Metamodeling of a deep drawing process using conditional Generative Adversarial Networks

Patrick Link*1, Johannes Bodenstab1, Lars Penter1,2 and Steffen Ihlenfeldt1,2

1 Fraunhofer Institute for Machine Tools and Forming Technology IWU, 09126 Chemnitz, Germany
2 Institute of Mechatronic Engineering, Technische Universität Dresden, 01062 Dresden, Germany

*patrick.link@iwu.fraunhofer.de

Abstract. Optimization tasks as well as quality predictions for process control require fast responding process metamodels. A common strategy for sheet metal forming is building fast data driven metamodels based on results of Finite Element (FE) process simulations. However, FE simulations with complex material models and large parts with many elements consume extensive computational time. Hence, one major challenge in developing metamodels is to achieve a good prediction precision with limited data, while these predictions still need to be robust against varying input parameters. Therefore, the aim of this study was to evaluate if conditional Generative Adversarial Networks (cGAN) are applicable for predicting results of FE deep drawing simulations, since cGANs could achieve high performance in similar tasks in previous work. This involves investigations of the influence of data required to achieve a defined precision and to predict e.g. wrinkling phenomena. Results show that the cGAN used in this study was able to predict forming results with an averaged absolute deviation of sheet thickness of 0.025 mm, even when using a comparable small amount of data.

1. Introduction
In the past decade machine learning for optimization of processes became a widely used method in manufacturing [1]. Machine learning algorithms are often used to develop data-driven metamodels, which predict product quality as a function of relevant process parameters. In deep drawing these metamodels can be used for process optimization [2] or can be applied as fast prediction models in process control schemes [3]. The amount and quality of available data for training significantly affects the predictive power of the metamodel. The prediction of entire part geometry is of particular interest for optimizing and controlling deep drawing processes. For instance, the optimization of local blank properties, such as friction (based on the amount of oil for lubrication) and sheet thickness, requires the computation of the final geometry including blank and these local properties. Such investigations require metamodels that represent the underlying deep drawing process appropriately. Typically, enhanced metamodels comprise a large number of parameters, and consequently, need a large amount of training data. Data acquisition in manufacturing in general but specifically in sheet metal forming is time consuming and costly before start of production. Hence, a common approach to train metamodels is to generate data with FE simulation models. However, data generation by means of numerical simulations can as well be costly and time consuming when dealing with complex material and contact models or
large parts with a great number of elements. It is desirable to keep the amount of FE simulations to a minimum. Therefore, it is necessary to develop metamodels for limited training data that still result in high predictive power.

Approaches reported in earlier literature can be distinguished into metamodels that predict scalars and metamodels that predict physical fields. For instance, Bonte et al. [2] described a methodology to optimize forming processes by using response surface methods and kriging models. Opposed to that, other authors described methods to predict physical fields. For example, Schwarz et al. used principle component analysis [4] and Fourier-based transformation methods [5] for dimension reduction with subsequent linear regression to predict physical fields of different sheet metal forming processes. Furthermore, Senn et al. [6] used a neuronal network to predict physical fields of a cup drawing process in dependency of the current time step. Moreover, Pfommer et al. [7] proposed a deep multilayer perceptron as a metamodel for process optimization in manufacturing. Whereby, the network was applied to predict shear angle fields of a composite textile draping part as a function of 50 process parameters based on 584 training samples. However, beside these previously used dimension reduction techniques and deep multilayer perceptrons, recent publications suggest to apply CNNs as metamodels of physical processes. The capability to consider and recognize spatial information is especially promising in focusing on physical processes.

In order to predict results from FE simulations, Zimmerling et al. [8] proposed to represent all results as images in a Cartesian grid so CNNs with convolution and transposed convolution layers could be used for metamodeling. Thereby, the metamodel predicts the resulting shear angle fields of a composite textile draping process depending on different geometries to be formed using 10,000 training samples. Furthermore, Jiang et al. [9] suggested that cGAN is able to predict a stress field of a two dimensional structure in dependence of loads and boundary conditions based on FE simulation data 38,400 FE simulation samples. Attar et al. [10] applied CNNs with U-Net architecture developed for image segmentation (see [11]) to separately predict displacement and thinning fields of a hot sheet metal forming process. Thereby, to develop fast metamodels during component design, critical radii and drawing depth of a square cup tool geometry were varied. Results show good predictive precision of the metamodels trained with 1.800 samples. In line with that, Zhou et al. [12] also applied CNNs with U-Net architecture to predict FE simulation results of a sheet metal forming process. Thereby, besides varying radii of the square cup and varying binder force additionally, sheet thickness has been varied to generate 1.080 samples.

