

# Neural Network Regression Approaches to Reconstruct Properties of Magnetic Configuration from Wendelstein 7-X Modeled Heat Load Patterns

Marko Blatzheim<sup>1, 2</sup>, Daniel Böckenhoff<sup>1</sup> and the Wendelstein 7-X Team<sup>1</sup>

<sup>1</sup>Max Planck Institute for Plasma Physics, Wendelsteinstraße 1, 17491 Greifswald, Germany

<sup>2</sup>Institute for Mathematics, University of Rostock, Ulmenstraße 69, 18057 Rostock, Germany

*marko.blatzheim@ipp.mpg.de, daniel.boeckenhoff@ipp.mpg.de*

## Abstract

Convective heat loads onto the plasma facing components of magnetic confinement devices contain information about edge magnetic field properties which are not yet fully exploited. Machine learning approaches are a promising technique to automatically extract information about such properties from heat load images. In this study, we present the successful reconstruction of proxies for two independent, important edge magnetic field properties given simulated heat load images on the Wendelstein 7-X divertor target plates.

Six different artificial neural network architectures from shallow and simple feed-forward fully-connected neural network to deep Inception ResNets with 24 223 to 804 804 free parameters are investigated. The relative reconstruction error is between 1 and 2% with calculation times on the order of milliseconds. A competing benchmark method without machine learning reaches slightly smaller errors but exceeds the calculation time by three orders of magnitude. The experiments demonstrate that machine learning is also a powerful tool in this particular field of nuclear fusion research and deep convolutional neural networks are identified as favorable algorithms for the stated problem. The findings of this paper build a basis for future real time discharge optimization and control by means of machine learning methods.

## 1 Introduction

In recent time artificial neural networks (NNs) have become a popular tool in data-driven scientific practice. Most NN tasks are classification problems, from the classic machine learning set of the MNIST database [1], to handwritten text recognition [2] and machine translation [3]. The NNs described in this study, on the other hand, solve regression problems. The general form [4] uses feed-forward fully-connected neural networks (FF-FCs), but more recent research on regression also considers convolutional neural networks (CNNs) when dealing with images [5]. Pioneering approaches with fusion relevance investigated the reconstruction of plasma equilibrium parameters [6] and plasma shape parameters [7], both considering tokamaks. More recent studies dealt

with neoclassical transport [8], real-time disruption predictions in tokamaks [9] or modelling turbulent transport [10]. While these NNs are all FF-FCs, the positive impact of CNNs on magnetic topology reconstruction performance is analyzed in [11]. With a new approach to unravel the complex 3D geometry of the plasma facing components (PFCs) monitored by the Infra-Red (IR) cameras onto a 2D picture-like input format [12], it is possible to investigate the performance of various NN architectures that have been optimized for 2D-input and proven good.

Wendelstein 7-X (W7-X) is the most advanced stellarator type nuclear fusion experiment optimized for nested magnetic surfaces with small islands, a small shafranov shift ( $\beta \approx 5\%$ ), small neoclassical transport, magneto hydro dynamics (MHD) stability, fast  $\alpha$  particle confinement and

small bootstrap current  $I_{bs}$  with feasible modular coils [13–16]. W7-X has already accomplished several major goals [17, 18], demonstrating the fusion relevance and high performance of the W7-X magnetic confinement concept.

One major advantage over the Tokamak confinement concept [19] is the steady state capability inherent to stellarators. With respect to technical restraints though, it is yet to be demonstrated that W7-X is capable of arbitrarily long operations at fusion-relevant plasma parameters (temperatures of the order of  $100 \times 10^6$  K at densities of  $1 \times 10^{20} / \text{m}^3$ ). Approaching such parameters, heat loads to the PFCs occur on the order of the critical heat loads of the components. Usually, experiments are planned such that actuator behaviour is determined in advance. However, going towards a minute and beyond in the discharge duration, a real time control system becomes crucial. Although, as mentioned before, W7-X is optimized for small  $I_{bs}$  as well as small Shafranov shift the impact of both quantities on machine performance and safety must not be neglected.  $I_{bs}$  is the origin of a toroidal current  $I_{tor}$ , establishing itself on time scales of minutes. The Shafranov shift, caused by the Pfirsch-Schlüter current, follows the time evolution of plasma pressure [20]. The rotational transform  $\iota$  and the radial plasma shift  $\Delta R$  are linearly dependent on  $I_{tor}$  and the plasma pressure respectively. Since the PFC heat load pattern is heavily influenced by  $\iota$  and  $\Delta R$ , significant changes of the pattern can occur during a discharge which are potentially critical [12, 21]. Thus, it is crucial to monitor and control the heat load onto the PFCs.

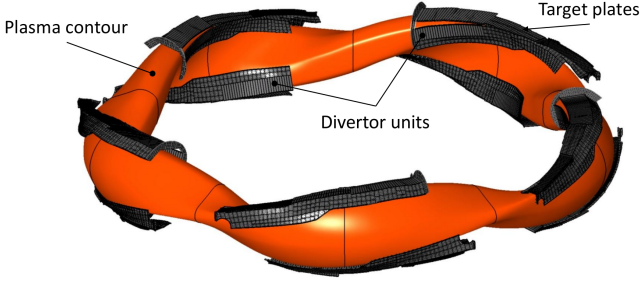
The heat load can be calculated from temperature data recorded by infra-red cameras [22]. Information about equilibrium properties such as  $\iota$  and shift contained within the heat load pattern by means of proxy measures can be extracted by NNs in real-time. The relations between the magnetic edge properties, and the heat load pattern on the PFCs are highly complex. The heat load patterns move in a non-trivial, non-linear way with the underlying parameters and may appear or disappear on certain divertor parts depending on the respective magnetic configuration (see the later introduced Figure 5). This is another reason to apply NNs.

<sup>1</sup>Either way, further direct diagnostics would be additional inputs for such systems. For example  $I_{tor}$  is directly measured by continuous Rogowski coils [25–27].

Feature reconstruction with NNs is relevant for the following reasons: The main purpose of the application of NNs to evaluate the infra-red diagnostic data is to identify heat load patterns in real-time for safety reasons. The applied feature reconstruction can, however, also connect heat load distributions to plasma parameters also known from other diagnostics, e.g. strike-line movements to toroidal current measurements with Rogowski-coils. Thus the applied feature reconstruction NN may be able to combine safety and control issues and help to steer divertor and discharge performance by safely controlling the heat load pattern on the divertor. The approach would be to relate the plasma parameters, which change the heat deposition patterns, i.e. the edge magnetic field topology (mainly the toroidal current and the plasma beta), to the heat deposition patterns in terms of proxies. These proxies could then be provided to classical control systems or even to a trained reinforcement learning [23, 24] control system <sup>1</sup>. In order to determine how successful a NN-based reinforcement agent can handle W7-X divertor heat load images, a first step would be to show the capability of NNs to reconstruct the aforementioned proxies for edge magnetic field properties. At this point we aim at testing and identifying which NN-setup is most suitable to our task in terms of its capabilities and its performance. We are therefore not using MHD-equilibrium fields but use the simpler approach of using certain variations of the vacuum fields allowed by the coil system of W7-X which can be somehow related to the previously mentioned plasma parameters. As said, the details of the relation is of no concern in the context of the present work.

In preceding work [11, 28], it has been demonstrated that the one-dimensional reconstruction of a proxy  $\tilde{\iota}$  for the rotational transform from heat load patterns is feasible. These experimental data and according simulations are heat load patterns on the graphite limiters which were the main PFC during the first operational phase. Since 2017 ten island divertors (see Figure 1) are installed, spatially separating the core plasma from the plasma edge and thus making W7-X suitable for high performance discharges [29–31]. For geometri-

cal reasons<sup>2</sup>, the limiter heat load patterns are much less diverse as compared to the heat load distribution on the divertor.



**Figure 1:** Ten divertor units surrounding the last closed flux surface (orange)

A synthetic data set based on a scan in the proxies  $\Delta\tilde{R}$  and  $\tilde{\iota}$  of the “plasma shift” and “rotational transform” respectively was introduced in [12]. In Section 2, we will give a brief overview of the training data, followed by an introduction of various NN architectures for the reconstruction of  $\tilde{\iota}$  and  $\Delta\tilde{R}$ . The results in Section 3 focus on the NN regression capability, differences for  $\Delta\tilde{R}$  and  $\tilde{\iota}$  reconstruction and dependencies on the region within the  $\tilde{\iota}$  -  $\Delta\tilde{R}$  field for regression.

