Key Engineering Materials
ISSN: 1662-9795, Vol. 926, pp 2247-2257
doi:10.4028/p-1o0007

Submitted: 2021-12-15
Revised: 2022-01-24
Accepted: 2022-01-25
Online: 2022-07-22

Formability Assessment of Variable Geometries Using Machine Learning - Analysis of the Influence of the Database

Clemens Zimmerling[1,a*], Benedikt Fengler[1,2,b] and Luise Kärger[1,c]

[1]Karlsruhe Institute of Technology (KIT), Institute of Vehicle System Technology, Karlsruhe, Germany

[2]Simutence GmbH, Karlsruhe, Germany

[a]clemens.zimmerling@kit.edu, [b]benedikt.fengler@simutence.de, [c]luise.kaerger@kit.edu

Keywords: Textile Forming, Composite Manufacturing, Machine Learning, Surrogate Modelling

Abstract. Surrogate modelling has proven to be an effective strategy for time-efficient analysis and optimisation of expensive functions such as manufacturing process simulations. However, most surrogate approaches generate problem-specific "one-off" models, which cannot be reused in other, even similar scenarios. Hence, variations of the problem, e.g. minor geometry changes, instantly invalidate the surrogate. Image-based machine learning (ML) techniques have been proposed as an option to train a surrogate for variable geometries. However, it is currently unclear how to construct a sufficiently diverse set of generic training geometries and what effect different databases have.

This work investigates the effect of different databases on the prediction accuracy of an ML-assessment of component manufacturability. The considered use-case is textile forming (draping) of a woven fabric. Sampling plans generate different numbers of training geometries, which are in turn evaluated in draping simulations. An image-based ML-algorithm is trained on these process samples and evaluated on a set of validation geometries. Results show that the diversity of the training geometries has a greater impact on the prediction accuracy than the number of samples. The results also hint that a comparably low number of geometry samples suffices to give meaningful results. With these findings, ML-techniques are considered a promising and time-efficient tool for manufacturability assessment at early stages of part and process design.

Introduction

Most industrial manufacturing processes require careful engineering of component design, manufacturing process and material to ensure acceptable part quality, cycle time and cost [1]. While for most processes the governing physics are well understood, efficiently determining a process optimum for a given geometry is still a profound question, though [2]. In many cases, companies rely on empirical best-practice guidelines in combination with experimental trials. Although such approaches often yield satisfactory results, they often entail significant rework for error correction and process fine-tuning [3].

This holds all the more for complex processes and materials such as composites and makes process optimisation a significant concern during part development. Flexible manufacturing technologies, shrinking lot sizes and shorter development cycles ("mass customisation") further compound this situation as they require ever more frequent process reconfigurations [4].

To reduce the experimental work during process optimisation, high-fidelity simulation methods, e.g. based on Finite Elements (FE), are used. They allow for a rigorous virtual analysis of the process dynamics prior to experimental trials and may provide additional information when experimental evidence is scarce [5]. In combination with optimisation algorithms, they also allow for automated search of optimal parameters – often referred to as "virtual process optimisation" [6-8]. However, the computational effort for iterative optimisation often renders them impracticable in practice.

One option to reduce the computational burden in virtual process optimisation is surrogate-based optimisation (SBO) [9]. Surrogates are numerically efficient, data-driven approximations of expensive simulations. Once trained, the surrogate guides the optimiser in the parameter space and concentrates costly simulations on the most promising variants. For material forming, as considered

in this work, a rich corpus of SBO applications examples exists [10-14], yet employing surrogates for component design is just at the beginning. For instance, Horton et al. relate geometry parameters to experimental and numerical forming results [15]. Similarly, FE forming simulations have been used to devise design maps for preselected geometry features in metal and textile forming [16,17].

Most of the current surrogate-approaches require an a-priori definition of geometry parameters such as corner radii or draft angles. Although remarkable results have been achieved, this limits their applicability to a pre-defined geometry class. This may be sufficient in case of recurring geometries, e.g. standard parts from norms, yet the evaluation of geometries outside the parameterisation scheme proves difficult. Also, most surrogate approaches predict scalar- or low-dimensional features only, e.g. maximum strain. Predictions of field quantities are rarely considered, although they provide valuable engineering information, e.g. the location of defects. See [14,14] or [18] for example implementations.

Machine Learning (ML) techniques using convolutional neural networks (CNN) have been proposed as a potential remedy, see e.g. [19,20] for examples from fluid mechanics and fabric forming. Unlike classical surrogates, CNNs interpret images and allow to describe the geometry through pixels or voxels in a grid-wise manner instead of distinct geometry parameters. This approach is more generalisable, but brings about sampling difficulties: For classical parameter spaces, so-called space-filling sampling plans have been developed to build a training database, e.g. Latin Hypercube Sampling. However, for a non-parametric problem, such space-filling strategies do not exist currently. In [19,20] for example, the authors randomly vary size, location and orientation of primitives (triangle, rectangles, ellipses, …), but give no justification as to why they proceeded this way. They neither study different training database compositions or sample sizes.

This work aims to illustrate the effect of different databases on prediction accuracy of an image-based ML-assessment of component manufacturability. Textile forming (draping) of a plain-weave woven fabric is taken as an example use case. To this end, two generic geometry classes (GC) are defined, which show different forming-relevant features, e.g. single vs. double curvature. From these GCs different numbers of samples are drawn and evaluated in textile forming simulations, which constitutes the database for ML-training. To keep the computation time t_{sim} within reasonable bounds for this study of different databases with hundreds of simulations, a simplified kinematic simulation ($t_{\mathrm{sim}}^{\mathrm{kin}} \approx 1.3\,\mathrm{min}$) approach is chosen over an FE-simulation ($t_{\mathrm{sim}}^{\mathrm{fem}} \approx 32\,\mathrm{min}$). However, as the algorithm's learning behaviour is unaffected of the underlying simulation approach, the results are deemed equally valid for both.

