IV International Conference on Simulation for Additive Manufacturing
Sim-AM 2023
E. Rank and S. Kollmannsberger (Eds)

# TOWARDS NEURONAL-NETWORK-ENHANCED FINITE ELEMENT SIMULATIONS IN ADDITIVE MANUFACTURING

**CHRISTOPH BEHRENS***, **VASILY PLOSHIKHIN***

*Airbus Endowed Chair for Integrative Simulation and Engineering of Materials and Processes
(ISEMP)
Bremer Center for Computational Materials Science
University of Bremen, Am Fallturm 1, 28359 Bremen
e-mail: behrens@isemp.de, www.isemp.de

**Key words:** numerical simulations, additive manufacturing (AM), machine learning, neural network, finite element method

## 1 INTRODUCTION

### 1.1 Additive Manufacturing

Complex, lightweight, and customized metal parts can be produced by additive manufacturing (AM). The base material can be fine metallic powder, as in powder bed fusion – laser beam/ metal (PBF-LB/M), or a wire feedstock, as in wire arc additive manufacturing (WAAM). The latter is an important fabrication process for large-scale components [1]. A wide range of materials can be manufactured [2]. However, residual stress will form during the build-up of metal components [2] and lead to critical defects, such as distortion and cracking [3]. Especially in large-scale components, distortion is a main concern for WAAM [1].

### 1.2 Inherent Strain Simulations

The research field of the prediction of stress and distortion in AM metal components splits into the research that improves the modeling of the simulations and the increase in computational efficiency [4]. One result of the aim for highly efficient AM simulations is the inherent strain method [5], in which the inherent strain method from welding was adapted to the AM process [6].

The inherent strain $\varepsilon^*$ is the resulting strain remaining in the part after the relaxation of a test volume [7]. Therefore, it can be described as

$$\varepsilon^* = \varepsilon - \varepsilon_{el}, \tag{1}$$

where $\varepsilon$ is the total strain and $\varepsilon_{el}$ the elastic strain. The inherent strain $\varepsilon^*$ is inserted layer-wise into the part, and the elastoplastic equilibrium is solved in each layer using the finite element method (FEM) [8].

Values for the inherent strain can be either calculated by calibration based on experimental displacement measurements or small-scale conventional thermomechanical process simulations [5,9].

The computational effort is reduced by transferring the complex thermomechanical problem to a mechanical problem.

Initially developed for PBF-LB/M, the inherent strain simulation was also successfully deployed for WAAM simulations by reactivating previous deposition material within the penetration depth of the melt pool. An adaptive increment-wise insertion of the inherent strain value is used to guarantee convergence. If the current increment does not converge, the inherent strain values of that increment are divided by two and solved in two steps. This process is self-repeating. The described method, material data, and calibrated values are taken from [10], and the results are used to train neural networks to predict the displacement within the next layer.

### 1.3 Newton-Raphson Iterations

The Newton-Raphson method (NR) is used for finding an approximation of the root of a real-valued function $f(x) = 0$. The tangent of the function is calculated in a point $x_i$ in each iteration. The root of the tangent is then used as the next point $x_{i+1}$. When the conditions of the convergence theorem of the NR are satisfied, a test solution being sufficiently close to the searched solution, the convergence rate is quadratic [11,12].

In inherent strain simulations, a new layer is activated without any displacement in the top node layer. The resulting strain is called a reference strain and is inserted in the element [13]. Hence, elements of the new layer activate in a stress-free state, despite having a strain from displacements in the previous active nodes and no displacements in the new nodes. The NR is necessary to solve the elastic-plastic equilibrium when the inherent strain is inserted in the new layer. The displacement field is searched, which solves the equilibrium. The NR starts with an initial displacement field as a test solution, which is the current displacement field from the stress-free activation. By improving the test solution, the iteration in the NR will be reduced.

Most simulations of WAAM specimens will show a self-similarity in the displacement field between each layer. Thus, a potential human observer could predict a better test solution than the displacement field from the stress-free activation after only seeing the displacement fields of a few layers.

The potential human observer uses an intuitive regression with a small training data set. This work substitutes the human observer and uses multivariate regression to estimate the three-dimensional displacement in each node of the FEM by training a neural network.

