Determining 1D Fast-Ion Velocity Distribution Functions from Ion Cyclotron Emission Data Using Deep Neural Networks

B. S. Schmidt,[1, a)] M. Salewski,[1] B. Reman,[2] R. O. Dendy,[3, 4] D. Moseev,[5] R. Ochoukov,[5] A. Fasoli,[6] M. Baquero-Ruiz,[6] and H. Järleblad[1]

[1)] *Technical University of Denmark, Department of Physics, Kgs. Lyngby, Denmark*
[2)] *Laboratoire Plasma et Conversion d'Energie, Université Toulouse, France*
[3)] *Centre for Fusion, Space and Astrophysics, University of Warwick, United Kingdom of Great Britain and Northern Ireland*
[4)] *Culham Centre for Fusion Energy, Culham Science Centre, Abingdon, Oxfordshire, United Kingdom of Great Britain and Northern Ireland*
[5)] *Max-Planck-Institut für Plasmaphysik, Greifswald and Garching, Germany*
[6)] *École Polytechnique Fédérale de Lausanne, Swiss Plasma Center, Switzerland*

The relationship between simulated ion cyclotron emission (ICE) signals s and the corresponding 1D velocity distribution function $f(v_\perp)$ of the fast ions triggering the ICE is modelled using a two-layer deep neural network. The network architecture (number of layers and number of computational nodes in each layer) and hyperparameters (learning rate and number of learning iterations) are fine-tuned using a bottom-up approach based on cross-validation. Thus, the optimal mapping $g(s;\theta)$ of the neural network in terms of the number of nodes, the number of layers, and the values of the hyperparameters, where θ is the learned model parameters, is determined by comparing many different configurations of the network on the same training and test set and choosing the best one based on its average test error. The training and test sets are generated by computing random ICE velocity distribution functions f and their corresponding ICE signals s by modelling the relationship as the linear matrix equation $Wf = s$. The simulated ICE signals are modelled as edge ICE signals at LHD. The network predictions for f based on ICE signals s are on many simulated ICE signal examples closer to the true velocity distribution function than that obtained by 0^{th}-order Tikhonov regularization, although there might be qualitative differences in which features one technique is better at predicting than the other. Additionally, the network computations are much faster. Adapted versions of the network can be applied to future experimental ICE data to infer fast-ion velocity distribution functions.

I. INTRODUCTION

In recent years, it has become possible to infer 2D fast-ion velocity distribution functions from fast-ion measurement data in magnetically confined plasmas[1–8]. This traditionally requires a formulation of the diagnostic forward models in terms of so-called weight functions that have been developed for collective Thomson scattering[9], fast-ion D-alpha spectroscopy[10–12], gamma-ray spectroscopy[13,14], neutron emission spectroscopy[15,16], and fast-ion loss detectors[17]. Here, we apply this approach to simulated ion cyclotron emission (ICE)[18] data and formulate a 1D inversion problem to infer a 1D fast-ion velocity distribution function. Further, we take an entirely new approach based on deep neural networks.

A deep neural network is a computational model comprising a network of functions capable of representing complex linear and non-linear relationships to translate input data into output data. Such networks are well-suited for analysing data from experiments where the quantities of interest are not directly measurable but must be inferred by modelling. The mathematical formulation of the problem may introduce systematic bias, and the inherent noise of the measurements increases the difficulty in obtaining a solution representative of the true solution by using traditional mathematical methods. Specifically, the linear inverse problem $Wf = s$ of finding a distribution function f given a forward model encoded in W and the measurement data s, is ill-posed. Regularization is needed in order to find an approximate solution. This is often done by Tikhonov regularization[2–8].

An alternative approach is to use deep neural networks to model the relationship between the measured signal s and the distribution function f. The networks are trained using a supervised training approach on datasets consisting of an input and known true solution such that the predictions by the network $g(s;\theta)$ come as close as possible to the true solutions. The network learns to predict f by optimizing its internal parameters θ. In effect, training a neural network is a calibration of the network (the equipment) by comparing its output values (the measurements) to known values (the calibration standard).