On each of the databases, an image-processing ML-algorithm is trained. After training, its performance is validated against actual simulations in two steps: At first, against separate validation geometries from the GC and, second, against a common benchmark geometry outside the GC. Thereby the generalisation capabilities of different databases can be compared.

Modelling Approach

Conceptual View. Formally, this work views a forming simulation as a function $\varphi_{\mathrm{sim}}: G \mapsto Q$, which maps an input geometry $g \in G$ to a part quality attribute $q \in Q$. In material forming, as considered in this work, the part quality is often quantified by material strain. In-plane shear is the main deformation mode for woven fabrics. Due to progressive yarn compaction and according in-plane compressive stress accumulation, excessive shear increases the likelihood of forming defects such as wrinkling or textile folding and poor resin permeability. Thus, the shear angle γ_{12} is a frequently used quality measure in fabric draping and is usually sought to be minimised

As the evaluation of φ_{sim} is generally costly, a numerically efficient substitute function μ_{ML} (the surrogate) is sought to approximate φ_{sim}. Please note that φ_{sim} can be evaluated but is otherwise unknown ('black-box'-conditions). This excludes analytical approximation techniques such as Taylor-Series and leaves data-driven approximations as the only viable option to build the surrogate.

For a data-driven approximation, a highly flexible model function μ_{ML} is selected and fitted to a dataset D_T with observation samples (training). In general, different model functions are at disposal

ranging from simple polynomials to advanced techniques such as Gaussian Process Regression, Support Vector Machines and Artificial Neural Networks (ANN) [21].

Neural Networks. Their extraordinary modelling capacity makes ANNs so-called universal approximators [22]. That is, given sufficient training data, they can reproduce any continuous function regardless of its complexity. ANNs are typically organised in a layer-structure with each layer containing a predefined number of neurons. When processing data, these neurons are activated or deactivated through weighting and summation operations. Thereby, complex activation patterns form within the layers. The last layer converts these patterns to an output quantity for engineering interpretation.

The weights and biases of the ANN are free network parameters $\boldsymbol{\theta}$ which are collectively tuned to match supplied samples in the training database D_T. The optimal parameters $\boldsymbol{\theta}^*$ are the ones which minimise the loss-function L, i.e. the error between μ_ML's estimation $\widetilde{\boldsymbol{\gamma}}_{12}$ and the ground truth $\boldsymbol{\gamma}_{12}$

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} L(\widetilde{\boldsymbol{\gamma}}_{12}, \boldsymbol{\gamma}_{12}) \,. \tag{1}$$

Please note that, in this case $\boldsymbol{\gamma}_{12}$ and $\widetilde{\boldsymbol{\gamma}}_{12}$ are field quantities stored in vector-format, e.g. $\boldsymbol{\gamma}_{12} = (\gamma_{12}^1, \gamma_{12}^2, \ldots, \gamma_{12}^{n_\mathrm{el}})$ with γ_{12}^k denoting the shear angle of element $k = 1,2,\ldots,n_\mathrm{el}$.

In general, different error metrics can be used, while this work employs the L_2-metric, also known as mean squared error MSE:

$$L = L_2 = \frac{1}{n_T}\sum_{i=1}^{n_T}\left(\widetilde{\boldsymbol{\gamma}}_{12_i} - \boldsymbol{\gamma}_{12_i}\right)^2 , \tag{2}$$

with n_T being the number of training samples in the database.

To evaluate μ_ML's performance on new data, it is common practice to hold out a separate set of data during training – the validation set D_V. While parameter optimisation is done on the training set D_T only, the loss on D_V informs about generalisation capabilities of the ANN. To ensure comparability between the algorithms, only the samples number n_T of D_T is varied. The validation database D_V is kept constant.

Data Representation. The following idea was originally proposed in [20] and is only briefly reiterated here: In stamp forming processes, component geometries must necessarily be undercut-free to avoid collision during tool-closure. This allows to describe the geometry by its elevation $h(x, y)$ above the tool plane. Analogous to a topographic map, h can then be efficiently described by greyscale-values as Fig. 1 shows.

The greyscale values GS range from 0 to 255, while 0 ('black') denotes $h = 0\,\mathrm{mm}$ and 255 ('white') corresponds a predefined maximum elevation h_{max}. In this work $h_{max} = 150\ \mathrm{mm}$ was chosen, but depending on the application other dimensions may be used.

The strain field can also be encoded in a greyscale-image: For the simulation, the fabric is discretised into n_{el} elements which undergo deformation during forming. Plotting the elemental strains into the undeformed fabrics then yields a 2D-representation of the forming result, which can again be conveniently represented as an image.

Both, the greyscale images for the geometry and the strain field constitute a pair of values for training of the network for end-to-end learning of the forming dynamics.