## 2 Methods

The recent W7-X operational phases provided experimental time for multiple proposals. Only a few of them focused on exploring the configuration space in detail, so for initial NN analysis with the divertor geometry, simulations are necessary to provide sufficient training data. Simulations allow to focus on the two most dynamic and thus most important parameters, the radial shift  $\Delta\tilde{R}$  and the edge rotational transform  $\tilde{\iota}$ . As mentioned before, a Shafranov shift  $\Delta R_S$ , induced by the Pfirsch-Schlüter current, corresponds to a radial shift and an establishing toroidal current  $I_{\text{tor}}$  influences the rotational transform. The two parameters  $\Delta\tilde{R}$  and  $\tilde{\iota}$  are proxies for the real radial shift and edge rotational transform assuming a superposition principle, i. e.  $\Delta R$  and  $\iota$  are linear

combinations

$$\begin{aligned}\Delta R &= c_1 \cdot \Delta\tilde{R}(I_A, I_B) + c_2 \cdot \Delta R_S \\ \iota &= c_3 \cdot \tilde{\iota}(I_A, I_B) + c_4 \cdot \iota_{I_{\text{tor}}}(I_{\text{tor}}) + \iota_0\end{aligned}$$

with constants  $c_1, c_2, c_3, c_4$  and  $\iota_0$ . In the case of  $\iota$  this superposition principle is supported by theory [32]. The dimensionless proxies are based on a linear transformation with

$$\begin{pmatrix} \Delta\tilde{R} \\ \tilde{\iota} \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 & 1 \\ -1 & -1 \end{bmatrix} \begin{pmatrix} I_A \\ I_B \end{pmatrix},$$

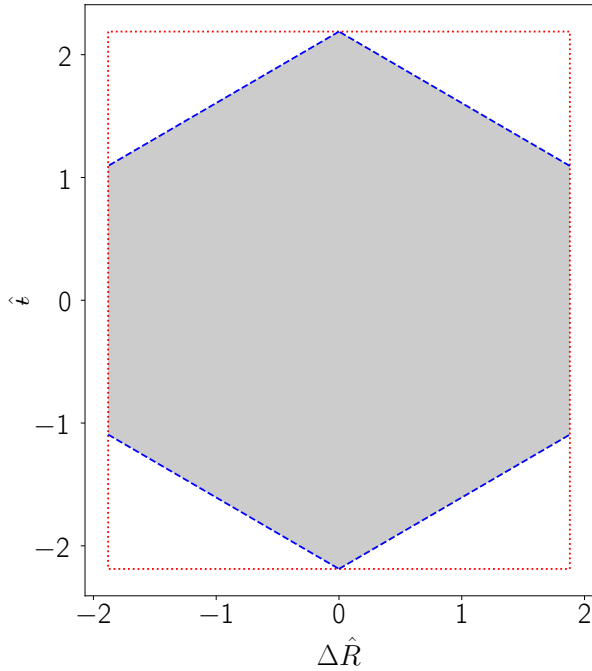
resulting in a rotation of  $\frac{3}{4}\pi$  with the W7-X planar coil currents  $I_A$  and  $I_B$  which are defined relative to the coil current of non-planar coil 1,  $I_{1,\text{true}}$  [12]. The simulation assumes a simplified vacuum configuration and therefore plasma pressure and  $I_{\text{tor}}$  are both zero, which reduces the simulation time by orders of magnitude. The planar coil currents are adjustable parameters of the field line diffusion (FLD) simulation [33] which is employed to compute the heat loads onto the W7-X PFCs by random diffusion processes which start sufficiently close to the plasma core. The non-planar coil currents  $I_{1.5}$  are all set to  $I_{1,\text{true}}$ . With this normalization the actual value of  $I_{1,\text{true}}$  is only a parameter for the magnetic field strength and does not influence the outcome of the simulation. The simulation is performed with a diffusion coefficient  $D_{\perp} = 1 \text{ m}^2 \text{ s}^{-1}$  based on the predecessor experiment Wendelstein 7-AS (W7-AS) [34], an estimated particle velocity of  $v = 1.4 \times 10^5 \text{ m s}^{-1}$  and a mean free path length of  $\lambda = 0.1 \text{ m}$ . A detailed discussion of the choice of these parameters in the FLD simulation can be found in [12]. The local heat loads are formed at the intersection of magnetic fluxes and PFCs as shown in Figure 3(a).

The  $\tilde{\iota}$  value range is twice as high as the  $\Delta\tilde{R}$  value range. Because of the different scaling, normalized radial shift and rotational transform are introduced as

$$\Delta\hat{R} = \frac{\Delta\tilde{R}}{\sigma_{\Delta\tilde{R}}}, \quad \hat{\iota} = \frac{\tilde{\iota}}{\sigma_{\tilde{\iota}}}$$

<sup>2</sup> The heat load pattern shape and relative intensity on the PFCs are in first order the result of the interaction between magnetic field geometry and PFC shape. The poloidally elongated limiter was installed on the inboard side of the center of each module and adjusted to limit the plasma radially. Each of the ten divertors on the other hand spans a toroidal angle of  $47^\circ$  with an approximately 20 times larger surface area as compared to the limiter. The divertors are designed to intersect the large magnetic resonances called magnetic islands which are highly sensitive to changes in the edge rotational transform. Thus, the limiter heat load pattern is a much weaker function of varying plasma properties and magnetic configurations as compared to the divertor heat load pattern (compare limiter heat load pattern variations [28] with Figure 5).

where  $\sigma_{\Delta\hat{R}}$  and  $\sigma_{\hat{t}}$  represent the standard deviation of  $\Delta\hat{R}$  and  $\hat{t}$  respectively. The data points grid is hexagon-shaped as shown in Figure 2 with spacing in the  $\Delta\hat{R}$ -axis between 1 and 121 points and in  $\hat{t}$ -axis between 151 and 301 points, so the data resolution in  $\hat{t}$  is higher.



**Figure 2:** The area enclosed by the red dotted lines represents 36421 grid points, with 121 evenly spaced in  $\Delta\hat{R}$  and 301 evenly spaced in  $\hat{t}$ . The simulated data set is reduced to 27181 points within the grey hexagon. This reflects technical limitations of the coil system, indicated by the blue dashed lines.

A distinction is made between two different pre-processings defined as extracted characteristics input (ECI) and pixel-based input (PBI). The result of a single FLD simulation is a set of points in 3D-space representing the impact location of particles onto terminating PFCs.

**ECI** The simulation generates a 3D hit point cloud of size 25 000 describing the heat load. In order to reduce the input dimensionality for ECI these points are clustered using DBSCAN [35] as shown in Figure 3<sup>3</sup>. Three characteristics are extracted from each of the  $n_c$  clusters, namely the number of points, the center of mass and eigenvectors of the covariance matrix. This leads to the number of extracted characteristics within each

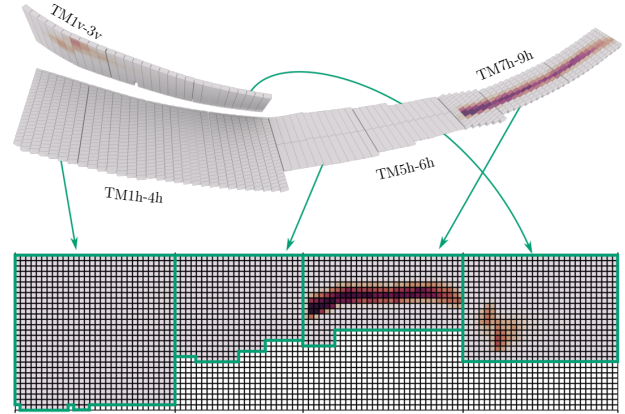
cluster  $m_c = 13$ . In this case the input vector  $u_{\text{ECI}} \in \mathbb{R}^{n'}$  with  $n' = 195$  is initialized with zeros and filled with  $n_c \cdot m_c$  entries, where  $n_c$  is the variable number of clusters found by DBSCAN and realized values are between 2 and 15.

**PBI** An approach to partition the complicated 3D shaped divertor heat flux onto a 2D array like structure, addressing the physics as well as the engineering constraints of the divertor presented in [12], can be utilized to generate the PBI with  $u_{\text{PBI}} \in \mathbb{R}^{m \times n}$ , where  $m = 113$  and  $n = 29$ . The geometric mapping as well as an exemplary PBI for a high iota reference magnetic configuration is depicted in Figure 4 with the heat load given in arbitrary units. In comparison with Figure 3 differences in the heat load patterns seem to appear. However, these discrepancies are artifacts of the misleading impact points representation in Figure 3 as it highly overrates very low point densities. Four exemplary PBI images at different plasma configurations are shown in Figure 5.

