Advanced Analytics and Predictive Sciences, Augsburg, Germany

ARTICLE INFO

Keywords:

Process simulation
Injection molding
Machine learning
Warpage
Finite element method

ABSTRACT

Injection molding is a popular mass production process for short fiber reinforced components. One of the main production defects is warpage, unwanted deformation, resulting from the thermal history of the injected polymer and the geometry of the part. Ideally, this deformation should be predicted and compensated for before mass production to minimize defective parts. Conventional process simulation is able to predict warpage, but is too computationally expensive to be used in iterative optimization procedures. Hence, we propose a fast approximation method based on machine learning and a custom finite element solver to predict warpage in arbitrary 3D geometries with any injection location. It combines nodal geodesic values and spatial moments to capture local geometry on different scales. Neural networks then predict fiber orientation and initial strain fields for a subsequent warpage calculation in a differentiable finite element software. The latter enables training of the neural network through the solver with warpage based losses, enabling the training with observable deformation data. The approximation pipeline exhibits relative warpage errors of less than 1% on typical injection molded geometries while being six times faster than the conventional simulation. Retraining through the solver with a warpage enhanced dataset to emulate real world data leads to significant improvements in the predictive accuracy.

1. Introduction

Thermoplastic injection molding is a widely adopted economic production process for large quantities of complex, three dimensional parts. A common challenge in injection molding is the unintended deformation of the part after ejection, called warpage, which needs to be understood and dealt with to produce products that are accurate in size and shape. This warpage originates from local variations in shrinkage, due to uneven pressure or temperature distributions, as well as material anisotropy [1]. Material anisotropy often results from process-induced orientation of fibers made from glass or carbon to enhance the mechanical performance. Other origins may be polymer chain alignment or anisotropic crystallinity in semi-crystalline polymers. Anisotropic behavior leads to accumulation of stresses while still constrained in the mold and concurrent deformation after ejection.

Expensive mold making and complex process design make a purely experimental approach to design and production in injection molding cumbersome. Therefore, ambitions to numerically predict the warpage behavior by calculating the thermally and pressure-induced residual stresses date back to the early 1980's [2,3]. They can be classified

mainly according to their description of material behavior, using a *viscoelastic* or a *viscous-elastic* model. The *viscoelastic* model describes the elastic moduli as a transient variable during the cooling process from liquid to solid [4]. This allows for an accurate prediction, but comes with higher complexity in material characterization [5] and higher computational cost. The *viscous-elastic* model simplifies this transient description into a sudden change at a set solidification temperature [6,7]. Above this temperature, the material is modeled as fully liquid, unable to sustain any elastic stresses. Below this temperature, the behavior changes to that of an elastic solid. Although neglecting in-mold stress-relaxation effects, it is a reasonably good approximation for residual stresses [8] and drastically simplifies simulations, making it a popular choice in process simulations.

In order to receive optimal shape accuracy in the final product, it is possible to change the input geometry to account for warpage. Kastelic et al. [9] proposed an iterative approach of locally changing nodal positions until convergence at the desired output shape. Tillmann et al. [10] proposed optimizing the mold cavity using Bayesian Optimization with a Gaussian process surrogate model. Both methods optimize arbitrary

[☆] This article is part of a Special issue entitled: 'AIComp25' published in Composites Part A.

* Corresponding author.

E-mail address: Julian.Greif@uni-a.de (J. Greif).

geometries but need a multitude of costly simulations in the process due to the non-linearity of warpage. Since these are the main computational bottleneck, they would greatly benefit from a fast approximation model that cuts the simulation time.

Numerous machine learning models have been published to predict warpage in injection molding to minimize overall deformation, e.g., by finding optimal process conditions. Several authors [11–13] employed neural networks trained on simulated or experimental datasets to predict the warping behavior of geometries for given processing parameters. Wenzel et al. [14] compared different hybrid machine learning approaches for the prediction of 1D part shrinkage on a single part, showing that the incorporation of feature learning drastically outperforms purely data-driven models. However, these methods focus on the optimization of process conditions for single parts, needing many simulations for each new part.

Fast surrogate models capable of generalizing across diverse geometries have not yet received significant attention in the academic literature. Kriging models have been used as warpage surrogate models [15], allowing the inclusion of parametric geometric changes into the design space [16]. While giving some variability in terms of geometry, this would still need a model retraining for different designs. Uglov et al. [17] computed the shortest paths from the injection point to nodes in different automotive dashboards and combined this information with a gradient boosting model to predict fill time and a subsequent convolutional neural network to predict warpage based on 2D images. While outperforming their naive estimator, the generalization to different geometries is limited by a downscaled pixelated representation of warpage. In previous work, we have proposed a fast approximation model that uses feature engineering such as encoding geodesic distances and the local geometric environment along with gradient boosted trees or feed-forward neural networks to predict nodal values such as fill time distribution, local fiber orientation, and cooling time in arbitrary 3D geometries [18]. This work generalizes well, predicting filling patterns on arbitrary unseen geometries with an average error of less than 5%.

While this previous work established a foundation for learning local geometric features, it remains insufficient for predicting final part warpage. Warpage is inherently non-local as localized shrinkage at one point can induce disproportionate deformations across the entire geometry through a structural leveraging effect. To address this, we move beyond local feature learning and introduce a differentiable finite element integration. This architectural shift is a key contribution of this study: it enables backpropagation of warpage errors through the entire prediction pipeline. It allows the model to further improve with real-world, observable warpage data for fine tuning and is not strictly limited to simulations for training.

2. Training data generation

Training and validation of fast approximation models requires a ground truth. In this case, we generate the ground truth by running state-of-the-art injection molding simulations on randomly sampled geometries from the ABC dataset [19], which is a large collection of Computer Aided Design geometries for geometric deep learning. We mesh the sample surfaces with triangles using the open-source software GMSH and choose a single random node on the surface as an injection location. Subsequently, we perform an automated AUTODESK MOLDFLOW 2023 analysis for each geometry to simulate the injection molding process and subsequent warpage.

The material for all simulations is polypropylene with 30 wt% glass fiber reinforcement (Celanese Factor PP GF 30) taken from the AUTODESK MOLDFLOW 2023 material database. The mold filling simulations are conducted as described in [18] with a packing pressure set to 80% of the filling pressure. The fiber orientation state is described via a second order fiber orientation tensor [20] and computed via the ARD-RSC model [21] using velocity gradients without two-way

coupling. Residual stresses are predicted using the anisotropic thermo-viscous-elastic residual stress model for 3D models assuming no in-mold shrinkage and no thermal expansion of the mold. Then, the warpage is calculated using small deflection analysis without mesh aggregation and upgrading all tetrahedral elements to second order to prevent warpage errors due to shear lock [22]. Further settings for the injection molding simulations can be found in Appendix D. As our machine learning pipeline operates on nodes, resulting stresses and strains must be mapped from integration points to nodes. Hence, elemental strain tensors are calculated from the process induced elemental stress tensors σ_{ij} and the fourth order elasticity tensors C_{ijkl} via $\epsilon_{ij} = C_{ijkl}^{-1} \sigma_{kl}$. Then, the nodal strain tensors $\epsilon_{ij}^{\text{nod}}$ are retrieved by solving

$$\epsilon_{ij}^{\text{nod}} = \underset{\epsilon_{ij}^{\text{nod}}}{\operatorname{argmin}} \|\mathbf{A} \epsilon_{ij}^{\text{nod}} - \epsilon_{ij}^{\text{elem}}\|_{\text{F}}^2 \quad (1)$$

with $\epsilon_{ij}^{\text{elem}}$ being the known elemental strain values and $\mathbf{A} \in \mathbb{R}^{n_{\text{elem}} \times n_{\text{nod}}}$ a connectivity matrix defining which nodes are part of which elements. Here, $\|\cdot\|_{\text{F}}$ denotes the Frobenius norm, n_{elem} and n_{nod} are the total number of elements and nodes in the mesh, respectively. Eq. (1) can be understood as the inverse operation of averaging the strains of all nodes in one element to the strain tensor for this element. In this work, two datasets are used, both originating from random geometries from the ABC dataset. The first comprises more than 600 models and a total of 18 million nodes, split into different pairs of training and test sets, each containing nine million data points. It contains per-node information on fiber orientation, warpage vectors, and additional process-related quantities. The second dataset contains approximately 200 geometries (distinct from those in Dataset 1) and is constructed in a similar manner, with one key difference: the packing process is simulated without any applied pressure. This modification increases the warpage compared to Dataset 1. The purpose of this variation is to introduce a controlled deviation from the idealized simulation conditions, thereby emulating a mismatch typically observed between simulations and real-world data. This synthetic deviation serves as a proxy for real 3D-scan data, which are currently unavailable in the required quantity or remain subject to industrial confidentiality. Both datasets are publicly available.¹