Deep neural networks can be applied to ICE data analysis to assist and improve current analysis techniques for measurements of velocities, densities, and energies of fast ions. Such analysis is of crucial importance for ITER[21,22]. Further, such measurements require no additional measurement equipment since the future ICRH antennas at ITER can be used as receivers to measure ICE[23]. Thus, the easy access to ICE data and the inclusion of such diagnostic systems in ITER provide a strong incentive to develop ICE data analysis techniques.

Typically, fast ions drive instabilities at the ion cyclotron frequency and higher harmonics in the $5 - 100$

[a)] Electronic mail: bossc@fysik.dtu.dk

MHz range. Such instabilities occur when the distribution function f becomes inverted in velocity space, i.e., when $\partial f/\partial v > 0$, since free energy becomes available to drive the instabilities[19]. The magnetoacoustic cyclotron instability (MCI) produces ICE[19,20], and compressional Alfvén eigenmodes (CAEs) and global Alfvén eigenmodes (GAEs) occur in the ion cyclotron frequency range[24–27]. Therefore, analysis of data from ICE is important since CAEs and GAEs may redistribute the energy from fusion-born alpha particles to the bulk plasma faster than Coulomb collisions. Furthermore, ICE may cause pitch angle scattering of beam ions when using NBI as heating source[28].

II. 1D ICE WEIGHT FUNCTIONS

We calculate 1D ICE weight functions by running a hybrid particle-in-cell (PIC) self-consistent Maxwell-Lorentz computation of the relaxation of an energetic ion population with given values of $v_\perp$ and $v_\parallel$ in the presence of an MCI. The computations result in power spectra of the excited fields in the non-linear saturated regime of the MCI. Each horizontal line in Fig. 1 constitutes such a power spectrum for the specified value of $v_\perp$ while $v_\parallel$ is held constant. Thus, Fig. 1 illustrates the signal intensities for a given $v_\perp$ and $v_\parallel$ of the ions for the frequencies ω. 1D weight functions for ICE appear as vertical lines in Fig. 1 and indicate the 1D velocity-space sensitivity of the detector to ions at a given ω. The 1D weight functions are analogous to the 2D weight functions for the previously mentioned fast-ion diagnostics[1,2,9,10,13,15–17]. The data array W displayed in Fig. 1 provides a mapping from ICE signals to distribution functions on the basis that $Wf = s$, where f is the distribution function and s the ICE signal. This approach assumes that the physics embodied in the PIC-hybrid simulations dominates the generation of the observed ICE signals, and this appears to be well grounded[29].

The relationship between the distribution function and the ICE signal is modelled as follows. Let $f \in \mathbb{R}^n$ be the distribution function and let $s \in \mathbb{R}^m$ be the signal. The relationship between f and s is then given by the linear equation

$$Wf = s, \tag{1}$$

where $W \in \mathbb{R}^{m\times n}$, $m = 451$, and $n = 22$. The matrix W contains the known physics of the problem and is calculated numerically from the Maxwell-Lorentz system of equations as follows: A non-linear 1D3V PIC-hybrid code follows the velocity-space trajectories (including the gyromotion of fully kinetic energetic and thermal ions) together with the three components of the electric and magnetic fields in an isothermal massless neutralising electron fluid. The kinetic ions, fluid electrons, and fields are coupled self-consistently through the Lorentz force and Maxwell's equations in Darwin's approximation[30]. These simulations are followed through the linear phase of the MCI and then deeply into its non-linear saturated phase.