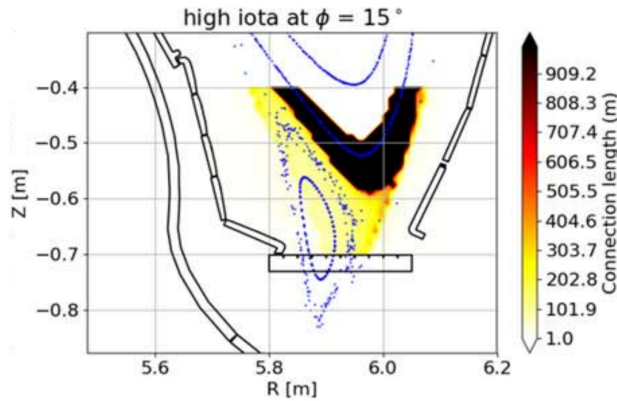
An advantage of ECI is, that features are already separated by the clustering and characteristic extraction. However, there are two important drawbacks. Each array entry within one ECI is of a different quantity (scalar intensity, vector direction), limiting the choice of NNs to FF-FCs. More importantly, an equivalent representation based on experimental data is not possible, so the performance of a trained NN on real data is not measurable and training sets cannot be mixed as shown in [11]. Although this study only considers simulated heat loads, it is of interest, whether the actual experimentally observed heat loads are roughly similar such that the results of this paper (concerning the optimized input and architecture) can be applied to experimental data. A comparison of the simulation with the experimentally found heat loads in the PBI representation is shown in Figure 6. Most importantly, the main heat load patterns are similar and the significant heat load is present on target modules 1h-4h. Deviations from the idealized simulated heat load pattern can be seen for example in the occurrence of increased heat load on target module 7h in the experiment as compared to the simulation. Furthermore the simulation assumes stellarator symmetry and error fields are not considered. However, experimental observations

<sup>3</sup>The two cluster parameters are chosen as  $\text{eps} = 0.065 \text{ m}$  and  $\text{min samples} = 40$

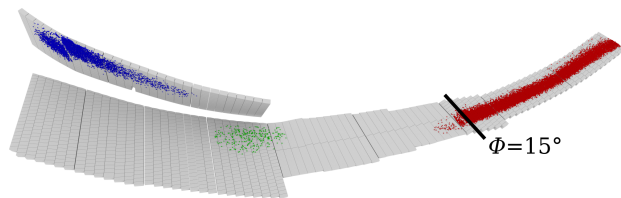
show, that the divertor heat loads are asymmetric (the deviation in the total power reaching upper and lower divertor in module 2 is approximately a factor 2). It is not in the scope of this work, to go into further detail about the differences. Such considerations are covered in much detail in other works [36–38].



**Figure 4:** Heat load pattern in arbitrary units retrieved from stellarator symmetric field line diffusion simulation in the so called high iota reference magnetic configuration on the divertor meshed model (top) and the mapping to the 2D array (bottom) which represents the PBI.

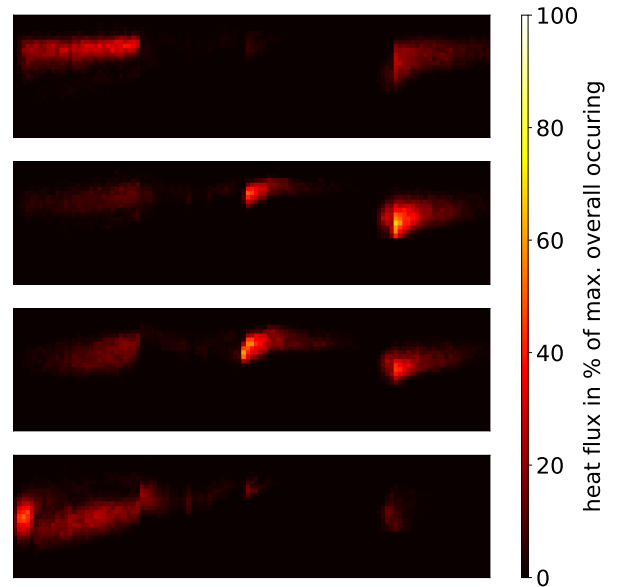


(a) Poincare plot at  $\Phi = 15^\circ$  (indicated in (b)). Adapted from [18]



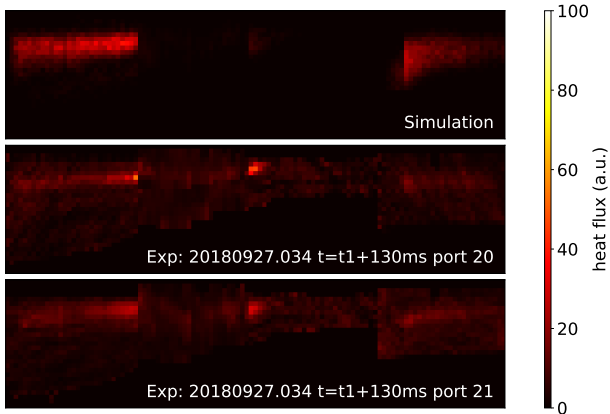
(b) Impact points of a stellarator-symmetric field line diffusion simulation

**Figure 3:** Convective heat exhaust in the high iota reference configuration. The plasma populating the edge island, carrying heat towards the intersecting divertor is shown exemplarily in (a) for a cross section at  $\Phi = 15^\circ$  which is a cut through divertor target TM7h-9h (compare Figure 4). In (b) the divertor with the impact points of a stellarator-symmetric field line diffusion simulation in the same reference configuration is shown. In blue, red and green, three clusters resulting from a DBSCAN clustering are color-coded. Based on these clusters statistical properties are taken as an input for the ECI.



**Figure 5:** An example of four PBI images representing a pair of  $(\Delta\hat{R}, \hat{i})$  with  $(0.0, 0.0)$ ,  $(-1.0, 1.0)$ ,  $(1.0, 1.0)$  and  $(0.0, -1.0)$  from top to bottom respectively. To compare the heat loads, a percentage with respect to the highest heat load within all images is given.





**Figure 6:** Comparison of three PBI images for standard reference case heat loads. From top to bottom: simulated standard case ( $\Delta\hat{R} = 0$  and  $\hat{\epsilon} = 0$ ), experimental measurements of the lower and upper divertor of module 2 for the same frame ( $t = t_1 + 130$  ms after trigger 1) of the same experiment (Program 20180927.034). It should be noted that the measured total power reaching the lower divertor is approximately twice the power as compared to the upper divertor. The experimental images show heat load on the high iota tail (center of the picture) whereas the simulation does not.

A brief introduction to the used NN architectures is given in the following, and a detailed layer-for-layer description for the respective architectures can be found in Appendix A. The FF-FC for both ECI and PBI consists of 3 layers following the input layer with 80, 100 and 2 nodes respectively. The performance with the PBI is analyzed not only on FF-FCs but on CNNs, deep convolutional neural networks (DCNNs), deep inception neural networks (DINNs) and Inception-ResNets (IRNNs). An increased depth of 8 additional layers and batch normalization [39] are the key differences between CNNs and DCNNs. The DINN starts with three convolutional layers followed by five inception blocks [40] to calculate the regression. Based on [41] and [42], the IRNN architecture is expanding the DINN by additional twelve residual-inception blocks. These two architectures are adaptations of *Inception-ResNet-v2* [42], with changed reductions in the image width and height induced by the rectangular input shape of  $113 \times 29$ , reduced repetitions of the residual blocks and a regression loss function. Batch normalization is applied to all convolutional layers within

DINN and IRNN architectures. To solve the regression task all NNs end with a fully-connected layer without an activation function. Unless otherwise stated, the activation functions are Rectified Linear Units (ReLU).

All input data is pre-processed such that the mean is 0 and the standard deviation is 1, which in our experience is necessary for convergence. In case of PBI the sections not corresponding to divertor parts are set to 0 afterwards (i. e. grey pixels in Figure 4). The desired NN outputs (targets) are pairs of  $\Delta\hat{R}$  and  $\hat{\epsilon}$ . The data set is shuffled and separated into five parts of almost equal size. By taking four parts as the training set and splitting the remaining part to validation and test set, it is possible to create five cross-validation sets, resulting in a training set size of 21745, validation set size of 2718 and test set size of 2718. The NNs are updated by the adam optimizer [43] with learning rates and batch sizes given in Table 1. Those two hyperparameters are chosen based on few trainings with learning rates of 0.001, 0.0005 and 0.0001 and batch sizes of 25 and 100 on cross-validation set 1. Batches are sampled randomly from the training set and every second training epoch the PBI training data is augmented by a Gaussian filter with a random Gaussian kernel standard deviation between 0 and 1 to increase the generalization of the NNs. The loss function is the mean squared error (mse) defined as

$$\text{mse} = \frac{1}{N} \sum_k^N (y_k - t_k)^2,$$

with reconstructions  $y_k$ , targets  $t_k$  and set size  $N$ . The error estimation is visualized by the root-mean-square error (rmse) defined as

$$\text{rmse} = \sqrt{\text{mse}}.$$

A scalar reconstruction performance measure  $\text{rmse}_{\text{total}}$  is defined based on the reconstructed values for  $\hat{\epsilon}$  and  $\Delta\hat{R}$ .

The total rmse is defined as

$$\text{rmse}_{\text{total}}(\Delta\hat{R}, \hat{\epsilon}) = 0.5 \left( \text{rmse}(\Delta\hat{R}) + \text{rmse}(\hat{\epsilon}) \right).$$

The rmse on the validation set suits as an early-stopping criteria if for 1000 consecutive NN weight updates the rmse is not improving. For each cross-validation distribution the NN training process is performed 32 times with different weight initializations. The software core is based on the TensorFlow library [44]. It runs on a workstation

with two Intel Xeon CPU E5 – 2650 and four NVIDIA GeForce GTX 1080 Ti. Each training is performed on a single GPU.

