PAPER • OPEN ACCESS

Image mapping the temporal evolution of edge characteristics in tokamaks using neural networks

To cite this article: Vignesh Gopakumar and D Samaddar 2020 *Mach. Learn.: Sci. Technol.* **1** 015006

View the article online for updates and enhancements.

MACHINE LEARNING
Science and Technology

Image mapping the temporal evolution of edge characteristics in tokamaks using neural networks

Vignesh Gopakumar[1] and D Samaddar

Culham Centre for Fusion Energy, Oxford, United Kingdom
[1] Author to whom correspondence should be addressed.

E-mail: vignesh.gopakumar@ukaea.uk and debasmita.samaddar@ukaea.uk

Abstract

We propose a method for data-driven modelling of the temporal evolution of the plasma and neutral characteristics at the edge of a tokamak using neural networks. Our method proposes a novel fully convolutional network to serve as function approximators in modelling complex nonlinear phenomenon observed in the multi-physics representations of high energy physics. More specifically, we target the evolution of the temperatures, densities and parallel velocities of the electrons, ions and neutral particles at the edge. The central challenge in this context is in modelling together the different physics principles encapsulated in the evolution of plasma and the neutrals. We demonstrate that the inherent differences in nonlinear behaviour can be addressed by forking the network to process the plasma and neutral information individually before integrating as a holistic system. Our approach takes into account the spatial dependencies of the physics parameters across the grid while performing the temporal mappings, ensuring that the underlying physics is factored in and not lost to the black-box. Having used the conventional edge plasma-neutral solver code SOLPS to build the synthetic dataset, our method demonstrates a computational gain of over 5 orders of magnitude over it without a considerable compromise on accuracy.

1. Introduction

Simulating magnetically confined fusion plasma in a tokamak is considered to be extremely challenging particularly at the edge of the device. Both perpendicular and parallel velocities (with respect to the magnetic field) are of significant importance in this region, also referred to as the scrape off layer (SOL). In addition to that, the interaction with the machine's plasma facing components introduces a lot of impurities into the plasma. Processes such as charge exchange and recombination contribute to the presence of neutral particles in the plasma that play a significant role in the dynamics —much of which is almost absent at the plasma core [1]. The additional complexities of the physics in the SOL lead to large differences in characteristic timescales in simulations of the plasma at the edge and the core [2]. Modelling the entire plasma within the tokamak (integrated tokamak modeling [3]) thus becomes particularly challenging. We suggest a novel approach that attempts to eradicate this problem by utilising the function approximation capabilities of neural networks to model the evolution of both the plasma and neutral behaviour at the edge.