3. Approximation pipeline

The proposed approximation pipeline is built by modifying a previously developed method [18] and interfacing to our custom differentiable FE program TORCH-FEM.² This chapter first briefly introduces the pre-existing model, before stating the required modifications and lastly explaining the FE modeling for deformation calculation. Fig. 1 shows an overview of the method as explained in the next sections.

3.1. Base model

First, distance features are calculated by solving the eikonal equation for different initial and boundary conditions, resulting in three different geodesic distances per node: The distance to the closest wall D_w , the shortest distance to the inlet node D_i and the flow distance D_f , which encapsulates the differing flow speeds of polymer melt in channels of different width by calculating the geodesic inlet distance with the wall distance D_w as approximation of travel speed. The distances are normalized to D_w^* , D_i^* , V_f^* and their normalized gradients $\nabla^*(\cdot)^*$. To describe the local material arrangement around a node, the spatial moment \mathbf{S} is included. This symmetric second order tensor describes the local directionality of the material distribution around a node on a characteristic length scale R . The distance features and

¹ Dataset 1: <https://doi.org/10.5281/zenodo.10027027>, Dataset 2: <https://doi.org/10.5281/zenodo.17831586>

² <https://github.com/meyer-nils/torch-fem>

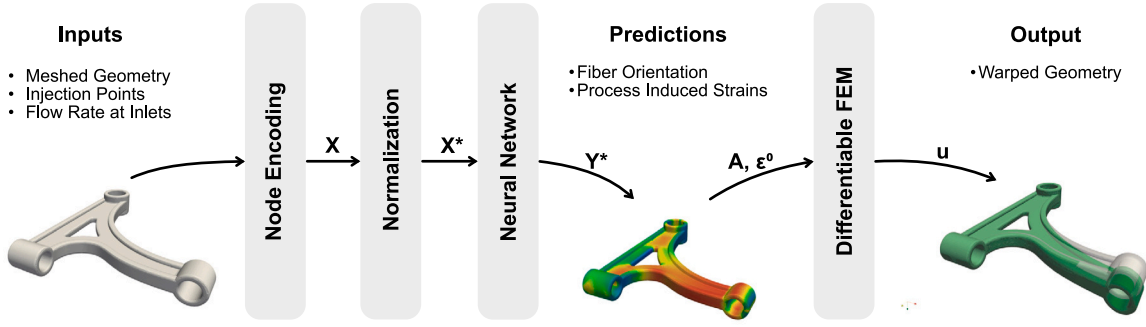


Fig. 1. Pipeline for the proposed fast approximation of warpage in the injection molding process.

spatial moments are combined to a nodal fingerprint vector \mathbf{X} for each node in a tetrahedrally meshed geometry. Subsequently, these are fed into a regression model that is trained to return a tuple containing the corresponding normalized nodal values for fill time t_{fill}^* , cooling time t_{cool}^* , volumetric shrinkage S_{vol}^* and fiber orientation \mathbf{A} . This work does not address the prediction of fill time, cooling time, or volumetric shrinkage, nor the corresponding denormalization steps required to obtain their final approximations; these details are provided in [18]. The model presented in the following section retains the capability to predict these variables, although they are not explicitly discussed here for the sake of clarity.

3.2. Required modifications to existing model

Due to differentiability requirements, feed-forward neural networks (FNNs) are used as trainable regression models in this work. In addition, several modifications have to be introduced to the base model in order to generate output suitable for subsequent warpage predictions.

Directly predicting the six independent components of \mathbf{A} , as in the previous model, does not strictly enforce the mathematical properties of \mathbf{A} . This may cause subsequent homogenization steps to yield non-physical mechanical properties. Therefore, a spectral decomposition is applied to split \mathbf{A} into its eigenvalues λ_i and eigenvectors \mathbf{a}_i . The eigenvectors, which span an orthonormal coordinate system, describe a rotation matrix \mathbf{R}_A that transforms a right-handed orthonormal coordinate system into the standard basis of \mathbb{R}^3 . As the third vector is implicitly defined by orthogonality, \mathbf{R}_A is encoded in a 6-D orthonormal way as described in [23]:

$$\mathbf{R}_A = \begin{bmatrix} | & | & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \\ | & | & | \end{bmatrix} \quad \mathbf{R}_A^{06D} = \begin{bmatrix} | & | \\ \mathbf{a}_1 & \mathbf{a}_2 \\ | & | \end{bmatrix} \quad (2)$$

According to definition [24], the eigenvalues satisfy

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \quad \text{and} \quad \lambda_1 + \lambda_2 + \lambda_3 = 1. \quad (3)$$

If λ_1 and λ_2 are predicted via the FNN and projected to their admissible space via

$$\frac{1}{3} \leq \lambda_1 \leq 1 \quad \text{and} \quad \frac{1}{2}(1 - \lambda_1) \leq \lambda_2 \leq \min(\lambda_1, 1 - \lambda_1), \quad (4)$$

the third eigenvalue can be calculated as $\lambda_3 = 1 - \lambda_1 - \lambda_2$ [25]. This changes the output of the fiber orientation prediction from $(A_{11}, A_{22}, A_{33}, A_{12}, A_{13}, A_{23})$ to $\mathbf{A}^{06D} = (a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23}, \lambda_1, \lambda_2)$, ensuring that resulting fiber orientation tensors always satisfy the mathematical properties.

The output of the surrogate model is extended by a nodal initial strain vector defined as $\boldsymbol{\varepsilon}^0 = (\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{33}, \gamma_{12}, \gamma_{23}, \gamma_{13})$, with $\gamma_{ij} = 2\varepsilon_{ij}$. Along with the prediction of the fiber orientation, this is the main input for the warpage calculation (see Section 3.3).

The previous model used only one scale per model to encode the geometric neighborhood in the spatial moment. Now, information on the geometric neighborhood on different scales is included

Table 1

Estimated material parameters.

Parameter	Value
Matrix modulus E_m	1400 MPa
Matrix Poisson ratio ν_m	0.45
Fiber modulus E_f	72 000 MPa
Fiber Poisson ratio ν_f	0.20
Fiber aspect ratio a	120.0
Fiber volume fraction V_f	0.1326

in the input fingerprint by calculating the spatial moment \mathbf{S} with three different radii R_i . Fig. 2 illustrates the difference that can be captured by this change: On a larger scale, the general orientation of the part is dominant, whereas on a smaller scale, the orientation of smaller individual ribs is taken into account. Spatial moment tensors are included in both a symmetric tensor description as components $(S_{11}, S_{22}, S_{33}, S_{12}, S_{13}, S_{23})$, as well as in their spectral decomposition into eigenvalues and eigenvectors $\mathbf{S}_{R_i}^{06D}$.