Existing literature shows that applying CNN based approaches leads to good performance in both precision and generalization of the trained metamodels. Thereby, two metamodels were trained separately to predict displacement and thinning fields. Nevertheless, the amount of data used for training differs in literature. Previous work demonstrated that cGANs achieve high performance in computer vision applications [13], in modelling aerodynamic data [14] as well as in predicting stress fields of FE simulations with high accuracy [9]. Therefore, in this study one cGAN architecture is used to train a metamodel that predicts both displacement and thinning fields of FE results as a function of process parameters of a sheet metal forming process. Furthermore, the effect of amount of data used to train the metamodels on predictive power will be investigated.

2. Method

cGANs are commonly used for tasks in computer vision, for instance image generation or image to image translation [13]. In comparison to conventional GANs [15], a condition is added to the training process to specify the picture generated by the GAN. Using cGAN architectures to predict results of FE simulations in sheet metal forming requires to represent results as images (see [8–10, 12]). Hence, results have to be structured in a Cartesian grid with a same number of pixels for each result.
2.1. Application example
The methodology was explored on the example of a sheet metal forming process with a cross cup geometry. Since the cross cup is double symmetric, only one quarter of the geometry is modeled in FE simulation, see Figure 1.

![Figure 1](image.png)

**Figure 1.** Form and dimensions of double symmetric (a) blank and punch of cross cup, (b) tool, and (c) exemplary deep drawn part.

2.2. FE simulation model
LS-DYNA explicit solver was used for numerical simulations of the deep drawing process. The blank was modeled with fully integrated Belytschko-Tsay shells. The tool was modeled with rigid shell elements. The blank mesh contains 4,844 elements, the die was meshed with 5,586 elements, the punch with 5,920 elements and the blank holder with 2,870 elements. The material used in this study was standard deep drawing steel DC01. The stress-strain curve $\sigma(\bar{\varepsilon}_p)$ in Figure 2 was used for modeling the material behavior during forming with material type 24 in LS-DYNA. Isotropic material behavior was assumed. Coloumb’s law of friction with friction coefficient $\mu$ was applied for the contact between blank and tool.

![Figure 2](image.png)

**Figure 2.** Stress-strain behaviour $\sigma(\bar{\varepsilon}_p)$ of DC01.

2.3. Data acquisition and preparation
The input space covers five parameters. Beside the process parameters drawing depth $h$ and blankholder force $F_{bh}$, boundary conditions were taken into account by varying the friction coefficients $\mu$ and different material properties were considered by varying the initial sheet thickness $s_0$ and scaling the stress-strain-curve with a factor $f_s$.
\[
\delta(\varepsilon_p) = f_s \sigma(\varepsilon_p)
\]  

To investigate the predictive power of the metamodel, large parameter ranges were used to ensure high variance of displacement and thickness fields, see Table 1.

| Parameter       | Initial sheet thickness | Drawing depth | Blankholder force | Friction coefficient | Scale factor |
|-----------------|-------------------------|---------------|-------------------|----------------------|--------------|
| Symbol          | s₀                      | h             | Fₘₜₜ              | \(\mu\)              | \(f_s\)      |
| Unit            | mm                      | mm            | kN                | -                    | -            |
| Minimum         | 1.2                     | 45            | 1                 | 0.05                 | 0.8          |
| Maximum         | 1.8                     | 80            | 360               | 0.20                 | 1.2          |