### 1.4 Neural Networks

Neural networks (NN) can perform the task of classification and regression for even complex problems [14]. In AM, the classification is used for pore detection from process monitoring data [15]. The regression capabilities can be used to predict the distortion of AM parts [16–18]. However, published results were limited in the geometrical variety of the specimens. Data generation and geometrical bias can be complex challenges for applying neural networks in AM.

Another possible application for NN in AM is improving the computational efficiency of process simulations. Not explicitly stated for AM, a NN was used in the material model of FEM simulations [19]. Elastic-plastic problems were solved with a recurrent NN inside a Monte-Carlo-Simulation, overcoming the high computational cost of such [20].

In this work, the NN is used to predict the displacement field of a new layer in an inherent strain simulation. The training data are the solved equilibriums of previous layers.

# 2 METHODS

## 2.1 Newton-Raphson-Procedure

The NR is commonly used when solving the nonlinear problem of finding the elastic-plastic force equilibrium.

For equilibrium, the displacement vector $u$ must be found, balancing internal $f^{int}$ and external forces $f^{ext}$. Thus, the problem can be described as finding the root of the residuum

$$r \coloneqq f^{int}(u_{l+1}) - f^{ext}. \tag{2}$$

The internal force $f^{int}$ depends on displacement. An initial guess is necessary to find the displacement field $u_{l+1}$ of the next layer $l$ in an inherent strain simulation. The initial displacement field is the displacement field of the equilibrium of the previous layer:

$$u_{l+1}^{0} \coloneqq u_l. \tag{3}$$

The updates during each Newton iteration $j$ consist of calculating the tangent stiffness matrix $K_T$, solving the linear equation

$$K_T \delta u^j = -r^{j-1}, \tag{4}$$

and updating the displacement

$$u_{l+1}^{j} = u_{l+1}^{j-1} + \delta u^j. \tag{5}$$

The new displacement results in a new residuum (Eq. (2)), leading to a new iteration of the Newton-Raphson until the residuum is sufficiently small. The procedure is taken from [21].

The convergence theorem of the NR is satisfied if the initial guess $u_{l+1}^{0}$ is sufficiently close to the searched solution $u_{l+1}$ [12,22]. As an initial guess $u_{l+1}^{0} = u_l$ is assumed in inherent strain simulation. If the influence of one layer is small, then $u_{l+1}^{0}$ is close to $u_l$. However, for other additive manufacturing processes, this assumption might fail. In WAAM, a weld track can be placed on the side of a thin wall structure, leading to a significant change in the displacement field.

By improving the initial guess $u_{l+1}^{0}$ convergence is ensured, and the number of iterations of the NR is reduced.

## 2.2 Neural Network

The neural network is built using the library OpenNN [23]. The architecture is a feed-forward neural network consisting of a scaling layer, followed by a perceptron layer of twenty neurons, another one with twenty, a third with nine, a last one with three, and an unscaling layer to the displacement output $u$. Figure 1 shows the architecture. A hyperbolic tangent is used as a gain function for all perceptron layers except the last one using a linear gain function. All perceptron layers have a bias node.