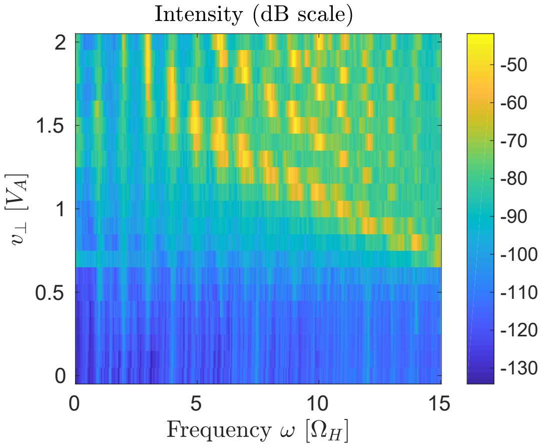

Figure 1. The weight function matrix W in the formulation $Wf = s$. V_A is the Alfvén speed, and Ω_H is the proton cyclotron frequency.

The calculations are initialized with physical parameters representative of tokamak and stellarator edge plasmas. In these simulations, we used parameters relevant for LHD. The thermal deuterons are loaded using a quiet start method[31] while the energetic protons are initialised uniformly and randomly in space with velocities following a cold-ring distribution $n_H\delta\left(v_\parallel\right)\delta\left(v_\perp - u_\perp\right)$ to generate the inversion[32]. The background magnetic field $B_0 = 1.75T$ is oriented at 89° with respect to the periodic simulation domain that consists of 2048 cells of length $\Delta x = 1.30 r_D$, where r_D is the deuteron Larmor radius. 500 macroparticles per energetic and per thermal ion species are loaded in each cell and $n_H/n_e = 0.002$ to achieve a high signal-to-noise ratio. The spatio-temporal fast Fourier transform of the perturbed magnetic field $\delta B_z(x,t)$ provides the 2D dispersion relation $\delta B_z(k,\omega)$, where k and ω are the wavenumber and frequency, respectively. In this specific case, the relaxation of the energetic protons through the MCI generates self-consistent excitations of the magnetic fields at multiple cyclotron harmonics on the fast-Alfvén branch. Summing over wavenumber-space k yields the synthetic power spectra. Each simulation corresponds to a different value of $v_\perp \in [0\,V_\text{A}, 2\,V_\text{A}]$. The power spectra are illustrated in Fig. 1.

III. INVERSION BY DEEP NEURAL NETWORKS

We use a trained deep neural network to predict the distribution functions from simulated ICE signals. Note that the simulated ICE signals closely mimic actual ICE signals from edge ICE in LHD. The network is trained using a stochastic gradient descent method based on adaptive estimation of first-order and second-order moments called Adam in the Keras API. The network is trained on a large number of simulated datasets where each ICE spectrum is directly matched with a distribution function. Gaussian noise is added to the measurements to obtain a signal-to-noise ratio of 5% which is similar to the expected noise level in ICE measurements at LHD.

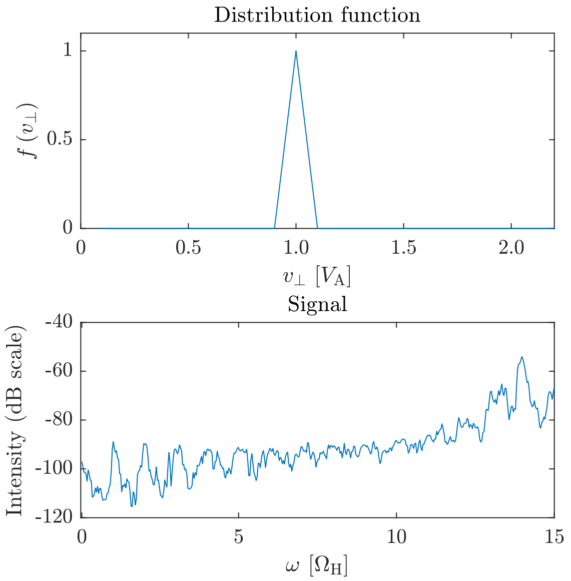

Figure 2. An example of a simulated distribution function $f(v_\perp)$ along with its corresponding simulated signal as measured by an ICE detector.

Fig. 2 shows a simulated ICE signal and the corresponding distribution function.