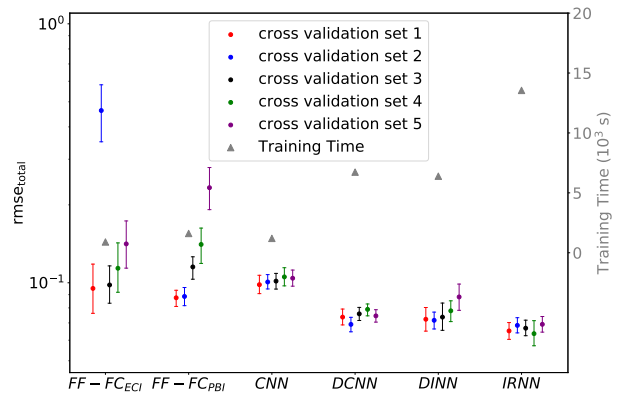
NN	Trainable Parameters	Learning Rate	Batch Size
FF – FC <sub>ECI</sub>	23 983	0.0005	25
FF – FC <sub>PBI</sub>	270 543	0.0010	25
CNN	63 671	0.0005	25
DCNN	89 073	0.0001	100
DINN	313 071	0.0001	100
IRNN	804 804	0.0001	100

**Table 1:** Comparing the number of trainable parameters as well as the chosen learning rate and batch size for different NNs

It is not possible to determine if human-level reconstruction ability is exceeded because an experimental setting to estimate the error of such a reconstruction can hardly be realized. The random noise of the Monte-Carlo simulation alone makes an error-free reconstruction impossible. To evaluate the NNs, the rmse for two additional benchmarks is defined that serve as examples of non-NN performance. The first one calculates the structural similarity (SSIM) [45] between each PBI of the test set with each of the training set and is referred to as SSIM rmse. The estimated  $\Delta\hat{R}$  and  $\hat{\epsilon}$  for the test set are those of the training set with the highest SSIM. The neighbor-rmse is the second benchmark which is the spacing distance in  $\Delta\hat{R}$  and  $\hat{\epsilon}$  direction. Note, that the SSIM method requires a high calculation time that depends on the size of the training set and the grid spacing is not considering any input data knowledge. However, the information that is actually asked for is required, so this method is not applicable in practice and only introduced for comparison purposes.

### 3 Results

The stated reconstruction problem is far more complex compared to the reconstruction of a single value in the W7-X limiter configuration. Due to non-linear effects between  $\Delta\hat{R}$  and  $\hat{\epsilon}$  the number of points in an equidistant grid of configurations increases exponentially with the number of considered parameters. There are NNs that converge to functions which solve the regression problem appropriately among all presented architectures.



**Figure 7:** Mean rmse with error bars and training times for each cross-validation set and NN type

An overview of all trained NN results is given in Figure 7. It shows the mean  $rmse_{total}$  with respect to all trainings and the 95% confidence interval of the mean calculated with bootstrapping [46] separately for the cross validation sets and architectures.

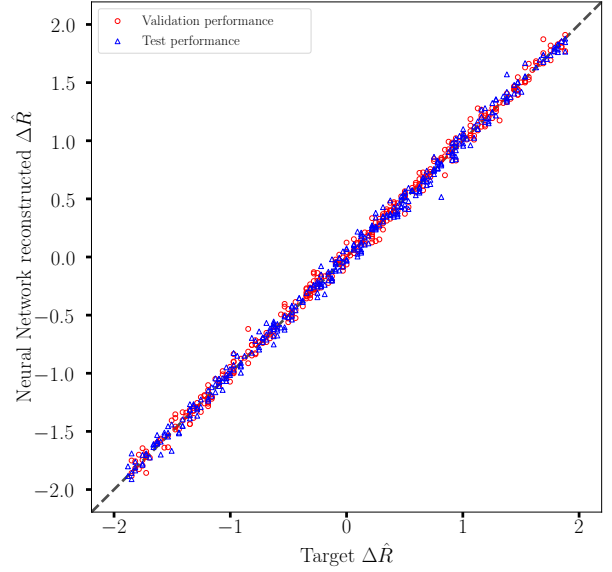
It can be seen that some FF-FCs with ECI perform well. However, the performance is not consistent across the five cross validation sets and especially cross validation set 3 shows poor performance. Although in [11] a similar concept was outperformed by CNNs on PBI, some of the trained FF-FCs with ECI behave surprisingly well. The FF-FC architectures with PBI show a similar inconsistent behavior. Again, the mean NN performance strongly varies when applied to the different cross validation sets. The FF-FC architectures are thus not recommendable, neither for ECI nor for PBI.

The mean performances of those NN architectures with convolutional layers on PBI are stable with respect to the cross-validation. Note that this stability hints to a sufficient generalization property, but because of the complex deep architectures used, it cannot be guaranteed. The three-layer CNN shows decent performance but it is clearly outperformed by deeper architectures. The first deeper structure we trained after the CNN is the IRNN because of its good performance on other image analysis problems [42]. The gain from CNN to IRNN is impressive but the training time as well as the calculation time increase approximately by a factor of 10, which is still real-time capable. By investigating architectures with

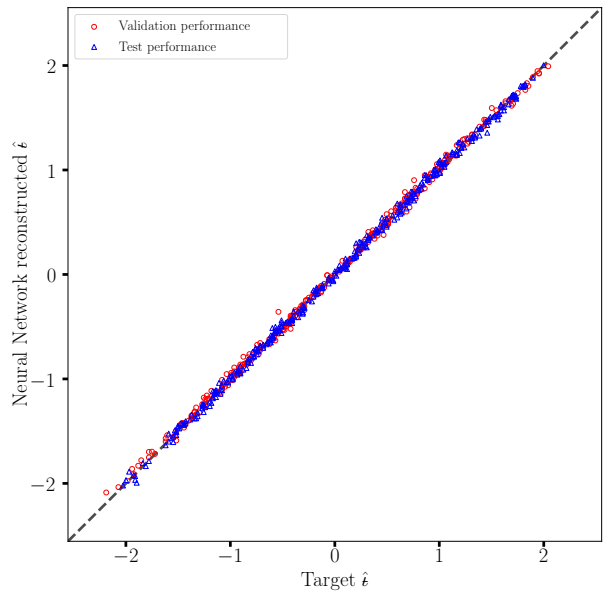
NN	rmse( $\Delta\hat{R}$ ) ( $10^{-2}$ )	rmse( $\hat{\epsilon}$ ) ( $10^{-2}$ )	$t_{\text{train}}$ ( $10^3$ s)	$t_{\text{calculation}}$ (ms)
FF – FC <sub>ECI</sub>	7.3	4.6	0.90	0.32
FF – FC <sub>PBI</sub>	9.5	15.8	1.61	1.06
CNN	10.2	7.1	1.20	1.74
DCNN	7.5	4.9	6.73	7.86
DINN	7.9	5.4	6.39	9.46
IRNN	6.7	5.3	13.55	18.39
SSIM	7.2	2.4	–	8489.27
Neighbor	3.1	1.5	–	–

**Table 2:** Comparing the different NNs best rmse

depths in between the previously mentioned ones, such as DCNN and DINN, we pursue a trade-off between reconstruction performance and run-time. As can be seen in Figure 7, the NN performance is almost similar to the much deeper IRNN whereas the training time is halved. In Table 2 we compare the NNs, SSIM rmse and neighbor rmse( $\hat{\epsilon}$ ) and rmse( $\Delta\hat{R}$ ) based on the worst performing cross-validation set and within those for the best converged NN. Additionally, the table lists the training times for each NN architecture and the necessary time to calculate the estimate for one input image. Note that these NNs are not optimized with respect to run time. When comparing the NN results with both benchmarks on the one hand it can be noticed that  $\hat{\epsilon}$  can be reconstructed a little better (rmse difference of  $2.9 \cdot 10^{-2}$ ) by the SSIM approach and clearly better by taking the next neighbor (rmse difference of  $3.8 \cdot 10^{-2}$ ). On the other hand for  $\Delta\hat{R}$  the NNs perform equally good or slightly better compared to the SSIM benchmark (rmse difference of  $0.5 \cdot 10^{-2}$ ) but the neighbor rmse is still smaller (rmse difference of  $3.6 \cdot 10^{-2}$ ). Since the neighbor benchmark does not depend on the input data but only on the grid spacing, it is not a competitive method but it shows that the errors of NNs are of the same order of magnitude. Note that the SSIM value between neighbors is not necessary the largest. These results show that there is still potential for better NN performance. The computation time for SSIM however is not sufficient for real-time applications while up to 18ms for NNs are acceptable.