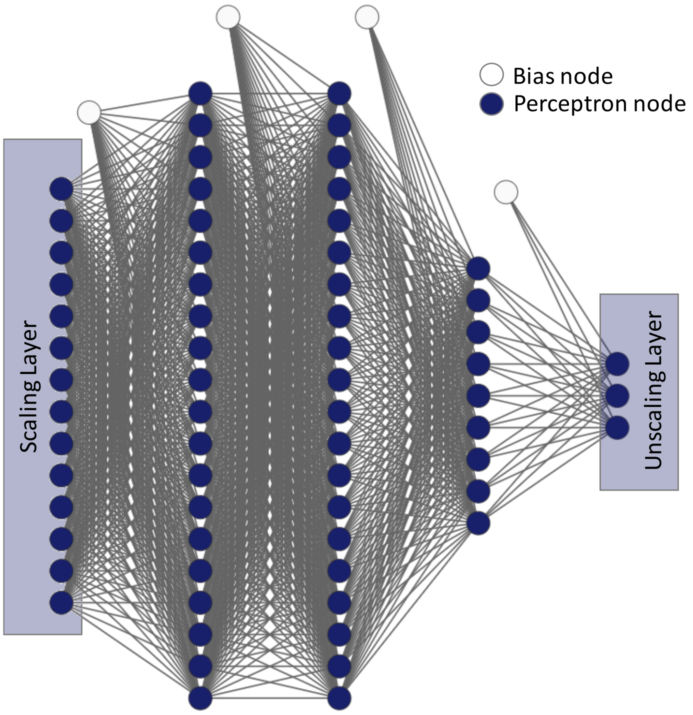

**Figure 1**: Neural Network Architecture.

The input has 17 inputs in total with values on different scales from $10^{-3}$ up to $10^{6}$. Therefore, a scaling layer is necessary. The scaling is done by using the maximum and minimum values of the training data for each input. The scaled input value $\eta$ is given by

$$\eta_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}, \tag{6}$$

when $x_i$ is the input value of a training set $i$. The same method applies to the unscaling layer. However, it might not be necessary because all outputs are displacement values and do not have values in different scales like the input.

## 2.3 Training and Validation Set

The training method is an adaptive momentum estimation optimization reducing the $L2$-norm [24]. Training data is taken from the elastic-plastic equilibriums from an inherent strain simulation calculated using the FEM. Each node provides one training set with the following input: the displacement from the last layer $u_i^{l-1}; i \in \{x, y, z\}$, the local stiffness matrix of the node with itself $K_{nn}^{l}$, the von-Mises-stress $\sigma_{vM}^{l-1}$, a distance $d_m^l$ to the mass point of the currently activated layer, a distance to the closest Dirichlet boundary condition $d_{bp}^l$, the components of the inherent strain tensor $\varepsilon_{xx}^*$, $\varepsilon_{yy}^*$, $\varepsilon_{zz}^*$ for the current layer $l$, the six components (Voigt notation) of the product of reference strain and inherent strain, called sum strain $\varepsilon_{sum}^{l} = \varepsilon^{*l} + \varepsilon_{ref}^{l}$, and a boolean if the node is part of the current layer $l$.

The output of the neural network is the three-dimensional displacement $u$: $u_x^l$, $u_y^l$, and $u_z^l$.

The specimen's complete inherent strain simulations were done according to [10] to generate the training and validation data sets. The training and validation are performed after the entire simulation is completed. The input of the training data is taken from the last layer $l-1$, and from the current layer $l$ the information that does not require solving the elastic-plastic equilibrium. The output of the training data is the displacement field $u_i^l$ of the current layer. The validation data uses the information of the current layer $l$ and the next layer $l+1$. Thus, the ability of the trained network is tested to predict the displacement field of the activation of the new layer $l+1$.

## 2.4 Geometries

In this work, two very different geometries were chosen. The first is a cylindric specimen (Figure 2a). All sides of the substrate were clamped to inhibit deformation in the substrate. Displacement is mainly present in the printed geometry itself.

The second specimen is a 13-layer high wall on a substrate clamped one-sided (Figure 2b). This specimen has an entirely different displacement behavior compared to the cylindric specimen. Primarily, the substrate will deform instead of the additively manufactured part. The similarity between each layer is loosely given, but the displacement will increase with each subsequent layer in the substrate.