To generate data for training the metamodels, a Latin hypercube sampling (LHS) was used. A total number of 10,000 samplings were created. As a first step of preprocessing data, 149 samples were excluded because of outliers and error terminations in FE simulation. 9,851 were stored in the database for training. In a second step, element results of the FE simulations were averaged to the corresponding nodes. Since FE simulations in solid mechanics mostly result in unstructured grids, an interpolation to a uniform Cartesian grid is required to use an image based cGAN. The results considered in this study contain node coordinates \(x = (x,y,z)\), node displacement \(u = (u_x,u_y,u_z)\) and averaged element thickness \(s\) at each node of the blank and the parts of the tool, respectively. This information is available for the configuration before and after forming. Linear interpolation in SciPy [16] was used to interpolate the results \(u\) and \(s\) into the Cartesian grid with \(256 \times 256\) pixels. This results in a pixel size of 0.625 mm. Areas on the Cartesian grid without blank or tool geometry were set to -1 and channels were normalized between 0 and 1, see Figure 3.

2.4. Metamodel

In general, GANs consist of a generator network and a discriminator network [15]. During GAN training, these networks are trained based on a minimax game, in which the generator learns to generate images as similar as possible compared to the training data. The discriminator learns to distinguish between real and fake images. Thereby, both networks are trained alternately until Nash equilibrium is reached.

Isola et al. [17] proposed a general solution for image to image translation problems with cGANs. It is based on a U-Net (see [11]) for the generator and a patch wise discriminator. Thereby, only patches of a full generated image are evaluated to be real or fake. The architecture of the generator is extended by adding residual squeeze and excitation (RES-SE) blocks in the bottleneck of the U-Net (see [18, 19]), because this architecture for the generator achieved good results when working with FE simulation data [10, 12]. Furthermore, SE blocks [19] are used to recalibrate the channels of the skip connections and in the decoder of the generator to improve generalization capability [20], see Figure 3. The reduction ratio for all SE blocks was set to 8. In this study the discriminator architecture is based on [17]. The input pixel grid consists of the target values with four channels (256,256,4) and the conditional information with three channels (256,256,3). The scalar values are concatenated after layer 5, see Figure 3.

Each convolution operation uses a kernel with size of 4, and a stride of 2. Padding was set to 1. Convolution to layer 5 and 6 of the discriminator uses stride of 1 and padding of 0. Kernel size of convolution to layer 6 of the discriminator was set to 1. Additionally, after each convolution and transposed convolution operation ReLU was chosen as activation function for the generator. LeakyReLU with a negative slope of 0.2 for values \(< 0\) was used for the discriminator, respectively. Except for the last layers of generator and discriminator, there was hyperbolic tangent (tanh) activation function set for generator and no activation function for discriminator, respectively. To provide noise
into the training of the generator, dropout with a probability of 0.5 for an element to be zero was applied after transposed convolution to layer 7.

Figure 3. Inputs, outputs and architecture of the discriminator and generator of the cGAN.

The inputs for the metamodel are represented in both an image and in a scalar manner. The initial sheet thickness $s_0$, drawing depth $h$ and blankholder force $F_{bh}$ are represented as images in the top view of the blank, punch and blankholder, respectively. This results in a pixel grid with three channels $(256,256,3)$. Friction coefficient $\mu$ and the scale factor $f_s$ of the stress-strain-curve were added to the feature map, see Figure 3. Thereby, each channel contains $16^2$ equal values of $\mu$ and $f_s$ for the generator and $29^2$ equal values for the discriminator, respectively. Targets and outputs are represented as an image in form of a pixel grid $(256,256,4)$ with four channels $(u_x, u_y, u_z, s)$. The cGAN is implemented using PyTorch [21]. In order to train the cGAN Adam algorithm was used with a learning rate of 0.00025 and a batch size of 128. The loss function from [17] for cGAN was used to alternately train the generator and the discriminator.

3. Results and discussion
The cGAN was trained with a different number of training samples and corresponding epochs to achieve the same number of training steps according to Table 2. Calculation was executed on one GPU (NVIDIA Quadro RTX 5000) (approx. 4.5h for each metamodel).
Table 2. Number of samples used for training of cGAN.

| Samples   | 125  | 500  | 2,000 | 8,000 |
|-----------|------|------|-------|-------|
| Epochs    | 16,000 | 4,000 | 1,000 | 250   |
| Best epoch| 14,385 | 3,971 | 349   | 186   |

The best metamodel was chosen based on generator loss to generate the results shown in Figure 4. Figure 4 depicts randomly picked predictions for three examples: standard part without failure, part with high thinning, and part with wrinkles. Thereby, high thinning and wrinkling are represented in the dataset as a course of the large process parameters chosen in for the sampling. Predictions show that the networks could generally be able to predict deep drawing results for standard parts without any failure as well as for parts with high thinning or wrinkling.