The network performs its computations according to a predetermined architecture, initial values of network parameters, and user-specified values of hyperparameters to produce the network output. The network output is compared to the true solution, and the internal parameters are updated by backpropagation. The network is then trained again on the same data with its updated internal parameters. The number of times this is repeated is called the number of iterations.

The cost function that the network minimizes during training is chosen to be the mean squared error

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(f_i - \hat{f}_i\right)^2, \tag{2}$$

where f_i corresponds to the ith entry in the true solution f, and $\hat{f}_i$ corresponds to the ith entry in the network output vector $\hat{f}$. Letting p_{data} be the true data generating distribution, the problem that the network solves is

$$g^* = \arg\min_g \mathbb{E}_{s,f\sim p_{\text{data}}} \|f - g(y)\|^2 \tag{3}$$

with the solution

$$g^*(s) = \mathbb{E}_{f\sim p_{\text{data}}(f|s)}(f), \tag{4}$$

derived using calculus of variations[33]. This result shows that by training the network on infinitely many samples from the true data generating distribution p_{data} by minimizing the MSE cost function, the network prediction is the mean of f for each s. Other possible cost functions include how well the predictions match the width and height of the peaks of the ice distribution or just focusing on the location of the peaks. Such measures will be explored in future works.

The dataset used for training the network consists of 400.000 simulated ICE signal-distribution function pairs. The dataset includes distribution functions with up to three normal distributions with random center, width, and amplitude to mimic possible ICE distributions. As an example, consider Fig. 2 but with three distributions with random locations, amplitude, and width.

The optimal network architecture is determined in a bottom-up manner by first comparing the MSE loss for a network consisting of a single layer and an output layer using the default learning rate of $\alpha = 0.0001$ for the Adam optimizer. This comparison is made using $K = 5$ cross-validation, thereby ensuring that the training and validation set is the same for all network models. The performance of a specific network architecture is determined by the mean and standard deviation of the MSE across the 5 tests. The number of nodes in the first layer is varied from 100 to 5000, and the number of nodes in the best performing network is chosen as the number of nodes in the first layer. A second layer is then added between the input and output layer and the optimal number of nodes determined in a similar manner. These calculations were performed for up to four layers excluding the output layer. See Fig. 3 for the performance of each network architecture. The optimal number of nodes in the input layer is 1300. Including a second layer improves the model, and its optimal number of nodes is 500 or 2000. Here, 500 is chosen since the mean value of the loss is slightly lower and to save computational resources. The model gets worse when including additional layers.

Similar cross-validation investigations show that adding a dropout layer improves model performance. The optimal choice of dropout fraction is 0.4 after the first layer and no dropout after the second layer. The mean value of the loss in 5-fold cross-validation is improved from 7.6×10^{-3} to 4.0×10^{-3} corresponding to an improvement of approximately 52 %.

Finally, the learning rate α is optimized by performing the learning rate range test as described by Leslie[34] and implemented by Wittmann[35]. The learning rate range test determines the values of the learning rate for which training improves the initial loss value. This range is then used to implement a triangular cyclical learning rate policy to achieve better and faster convergence of the model towards the optimal training state. The training for the range test is generally run for a low number of iterations. Here a single iteration is used with a batch size of 500. The results for the MSE loss is shown in Fig. 4. A good value of the learning rate occurs when the loss decreases. This occurs approximately for any $\alpha \in \left[10^{-5}, 2\times 10^{-3}\right]$, which is chosen to be the lower and upper bounds for the learning rate in the cyclical learning rate policy. The cyclical learning rate is implemented by using the code developed by Kenstler[36].