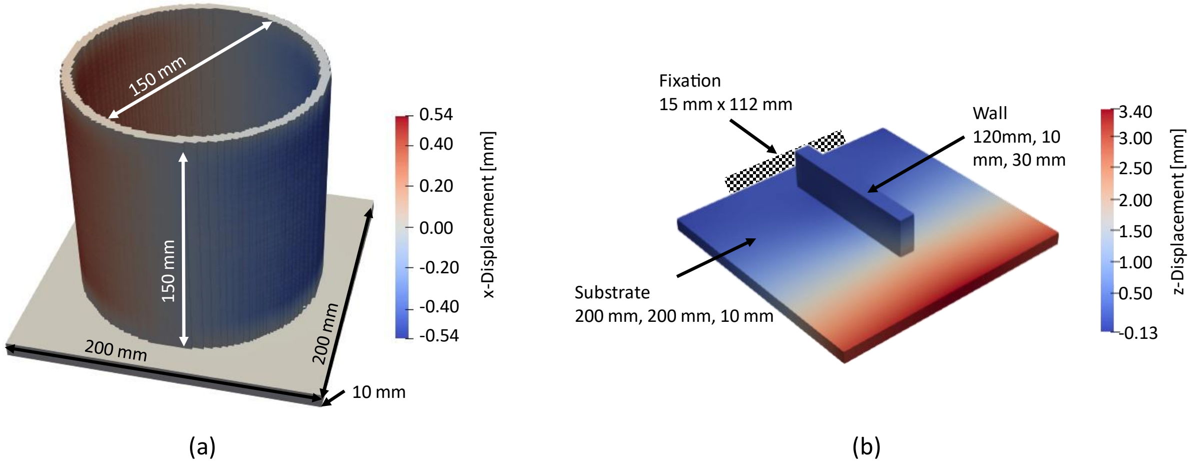

**Figure 2**: Displacement calculated by the Inherent strain simulation [10] for a cylinder specimen (a) with Fixation condition beneath the substrate and a wall (b) on substrate clamped on one side (Fixation).

# 3 RESULTS & DISCUSSION

The number of iterations in the NR is shown in Figure 3. Each point corresponds to one NR iteration, which is not equivalent to the number of calculated layers. The adaptive insertion of inherent strains can lead to multiple NR in a single layer. The total number of iterations necessary to compute the displacement field of the cylindric geometry (Figure 2a) was 631. This number leads to an average of 4.6 iterations per layer. Therefore, the calculation time can be reduced if the goal of the NN-based displacement prediction, reducing the NR iterations to one, is achieved.

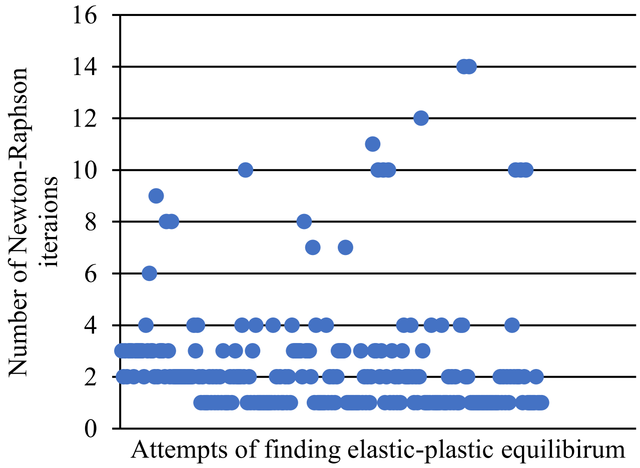

**Figure 3:** Needed Newton-Raphson iterations to calculate the elastic-plastic equilibrium in an inherent strain simulation of a wire arc additively manufactured cylinder.

The training of the network shown in Figure 1 was done using the adaptive momentum estimation optimization [24]. The Training Error over each epoch (complete learning cycle for all FEM nodes) is given in Figure 4 for three layers. Layers 20 and 30 have a minimum in which the training does not converge. An "early-stop"-method [25] could be utilized to optimize the NN. Overfitting could potentially be happening, depending on the definition. The NN needs to predict the displacement in the next layer and is not designed to give general displacement predictions (for now). Thus, overfitting is not a concern, and for easy implementation, training was done for up to 10000 epochs.

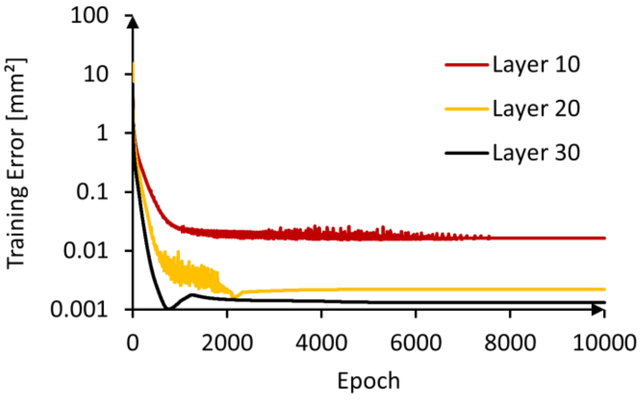

**Figure 4**: Training error or L2 error over the epochs of the training of a neural network for predicting the node displacement in layers 10, 20, and 30.