The normalized features are computed for each geometry in the dataset. For each node, the full normalized input vector is defined as

$$\mathbf{X}^* = \left(V_f^*, \nabla^* V_f^*, D_1^*, \nabla^* D_1^*, D_w, D_w^*, \nabla^* D_w^*, \mathbf{S}_{R_1}, \mathbf{S}_{R_2}, \mathbf{S}_{R_3}, \mathbf{S}_{R_1}^{06D}, \mathbf{S}_{R_2}^{06D}, \mathbf{S}_{R_3}^{06D} \right) \quad (5)$$

Ideally, this fingerprint allows one to infer approximate information about the normalized injection molding result at the corresponding node, represented by the tuple

$$\mathbf{Y}^* = (\mathbf{A}^{06D}, \boldsymbol{\varepsilon}^0). \quad (6)$$

The relation between the input features \mathbf{X}^* and the outputs \mathbf{Y}^* is modeled using a feed-forward neural network (FNN).

3.3. Warpage calculation via FEM

We compute warpage assuming a linear small-strain elasticity problem without body forces, i.e. solving the momentum balance

$$\frac{\partial \sigma_{ij}}{\partial x_j} = 0 \quad (7)$$

with the material model

$$\sigma_{ij} = C_{ijkl}(\varepsilon_{kl} - \varepsilon_{kl}^0), \quad (8)$$

where C_{ijkl} is the stiffness tensor, ε_{kl} the infinitesimal strain tensor, and ε_{kl}^0 is the initial strain tensor resulting from the molding process.

To solve this problem, the effective stiffness tensor \mathbf{C} is obtained from the predicted fiber orientation tensor and material parameters using mean-field homogenization [26] as well as the invariant-based optimal fitting (IBOF) closure [27]. The material parameters are selected as physically reasonable estimates (listed in Table 1) for PP GF30.

The initial strain tensor is obtained directly from the network prediction $\boldsymbol{\varepsilon}_{ij}^0$. For each finite element, the fiber orientation and initial

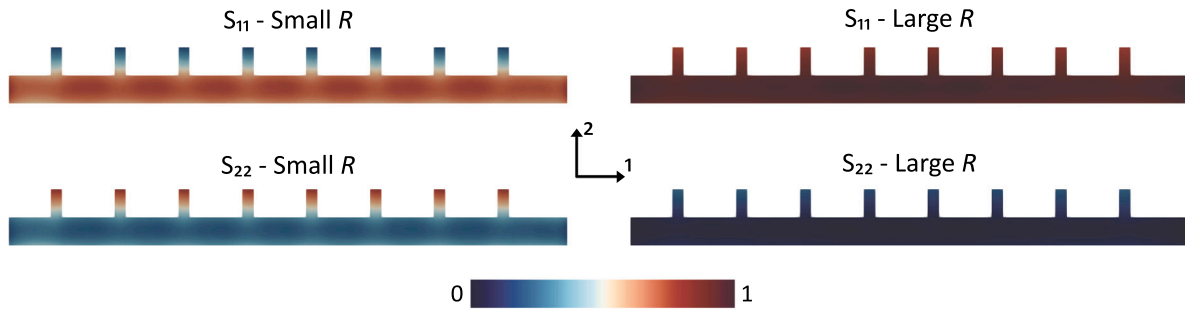


Fig. 2. Exemplary visualization for capturing differently sized detail by spatial moment with different radius.

strain tensors are computed as the average of the corresponding nodal values. Before solving, the mesh is converted to second-order tetrahedral elements. The homogenized material model and the predicted initial strain tensors are then supplied to TORCH-FEM, which employs the direct Pardiso method to solve for the displacement field. Rigid-body rotations are suppressed by applying six constraints to three anchor points according to the 3-2-1 rule. To ensure that the part is well constrained and that the anchor points are spatially distributed, these points are systematically selected for each geometry (see Appendix A for details). For comparison with reference displacements, a rigid-body rotation that minimizes the overall error is determined via a singular value decomposition. This postprocessing step ensures that the error evaluation reflects only the actual deformation and not contributions from rigid-body motion.

To verify the accuracy of the warpage calculation, Table 2 lists the mean and maximum errors for displacements of an exemplary geometry against the reference solution obtained with AUTODESK MOLDFLOW 2023. An additional ANSYS calculation was performed based on an export via the MPI2ANS macro. It is apparent that using the same initial strains and material parameters, both errors for ANSYS and TORCH-FEM are negligible. The slightly larger error in the right column originates mainly from the material homogenization with approximated parameters and a different closure.

Table 2

Mean and maximum nodal displacement errors over all 11641 nodes of an example geometry of ANSYS and TORCH-FEM in comparison to the AUTODESK MOLDFLOW 2023 reference result. The reference geometry (gray) and its warped shape (amplified x10, blue) is shown on top left inset.

	ANSYS (exported material)	TORCH-FEM (exported material)	TORCH-FEM (homogenized material)
Mean Error	0.0025%	0.0025%	0.0181%
Maximum Error	0.0038%	0.0044%	0.0265%

4. Training procedures and results

The overall training procedure is divided into two stages: In the first stage, a neural network is trained by minimizing the prediction error on pairs $(\mathbf{X}^*, \mathbf{Y}^*)$ and warpage is computed subsequently employing the FE solver. In the second stage, we exploit the differentiability of the FE solver to retrain the model on a different dataset through the complete simulation pipeline, using the resulting warpage to compute the loss. This setup emulates retraining with experimental data, in which strain fields are typically unavailable, whereas warpage vectors can be obtained more easily via 3D scanning.

4.1. Stage 1: Training with process data and subsequent warpage computation

4.1.1. Training procedure

In the first training phase, the mean squared error (MSE) between the network predictions \mathbf{Y}^* and the simulation ground truth is minimized on a training dataset containing more than nine million node pairs. Training is performed using the Adam optimizer implemented in PyTorch with a learning rate of 10^{-4} , a batch size of 2048 node pairs, and for a total of 10 epochs. Three different combinations of training set and test set are used to train FNNs consisting of five layers containing 500 neurons in each hidden layer.

To ensure generalization and prevent bias towards particular part orientations, each model, its nodes, and its properties are randomly rotated before training. This strategy mitigates the risk of the network learning spurious orientation-dependent patterns and enforces focus on the intrinsic correlations between input and output quantities. During the second half of the training, the fiber orientation is projected to the admissible space determined by Eq. (4) as described in Section 3.2. Enforcing projection too early leads to convergence towards isotropic predictions of \mathbf{A} , whereas the later application preserves meaningful anisotropic behavior. The mean squared error (MSE) is computed after the prediction is rotated back into the initial coordinate system, and the predicted fiber orientation is transformed to its tensor representation.

4.1.2. Evaluation on ABC test sets

This section evaluates the performance of the warpage approximation pipeline after training. We introduce the root mean squared error (RMSE) of the nodal warpage vectors, normalized by the largest dimension L of the respective part, as a suitable error metric. It is defined as

$$\text{Relative Warpage Error}(u_i, \hat{u}_i) = \sqrt{\frac{1}{I} \sum_{i=1}^I \left\| \frac{u_i}{L} - \frac{\hat{u}_i}{L} \right\|_2^2}, \quad u_i, \hat{u}_i \in \mathbb{R}^3 \quad (9)$$

with u_i and \hat{u}_i being the approximated and reference nodal warpage vectors for a given geometry. Normalization by size L is required for an error metric that is invariant to the scaling of a part. This is crucial as our dataset contains a large variety of part sizes.

For analyzing the performance of the approximation model, we calculate the warpage errors on test sets for three different training runs with different, randomly drawn combinations of training and test sets sampled from our dataset. The errors are then sorted in ascending order such that parts with small errors are shown on the left in Fig. 3. Exemplary warped geometries are added as insets to visualize geometries and their shape deviation corresponding to their error. The majority of parts have a relative warpage prediction error of less than 1% by the definition given in Eq. (9).