**Figure 8:**  $\Delta\hat{R}$  reconstruction with best IRNN and a rmse of 0.056



**Figure 9:**  $\hat{\epsilon}$  reconstruction with best IRNN and a rmse of 0.034



The relative error in the  $\Delta\hat{R}$  reconstruction is larger compared to that of the  $\hat{\epsilon}$  reconstruction. This can be seen in Table 2 but also when visualizing a single NN result in more detail. In Figure 8 and Figure 9 the grey dashed lines mark the identity and thus perfect reconstruction. The points represent the estimated value for each target value and the distance to the grey dashed line resembles the error. In this case, as an example the performance of the best IRNN with a total rmse of 0.045 is shown. When comparing the  $\Delta\hat{R}$  and  $\hat{\epsilon}$  reconstruction (Figures 8 and 9 respectively), it can be seen that  $\hat{\epsilon}$  is estimated better than  $\Delta\hat{R}$ . However, the NN calculation is appropriately following the identity for both values, indicating that a connection between heat flux image and the respective property has been found. The relative error with respect to the target range is 1.4% for the radial shift and 0.9% for the rotational transform. The reconstruction appears to be symmetric, so the targets are not systematically over- or underestimated.

In the following, the uniformity of the reconstruction error in the  $\Delta\hat{R}$ - $\hat{\epsilon}$  space is investigated. When comparing the best-performing NN of each architecture, there are two different types of error distributions, which are represented by the CNN and IRNN results in Figure 10. The four sub-figures depict the difference in the NN performance in a detailed 2D scatter, where the absolute residuals of  $\hat{\epsilon}$  and  $\Delta\hat{R}$  separately are color-coded. The first type, seen in Figures 10(a) and 10(b), shows clear clusters of higher residuals, especially along the boundary of available data. This phenomenon occurs for  $\Delta\hat{R}$  and  $\hat{\epsilon}$  reconstruction, however, the cluster of higher residuals are not located at the same region for those two values. This is also the case for PBI FF-FCs. The deeper structures DCNN, DINN and IRNN show a homogeneous error distribution except for a few outliers but no specific structure can be identified. In Figures 10(c) and 10(d) this is shown for the example of IRNN. Notably these residuals do not increase when moving towards the boundary. The ECI FF-FCs show error clusters for  $\Delta\hat{R}$  and a more homogeneous error distribution for  $\hat{\epsilon}$ .

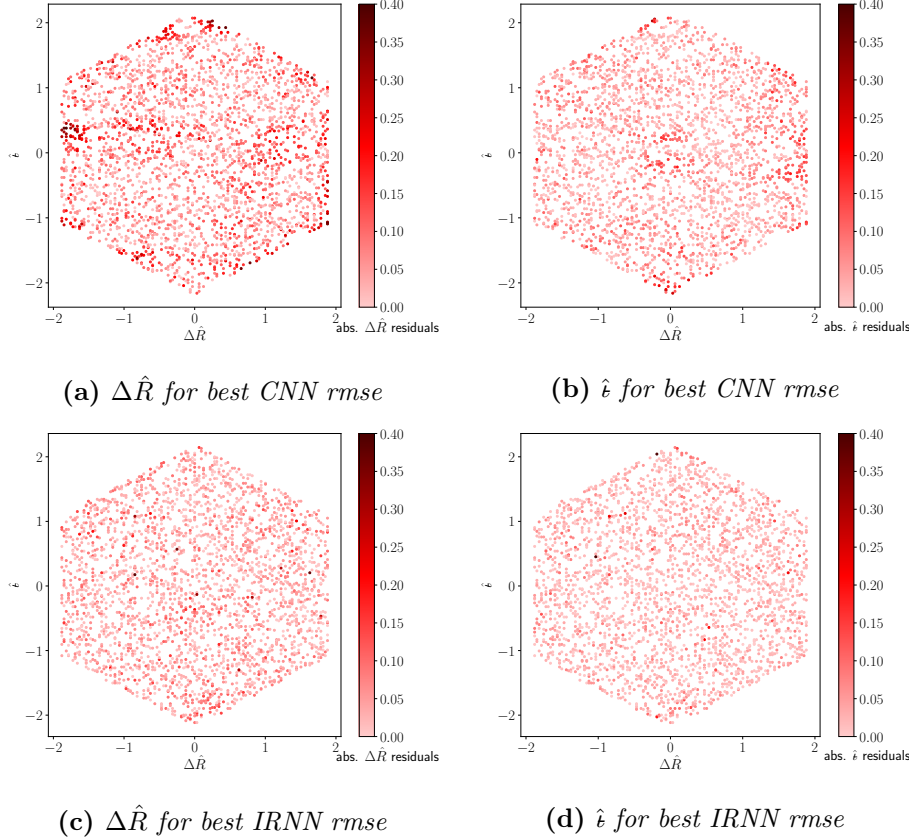
## 4 Conclusion

The work presented in this article demonstrates the capability of NNs to find a connection be-

tween simulated W7-X divertor heat flux images and proxies for edge magnetic field properties. Former experiments showed similar promising results for less complex limiter heat flux images and a single plasma parameter. The choice of the suiting NN architecture is a trade-off between performance and calculation times. Deeper and more branched NN architectures are associated with decreased regression errors. When stable convergence in combination with small regression errors is the main focus, a customized IRNN can be recommended. FF-FCs are not advisable due to their unpredictable performance on various cross validation sets. A benchmark approach without using machine learning achieves slightly smaller errors but fails to be real-time capable.

There are several possible future investigations arising from the results of this first step. Since the SSIM benchmark example is not outperformed by a machine learning application, further architecture and hyperparameter adjustments may reduce the reconstruction errors. With NNs found feasible to reconstruct proxies for edge magnetic field properties from simulated heat load patterns, it would be the most interesting and consequent pursue to apply the NN to experimental data, which is ongoing work. The PBI allows effortless interchange of simulation and experimental data [12]. This opens the possibility to support experimental data with simulations with the aim to improve the performance of the proxy reconstruction as compared to the proxy reconstruction on NNs trained on experimental data only. Such an approach is inspired by the results in [11] and can be improved by techniques such as transfer learning and generative adversarial networks to reduce necessary experimental data in the training set as much as possible.

More advanced open problems are the consideration of time-dependent effects and using the safety criterion developed in [12] combined with NN architectures from this study for a deep Q-value estimator [24] in reinforcement learning approaches to further approach the long-term objective of real-time heat load control and performance optimization. Besides reinforcement learning, common control theory approaches are candidates for a control loop which would directly use inputs such as the features which have been herein proven to be reconstructable from synthetic heat load images in addition to more direct diag-



**Figure 10:** Absolute residuals color-coded depending on *iota* and shift position

nostic inputs such as Rogowski coils and further. Beyond the scope of heat load control, the possibility to reconstruct proxies for the rotational transform, radial axis shift or further edge magnetic field properties is a desirable goal in itself since it can provide important status information to the operators of W7-X in real-time.

## 5 Acknowledgments

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training program 2014-2018 and 2019-2020 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

We wish to acknowledge the helpful discussions with J. Geiger, T. Sunn Pedersen, R. Labahn, G. Leifert, T. Grüning, T. Strauss, M. Weidemann and J. Michael. We are also thankful for the support from the CITlab (University of Rostock) and Planet-AI teams.

## Bibliography

- [1] Yann LeCun. *The MNIST database of handwritten digits*. 1998.
- [2] Gundram Leifert et al. “Cells in Multidimensional Recurrent Neural Networks”. In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 3313–3349. arXiv: arXiv:1412.2620v2.
- [3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural Machine Translation by Jointly Learning to Align and Translate”. In: *Proceedings of 3rd International Conference on Learning Representation*. San Diego, USA, 2015, pp. 1–15. arXiv: arXiv:1409.0473v7.
- [4] Donald F Specht. “A general regression neural network”. In: *IEEE transactions on neural networks* 2:February 1991 (1991), pp. 568–576. DOI: 10.1109/72.97934.
- [5] Philipp Fischer, Alexey Dosovitskiy, and Thomas Brox. “Image Orientation Estimation with Convolutional Networks”. In: *German Conference on Pattern Recognition*.