The training errors of layer 10 and layer 20 have some noise before converging, and the error of layer 30 has not. Additionally, the limit of the training error reduces and is reached earlier with increasing layer number. Both effects are a result of an increased training set size. More layers lead to more FEM nodes and larger sample sizes for each layer. Thus, the training speed does not increase, which could be concluded from the minima reached for fewer epochs with increasing layer number. Instead, the training time increases drastically. For that reason, only the first 40 layers were used.

It seems sufficient for a NN to have the sum strain as input data, which is the entity distorting the elastic-plastic equilibrium. The input nodes for the inherent strain, which are part of the sum strain, were dropped during training. Therefore, it is to assume that only data directly impacting the linear equation of the NR (Eq. (4)) should be used as input. The plastic strain will be investigated in future architectures.

The predicted displacement field is mapped to a three-dimensional structure using a Delaunay filter in the software Paraview (Figure 5a). In comparison, the FEM result for the same layer is shown in Figure 5b. The transition from the substrate to the printed structure in Figure 5a is an artifact from the filter. Visually, there is little to no difference between the predicted results and the results solving the elastic-plastic equilibrium using the FEM.

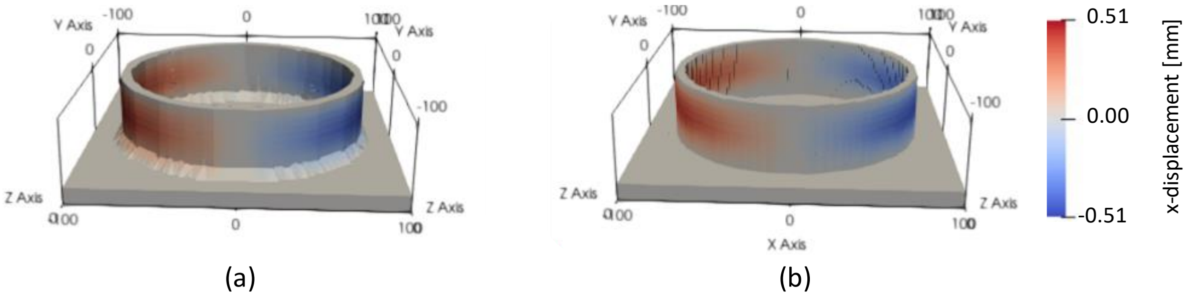

**Figure 5**: Comparison of a neural network prediction (a) and an elastic-plastic finite element simulation (b). The x-displacement is displayed on the color map. The node data of the neural network was transformed into a surface plot by a Delaunay filter.

Figure 6 displays the predicted displacement values over the target values of the FEM for a single layer. The values of the NN prediction are shown. Figure 6a shows these values for all nodes and Figure 6b only for the nodes of the new layer. For an optimal prediction, the points must align with a gradient of 1. The results of the NN follow this line with a slight deviation. For comparison, the standard method is plotted as well, where the displacement field of the last layer is used as a test displacement for the current layer. The prediction of the NN is overall better than the standard method, especially for the FEM nodes in the new layer. The linear regression slopes for the first 40 layers are shown in Figure 7a/b. The average slope over all nodes is $1.00 \pm 0.05$ for the NN prediction and $0.88 \pm 0.08$ for the standard method. A strong potential of the NN to be used to predict the test solution in the NR is indicated by the average slope values.

Significant improvements are primarily located in the new layer FEM nodes, where the slope was increased from $0.75 \pm 0.06$ to $0.97 \pm 0.08$. The prediction of the NN can even lead to errors when applied to all nodes, which can be seen in Figure 6a. The displacement values smaller than $-0.5$ mm practically do not change, whereby the NN predicts small changes. For each FEM node, the effect of subsequent layers decreases until new layers have nearly no influence on the displacement field in that node. The NN prediction, as imperfect as a NN regression is, will slightly alter the displacement. Therefore, the NN should only be used to predict displacements close to the current layer to avoid such effects or introduce a cut-off value when the new displacement is set to zero.