Insets (a) to (c) show typical injection molded geometries, with maximum warpage errors of 0.1 mm for a part with 30 mm width (a), 0.2 mm for another part with 30 mm width (b), and 0.43 mm for a part

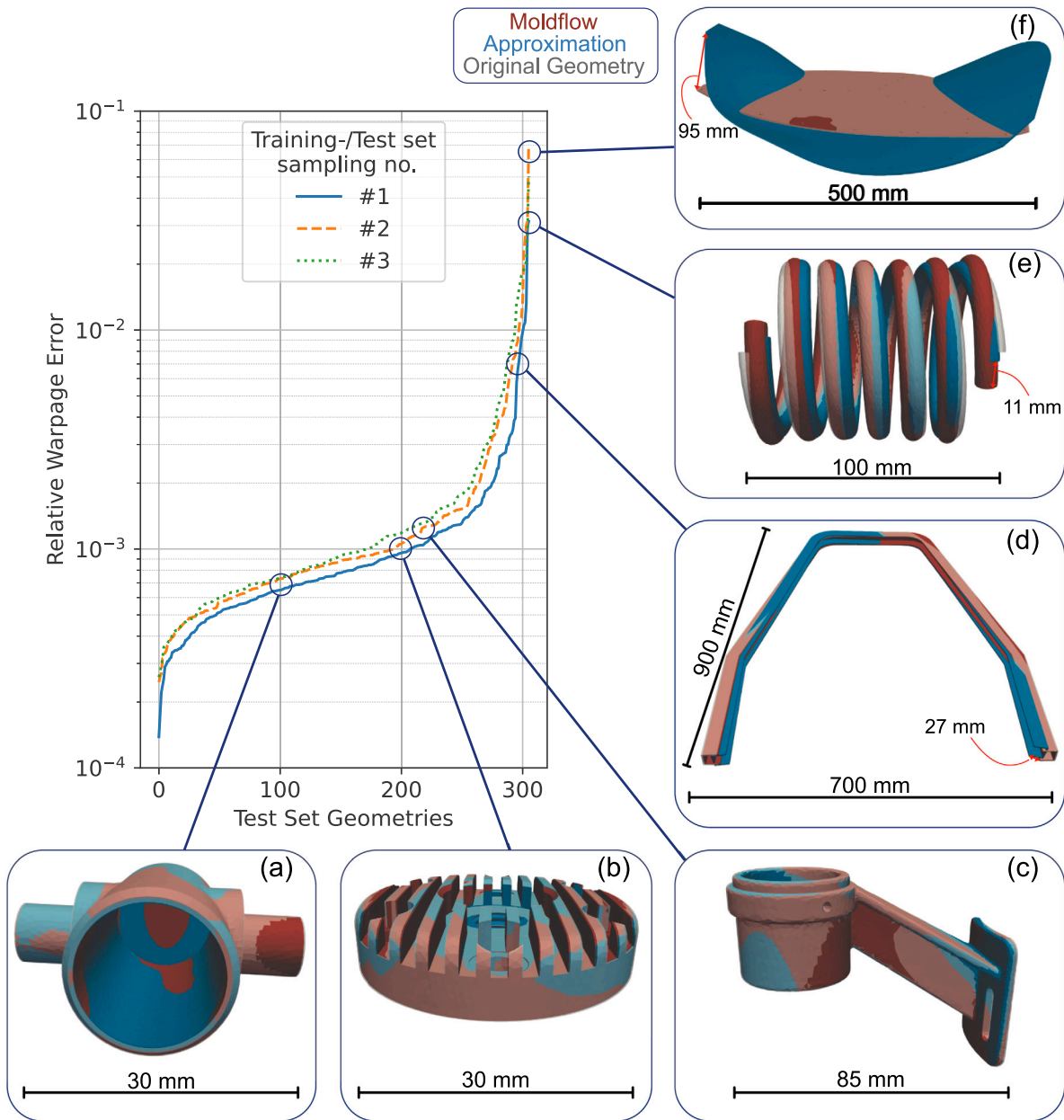


Fig. 3. Sorted distribution of warpage prediction errors on three different, randomly sampled training and test set combinations, along with exemplary geometries representing the error values at the respective point of the curves. The insets feature the original geometry (gray), the result of the approximation pipeline (blue) and the reference result (red). Red arrows indicate absolute errors in parts with significant prediction errors.

with 85 mm width (c). Geometries with the largest errors are usually either long with a lever as structures as in (d) and (e) or very thin structures as in (d) and (f). Large, thin walled geometries suffer from two problems: Small prediction differences on both sides such as on top and bottom of example (f) lead to unrealistic distortion of the parts. Furthermore, due to computational constraints, these parts may be insufficiently meshed in the thickness direction, amplifying the error both in the strain prediction as well as in the FE calculation. For such thin parts, shell-based meshes would be much more appropriate to capture through-thickness bending behavior accurately and should be investigated in the future.

In general, prediction of geometries that contain very long, lever-like areas (like long bars or springs) suffers from the fact that warpage errors are accumulated and amplified, leading to larger errors at the ends of the part. However, these parts are unlikely to be produced using injection molding due to unfavorable aspect ratios, and other

techniques, such as extrusion or compression molding, might be preferential in those cases. Since the trend is still captured correctly by the FNN, these models are kept in the dataset in the pursuit of building an approximation model that generalizes to other geometries as much as possible.

4.2. Stage 2: Training with observable warpage through FE solver

4.2.1. Training procedure

The second training phase aims to simulate the retraining of the neural network using information on warpage and fiber orientation, as would be the case with real-world measurements where nodal initial strain values cannot be directly observed. This retraining step, using experimentally observable data, can be interpreted as fine-tuning an FNN that was previously trained on synthetically generated data. Training the network exclusively on measured data would eliminate its

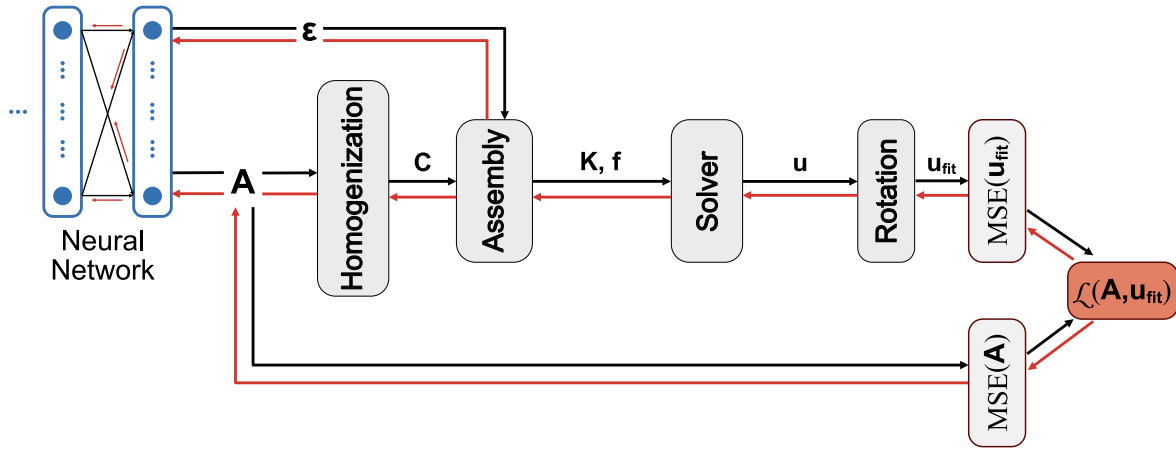


Fig. 4. Schematic of the FE pipeline: Red arrows indicate gradient flow during the backward pass, propagating from the total loss through all components to the weights and biases. The gradient flow through the Solver operation is detailed in B.