- Aachen, Germany: Springer, 2015, pp. 368–378.
- [6] J B Lister and H Schnurrenberger. “Fast non-linear extraction of plasma equilibrium parameters using a neural network mapping”. In: *Nuclear Fusion* 31.7 (1991), pp. 1291–1300. DOI: 10.1088/0029-5515/31/7/005.
- [7] E Coccoresse, C Morabito, and R Martone. “Identification of noncircular plasma equilibria using a neural network approach”. In: *Nuclear Fusion* 34.10 (1994), pp. 1349–1363. DOI: 10.1088/0029-5515/34/10/i05.
- [8] Arimitsu Wakasa, Sadayoshi Murakami, and Shun-ichi Oikawa. “Study of Neoclassical Transport in LHD Plasmas by Applying the DCOM/NNW Neoclassical Transport Database”. In: *Plasma and Fusion Research* 3 (2008), S1030. DOI: 10.1585/pfr.3.S1030.
- [9] B. Cannas et al. “An adaptive real-time disruption predictor for ASDEX upgrade”. In: *Nuclear Fusion* 50.7 (2010), p. 075004. DOI: 10.1088/0029-5515/50/7/075004.
- [10] J. Citrin et al. “Real-time capable first principle based modelling of tokamak turbulent transport”. In: *Nuclear Fusion* 55.9 (2015), p. 092001. DOI: 10.1088/0029-5515/55/9/092001. arXiv: 1502.07402.
- [11] Marko Blatzheim et al. “Neural network performance enhancement for limited nuclear fusion experiment observations supported by simulations”. In: *Nuclear Fusion* 59.1 (2019), p. 016012. DOI: 10.1088/1741-4326/aaefaf.
- [12] Daniel Böckenhoff and Marko Blatzheim. “Application of improved analysis of convective heat loads on plasma facing components to Wendelstein 7-X”. In: *Nuclear Fusion* 59.8 (2019), p. 086031. DOI: 10.1088/1741-4326/ab201e.
- [13] J. Nührenberg and R. Zille. “Stable stellarators with medium  $\beta$  and aspect ratio”. In: *Physics Letters A* 114.3 (1986), pp. 129–132. DOI: 10.1016/0375-9601(86)90539-6.
- [14] Craig Beidler et al. “Physics and Engineering Design for Wendelstein VII-X”. In: *Fusion Technology* 17.1 (1990), pp. 148–168. DOI: 10.13182/FST90-A29178.
- [15] J. Nührenberg et al. “Overview on Wendelstein 7-X Theory”. In: *Fusion Technology* 27.3T (1995), pp. 71–78. DOI: 10.13182/FST95-A11947048.
- [16] H.-S. Bosch et al. “Technical challenges in the construction of the steady-state stellarator Wendelstein 7-X”. In: *Nuclear Fusion* 53.12 (2013), p. 126001. DOI: 10.1088/0029-5515/53/12/126001.
- [17] Thomas Sunn Pedersen et al. “Key results from the first plasma operation phase and outlook for future performance in Wendelstein 7-X”. In: *Physics of Plasmas* 24.5 (2017), p. 055503. DOI: 10.1063/1.4983629.
- [18] Thomas Sunn Pedersen et al. “First results from divertor operation in Wendelstein 7-X”. In: *Plasma Physics and Controlled Fusion* 61.1 (2019), p. 014035. DOI: 10.1088/1361-6587/aaec25.
- [19] LA. Artsimovich. “Tokamak Devices”. In: *Nuclear Fusion* 12.2 (1972), p. 215. DOI: 10.1088/0029-5515/12/2/012.
- [20] Vitaly Dmitrijevitsch Shafranov. “Equilibrium of a toroidal plasma in a magnetic field”. In: *Journal of Nuclear Energy. Part C, Plasma Physics, Accelerators, Thermonuclear Research* 5.4 (1963), pp. 251–258. DOI: 10.1088/0368-3281/5/4/307.
- [21] Arnold Lumsdaine et al. “Design and analysis of the W7-X divertor scraper element”. In: *Fusion Engineering and Design* 88.9–10 (2013), pp. 1773–1777. DOI: 10.1016/j.fusengdes.2013.05.075.
- [22] M.W. Jakubowski et al. “Development of infrared and visible endoscope as the safety diagnostic for steady-state operation of Wendelstein 7-X”. In: *Proceedings of the 2014 International Conference on Quantitative InfraRed Thermography*. Vol. 100. Bordeaux, France: QIRT Council, 2014. DOI: 10.21611/qirt.2014.100.
- [23] P. Read Montague. *Reinforcement Learning: An Introduction, by Sutton, R.S. and Barto, A.G.* 2nd ed. Vol. 1. MIT Press, 2017, p. 360. DOI: 10.1016/S1364-6613(99)01331-5. arXiv: 1603.02199.

- [24] Volodymyr Mnih et al. “Human-level control through deep reinforcement learning”. In: *Nature* 518.7540 (2015), pp. 529–533. DOI: 10.1038/nature14236.
- [25] K. Rahbarnia et al. “Diamagnetic energy measurement during the first operational phase at the Wendelstein 7-X stellarator”. In: *Nuclear Fusion* 58.9 (2018), p. 096010. DOI: 10.1088/1741-4326/aacab0.
- [26] T. Andreeva et al. “Equilibrium evaluation for Wendelstein 7-X experiment programs in the first divertor phase”. In: *Fusion Engineering and Design* 146.October 2018 (2019), pp. 299–302. DOI: 10.1016/j.fusengdes.2018.12.050.
- [27] Andreas Werner et al. “W7-X magnetic diagnostics: Rogowski coil performance for very long pulses”. In: *Review of Scientific Instruments* 79.10 (2008), 10F122. DOI: 10.1063/1.2957933.
- [28] Daniel Böckenhoff et al. “Reconstruction of Magnetic Configurations in W7-X using Artificial Neural Networks”. In: *Nuclear Fusion* 58.5 (2018), p. 056009. DOI: 10.1088/1741-4326/aab22d.
- [29] Y Feng et al. “Impact of Island Geometry on Island Divertor Performance”. In: *30th EPS Conference on Plasma Physics and Controlled Fusion*. Ed. by M. Pick and P. Helfenstein. St. Petersburg: European Physical Society, 2003.
- [30] Hermann Renner et al. “Physical Aspects And Design of the Wendelstein 7-X Divertor”. In: *Fusion Science and Technology* 46.2 (2004), pp. 318–326. DOI: 10.13182/FST04-A570.
- [31] Y. Gao et al. “Approaches for quantitative study of divertor heat loads on W7-X”. In: *Proceedings of the 2018 International Conference on Quantitative InfraRed Thermography*. Berlin, Germany: QIRT Council, 2018, pp. 6–7. DOI: 10.21611/qirt.2018.p23.
- [32] Yu Turkin. “Current Control by ECCD for W7-X”. In: *Fusion Science and Technology* 50.3 (2006), pp. 387–394. DOI: 10.13182/FST06-5.
- [33] S.A. Bozhenkov et al. “Service oriented architecture for scientific analysis at W7-X. An example of a field line tracer”. In: *Fusion Engineering and Design* 88.11 (2013), pp. 2997–3006. DOI: 10.1016/j.fusengdes.2013.07.003.
- [34] F. Wagner et al. *Overview on W7-AS Results with Relevance for Wendelstein 7-X and the Low-Shear Stellarator Line*. Tech. rep. 1998, p. 33029492.
- [35] Martin Ester et al. “Density-Based Clustering Algorithms for Discovering Clusters”. In: *Comprehensive Chemometrics 2* (2010), pp. 635–654. DOI: 10.1016/B978-044452701-1.00067-3. arXiv: 10.1.1.71.1980.
- [36] Samuel A. Lazerson et al. “First measurements of error fields on W7-X using flux surface mapping”. In: *Nuclear Fusion* 56.10 (2016), p. 106005. DOI: 10.1088/0029-5515/56/10/106005.
- [37] Y. Gao et al. “Methods for quantitative study of divertor heat loads on W7-X”. In: *Nuclear Fusion* 59.6 (2019), p. 066007. DOI: 10.1088/1741-4326/ab0f49.
- [38] Samuel A Lazerson et al. “Error fields in the Wendelstein 7-X stellarator”. In: *Plasma Physics and Controlled Fusion* 60.12 (2018), p. 124002. DOI: 10.1088/1361-6587/aae96b.
- [39] Sergey Ioffe and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *Proceedings of the 32nd International Conference on Machine Learning*. Lille, France, 2015. DOI: 10.1007/s13398-014-0173-7.2. arXiv: 1502.03167.
- [40] Christian Szegedy et al. “Rethinking the Inception Architecture for Computer Vision”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. Las Vegas USA, 2016, pp. 2818–2826. DOI: 10.1109/CVPR.2016.308. arXiv: 1512.00567.
- [41] K He et al. “Deep residual learning for image recognition”. In: *IEEE conference on computer vision and pattern recognition*. Las Vegas USA, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90. arXiv: 1512.03385.

- [42] Christian Szegedy et al. "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning". In: *AAAI*. San Francisco, USA, 2017, p. 12. DOI: 10.1016/j.patrec.2014.01.008. arXiv: 1602.07261.
- [43] Diederik P. Kingma and Jimmy Lei Ba. "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations 2015*. San Diego, USA, 2015, pp. 1–15. DOI: 10.1145/1830483.1830503. arXiv: 1412.6980.
- [44] Martín Abadi et al. "TensorFlow: A System for Large-Scale Machine Learning". In: *USENIX Symposium on OSDI*. Savannah, USA, 2016, pp. 265–283.
- [45] Zhou Wang et al. "Image Quality Assessment: From Error Visibility to Structural Similarity". In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612. DOI: 10.1109/TIP.2003.819861.
- [46] Thomas J Diccio and Bradley Efron. "Bootstrap Confidence Intervals". In: *Statistical Science* 11.3 (1996), pp. 189–228. DOI: 10.1214/ss/1032280214.