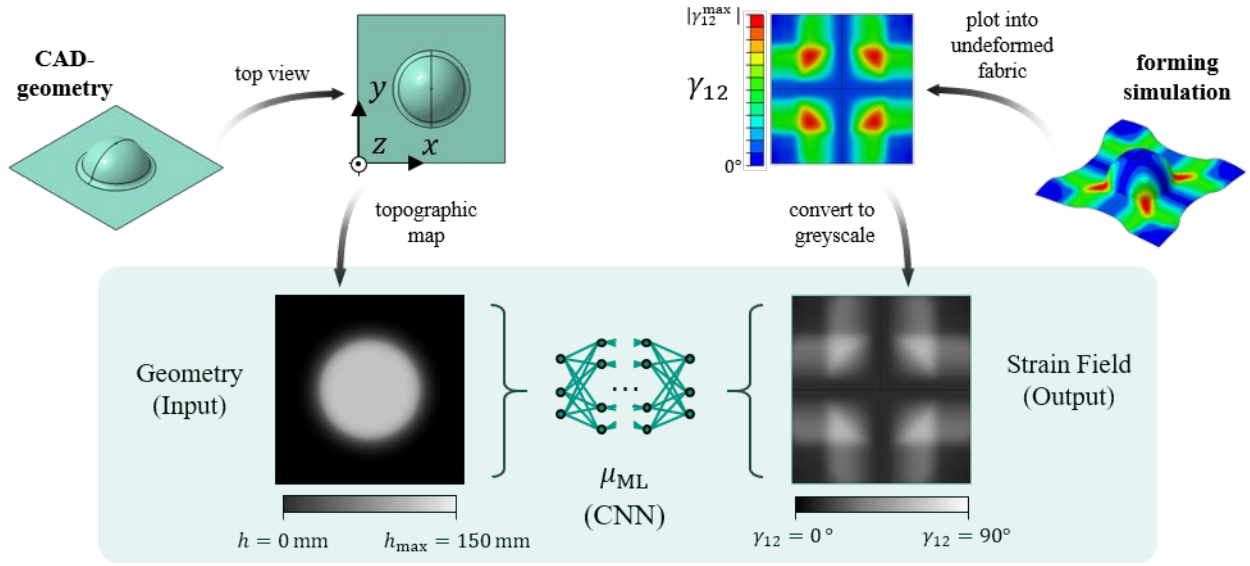


Fig. 1: Image-based data representation as proposed in [20].

Neural Network Architecture. Convolutional neural networks (CNN), are deliberately designed to take advantage of spatially structured data like images and are thus used in this work. More specifically, this work employs a CNN-architecture from prior work for end-to-end training of μ_{ML}. See [20] for details. In essence, CNNs slide ('convolve') a pre-defined number of templates ('kernels') across an input image and continuously measure the degree of coincidence. Depending on the local degree of coincidence, neurons in the subsequent layer are activated and so-called 'feature maps' form. The repeated application on subsequent network layers ('deep learning') forms higher-order feature maps, which are essentially a compressed representation of forming-relevant information. The second part of the network then interprets this compressed information through repeated transpose-convolutional operations. One evaluation of μ_{ML} plus data loading into memory takes about $t_{\mathrm{evl}} \approx 0.45\,s$ on an *nVidia™ RTX 2080 Ti* graphics card – a significant reduction compared to simulation efforts $t_{\mathrm{sim}}^{\mathrm{kin}}$ and $t_{\mathrm{sim}}^{\mathrm{fem}}$, respectively.

Study procedure

Geometry parameterisation. Two separate classes of generic geometries are considered – GC 1 and GC 2. To study the effect of different databases, μ_{ML} is at first trained separately on GC 1 and GC 2 and subsequently on a combination of both. In order to generate an arbitrary number of training samples n_{T}, a parametric CAD-model is generated. For both geometries the equation

$$1 = \left(\frac{y}{y_{\max}}\right)^{p} + \left(\frac{z}{z_{\max}}\right)^{p} \tag{3}$$

describes the component surface contour line. Therein, $y_{\max}$ and $z_{\max}$ denote the largest y- and z-coordinate of the component. Fig. 2 visualises equation (3) along with some contour examples. For GC 1 the contour is rotated about the z-axis (360° rotation); for GC 2 the contour is at first mirrored at the xz-plane, indicated by dashed lines and then rotated about the y-axis by $\pm 90°$.

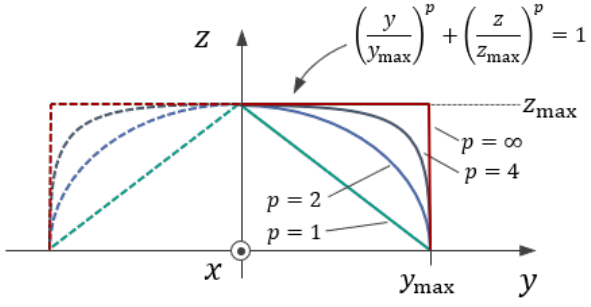


Fig. 2: Visualisation of the contour line parameterisation for geometry sampling.

By variation of p, Equation (3) yields a set of highly diverse geometries, ranging from cones ($p = 1$), spheres/ellipses ($p = 2$) to near-cylindrical structures ($p \gg 2$). Their height and aspect ratio can be varied through $z_{\max}$ and $y_{\max}$, respectively. In this work, the parameter range $1 \leq p \leq 6$, $50\,\text{mm} \leq y_{\max} \leq 150\,\text{mm}$ and $0.05 \leq z_{\max}/y_{\max} \leq 1.0$ has been investigated, while any sharp edges are rounded with the radius $r = 10\,\text{mm}$. Fig. 3 a) highlights the axis of rotation and Fig. 3 b) shows some geometry samples. For comparison of the two geometry classes, each sample from GC 1 (top) has a counterpart from GC 2 obtained with the same values for p, $y_{\max}$, $z_{\max}$.