ability to predict non-observable process variables, such as flow-front evolution and cooling time. Moreover, such an approach would require substantially higher computational effort and a significantly larger dataset. The dataset consists of new geometries, which are simulated without packing pressure to enhance warpage effects (as discussed in Section 2). Both training set and test set comprise 75 mutually exclusive geometries randomly drawn from this dataset, each of which contains on average 23000 nodes. Three different combinations of training set and test set are used to train neural networks. As explained in the previous section, all coordinates and input features are randomly rotated for each model, while predictions are rotated back prior to warpage computation. The warpage training loss \mathcal{L}_{warp} is calculated from the predicted warpage values as the squared relative warpage error in Eq. (9), being its MSE equivalent. The final loss function combines \mathcal{L}_{warp} computed through the approximation pipeline with the loss for fiber orientation \mathcal{L}_{ori} . Including fiber orientation ensures that the network learns the appropriate initial strain fields necessary to reproduce the warpage rather than compensating for incorrect strains through more inaccurate fiber orientation predictions or material behavior. The two loss terms are balanced through dynamic weighting, updated at the end of each epoch according to the average loss values. The initial weights are set to $w_{ori} = 10^{-2}$ and $w_{warp} = 1$, and their updates follow the rules

$$\begin{aligned} w_{ori}^{current} &= \frac{\bar{\mathcal{L}}_{warp} + \bar{\mathcal{L}}_{ori}}{\bar{\mathcal{L}}_{warp}} & w_{ori}^{new} &= 0.9w_{ori}^{old} + 0.1w_{ori}^{current} \\ w_{warp}^{current} &= \frac{\bar{\mathcal{L}}_{warp} + \bar{\mathcal{L}}_{ori}}{\bar{\mathcal{L}}_{ori}} & w_{warp}^{new} &= 0.9w_{warp}^{old} + 0.1w_{warp}^{current} \end{aligned} \quad (10)$$

which gradually stabilizes the relative importance of both terms during training. $\bar{\mathcal{L}}_{warp}$ and $\bar{\mathcal{L}}_{ori}$ are the averaged loss values for warpage and fiber orientation, respectively, over the epoch passed. Fig. 4 illustrates the connection of the neural network and the components of the FE pipeline along with the gradient flow from the combined loss back to the parameters of the neural network. Loss gradients are evaluated and used for optimization steps in batches of five models. This is necessary due to the wide variety of different geometries and the random rotation strategy leading to changing error magnitudes and noisy training if steps are taken after every model (see C for further information on training loss over the epochs). The network is trained for 100 epochs, with the learning rate starting at $5 \cdot 10^{-5}$ for the first 30 epochs and halved every ten epochs thereafter to improve convergence in the later training stages.

4.2.2. Evaluation on ABC test set

Fig. 5 presents the sorted error distributions for warpage, fiber orientation, and initial strain prediction in three independent train/test combinations sampled from the dataset. The Relative Warpage Error as defined in Eq. (9) and the RMSE for fiber orientation and initial strain are chosen as error metrics. The errors are evaluated before and after model retraining through the FE solver with warpage data. Some representative geometries taken from one combination (indicated with thick lines) are visualized together with the warpage fields both pre- and post-training to illustrate the qualitative improvement.

A substantial reduction in the warpage prediction error on the test sets is observed consistently across all runs. This outcome aligns with the primary objective of the retraining strategy, which explicitly minimizes the warpage deviation. The improvement is particularly visible for components that feature lever-like geometries, where large initial discrepancies are significantly reduced (see Fig. 5). It is noted that a small subset of parts experiences slight increases in error during the retraining process; however, this behavior is limited and does not affect the overall improvement trend.

The fiber orientation error decreases only marginally with retraining. This behavior is expected, as the fiber orientation loss is explicitly incorporated and dynamically balanced against the warpage loss. The limited improvement suggests that the current input fingerprint may no longer provide sufficient descriptive capacity for further enhancement of orientation prediction. Consequently, achieving additional progress would likely require incorporating supplementary or alternative process- or geometry-based descriptors.

In contrast, the initial strain prediction error increases notably with retraining. This trend is also anticipated, as the initial strains constitute the only freely adjustable internal variables available to the model to improve the accuracy of warpage prediction. Given that the fiber orientation prediction remains imperfect, the model compensates by deviating from the reference initial-strain fields. This effect is reinforced by the fact that initial strain accuracy is not explicitly included in the loss formulation.

Fig. 6 demonstrates another benefit of training through the automatically differentiable FE solver by visualizing the sensitivities of the warpage loss with respect to the components of nodal initial strain, defined as

$$g_{ij}^{nod} = \frac{\partial \mathcal{L}_{warp}}{\partial \epsilon_{ij}^{nod}}. \quad (11)$$

The images highlight that different regions of the component, as well as different strain components, contribute disproportionately to the

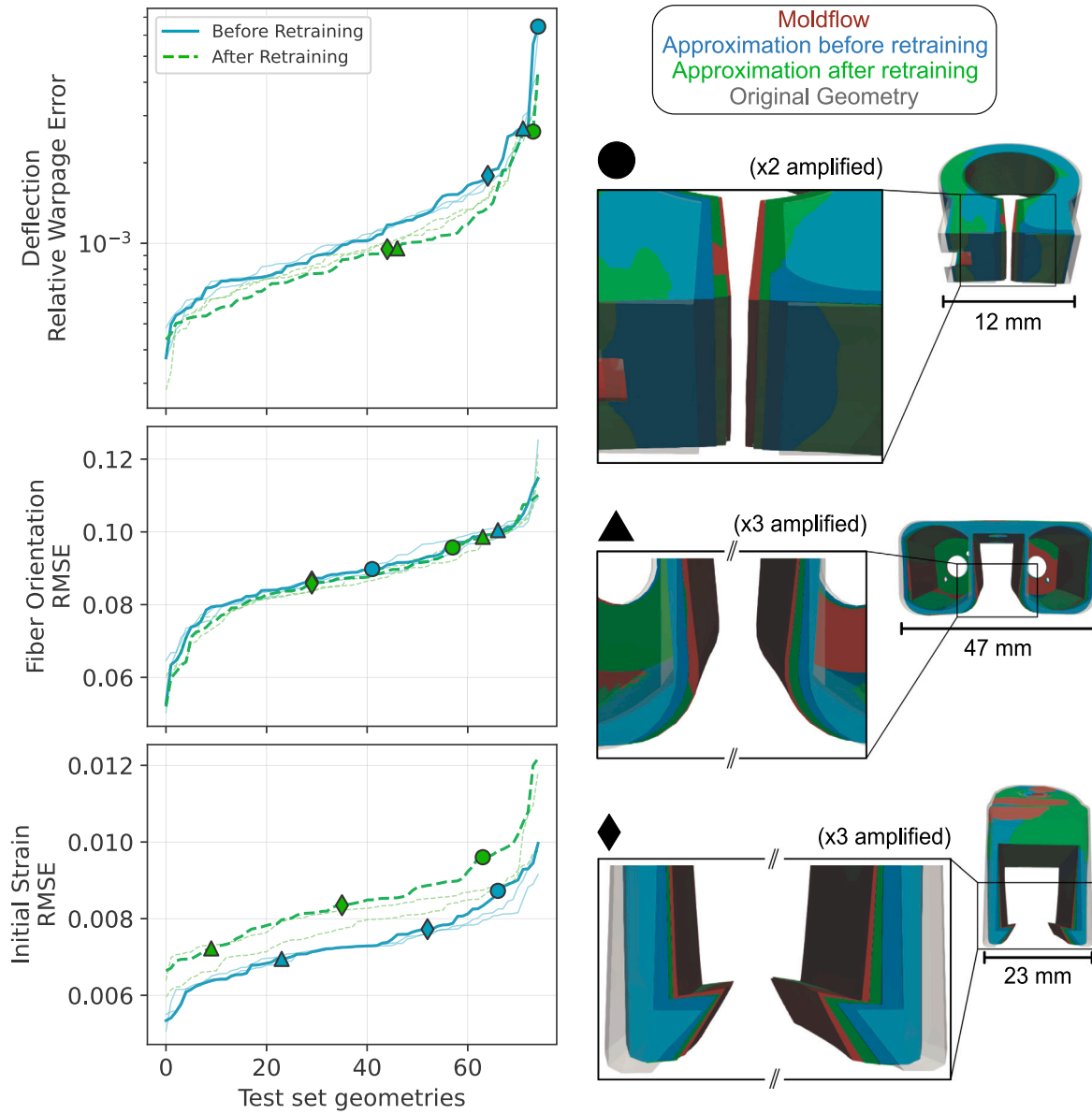


Fig. 5. Sorted distribution of warpage errors, along with exemplary geometries representing the error values at the respective point of the curves. The images feature the original geometry (gray), the result of the approximation pipeline before retraining (blue), the result of the approximation pipeline after retraining (green) and the reference result (red). Warpage is amplified with respective factor for better visibility. Two other random combinations of training set and test set are plotted with lower opacity for reference.