We have demonstrated the impact of our approach on the SOLPS (Scrape Off Layer Plasma Simulator) framework, a comprehensive code suite that aims to solve the edge plasma physics, as represented in figure 1 [4]. By utilising synthetic data generated by the SOLPS code, we have shown how our model is capable of mapping the evolution of both the edge plasma and the neutral parameters. We model two interlinked physical systems characterised by two different nonlinear behaviours with a computational gain of over 5 orders of magnitude compared to the traditional numerical SOLPS approach.
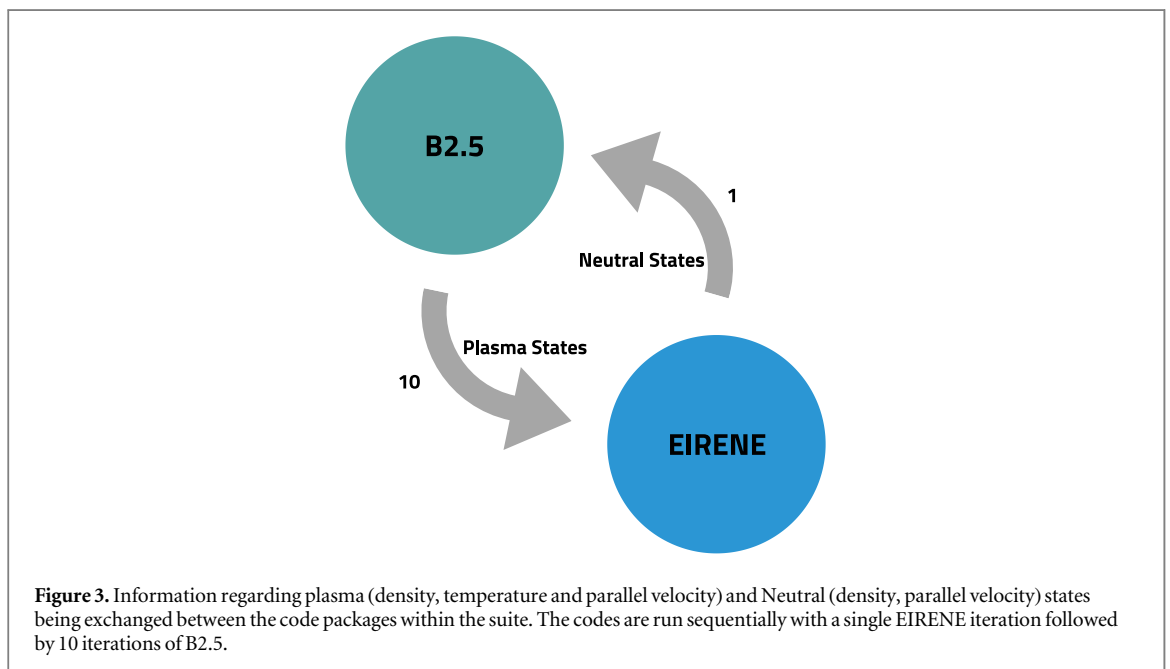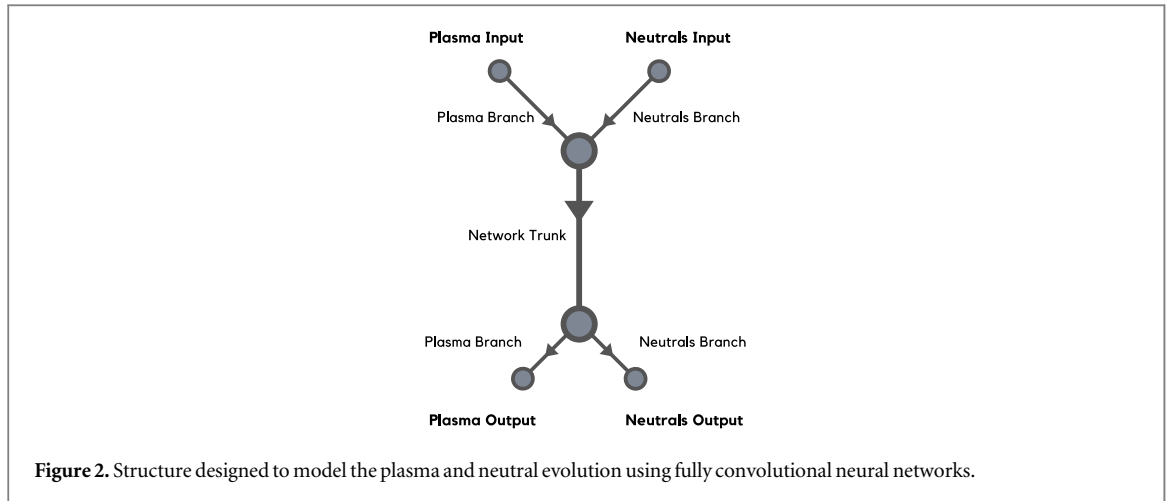
Figure 1. Mapping the temporal evolution of electron density. The set of images on top depicts the evolution from the initial state to the final state as solved by the SOLPS framework. Similarly the set in the bottom, represents the evolution characterised by our novel FCN approach.
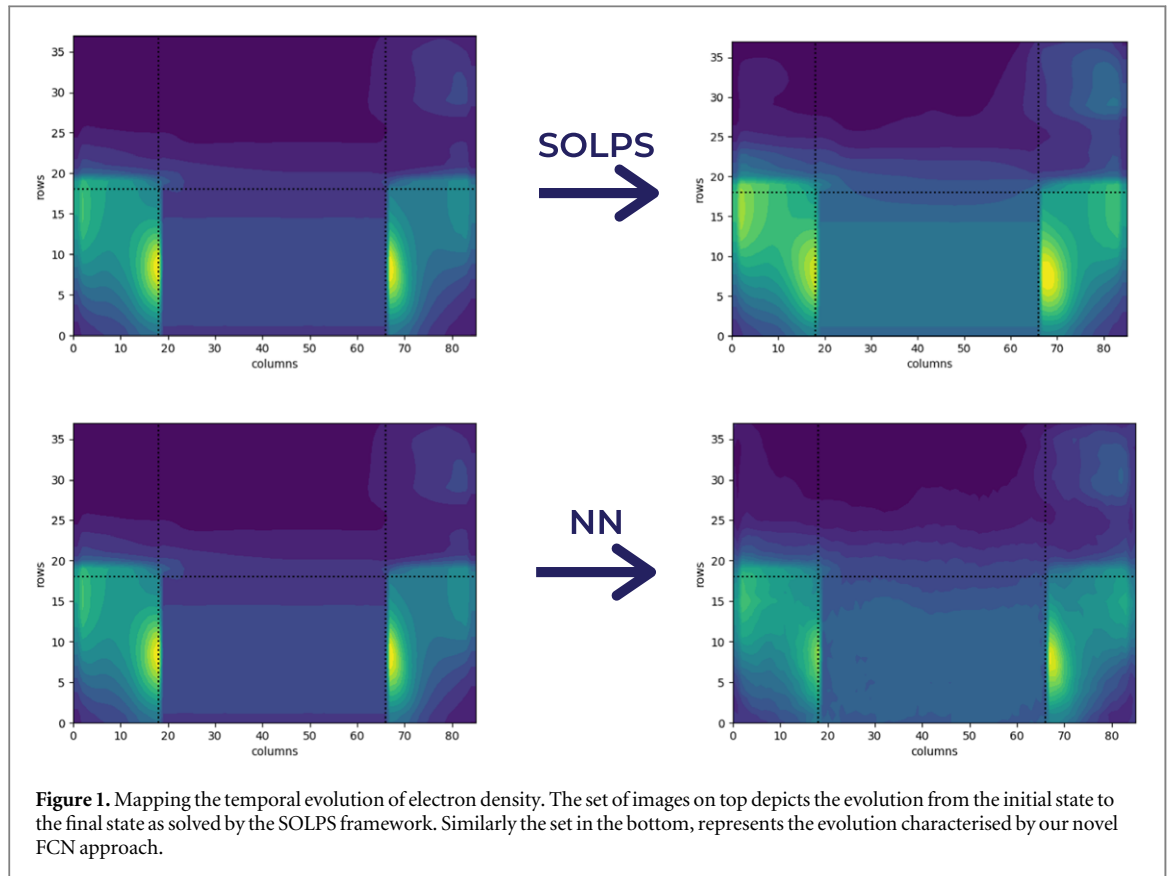
2. Concept