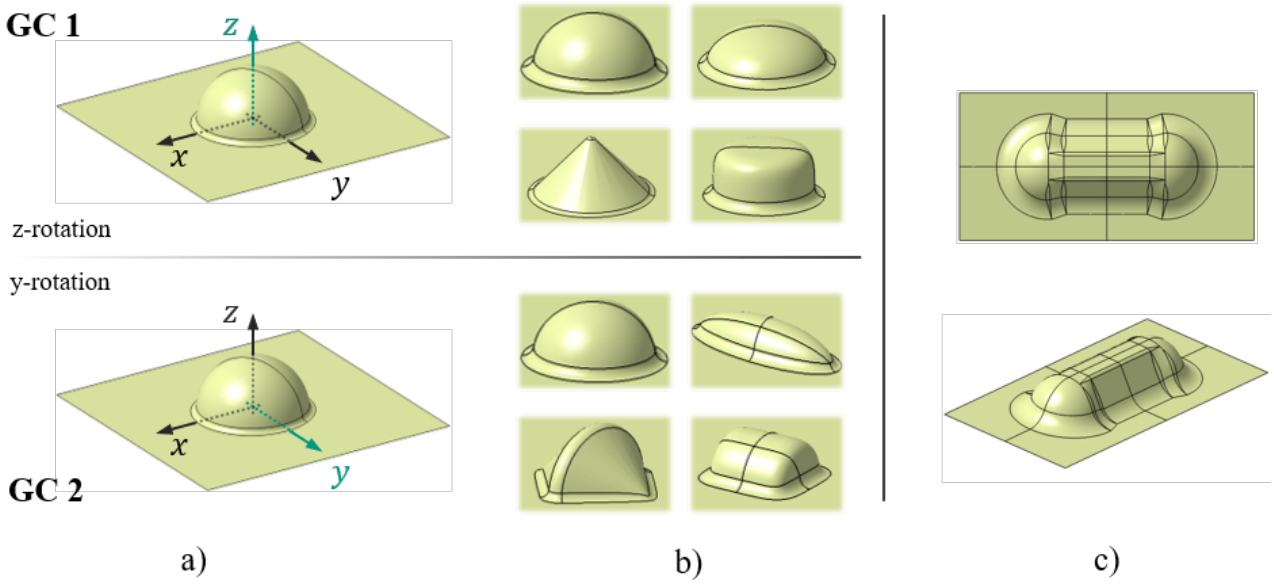


Fig. 3: a) Geometry sampling by rotation of the contour line around the z-axis (GC 1) and y-axis (GC 2), b) geometry samples for GC 1 and GC 2 for visualisation, c) Validation geometry outside GC 1 and GC 2 ('double-dome').

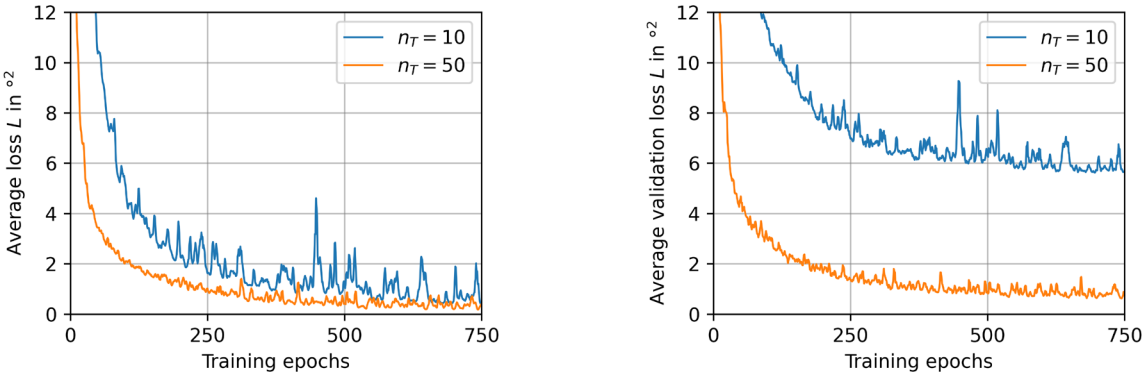
Database and validation geometries. Using Latin Hypercube Sampling, from both geometry classes five different-sized training databases D_T are sampled with $n_\text{T} = \{5, 10, 25, 50, 100\}$ samples. Additionally, for each GC a separate validation D_V is sampled with 20 validation geometries to assess μ_ML's performance on unseen geometry samples.

Ultimately, a test geometry outside GC 1 and GC 2 is evaluated, the so-called 'double-dome' as shown in Fig. 3 c). It is a common benchmark geometry in stamp forming as it features several forming-relevant characteristics such as single- and double-curved areas as well as convex and concave regions. Please note that – despite some similarity – neither GC 1 nor GC 2 can exactly reproduce the double-dome.

Network training. As the network's parameters $\boldsymbol{\theta}$ are initialised randomly, individual training runs show scatter. Thus, for each training database D_T ten independent training runs are performed and their average and 95 % confidence interval evaluated. The prediction quality is measured by the loss L, the mean absolute error of the shear angle maximum $MAE(\gamma_{\max})$, the mean absolute error of the maximal shear difference $MAE(\max(\Delta\boldsymbol{\gamma}))$, the coefficient of determination c_d and the Pearson correlation coefficient r. See the appendix for the metrics' definition. The metrics serve different purposes: While L, $MAE(\gamma_{\max})$ and $MAE(\max(\Delta\boldsymbol{\gamma}))$ measure absolute discrepancies, c_d and r inform about the spatial correlation between ground truth and estimated strain fields. Loosely speaking, the latter measure how well μ_ML is able to locate peaks in strain field. Training of μ_ML is done using the ADAM optimiser for $k_\text{ep} = 750$ epochs. A pre-study showed that a learning rate $\alpha_\text{LR} = 5 \cdot 10^{-5}$ and a batch size of 1 gave best results. With these settings one gradient-descent takes at average $t_\text{gd} \approx 0.027\,\text{s}$ per batch in each epoch. The total training time t_trn depends on the number of training samples n_T and amounts to $t_\text{trn} = t_\text{gd} \cdot k_\text{ep} \cdot n_\text{T} = 0.027\,\text{s} \cdot 750 \cdot n_\text{T} = 20{,}25\,\text{s} \cdot n_\text{T}$. Thus, it ranges between $101.25\,\text{s} \leq t_\text{trn} \leq 2025\,\text{s}$ for $n_\text{T} = 5$ and 100 geometries, respectively.

Results & Discussion

Training progress. Fig. 4 visualises μ_{ML}'s training progress as measured by the loss L for two exemplary database sizes of GC 1, namely $n_{\mathrm{T}} = 10$ and 50 samples.