warpage response. Areas exhibiting higher absolute values of g_{ij}^{nod} indicate locations where deviations in initial strain have a greater influence on the resulting deformation. By backpropagation of these losses through the FE model, the optimization process is therefore naturally directed to adjust strain values predominantly in the most influential regions. This behavior contrasts with the training procedure described in Section 4.1, in which all components of the nodal strain are uniformly weighted. Although such an approach offers faster training due to the missing FE computation, it lacks the spatial selectivity enabled by gradient-aware optimization.

5. Computational performance

Training and evaluation were performed on a workstation equipped with an Intel Core Ultra 9 285K CPU, an NVIDIA RTX5090 GPU, and

64 GB of RAM. The initial training of the neural network takes approximately 29 min. The retraining process through the differentiable FE solver is strongly influenced by the mesh size of the geometries in the respective randomly sampled training set, since the FE computation time scales with the number of nodes. A representative retraining run requires roughly 110 h, including approximately 7500 FE forward and backward evaluations. Although computationally expensive, this step is performed only once, after which inference is significantly faster than conventional simulation workflows.

Inference performance of the full approximation pipeline was benchmarked using an arbitrary part on a workstation with an AMD Ryzen Threadripper PRO 5995WX CPU (64 cores), an NVIDIA RTX4090 GPU, and 512 GB of RAM. Table 3 lists the corresponding run times for different mesh densities. The neural network inference time is negligible compared to geometric preprocessing and FE evaluation. The meshing time is excluded, as both the reference and surrogate

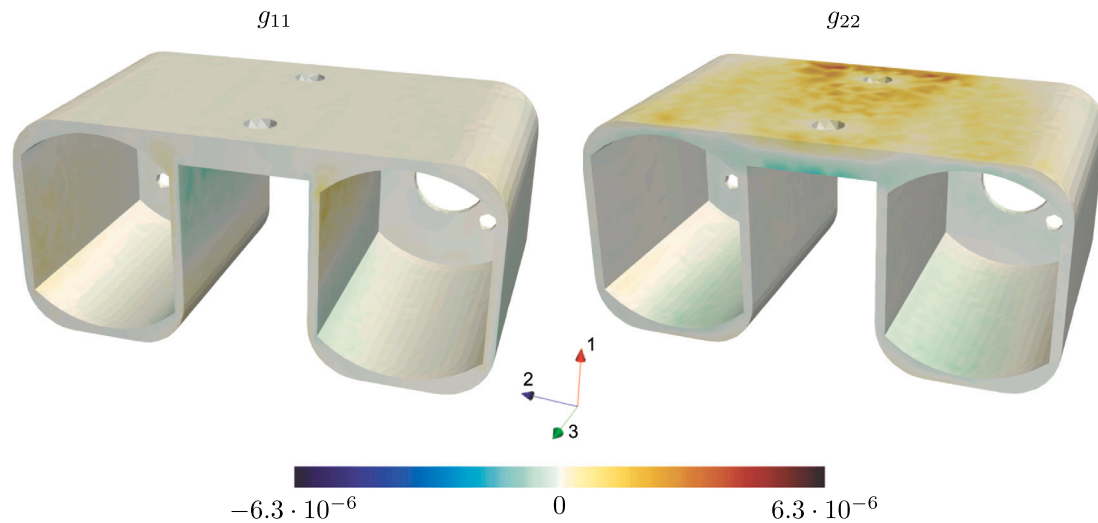


Fig. 6. Sensitivity of warpage loss with respect to nodal strain values ϵ_{11} and ϵ_{22} , showing the relative importance of different areas in the strain prediction.

Table 3

Table of run times for the different steps in the approximation for varying mesh densities on an arbitrary part from the ABC dataset.

# Nodes	# Elements	Approximation			Simulation	
		Geodesics	Spatial moments	TORCH-FEM	Fill	Warp
20203	110 965	7.1 s	5.7 s	18.7 s	185 s	24 s
66932	364 366	14.7 s	12.3 s	76.3 s	615 s	82 s
102045	559 748	18.9 s	31.5 s	141.3 s	1136 s	128 s

approaches use identical meshes. It can be seen that the approximation without FE calculation is around 20 times faster than the full filling simulation. The custom FE solver is about as fast as the commercial tool for the investigated part but exhibits a higher RAM usage due to the chosen Pardiso solver. However, it comes with the benefit of being fully differentiable, allowing gradient-based optimization procedures which would not be possible with most other commercial software.

6. Conclusion

We presented a fast approximation pipeline capable of predicting warpage in injection-molded 3D geometries by combining nodal encoding, machine-learning-based field prediction, and a custom differentiable FE solver. The prediction accuracy was evaluated using the RMSE of size-normalized nodal warpage vectors. For typical injection-molded components, the method achieves a mean relative warpage prediction error of less than 1%. Reduced accuracy is observed primarily for very thin or lever-shaped geometries. The latter may be addressed in future work by employing an approximation model based on shell elements, which would further improve both accuracy and computational speed. Training through the differentiable FEM solver on a warpage based loss with a warpage-enhanced dataset lead to significant improvement in the prediction. This serves as an example of the potential to train the neural network on a dataset of real world 3D scans, where initial strains, which serve as the basis for our warpage calculation, are not available. Furthermore, the differentiability makes the training process automatically focus on the most relevant sections of geometries for the resulting warpage due to higher loss gradients in these areas.

The proposed approach provides a substantial reduction in computation time compared to conventional simulation workflows. This enables efficient integration into automated design and optimization procedures in which the objective function must be evaluated repeatedly. For example, both the part geometry and injection locations

could be optimized with respect to the final shape using the surrogate, requiring only a single final high-fidelity simulation for validation.

CRediT authorship contribution statement

Julian Greif: Writing – original draft, Software, Methodology, Investigation, Conceptualization. **Nils Meyer:** Writing – review & editing, Supervision, Software, Conceptualization.

Declaration of Generative AI and AI-assisted technologies in the writing process

In the preparation of this work, the authors used Github Copilot and ChatGPT for the efficient programming of helper functions (e.g. randomized rotation of objects) as well as for rephrasing of individual text sections. After using these tools, we reviewed and modified the content as necessary and take full responsibility for the content of the published article.

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.

Acknowledgments

This research was carried out within the framework of the AI production network Augsburg funded by the Bavarian State Ministry of Science and the Arts inside the Hightech Agenda Plus. In addition, the research documented in this manuscript has been supported by the Hightech Agenda Bavaria. The support of the Bavarian State Government is gratefully acknowledged.

Table A.1
Equations to find constrained points and which coordinate to constrain.

Anchor Node	Mathematical Description	Constrained directions
P_1	$\underset{x}{\operatorname{argmax}}(x)$	1, 2, 3
P_2	$\underset{x}{\operatorname{argmin}}(x)$	2, 3
P_3	$\underset{x}{\operatorname{argmax}} \frac{1}{2} \ (P_2 - P_1) \times (x - P_1)\ $	3

Appendix A. Constraining rigid body motions

To ensure that the part is well constrained and that the anchor points are spatially distributed, the anchor points are systematically selected from all 3D node coordinates x . The node P_1 with the maximum distance from the origin, the node P_2 with the minimum distance from the origin, and the one P_3 that maximizes the area of the triangle spanned by all three anchors. Table A.1 lists the mathematical descriptions and set constraints for all three nodes.