Neural Networks have been consistently proven to perform as 'universal function approximators' over the years, performing exceptionally well in modeling hefty nonlinear functions [5, 6]. The temporal evolution of plasma and neutrals within a specified spatial grid is described through highly complex nonlinear transformations [7]. The charged species of the plasma and the neutrals are modelled differently (as described in section 3). A time instant of the plasma and neutral evolution is characterised by various physics profiles spread across the poloidal cross-section. Each point in this grid represents the value of a physical parameter at that point in space. The evolution of this parameter in each of the grid points across time is represented by a strong nonlinear function that is dependent not only on itself but also on the nonlinearities of the physical parameters encapsulated in the neighboring grid points. To capture the two different physics models at the edge described as nonlinearities with local relational dependencies, we have designed a fully convolutional neural network with forked inputs and outputs.

In order to account for differing physics from which the plasma and neutral grids emerge from we designed a network to have bifurcated inputs and outputs as shown in figure 2. At the input, grids characterising the plasma were fed into one branch, while those of the neutrals were fed into the other. Each branch is composed of the same number of layers, performing convolutions and pooling activities on the input grids. The convoluted plasma and neutral grids would be later merged at the end of the branches and fed to the trunk of the network, where the interlinked holistic system, the plasma along with its neutral interactions will be modelled. At the output, the trunk, branches into two again, one for the plasma grids and the neutral grids. By having branched inputs and outpus, the network becomes capable of modelling both the plasma and neutral behaviour while allowing for information exchange between them.

3. SOLPS

The scrape off layer plasma simulator (SOLPS) is a state of the art code framework used to model the edge transport and study the plasma surface interactions. The SOLPS-ITER version which has been used in this research, is the basis of design of the divertor in ITER, as well as in simulating detachment and studying the plasma-wall interactions [8–10]. Edge modeling in SOLPS is performed by two sub codes within the framework : B2.5 and EIRENE [11].

Figure 2. Structure designed to model the plasma and neutral evolution using fully convolutional neural networks.

Figure 3. Information regarding plasma (density, temperature and parallel velocity) and Neutral (density, parallel velocity) states being exchanged between the code packages within the suite. The codes are run sequentially with a single EIRENE iteration followed by 10 iterations of B2.5.
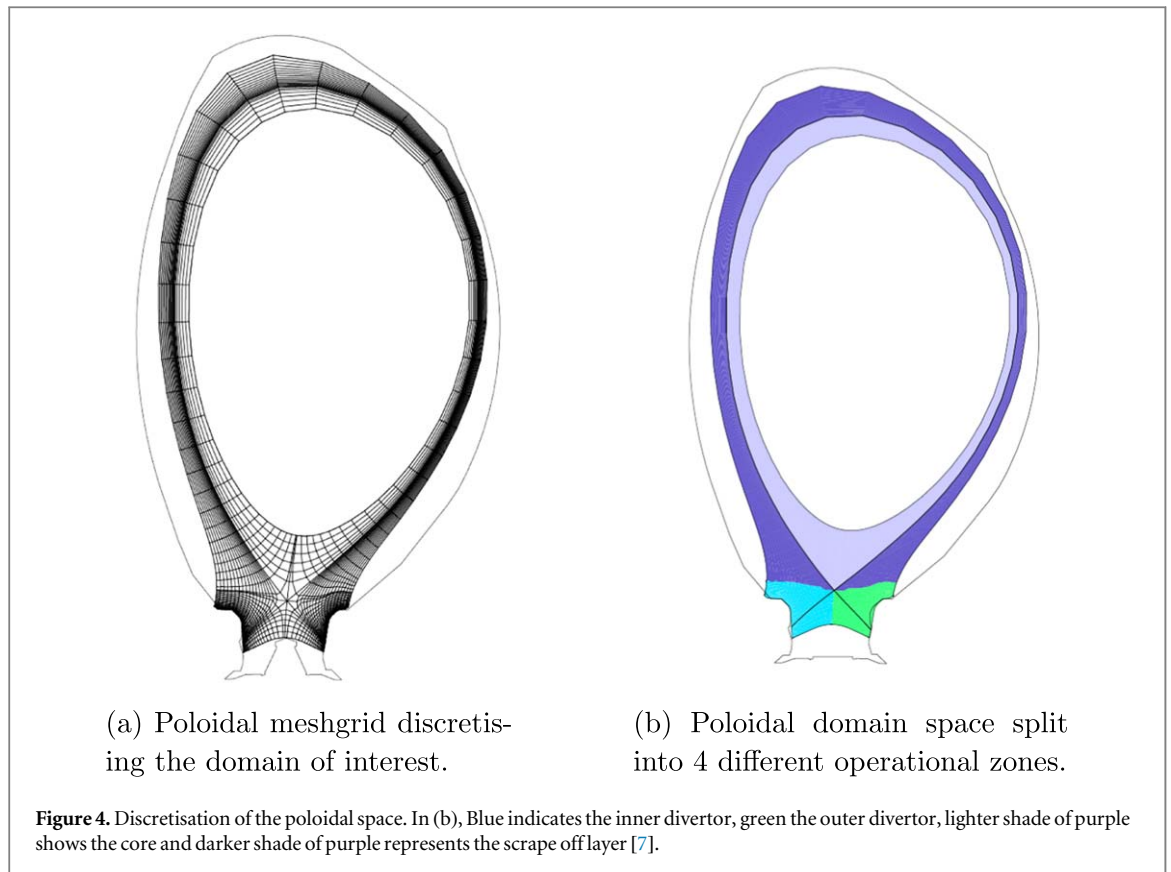
Being a multi-fluid transport code, B2.5 solves the behaviour of multi-species plasma along a 2-dimensional geometry. It models the plasma evolution by numerically solving the Braginskii equations in magnetically aligned curvilinear coordinates [7, 12]. Conservation equations of mass, momentum and energy are taken into account within the code package [11]. Modeling the plasma behaviour using B2.5 contributes consiserably to the wall clock time as it is implemented as a serial code within the SOLPS framework [13]. EIRENE is a 3-dimensional Monte Carlo transport code used to solve the evolution of neutrals by solving a multi-species set of coupled Boltzmann equations utilising a Bayesian method [14].