Figure 4. Three exemplary predictions from the test set (deformation and thickness) of the four trained metamodels and the corresponding ground truth.
Furthermore, the network achieved good predictions for standard parts and parts with high thinning, even with only 125 samples for training. Moreover, the draw-in is also mainly predicted correctly in examples shown in Figure 4 and Figure 5, even for small data sets. Opposed to that, the prediction of wrinkling only tended to be acceptable when using 8,000 samples for training, see Figure 4 and Figure 5 (a) and (c). Similar to that, 8,000 samples were needed for an acceptable prediction of the thickness field of the part with wrinkles.

In order to show the performance of the cGANs, Figure 5 depicts additional exemplary predictions with high and low predictive power of parts were high thinning and wrinkling occurred. The high predictive power results are mostly independent of the data used for training, except for predicting wrinkles. In contrast, the results with low predictive power show dependency on data used for training and blurriness.

![Figure 5. Exemplary predictions with high and low predictive power.](image)

In general, most predictions were of high predictive power, as shown in Figure 5 (a) and (b). Nevertheless, some predictions were only of lower predictive power that can be observed in Figure 5 (c) and (d) showing deviations between predictions and ground truth. However, the overall high predictive power of the models can be seen in Table 3. Table 3 shows the average over all 1,951 test samples of the mean absolute error (MAE) between all pixels of the ground truth and the predictions.
from the metamodels. This shows, that with increasing number of training samples the deviation can be reduced. The largest effects are shown for $u_y$ and $u_z$. Moreover, the values for $z$ displacement are comparably high. In comparison to that, the average MAE of the sheet thickness is small and does not significantly decrease with increasing number of samples.

**Table 3. Average MAE over all pixels for all four trained metamodels on test set with 1,951 samples.**

| Samples | 125  | 500  | 2,000 | 8,000 |
|---------|------|------|-------|-------|
| Average MAE $u_x$ [mm] | 0.482 | 0.391 | 0.406 | 0.364 |
| Average MAE $u_y$ [mm] | 0.727 | 0.584 | 0.578 | 0.504 |
| Average MAE $u_z$ [mm] | 0.870 | 0.724 | 0.688 | 0.649 |
| Average MAE $s$ [mm]   | 0.027 | 0.024 | 0.024 | 0.022 |

The predictions as well as the performance indices of all four networks show overall good results. The comparably high average MAEs for all four channels may result from the large range of the process parameters and from the consequent samples with high thinning as seen in Figure 5 (d). Furthermore, it is also possible that there are some parameter combinations of the test dataset that caused large deviations between predictions and the ground truth, for instance the predictions from the cGAN trained with 500 and 2,000 samples in Figure 5 (d). Since the accuracy of predictions always depends on the application, the suggested approach is suitable to predict FE results of a sheet metal forming process with cGAN training and the suggested architecture in general. The predictions of the metamodels in dependency of the amount of training data show that even with very small datasets a good quality prediction can be made for parts without failures. In this context, it is questionable how robust the predictions of the metamodels are against a test dataset containing more than 1,951 samples, as neuronal networks can respond very sensitively. The metamodels based on small datasets are only able to predict tendencies of occurring wrinkling. To address these effects, further investigations on different architectures of both generator and discriminator have to be conducted. Especially, the complexity of the generator in comparison with the discriminator is of high importance to ensure maximum benefit of the adversarial training.

**4. Conclusion**

The investigations prove that cGANs are capable of predicting deep drawing results. The proposed architecture achieves good results for calculating sheet draw-in and thinning based on a small amount of training data. The authors show that the prediction quality of wrinkling depends much stronger on the amount of available data for training than does the forecasting of thinning and draw-in. If wrinkles are to be accurately mapped, more data is required than if only the tendency towards wrinkling is to be predicted. Future research will include adapting the architecture of the generator and discriminator. In addition, a more detailed investigation of the generated metamodels will take place.