All nodes, strain tensors, and stiffness tensors are rotated into a local right-handed coordinate system (u, v, w) , which is defined by $u = P_2 - P_1$, $w = P_3 - P_1$, $v = w \times u$.

After assembly of the global stiffness matrix, the resulting linear system of equations is solved in the rotated configuration and subsequently transformed back into the global coordinate system.

Appendix B. Automatic differentiation of the solve operation

To train the model with warpage, gradients must be able to flow backwards through the entire Finite Element pipeline illustrated in Fig. 4. In particular, this includes the *Solver* operation of the assembled linear system

$$Ku = f, \quad (B.1)$$

where K denotes the global stiffness matrix, u the unknown global displacement vector, and f the global force vector. As K is sparse, symmetric, and positive definite, the system is solved most efficiently with dedicated solvers, which naturally do not allow backpropagation via automatic differentiation. To enable automatic differentiation, the *Solver* is wrapped in a custom `TORCH` function which computes

$$u = \text{SOLVE}(K, f) \quad (B.2)$$

on the forward path, where `SOLVE` denotes applying a non-differentiable solver (in this case Pardiso, but the backend also enables other direct and iterative solvers). During the backward pass, the wrapper has to compute the gradients of this operation with respect to its inputs to keep the gradient flowing. Some straightforward calculations in index notation yield

$$\frac{\partial \mathcal{L}}{\partial b} = K^{-T} \frac{\partial \mathcal{L}}{\partial u} = \text{SOLVE}(K^T, \frac{\partial \mathcal{L}}{\partial u}) \quad (B.3)$$

and

$$\frac{\partial \mathcal{L}}{\partial K} = -\frac{\partial \mathcal{L}}{\partial f} u^T \quad (B.4)$$

with the incoming gradient on the backward pass $\frac{\partial \mathcal{L}}{\partial u}$. This adjoint approach enables the use of arbitrary non-differentiable backends for solving the linear system efficiently without potentially tracking gradients through the entire iterative solver. Furthermore, since $K = K^T$, any factorization or preconditioner generated during the forward pass can be reused to compute the adjoint variables in the backward pass, significantly reducing computational cost.

Appendix C. Training loss

Fig. C.7 shows an exemplary training loss over the training epochs, corresponding to the thick plotted test error curves in Fig. 5 over the training epochs. The shown losses are the pure prediction losses during the training which are then dynamically weighted to compute the total loss.

Table D.2
Solver Setup.

Parameter	Value
Solver	Coupled 3D
Solution type	Stokes
Simulate inertia effect	No
Simulate gravity effect	No
Simulate wall slip	No

Table D.3
Material Metadata and Key Properties.

Field	Value
Manufacturer	Celanese
Trade name	Factor PP GF 30
pvT model	2-domain modified Tait
Viscosity model	Cross-WLF
Filler properties	Confidential

Table D.4
Viscosity Model Coefficients.

Coefficient	Value
n	0.3251
τ^* (Pa)	$2.0200 \cdot 10^4$
$D1$ (Pa s)	$5.8200 \cdot 10^{11}$
$D2$ (K)	263.1500
$D3$ (K/Pa)	0.0000
$A1$	26.1370
$A2T$ (K)	51.6000

Table D.5
PvT Model Coefficients.

Coefficient	Liquid phase	Solid phase
$b5$ (K)	408.1500	
$b6$ (K/Pa)	$2.3100 \cdot 10^{-7}$	
$b1$ (m ³ /kg)	0.0010	0.0009
$b2$ (m ³ /kg-K)	$5.5220 \cdot 10^{-7}$	$2.9510 \cdot 10^{-7}$
$b3$ (Pa)	$1.2036 \cdot 10^8$	$2.6346 \cdot 10^8$
$b4$ (1/K)	0.0038	0.0058
$b7$ (m ³ /kg)		$7.8230 \cdot 10^{-5}$
$b8$ (1/K)		0.1351
$b9$ (1/Pa)		$3.3700 \cdot 10^{-8}$

Table D.6
Filling and Packing Parameters.

Parameter	Value
Maximum % volume per time step (Filling)	1.0%
Flow front scheme	Level set
Maximum iterations per time step	50
Maximum time step (Packing)	2.0 s

Appendix D. Moldflow settings

Tables D.2–D.13 list the `MOLDFLOW` simulation parameters used for creating the datasets.

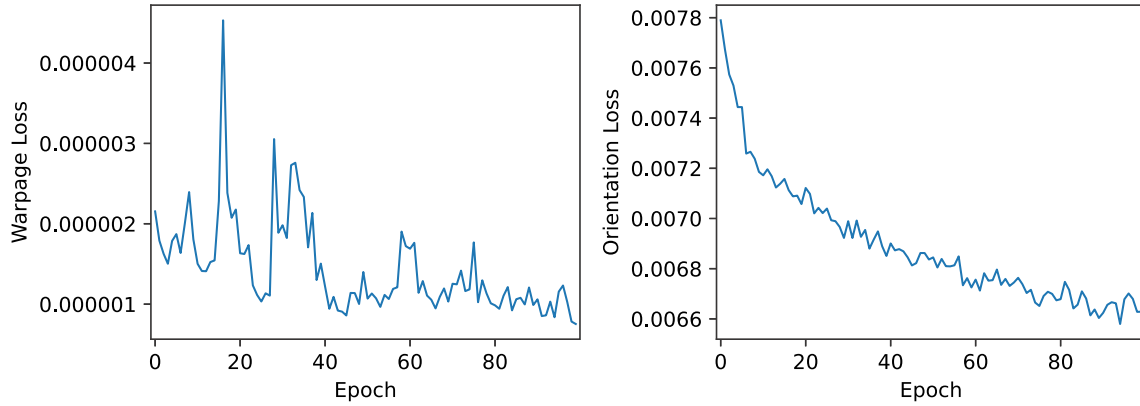


Fig. C.7. Training loss over the training epochs for exemplary training run.

Table D.7
Specific Heat.

T (K)	C_p (J/kg-K)
293.150	1540.000
373.150	2280.000
403.150	2820.000
413.150	3210.000
423.150	4090.000
433.150	6300.000
443.150	3940.000
453.150	2770.000
473.150	2880.000
503.150	3060.000
523.150	3190.000
543.150	3350.000

Table D.8
Thermal Conductivity.

T (K)	k (W/m-K)
320.150	0.200
330.150	0.205
371.150	0.247
392.150	0.237
413.150	0.227
432.150	0.214
452.150	0.223
471.150	0.228
481.150	0.233
501.150	0.240
521.150	0.244

Table D.9
Mechanical Properties and Thermal Expansion.

Parameter	Value
E_1 (Pa)	$7.1461 \cdot 10^9$
E_2 (Pa)	$3.3514 \cdot 10^9$
ν_{12}	0.4459
ν_{23}	0.4895
G_{12} (Pa)	$1.5830 \cdot 10^9$
CTE α_1 (1/K)	0.0000182
CTE α_2 (1/K)	0.0000509
Matrix E (Pa)	$1.3400 \cdot 10^9$
Matrix ν	0.3920
Matrix α_1 (1/K)	0.0000905

Table D.10
Fiber Ori. Calculation Parameters.

Parameter	Value
Fiber orientation model	ARD-RSC
Red. Strain Clos. factor	0.1000
ARD parameter b_1	0.000192
ARD parameter b_2	0.005839
ARD parameter b_3	0.040000
ARD parameter b_4	0.000012
ARD parameter b_5	0.000000
Fiber inlet condition	Aligned at skin, random at core
Micro-mechanics model	Tandon-Weng
Thermal expansion coeff.	Rosen-Hashin
Closure approximation	Orthotropic 3

Table D.11
Machine and Process Parameters.