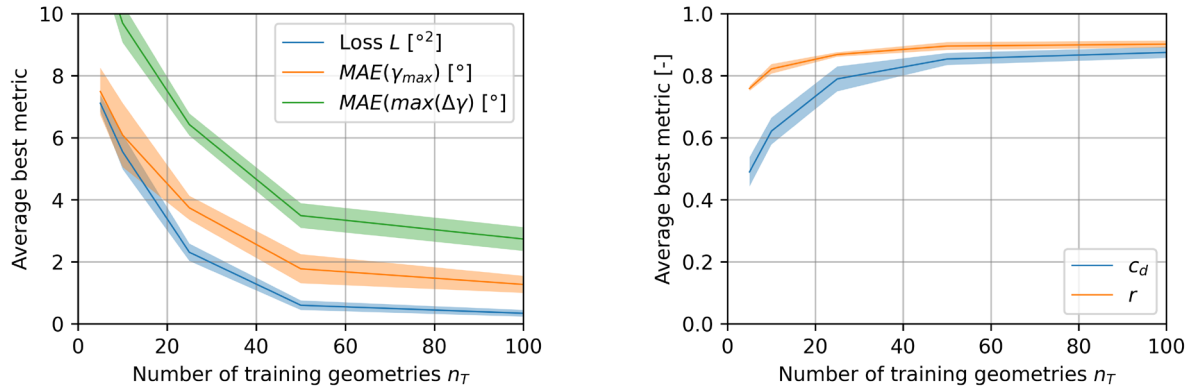


a) Loss on training geometries b) Loss on validation geometries

Fig. 4: Training progress of μ_{ML} during training on GC 1.

Fig. 4 a) shows the loss on training data averaged over ten training runs. Like a regular surrogate, for both $n_{\mathrm{T}} = 10$ and $n_{\mathrm{T}} = 50$ the loss L approaches zero in a roughly monotonous manner. Loosely speaking, the networks learns to reproduce the training data. The loss on the validation data (Fig. 4 b)), however, reveals a difference: While the graph still approaches almost zero ($L \approx 1°^2$) for $n_{\mathrm{T}} = 50$, for $n_{\mathrm{T}} = 10$ an error of $L \approx 6°^2$ persists. This implies that 10 training samples do not contain sufficient information for an accurate generalisation to new geometries from GC 1, but 50 do. Consequently, a threshold number of training geometries must exist, beyond which the prediction accuracy stays approximately constant. This threshold is closer examined in the following.

Performance on validation data. Fig. 5 shows the evolution of the performance metrics on the validation data for different training samples n_T. More specifically, it shows the average of the best achieved values and their 95 % confidence intervals across ten separate training runs. Fig. 5 splits into two separate plots. Plot a) shows metrics measuring absolute differences, whereas plot b) shows statistical correlation measures between prediction and ground truth.



a) b)

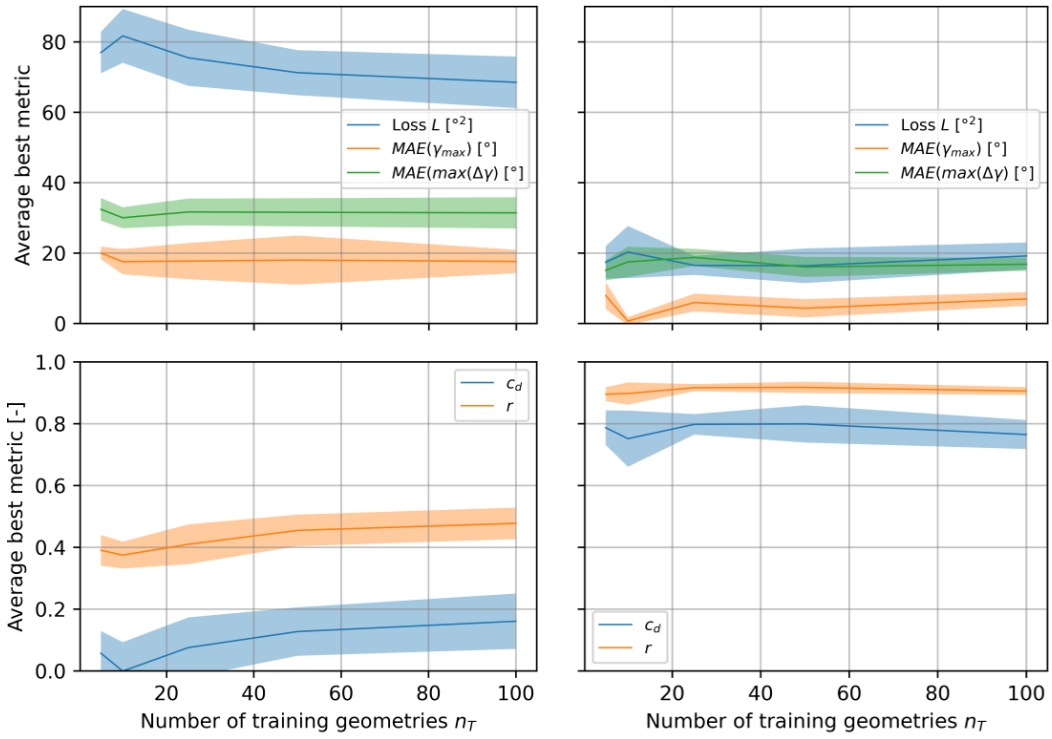
Fig. 5: Performance metrics of μ_{ML} on validation data (inside of geometries) for different sizes of the training database after training on GC 1. Clearly, μ_{ML}'s performance improves with n_T.
Similar behaviour is found for training on GC 2.

Please note that the sense of prediction quality is opposite in both: In plot a), a perfect estimation implies zero error metric, in plot b) $c_{\mathrm{d}} = r = 1$ denotes a perfect estimation.