**Acknowledgements**

We gratefully acknowledge the funding provided by the Fraunhofer Society as part of the lighthouse project *Machine Learning for Production (ML4P)*.

**References**

[1] Weichert D, Link P, Stoll A, Rüping S, Ihlenfeldt S and Wrobel S 2019 A review of machine learning for the optimization of production processes *Int J Adv Manuf Technol* **104** 1889–902

[2] Bonte M H A, van den Boogaard A H and Huétink J 2008 An optimisation strategy for industrial metal forming processes *Struct Multidisc Optim* **35** 571–86

[3] Allwood J M, Duncan S R, Cao J, Groche P, Hirt G, Kinsey B, Kuboki T, Liewald M, Sterzing A and Tekkaya A E 2016 Closed-loop control of product properties in metal forming *CIRP Annals* **65** 573–96
[4] Schwarz C, Ackert P and Mauermann R 2018 Principal component analysis and singular value decomposition used for a numerical sensitivity analysis of a complex drawn part Int J Adv Manuf Technol 94 2255–65

[5] Schwarz C, Link P, Ihlenfeldt S and Drossel W-G 2021 Application of Fourier-related data reduction methods in sheet metal forming Procedia CIRP 99 260–5

[6] Senn M, Jöchen K, Van T P, Böhlike T and Link N 2013 In-depth online monitoring of the sheet metal process state derived from multi-scale simulations Int J Adv Manuf Technol 68 2625–36

[7] Pfrommer J, Zimmerling C, Liu J, Kärger L, Henning F and Beyerer J 2018 Optimisation of manufacturing process parameters using deep neural networks as surrogate models Procedia CIRP 72 426–31

[8] Zimmerling C, Trippe D, Fengler B and Kärger L 2019 An approach for rapid prediction of textile draping results for variable composite component geometries using deep neural networks AIP Conference Proceedings 2113 20007

[9] Jiang H, Nie Z, Yeo R, Farimani A B and Kara L B 2021 StressGAN: A Generative Deep Learning Model for Two-Dimensional Stress Distribution Prediction Journal of Applied Mechanics 88

[10] Attar H R, Zhou H and Li N 2021 Deformation and thinning field prediction for HFQ® formed panel components using convolutional neural networks IOP Conf. Ser.: Mater. Sci. Eng. 1157 12079

[11] Ronneberger O, Fischer P and Brox T 2015 U-Net: Convolutional Networks for Biomedical Image Segmentation Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015 (Lecture Notes in Computer Science) ed N Navab et al (Cham: Springer International Publishing) pp 234–41

[12] Zhou H, Xu Q, Nie Z and Li N 2022 A Study on Using Image-Based Machine Learning Methods to Develop Surrogate Models of Stamp Forming Simulations Journal of Manufacturing Science and Engineering 144

[13] Creswell A, White T, Dumoulin V, Arulkumaran K, Sengupta B and Bharath A A 2018 Generative Adversarial Networks: An Overview IEEE Signal Process. Mag. 35 53–65

[14] Hu L, Zhang J, Xiang Y and Wang W 2020 Neural Networks-Based Aerodynamic Data Modeling: A Comprehensive Review IEEE Access 8 90805–23

[15] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y 2014 Generative Adversarial Nets Advances in Neural Information Processing Systems ed Z Ghahramani et al (Curran Associates, Inc)

[16] Virtanen P, Gommers R, Haberland M and Hilboll A 2020 SciPy 1.0: Fundamental algorithms for scientific computing in Python Nature methods

[17] Isola P, Zhu J-Y, Zhou T and Efros A A 2017 Image-To-Image Translation With Conditional Adversarial Networks Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

[18] He K, Zhang X, Ren S and Sun J 2016 Deep Residual Learning for Image Recognition Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

[19] Hu J, Shen L and Sun G 2018 Squeeze-and-Excitation Networks Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

[20] Rundo L et al 2019 USE-Net: Incorporating Squeeze-and-Excitation blocks into U-Net for prostate zonal segmentation of multi-institutional MRI datasets Neurocomputing 365 31–43

[21] Paszke A et al 2019 PyTorch: An Imperative Style, High-Performance Deep Learning Library Neural Information Processing Systems 32