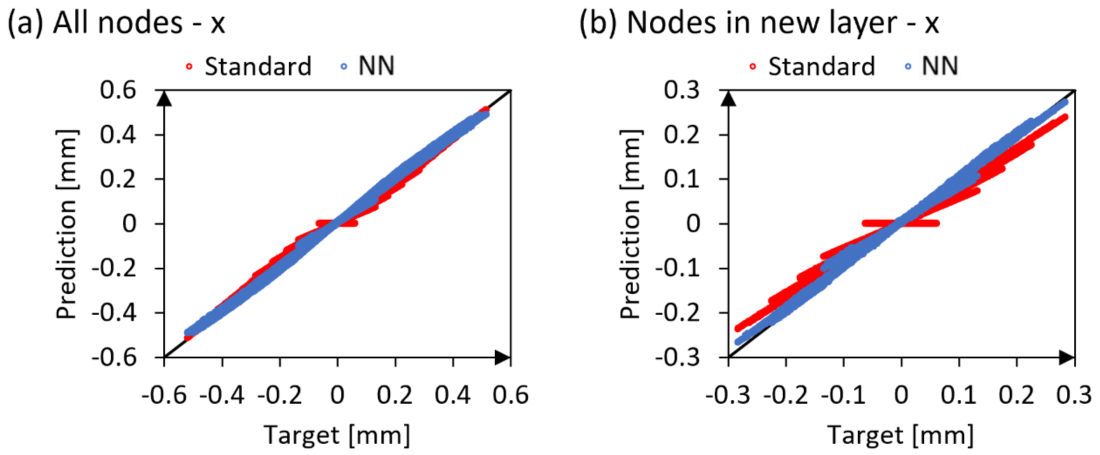

**Figure 6**: Comparison of test solutions of the elastic-plastic Newton-Raphson iteration in an inherent strain simulation for wire arc additive manufacturing of a cylinder specimen. Standard method is the displacement field of the last layer (red), and NN is a prediction for the displacement based on a neural network. (a) shows the results for the x -displacement for all nodes and (b) for the nodes in the new layer.

The simulation results of the second specimen have the most considerable displacement in the substrate material instead of the new layer. The cylindric specimen has a self-similarity of the displacement field between each layer. In contrast, the displacement in the wall with a one-sided clamped substrate will increase over all 13 layers. The slopes of linear regression for this specimen are shown in Figure 7c/d. The average slope is still around 1 with $1.01 \pm 0.13$, but the high standard deviation is over 10 %. Primarily the first layers seem to scatter around the slope of 1, and no noticeable improvement to the standard method can be seen. However, after the eighth layer, the NN outperforms the standard method. When the NN learns from the last layer what the change in the displacement field of the substrate plate is, it will overestimate the displacement for the new layer. After the eighth layer, the displacement change in the substrate gets smaller, which can also be seen by the slope of the standard method for all nodes getting closer to 1 (Figure 7c). It follows that the prediction in the newly added layer is more critical than the displacement prediction in the substrate material. In conclusion, training on multiple equilibria for different layers or instead of training a new NN updating the existing NN from the last layer should be considered.