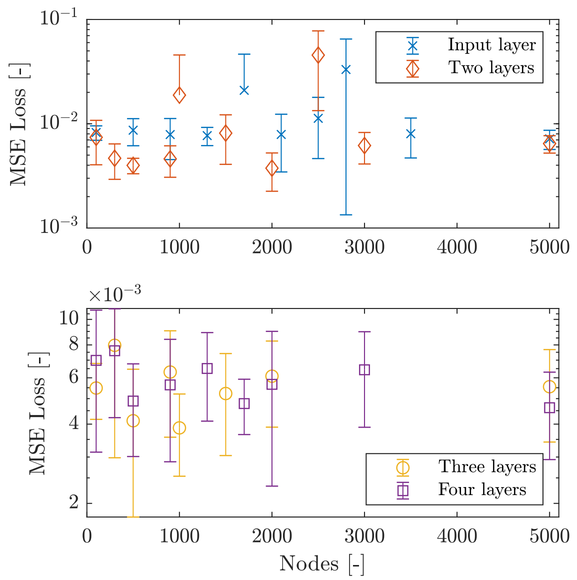

Figure 3. MSE loss of the network for a different number of layers and nodes.

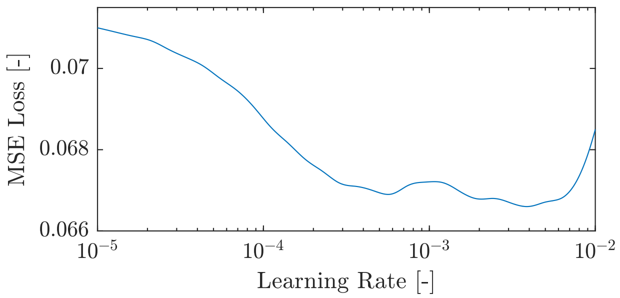

Figure 4. MSE loss from the learning rate range test to determine the optimal minimum and maximum values for the learning rate when employing a cyclical learning rate strategy.

IV. RESULTS AND DISCUSSION

Based on the investigation described in the previous section, the optimal network has the following properties. The network has three layers: one layer with 1300 nodes, a second layer with 500 nodes, and an output layer with 22 nodes. A 40 % dropout layer is placed between the first and second layer. Finally, the Adam optimizer is used in training the network with a cyclical learning rate policy with $\alpha \in \left[10^{-5}, 2\times 10^{-3}\right]$. This network is illustrated in Fig. 5.

The network is trained for 1500 iterations as the network has converged to the best possible model at this point. This is shown by the training loss curve, see Fig. 6. Two signals and their distribution function predictions by the network, the best Tikhonov inversions, and the true solutions are displayed in Fig. 7. The network finds a more accurate distribution function than Tikhonov regularization as shown by the lower MSE value, although the distribution functions are quantitatively similar.

The neural network is trained to analyse simulated ICE spectra from LHD. In order to apply the trained neural network to other ICE spectra such as ICE spectra from JET and TFTR, the neural network needs to be trained for these scenarios. Since the weight function matrix depends on the ion species, the magnetic field, and other parameters, the weight function matrix for other machines, and even for other discharges in LHD, will differ from the ones used here. However, we expect the same methods to be applicable.

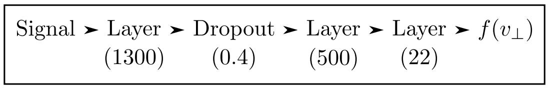

Figure 5. The structure of the neural network to obtain distribution functions from simulated ICE data.

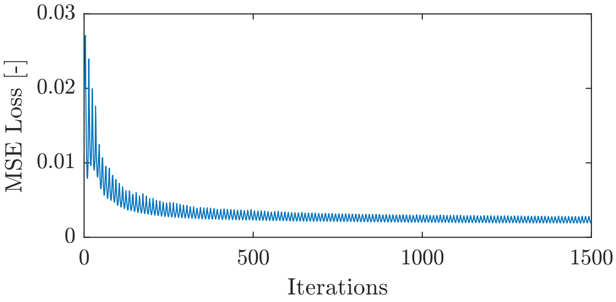

Figure 6. Convergence of the network as a function of the number of training iterations for the MSE loss using the cyclical learning rate policy.