## A Neural Network Architectures

There are many types of NNs and by combining them and modifying parameters the number is infinite. The analysis done in "Neural Network Regression Approaches to Reconstruct Plasma Properties from Wendelstein 7-X Heat Load Images" is limited to five different types introduced in detail in the following. The hyperparameter of a convolutional layer are shown

### A.1 Feed-Forward Fully-Connected Network

The first hidden layer consists of 80 neurons, the second layer consists of 100 neurons leading to the output of 2 values. The hidden layer activation functions are ReLU, the last layer activations are the identity.

### A.2 Convolutional Network

The convolutional layer hyperparameter are given as shown in Figure 11. A visual representation of the three layers CNN is given in Figure 12. The architecture starts with a convolutional layer with  $5 \times 5$  kernel followed by one with a  $3 \times 3$  kernel. The dimension is reduced by a  $2 \times 2$  Max-Pooling with stride  $2 \times 2$ . The last convolutional layer has a  $3 \times 3$  kernel and is again followed by  $2 \times 2$  Max-Pooling with stride  $2 \times 2$ . The last two layers are fully-connected, the first one has 24 neurons and the last one leads to the two output values. All activation functions are ReLU, except for the last layer, where activations are the identity.

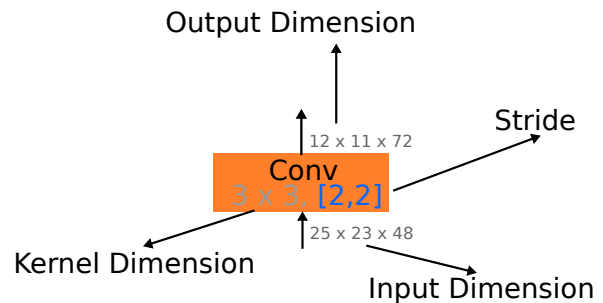


Figure 11: The hyperparameter for convolutional layer are given as shown here. The padding is given indirectly by the input and output dimension. If the width and height stay the same, padding is applied, otherwise, it is not applied.

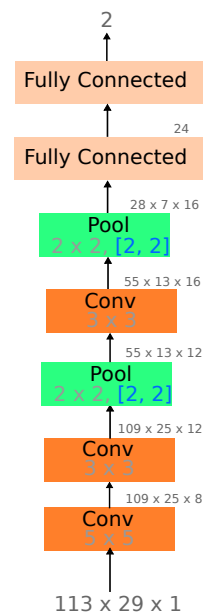


Figure 12: Overview of the CNN



### A.3 Deep Convolutional Network

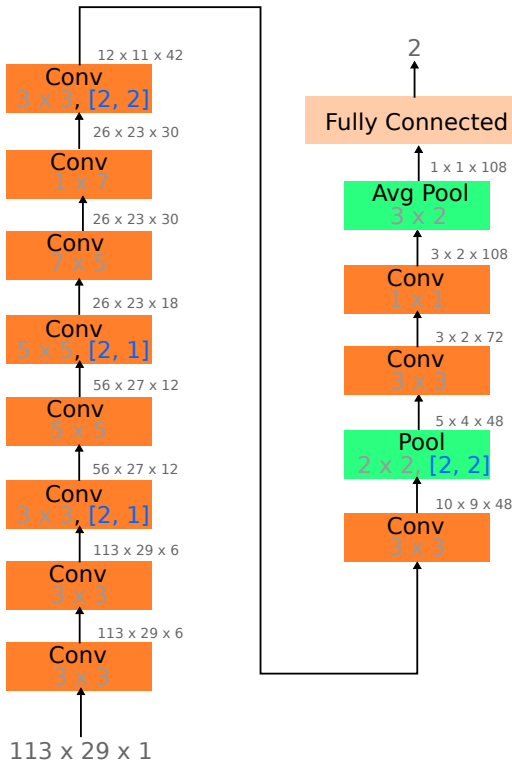


Figure 13: Overview of the DCNN.

The DCNN with 11 convolutional layers is shown in Figure 13. It starts with 2 convolutional layers and a  $3 \times 3$  kernel. A third  $3 \times 3$  kernel with  $2 \times 1$  dimension reduction follows. The next two layers consists of a  $5 \times 5$  kernel with the latter reducing the dimension again by  $2 \times 1$ . The next four kernels are  $7 \times 5$ ,  $1 \times 7$ ,  $3 \times 3$  and  $3 \times 3$  respectively leading to a total dimensionality of  $10 \times 9 \times 48$ . A Max-Pooling with  $2 \times 2$  kernel reduces it to  $5 \times 4 \times 48$ . Two more convolutional layers with  $3 \times 3$  and  $1 \times 1$  kernel followed by average pooling lead to a dimensionality of  $1 \times 1 \times 108$ . Batch normalization is applied to each convolutional layer and up to here all activation functions are ReLU. The last layer is fully-connected and without an activation function.

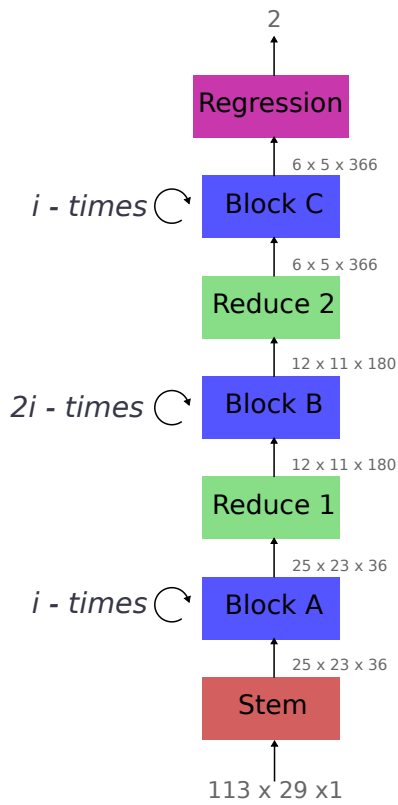
### A.4 Inception Residual Network

The basic structure of the residual inception regression network is shown in Figure 14. It begins with a block called stem, continues with an alternating structure of repeated blocks and a reduction block and ends with a regression block. The number of repetitions depends on the parameter  $i$ , which is set to 3 in the IRNN. The stem (see Fig-

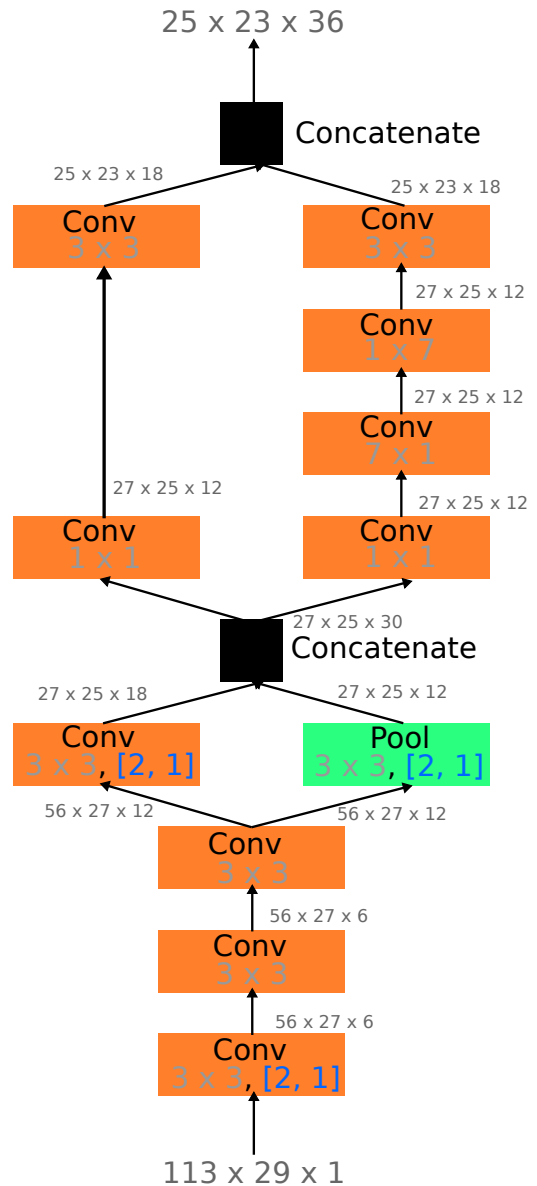
ure 15) consists of three convolutional layers and two consecutive inception modules. The input picture dimensionality is reduced from  $113 \times 29$  to  $25 \times 23$  while the number of channels increases from 1 to 36. In the classic Inception-ResNet v2 the dimension reduction is done on both dimensions but there the input image dimensions are larger. Block A (see Figure 16) is the first inception residual block. In each of the blocks A–C the block input is processed along different paths and concatenated along the channel dimension. Afterwards a  $1 \times 1$  convolution without an activation function is applied. In block A the result is scaled by the factor 0.17 and added to the block input followed by the ReLU activation. The output dimension does not change so stacking these blocks is possible.