In both plots the performance metrics constantly improve with increasing n_T. However, for $n_\mathrm{T} \geq 50$, the metrics improves only moderately, which indicates the aforementioned threshold. Interestingly, for the correlation metrics c_d and r this threshold seems to appear a little earlier ($n_\mathrm{T} \geq 25$). This may indicate that it is easier to learn the spatial relation between shear strains rather than absolute values but requires further investigation for confirmation. Similar behaviour is found for GC 2, although better metrics are observed for c_d ($0.75 < c_\mathrm{d} < 1.0$) and r ($0.85 < r < 1.0$). To avoid redundancy, the graphs are omitted here.

Performance on test data outside training geometries. Similar to a classical surrogate, when evaluating geometries inside the training geometry classes, μ_ML's accuracy improves with the size of the training database (Fig. 5). However, as outlined in the introduction, when aiming for generalisation to arbitrary geometries, the accuracy on geometries outside the class of generic training geometries is important.

To this end, the double-dome (cf. Fig. 3 c) is taken as a benchmark geometry. Fig. 6 plots the according performance metrics against n_T: a) after training on GC 1, b) after training on GC 2. Surprisingly, in both cases varying n_T has little effect and the graphs remain approximately constant; at most the accuracy increases moderately in sub-image a), i.e. when training on GC 1.



a) Training on GC 1 b) Training on GC 2

Fig. 6: Performance metrics of μ_{ML} on the double-dome geometry (outside of geometry class) for different sizes of the training database. Unlike Fig. 5, μ_{ML}'s performance hardly improves with more training samples but stays constant. At the same time, a large difference can be observed between GC 1 and GC 2, i.e. the composition of the training data.

However, comparing the sub-images a) and b) reveals a great difference: When training on GC 1, the loss L amounts at best to $L \approx 68°^2$, while only $L \approx 20°^2$ is found for GC 2 – a ≈ 70 % difference. Equally large differences are observed for the other metrics. This hints that GC 2 holds more useful information than GC 1 for training of μ_ML. This statement holds independent of the number of samples since $n_T = 5$ (GC-2) already outperforms $n_T = 100$. For completeness, μ_ML has also been trained on a combined database, i.e. GC 1+2. However, the resulting plots are practically identical to Fig. 6 b) and are omitted to avoid redundancy.

Overall, the results show that samples from GC 1 contribute only marginally to μ_{ML}'s performance. Thus, when aiming for full generalisation, carefully selecting the training geometries classes is far more important than the sheer number of samples per class. In surrogate modelling and deep learning this phenomenon is sometimes casually summarised by "smart data outperforms big data". Although it is admitted that this work does not truly use "big data", i.e. overwhelmingly large and complex datasets, but well-defined and carefully curated data sets.

For a visual assessment of the μ_{ML}'s estimations, Fig. 7 shows the estimated and the simulated strain field for the double-dome in three sub-images a)-c): It visualises the results after training a) on GC 1, b) on GC 2 and c) on both GC 1+2. In each case $n_{\mathrm{T}} = 100$ samples were in the database. The strain field difference $\Delta\boldsymbol{\gamma} = \mu_{\mathrm{ML}}(g_{\mathrm{dd}}) - \varphi_{\mathrm{sim}}(g_{\mathrm{dd}})$ is also given and allows the analysis of local over- and underestimations. For ease of readability, each plot also quantifies its maximum absolute values.