The computation speed of the network surpasses that of the Tikhonov inversions by at least $\mathcal{O}\left(10^3\right)$: the neural network performs one prediction on the order of ms, and Tikhonov inversions on the order of a second or slower. The problem with Tikhonov regularization is the need for determining a value of the regularization parameter in order to obtain a good inversion. This either costs extra computation time or must be done manually. Computation speed is relevant for both on-site and off-site analyses since the data analysis speed is often a bottleneck in determining how to proceed with a given experiment. Fast inversions for in-between shot analysis would be a large advantage.

V. CONCLUSION

A trained deep neural network can be used to model the relationship between simulated edge ICE signals from LHD due to fast ions and their 1D velocity distribution function in $v_\perp$. The network is more accurate than Tikhonov regularization on many of the ICE signal validation examples. Application of the neural network to ICE measurements and not simulated ICE data requires specific training of the network and the sensitivity of how exactly the simulated training data must mimic the measurement data has not been studied here. Thus, criteria for when a training set is good enough needs to be investigated further. Future work also involves expanding the neural network to predict the 2D velocity distribution in $v_\perp$ and $v_\parallel$ of fast ions, or even the 3D phase-space distribution[37], from ICE signals.

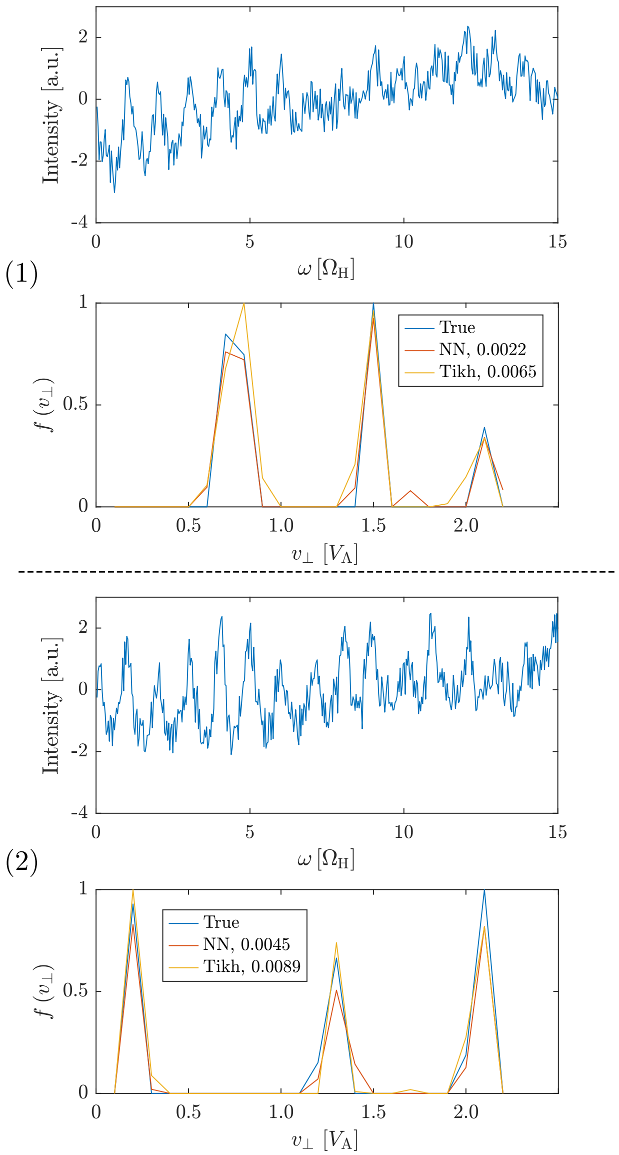

Figure 7. Two signals with the deep neural network ("NN") and Tikhonov inversion ("Tikh") predictions compared to the true solutions ("True"). The MSE is indicated. Note that the signals are normalized so that the y-axis is the normalized intensity in arbitrary units.