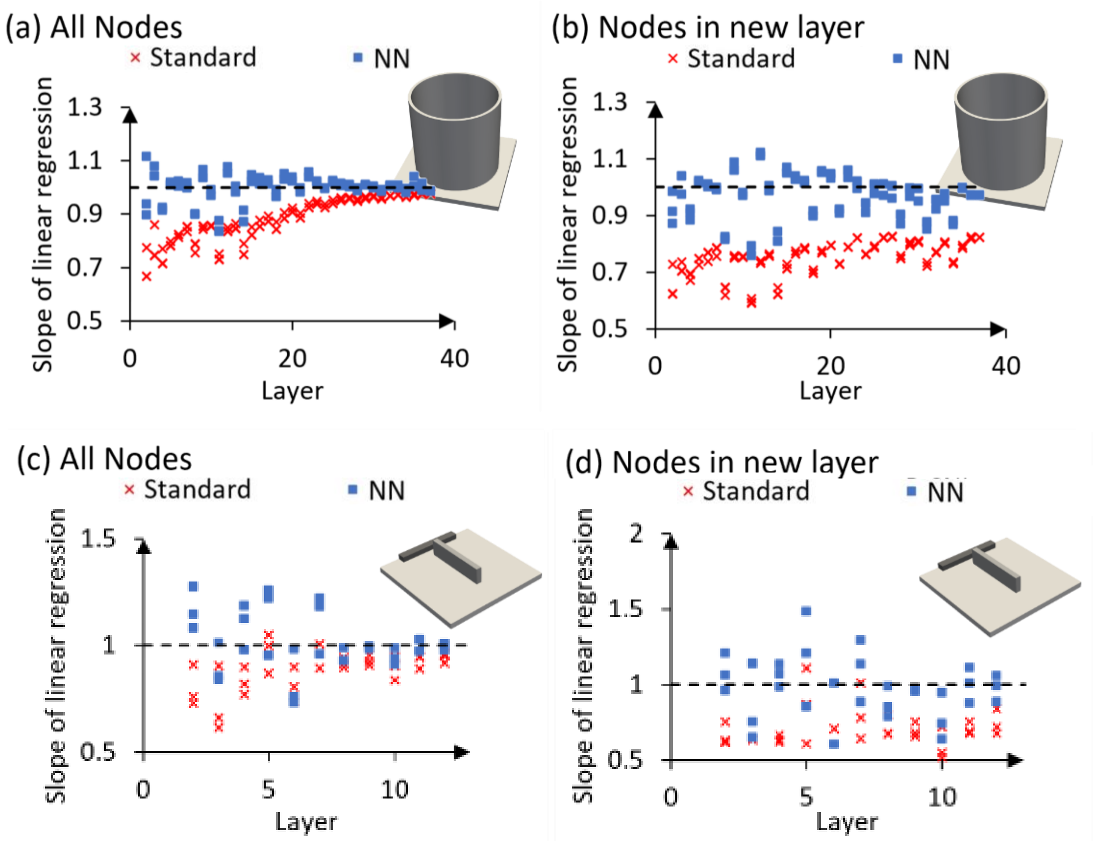

**Figure 7**: Comparison of a neural network-based prediction of the elastic-plastic equilibrium and using the displacement field of the equilibrium of the last layer in an additive manufacturing inherent strain simulation. Slopes of linear regression taken from the test solution over the calculated equilibrium of the next layer for all nodes (a)/(c) and for only new layer nodes (b)/(d). (a)/(b) for a cylindric specimen, and (c)/(d) for a wall on a substrate clamped on one side.

## 4 CONCLUSION & OUTLOOK

In this work, a NN framework has been presented, which can predict the displacement field of the next layer of an additive manufacturing process using the simulated displacement of previous layers. The NN prediction matches the FEM results with high accuracy, especially for the new layer nodes. The work contributes to the field of additive manufacturing simulation by proposing a novel and efficient way of using machine learning. The NN can capture the complex nonlinear relationship between geometry and displacement.

However, this work also shows some limitations. Only two thin-walled specimens were considered for testing the NN. It is unclear how well the model can handle more complex structures with more bulk material. Secondly, only one NN architecture and training method was investigated. Possibly, other architectures and training algorithms can lead to better results.

Future work will focus on implementing the NN prediction in inherent strain simulations to improve the calculation time. Different training methods must be investigated, the training set size has to be limited, and the trained NN should be updated. Else, the potential benefit of reducing NR iteration will be time-wise compensated by training of the NN. Another critical point in the future is a limit test of the model: Can it be used to skip the NR in some layers altogether? Can the NN be designed to work process-wise and not specimen-wise?

The main takeaway message of this work is that NN can be a powerful tool for AM simulation with the potential to reduce computational time in the future significantly.

## 5 REFERENCES

[1] D. Ding, Z. Pan, D. Cuiuri, H. Li, Wire-feed additive manufacturing of metal components: technologies, developments and future interests, The International Journal of Advanced Manufacturing Technology 81 (2015) 465–481. https://doi.org/10.1007/s00170-015-7077-3.

[2] K. Treutler, V. Wesling, The Current State of Research of Wire Arc Additive Manufacturing (WAAM): A Review, Applied Sciences 18 (2021) 8619. https://doi.org/10.3390/app11188619.

[3] B. Wu, A review of the wire arc additive manufacturing of metals: properties, defects and quality improvement, Journal of Manufacturing Processes 35 (2018) 127–139. https://doi.org/10.1016/j.jmapro.2018.08.001.

[4] L.-E. Lindgren, A. Lundbäck, Approaches in computational welding mechanics applied to additive manufacturing: Review and outlook, Comptes Rendus Mécanique 346 (2018) 1033–1042. https://doi.org/10.1016/j.crme.2018.08.004.

[5] N. Keller, V. Ploshikhin, New Method for fast predictions of residual stress and distortion of AM parts, Austin, 2014.