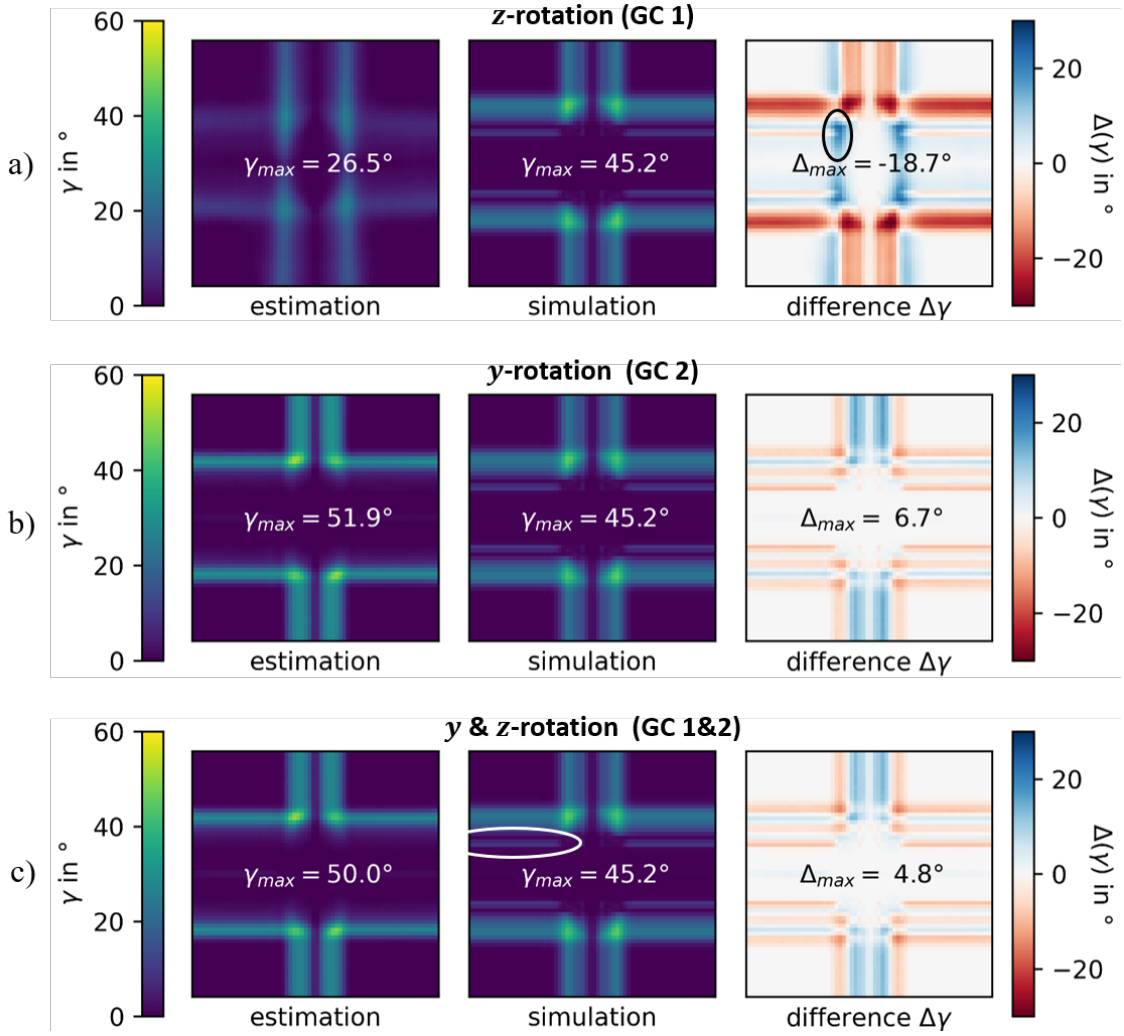


Fig. 7: Comparison of the shear strain estimation by μ_{ML} and a forming simulation along with a plot of the differences.

Fig. 7 a) clearly shows that μ_{ML}'s estimation resembles the simulation result only regarding the vertical and horizontal shear bands with a local maximum at their intersection. Their location and width, however, differ significantly. Also, μ_{ML} substantially underestimates the shear angles (red areas in difference plot). At the same time, it predicts shear strains in the image centre (cf. black ellipsis in difference plot). However, no shear deformation can occur there since the double-dome only shows single curvature in its centre. This overestimation takes place because all geometries in GC 1 are rotation-symmetric and none of them feature straight areas. As a consequence, μ_{ML} remains uninformed, how straight parts behave during forming.

In Fig. 7 b), i.e. training on GC 2, the results have significantly improved: The horizontal and vertical shear bands and their intersection with a local maximum match far better the simulation results – not just qualitatively but also quantitatively. Also, the erroneous shear deformation in the centre do not occur. Since GC 2 encompasses samples with straight areas (cylinder form $p \gg 2$), μ_{ML} learns that straight component features evoke no shear.

Training μ_{ML} on a combination of the available samples of GC 1 and GC 2 improves the results only marginally as Fig. 7 c) shows. It is to the widest extent identical to sub-image b) except for a minimally improved accuracy regarding the maximum shear strain. From the perspective of surrogate-modelling, this has the advantage that combining geometry classes does not deteriorate the surrogate performance, even if the added geometry class introduces no relevant information. Consequently, adding geometries to the database at best helps improve the surrogate and has at worst no effect, i.e. is conservative.

It may be noted that neither of the estimations in a)-c) is able to reproduce the thin shear bands near the main deformation zone (cf. white ellipsis in c)). This shear band stems from a doubly-curved, concave indentations of the double-dome (cf. Fig. 3 c)). Yet, like with the straight geometry features, neither GC 1 nor GC 2 comprise concave geometry samples thus μ_{ML} remains unaware of their effect on shear deformations.

Summary and Conclusion

This work studies the effect of two different classes of generic geometries to train an image-based ML-surrogate for manufacturability assessment of variable geometries. The first geometry class primarily varies the out-of-plane curvature, the second in-plane curvature. Additionally, a combined database with samples from both geometry classes is studied. The considered use-case is forming of a plain-weave woven fabric. For each geometry class five different-sized sets of training geometries are sampled and their forming results evaluated in a forming simulation. On these process samples, an image-based ML-algorithm μ_{ML} from prior work [20] is trained. After training, μ_{ML} is evaluated on a set of validation geometries from the class of training geometries and on a benchmark geometry outside the geometry class.

Results show that μ_{ML} consistently improves its accuracy for new geometries within the class of training geometries, when the number of training geometries increases. However, this does not hold for geometries outside the training geometry class. In this case, the prediction accuracy is nearly independent of the number of training samples. Instead, the geometry class itself – more precisely, their geometry features – has by far the largest impact on the prediction accuracy. The results also show, that geometry classes can be combined without impairing the prediction accuracy, even if the added geometry class provides little additional information. That is, adding geometries is conservative.

The findings suggest that carefully engineering and selecting the classes of generic training geometries is more important than the sheer number of samples. Loosely speaking, the findings support the notion of “smart data outperforms big data”. When aiming for a fully generalised surrogate, follow-up studies need to investigate, which geometry characteristics are most important to be covered during training. This work already highlights the importance of single and double curvature areas as well as convex and concave features, but other features – and according test geometries – may be added in the future.

Acknowledgment

This work is part of the IGF research project *OptiFeed* (21949 N) of the research association Forschungskuratorium Textil e.V., Reinhardtstraße 14-16, 10117 Berlin, funded via the AiF within the program for supporting the “Industrial Collective Research” (IGF) from funds of the Federal Ministry of Economic Affairs and Climate Protection (BMWK). It is also part of the Young Investigator Group (YIG) *Green Mobility*, generously funded by the *Vector Stiftung*. All support is gratefully acknowledged.