ACKNOWLEDGEMENTS

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 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. Also, the work received support from the RCUK Energy Programme [grant number EP/T012250/1].

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] M. Salewski et al. (2014). *Nucl. Fusion*, 54, 023005.
[2] M. Salewski et al. (2016). *Nucl. Fusion*, 56, 106024.
[3] M. Salewski et al. (2017). *Nucl Fusion*, 57, 056001.
[4] M. Weiland et al. (2016). *Plasma Phys. Control. Fusion*, 58, 025012.
[5] M. Weiland et al. (2017). *Nucl. Fusion*, 57, 116058.
[6] B. Madsen et al. (2018). *Rev. Sci. Instrum.*, 89, 10D125.
[7] B. Madsen et al. (2020). *Nucl. Fusion*, 60, 066024.
[8] B. Madsen et al. (2020). *Plasma Phys. Control. Fusion*, 62, 115019.
[9] M. Salewski et al. (2011). *Nucl. Fusion*, 51, 083014.
[10] W. W. Heidbrink et al. (2007). *Plasma Phys. Control. Fusion*, 49, pp. 1457-1475.
[11] L. Stagner and W. W. Heidbrink (2017). *Phys. of Plasmas*, 24, 092505.
[12] M. Salewski et al. (2014). *Plasma Phys. Control. Fusion*, 56, 105005.
[13] M. Salewski et al. (2015). *Nucl. Fusion*, 55, 093029.
[14] M. Salewski et al. (2016). *Nucl. Fusion*, 56, 046009.
[15] A. S. Jacobsen et al. (2015). *Nucl. Fusion*, 55, 053013.
[16] A. S. Jacobsen et al. (2017). *Rev. Sci. Instrum.*, 88, 073506.
[17] J. Galdon-Quiroga et al. (2018). *Plasma Phys. Control. Fusion*, 60, 105005.
[18] K.G. McClements et al. (2015). *Nucl. Fusion*, 55, 043013.
[19] R. Ochoukov et al. (2019). *Nucl. Fusion*, 59, 086032.
[20] L. Liu et al. (2021). *Nucl. Fusion*, 61, 026004.
[21] A J.H. Donne et al. (2007). IAEA. doi:10.1088/0029-5515/47/6/S07.
[22] M. Salewski et al. (2018). *Nucl. Fusion*, 58, 096019.
[23] R.O. Dendy et al. (1994). *Plasma Phys. Control. Fusion* 36 B163.
[24] N. N. Gorelenkov (2016). *New J. Phys.*, 18, 105010.
[25] W. W. Heidbrink et al. (2006). *Nucl. Fusion*, 46, 324.
[26] E. D. Fredrickson et al. (2001). *Phys. Rev. Lett.*, 87, 145001.
[27] L. C. Appel et al. (2008). *Plasma Phys. Control. Fusion*, 50, 115011.
[28] R. Ochokov et al. (2020). *Nucl. Fusion*, 60, 126043.
[29] J. W. Cook et al. (2013). *Plasma Phys. Control. Fusion* 55.6 065003.
[30] C.G. Darwin (1920). *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 39, 233, 537–551.
[31] R. D. Sydora (1999). *Journal Comp. Appl. Math.*, 109, 243-259.
[32] D. Moseev and M. Salewski (2019). *Phys. of Plasmas*, 26, 020901.
[33] I. Goodfellow et al. (2016), *Deep Learning*, MIT Press.
[34] L. N. Smith, *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Santa Rosa, CA, 2017, pp. 464-472.
[35] F. Wittmann, Learning Rate Finger as a Keras Callback, (2019), Github repository, `https://gist.github.com/WittmannF/c55ed82d27248d18799e2be324a79473`
[36] B. Kenstler, Cyclical Learning Rate, (2016), Github repository, `https://github.com/bckenstler/CLR`
[37] H. Järleblad et al., *Fast-ion Orbit Sensitivity of Neutron Emission Spectroscopy Diagnostics*, in press.