[6] H. Murakawa, Y. Luo, Y. Ueda, Prediction of welding deformation and residual stress by elastic FEM based on inherent strain, Journal of the society of Naval Architects of Japan 1996 (1996) 739–751.

[7] X. Liang, Q. Chen, L. Cheng, D. Hayduke, A.C. To, Modified inherent strain method for efficient prediction of residual deformation in direct metal laser sintered components, Computational Mechanics 64 (2019) 1719–1733. https://doi.org/10.1007/s00466-019-01748-6.

[8] D. Braess, Finite Elemente: Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie, Springer-Verlag, 2013.

[9] X. Liang, L. Cheng, Q. Chen, Q. Yang, A.C. To, A modified method for estimating inherent strains from detailed process simulation for fast residual distortion prediction of single-walled structures fabricated by directed energy deposition, Additive Manufacturing 23 (2018) 471–486.

[10] C. Behrens, S. Neubert, M. Siewert, M.S. Mohebbi, V. Ploshikhin, Integration of annealing into the inherent strain simulation of wire arc additive manufacturing, Additive Manufacturing Letters 4 (2023) 100115. https://doi.org/10.1016/j.addlet.2022.100115.

[11] A. Antoniou, W.-S. Lu (Eds.), Practical Optimization: Algorithms and Engineering Applications, Springer US, Boston, MA, 2007.

[12] E. Süli, D.F. Mayers, An introduction to numerical analysis, Cambridge University Press, 2003.

[13] Q. Lu, E. Beauchesne, T. Liszka, Enhancements to the inherent strain method for Additive Manufacturing analysis, International Journal for Multiscale Computational Engineering 17 (2019).

[14] A.J. Izenman, Modern multivariate statistical techniques, Springer, 2008.

[15] M. Valizadeh, S.J. Wolff, Convolutional Neural Network applications in additive manufacturing: A review, Advances in Industrial and Manufacturing Engineering 4 (2022) 100072. https://doi.org/10.1016/j.aime.2022.100072.

[16] C. Wacker, M. Köhler, M. David, F. Aschersleben, F. Gabriel, J. Hensel, K. Dilger, K. Dröder, Geometry and distortion prediction of multiple layers for wire arc additive manufacturing with artificial neural networks, Applied Sciences 11 (2021) 4694.

[17] Z. Zhu, K. Ferreira, N. Anwer, L. Mathieu, K. Guo, L. Qiao, Convolutional Neural Network for geometric deviation prediction in Additive Manufacturing, Procedia CIRP 91 (2020) 534–539. https://doi.org/10.1016/j.procir.2020.03.108.

[18] O. Aljarrah, J. Li, A. Heryudono, W. Huang, J. Bi, Predicting part distortion field in additive manufacturing: a data-driven framework, Journal of Intelligent Manufacturing 34 (2023) 1975–1993.

[19] Y.M.A. Hashash, S. Jung, J. Ghaboussi, Numerical implementation of a neural network based material model in finite element analysis, Int. J. Numer. Meth. Engng. 59 (2004) 989–1005. https://doi.org/10.1002/nme.905.

[20] A. Koeppe, F. Bamer, B. Markert, An efficient Monte Carlo strategy for elasto-plastic structures based on recurrent neural networks, Acta Mechanica 230 (2019) 3279–3293. https://doi.org/10.1007/s00707-019-02436-5.

[21] E.A. de Souza Neto, D. Peric, D.R.J. Owen, Computational methods for plasticity: theory and applications, John Wiley & Sons, 2011.

[22] Basic Multidimensional Gradient Methods, in: A. Antoniou, W.-S. Lu (Eds.), Practical Optimization: Algorithms and Engineering Applications, Springer US, Boston, MA, 2007, pp. 119–144.

[23] P. Martin, C. Barranquero, J. Sanchez, G. Gestoso, J. Andres, D. Refoyo, C. Garcia, A. Fernadez, R. Lopez, OpenNN: Open Neural Network Library, 2021.

[24] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[25] L. Prechelt, Early Stopping - But When?, in: G.B. Orr, K.-R. Müller (Eds.), Neural Networks: Tricks of the Trade, Springer Berlin Heidelberg, Berlin, Heidelberg, 1998, pp. 55–69.