After block A the following reduce block is shown in Figure 17, which is an inception module. The last layer has a stride of  $[2, 2]$  which reduces the dimension from  $25 \times 23$  to  $12 \times 11$  while increasing the channel size from 36 to 180.

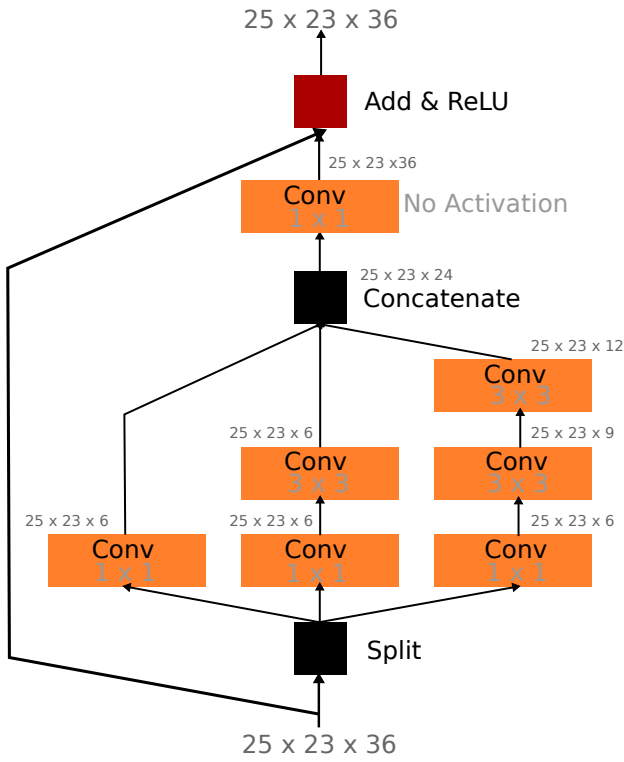
Despite the different input size to block B (see Figure 18) and varied convolutions within the inception module, the overall structure is similar to block A. Here the scaling factor after the inception is 0.2.



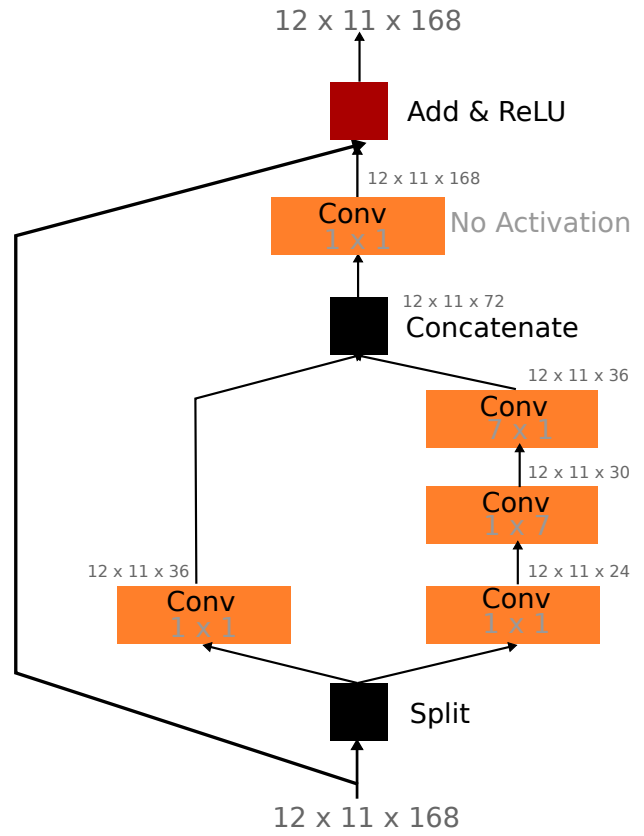
**Figure 14:** Overview of the IRNN. The parameter  $i$  determines the NN depth by giving the numbers of repetitions of each block. The individual blocks are described in Figures 15 to 21.



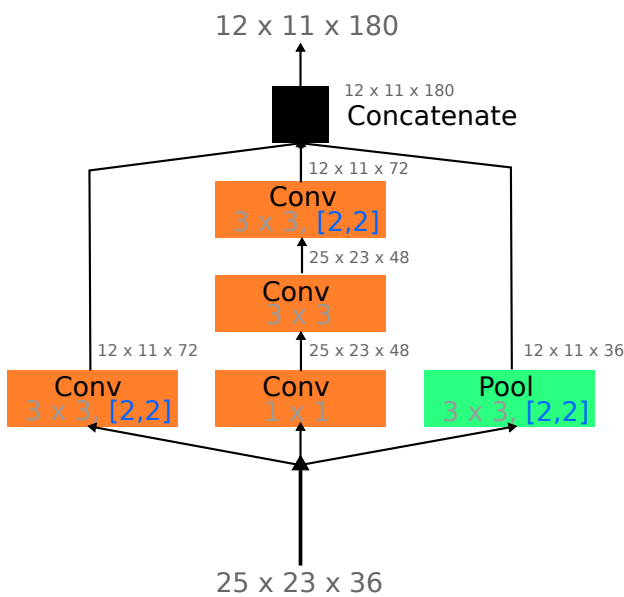
**Figure 15:** Stem of IRNN which structure overview is given in Figure 14



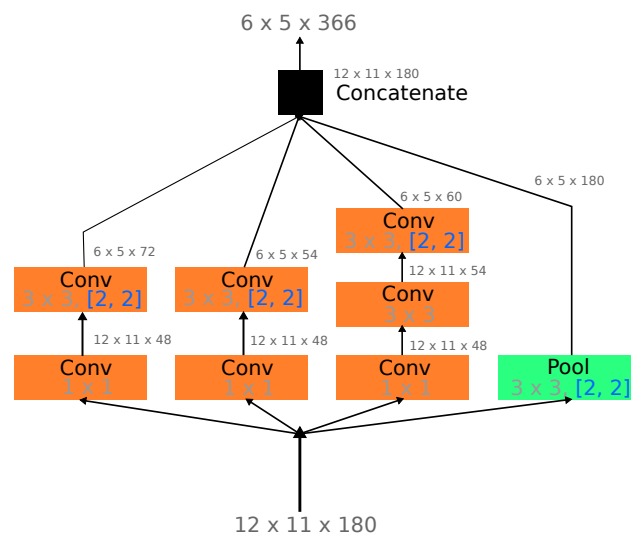
**Figure 16:** Residual block A of residual-inception NN which structure overview is given in Figure 14



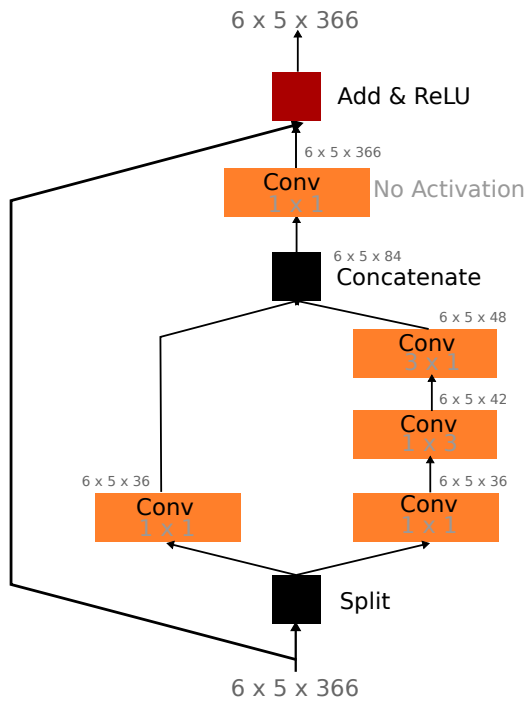
**Figure 18:** Residual block B of IRNN which structure overview is given in Figure 14



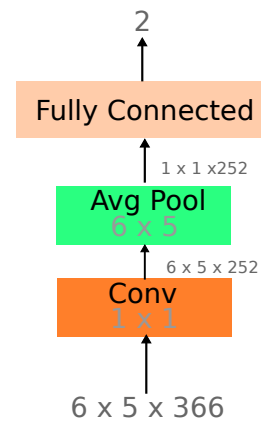
**Figure 17:** Reduce 1 module of IRNN which structure overview is given in Figure 14



**Figure 19:** Reduce 2 module of IRNN which structure overview is given in Figure 14



**Figure 20:** Residual block C of IRNN which structure overview is given in Figure 14



**Figure 21:** Reduce 3 module and regression of IRNN which structure overview is given in Figure 14

### A.5 Deep Inception Network

The DINN is a special case of the IRNN, where the residual parameter  $i$  is equal to 0 which can be seen as skipping block A – C in Figure 14. Even without the residual blocks this NN has a deep and complex structure with multiple inception modules.