Parameter	Value
Max. injection pressure (Pa)	$1.800 \cdot 10^8$
Max. machine clamp force (N)	$6.867 \cdot 10^7$
Max. machine injection rate (m ³ /s)	$5.0 \cdot 10^{-3}$
Machine hydr. resp. time (s)	$1.0 \cdot 10^{-2}$
Melt temperature (K)	543.15
Mold temperature (K)	318.15
Atmospheric temperature (K)	298.15
Mold-Melt Heat Trans. Fill. (W/m ² K)	5000.0
Mold-Melt Heat Trans. Pack. (W/m ² K)	2500.0
Mold-Melt Heat Trans. Detach. (W/m ² K)	1250.0
Filling control type	Automatic
Stroke volume det.	Automatic
V/P switchover Control type	Automatic

Table D.12
Pack/Holding Pressure Profile.

Duration (s)	% Filling pressure
0.00	80.00
10.00	80.00

Table D.13
Warp Analysis Settings.

Parameter	Value
Stress calculation model	Residual stress model
Matrix solver	AMG

Data availability

Data will be made available on request.

References

- [1] Kikuchi H, Koyama K. Warpage, anisotropy, and part thickness. *Polym Eng Sci* 1996;36(10):1326–35. <http://dx.doi.org/10.1002/pen.10527>.
- [2] Isayev AI, Hieber CA. Toward a viscoelastic modelling of the injection molding of polymers. *Rheol Acta* 1980;19:168–82. <http://dx.doi.org/10.1007/BF01521928>.
- [3] Baaijens F. Calculation of residual stresses in injection molded products. *Rheol Acta* 1991;30:284–99. <http://dx.doi.org/10.1007/BF00366642>.
- [4] Kabanemi KK, Crochet MJ. Thermoviscoelastic Calculation of Residual Stresses and Residual Shapes of Injection Molded Parts. *Int Polym Process* 1992;7(1):60–70. <http://dx.doi.org/10.3139/217.920060>.
- [5] Zheng R, Tanner RI, Fan X-J. Injection molding: integration of theory and modeling methods. Berlin Heidelberg: Springer; 2011. <http://dx.doi.org/10.1007/978-3-642-21263-5>.
- [6] Titomanlio G, Drucato V, Kamal MR. Mechanism of Cooling Stress Build-Up in Injection Molding of Thermoplastic Polymers. *Int Polym Process* 1987;1(2):55–9. <http://dx.doi.org/10.3139/217.870055>.
- [7] Struik LCE. Internal stresses, dimensional instabilities, and molecular orientations in plastics. Chichester: Wiley; 1990.
- [8] Zoetelief WF, Douven LFA, Housz AJI. Residual thermal stresses in injection molded products. *Polym Eng Sci* 1996;36(14):1886–96. <http://dx.doi.org/10.1002/pen.10585>.
- [9] Kastelic T, Starman B, Cafuta G, Halilović M, Mole N. Correction of mould cavity geometry for warpage compensation. *Int J Adv Manuf Technol* 2022;123:1957–71. <http://dx.doi.org/10.1007/s00170-022-10337-6>.
- [10] Tillmann S, Behr M, Elgeti S. Using Bayesian optimization for warpage compensation in injection molding. *Mater Werkst* 2024;55(1):13–20. <http://dx.doi.org/10.1002/mawe.202300157>.
- [11] Liao SJ, Hsieh WH, Wang JT, Su YC. Shrinkage and warpage prediction of injection-molded thin-wall parts using artificial neural networks. *Polym Eng Sci* 2004;44(11):2029–40. <http://dx.doi.org/10.1002/pen.20206>.
- [12] Yin F, Mao H, Hua L, Guo W, Shu M. Back Propagation neural network modeling for warpage prediction and optimization of plastic products during injection molding. *Mater Des* 2011;32(4):1844–50. <http://dx.doi.org/10.1016/j.matdes.2010.12.022>.
- [13] Song Z, Liu S, Wang X, Hu Z. Optimization and prediction of volume shrinkage and warpage of injection-molded thin-walled parts based on neural network. *Int J Adv Manuf Technol* 2020;109:755–69. <http://dx.doi.org/10.1007/s00170-020-05558-6>.
- [14] Wenzel M, Raisch SR, Schmitz M, Hopmann C. Comparison of Hybrid Machine Learning Approaches for Surrogate Modeling Part Shrinkage in Injection Molding. *Polymers* 2024;16(17):2465. <http://dx.doi.org/10.3390/polym16172465>.
- [15] Baum M, Anders D, Reinicke T. Optimizing injection molding simulations: Comparative performance of Kriging and RSM surrogate models for process efficiency. *Discov Mech Eng* 2025;4:31. <http://dx.doi.org/10.1007/s44245-025-00115-5>.
- [16] Gao Y, Turng L-S, Wang X. Adaptive geometry and process optimization for injection molding using the kriging surrogate model trained by numerical simulation. *Adv Polym Technol* 2008;27(1):1–16. <http://dx.doi.org/10.1002/adv.20116>.
- [17] Uglov A, Nikolaev S, Belov S, Padalitsa D, Greenkina T, Biagio MS, Cacciatori FM. Surrogate modeling for injection molding processes using deep learning. *Struct Multidiscip Optim* 2022;65:305. <http://dx.doi.org/10.1007/s00158-022-03380-0>.
- [18] Greif J, Lechner P, Meyer N. Fast approximation of fiber reinforced injection molding processes using Eikonal equations and machine learning. *Compos-A: Appl Sci Manuf* 2024;185:108340. <http://dx.doi.org/10.1016/j.compositesa.2024.108340>.
- [19] Koch S, Matveev A, Jiang Z, Williams F, Artemov A, Burnaev E, Alexa M, Zorin D, Panozzo D. ABC: A Big CAD Model Dataset For Geometric Deep Learning. In: The IEEE conference on computer vision and pattern recognition (CVPR). 2019, p. 1–15. <http://dx.doi.org/10.48550/arXiv.1812.06216>.
- [20] Advani SG, Tucker CL. The Use of Tensors to Describe and Predict Fiber Orientation in Short Fiber Composites. *J Rheol* 1987;31(8):751–84. <http://dx.doi.org/10.1122/1.549945>.
- [21] Phelps JH, Tucker CL. An anisotropic rotary diffusion model for fiber orientation in short- and long-fiber thermoplastics. *J Non-Newton Fluid Mech* 2009;156(3):165–76. <http://dx.doi.org/10.1016/j.jnnfm.2008.08.002>.
- [22] Bathe KJ. *Finite Element Procedures*. New Jersey: Prentice-Hall; 1996.
- [23] Hempel T, Abdelrahman AA, Al-Hamadi A. 6D Rotation Representation For Unconstrained Head Pose Estimation. In: 2022 IEEE international conference on image processing (ICIP). 2022, p. 2496–500. <http://dx.doi.org/10.1109/ICIP46576.2022.9897219>.
- [24] Tucker CL. *Fundamentals of Fiber Orientation - Description, Measurement and Prediction*. Munich: Hanser; 2022. <http://dx.doi.org/10.3139/9781569908761>.
- [25] Bauer JK, Böhlke T. Variety of fiber orientation tensors. *Math Mech Solids* 2022;27(7):1185–211. <http://dx.doi.org/10.1177/10812865211057602>.
- [26] Tandon GP, Weng GJ. The effect of aspect ratio of inclusions on the elastic properties of unidirectionally aligned composites. *Polym Compos* 1984;5(4):327–33. <http://dx.doi.org/10.1002/pc.750050413>.
- [27] Chung DH, Kwon TH. Invariant-based optimal fitting closure approximation for the numerical prediction of flow-induced fiber orientation. *J Rheol* 2002;46(1):169–94. <http://dx.doi.org/10.1122/1.1423312>.