Appendix

Metric definitions:

$MAE(\gamma_{\max}) = \text{mean}[\max(\hat{\boldsymbol{\gamma}}_{12_i}) - \max(\boldsymbol{\gamma}_{12_i})]$	Mean absolute error of the maximal true and estimated strain
$MAE(\max(\Delta\boldsymbol{\gamma})) = \text{mean}[\max(\lvert\hat{\boldsymbol{\gamma}}_{12_i} - \boldsymbol{\gamma}_{12_i}\rvert)]$	Mean absolute error of the maximal strain difference
$c_{\mathrm{d}} = 1 - \frac{\sum_i \left(\boldsymbol{\gamma}_{12_i} - \hat{\boldsymbol{\gamma}}_{12_i}\right)^2}{\sum_i \left(\boldsymbol{\gamma}_{12_i} - \text{mean}(\boldsymbol{\gamma}_{12})\right)^2}$	Coefficient of Determination [23]
$r = \frac{\sum_i \left(\hat{\boldsymbol{\gamma}}_{12_i} - \text{mean}(\hat{\boldsymbol{\gamma}}_{12})\right)\left(\boldsymbol{\gamma}_{12_i} - \text{mean}(\boldsymbol{\gamma}_{12})\right)}{\sqrt{\sum_i \left(\hat{\boldsymbol{\gamma}}_{12_i} - \text{mean}(\hat{\boldsymbol{\gamma}}_{12})\right)^2}\sqrt{\sum_i \left(\boldsymbol{\gamma}_{12_i} - \text{mean}(\boldsymbol{\gamma}_{12})\right)^2}}$	Pearson correlation coefficient [23]

References

[1] D.M. Anderson, Design for Manufacturability: How to Use Concurrent Engineering to Rapidly Develop Low-Cost, High-Quality Products for Lean Production, Taylor & Francis, United Kingdom, 2014.

[2] H.S. Jagdev, J. Browne, J. Keogh, Manufacturing Process Optimisation – A Survey of Techniques, in: B.J. Davies (Eds), Proc. 28th Intl. Matador Conf., Palgrave, London. 1990, pp. 205-215

[3] I. Dostaler, Avoiding rework in product design: evidence from the aerospace industry, Int. J. Qual. Reliab. Manage. 27 (2010) 5–26.

[4] Y. Koren, The Global Manufacturing Revolution: Product-Process-Business Integration and Reconfigurable Systems. Wiley, USA, 2010.

[5] L. Kärger, A. Bernath, F. Fritz, S. Galkin, D. Magagnato, A. Oeckerath, K.Wolf, Development and validation of a CAE chain for unidirectional fibre reinforced composite components, Compos. Struct. 132 (2015) 350–358.

[6] L. Kärger, S. Galkin, C. Zimmerling, D. Dörr, J. Linden, A. Oeckerath, et al., Forming optimisation embedded in a cae chain to assess and enhance the structural performance of composite components, Compos Struct 192 (2018) 143–152.

[7] S. Chen, L.T. Harper, A. Endruweit, N.A. Warrior, Formability optimisation of fabric preforms by controlling material draw-in through in-plane constraints, Composites Part A 76 (2015) 10–19

[8] B. Fengler, M. Schäferling, B. Schäfer, L. Bretz, G. Lanza, B. Hafner, A. Hrymak, L. Kärger, Manufacturing uncertainties and resulting robustness of optimized patch positions on continuous-discontinuous fiber reinforced polymer structures, Compos. Struc. 213, 47-57, 2019

[9] S. Koziel, L. Leifsson, Surrogate-based Modeling and Optimization, first ed., Springer, New York, 2013

[10] J. Jakumeit, M. Herdy, M. Nitsche, Parameter optimization of the sheet metal forming process using an iterative parallel Kriging algorithm, Structural and Multidisciplinary Optimization (29) (2005), 498-507

[11] M.H.A. Bonte, A.H. van den Boogaard, J. Huétink, A Metamodel Based Optimisation Algorithm for Metal Forming Processes, in: Adv. Methods in Material Forming, Springer Berlin/Heidelberg, (2007), 55-72

[12] H.Wang, F. Ye, L. Chen, E. Li, Sheet metal forming optimization by using surrogate modeling techniques. Chinese Journal of Mechanical Engineering 30, 22–36, 2017

[13] J. Pfrommer, C. Zimmerling, J. Liu, L. Kärger, F. Henning, J. Beyerer, Optimisation of manufacturing process parameters using deep neural networks as surrogate models, Proc. CIRP 72, 2018, 426–431

[14] C. Zimmerling , P. Schindler, J. Seuffert, L. Kärger, Deep neural networks as surrogate models for time-efficient manufacturing process optimization, Proceedings of ESAFORM 2021, Liège/Belgium, 2021

[15] P.M. Horton, J.M. Allwood, C. Cleaver, A. Nagy-Sochacki, An experimental analysis of the relationship between the corner, die and punch radii in forming isolated flanged shrink corners from Al 5251, Journal of Materials Processing Technology 278 (2020) 116486

[16] H.R. Attar, N. Li, A. Foster, A new design guideline development strategy for aluminium alloy corners formed through cold and hot stamping processes, Materials & Design 207, 2021,109856

[17] C. Zimmerling, D. Dörr, F. Henning, L. Kärger, A Machine learning assisted approach for textile formability assessment and design improvement of composite components, Composites Part A 124, 2019

[18] G. Misiun, C. Wang, H. Geijselaers, A. van den Boogaard, Interpolation of final geometry and result fields in process parameter space, Numiform 80, 2016, 16010-1 – 16010-6

[19] X. Guo, W. Li, F. Iorio, Convolutional Neural Networks for Steady Flow Approximation. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data, 2016, 481-490

[20] C. Zimmerling, D. Trippe, B. Fengler, L. Kärger, An approach for rapid prediction of textile draping results for variable composite component geometries using deep neural networks, AIP Conf. Proc. 2113, AIP Publishing, 2019

[21] A.I.J. Forrester, A. Sóbester, A.J. Keane, Engineering design via surrogate modelling: A practical guide, Wiley, USA, 2008.

[22] K. Hornik, Approximation capabilities of multilayer feedforward networks, Neural Networks 4, 1991, 251–257

[23] P.S. Mann, Mann’s Introductory Statistics, ninth ed., Wiley, Hoboken, 2017