As the plasma and neutral evolution are modelled by solving two sets of equations encapsulating different physics principles, both B2.5 and EIRENE are run in an independent yet sequential manner as shown in figure 3. Between the two code packages information characterising the plasma and neutral state is exchanged after a predetermined number of iterations within the simulation [15, 16].

3.1. Physics parameters
Within the SOLPS framework, we have identified 7 fundamental physics parameters characterising the plasma and neutral state at any instant in time. The network models the behaviour of plasma and neutrals by mapping the evolution of these 7 parameters.

(i) Neutral density.

(ii) Ion density.

(a) Poloidal meshgrid discretising the domain of interest.

(b) Poloidal domain space split into 4 different operational zones.

Figure 4. Discretisation of the poloidal space. In (b), Blue indicates the inner divertor, green the outer divertor, lighter shade of purple shows the core and darker shade of purple represents the scrape off layer [7].

(iii) Electron density.

(iv) Ion temperature.

(v) Electron temperature.

(vi) Neutral parallel velocity.

(vii) Ion parallel velocity.

Each parameter is represented by a profile in discretised grids across the poloidal cross section of the tokamak focusing only on the edge regions as shown in figure 4. We have chosen 5 grids (ion density, temperature, parallel velocity along with electron density and temperature) to characterise the plasma state and 2 grids (neutral density and parallel velocity) to characterise the state of the neutrals.

3.2. Grid representation

The domain of interest within the SOLPS framework is the outer edges of the poloidal cross-section. For both the neutrals and plasma, the physical domain space under observation is that shown in figure 4. The domain is discretized into 3268 grid cells of varying size, whose compact image is shown in figure 4(a). Spatial coverage for each cell is determined so as to ensure that the physics parameters are consistent within each cell [7]. The meshgrid is composed of 4 parts as show in figure 4(b), with each part categorising the grid areas into different operational domains within a tokamak with lower single null geometry [7].

To perform numerical simulation and model the time evolution of the plasma and the neutrals, the discretised grid shown in figure 4 is mapped onto quadrangular grids. The DivGeo, CARRE, and Triageom packages within the SOLPS framework would rearrange the poloidal meshgrid into a rectangular meshgrid as shown in figure 5 [7].

SOLPS rearranges the poloidal meshgrid into a rectangular grid with 38 rows and 86 columns. The transformation ensures that the local relations and dependencies within the poloidal meshgrid are not lost by maintaining the adjacency of the grid cells. These rectangular grids would be fashioned into inputs and outputs for the FCN model. Each of the 7 parameters are represented by 38×86 grids as shown in figure 5. Out of the 7 grids, 5 characterises the plasma state and the remaining 2 defines the neutral state.
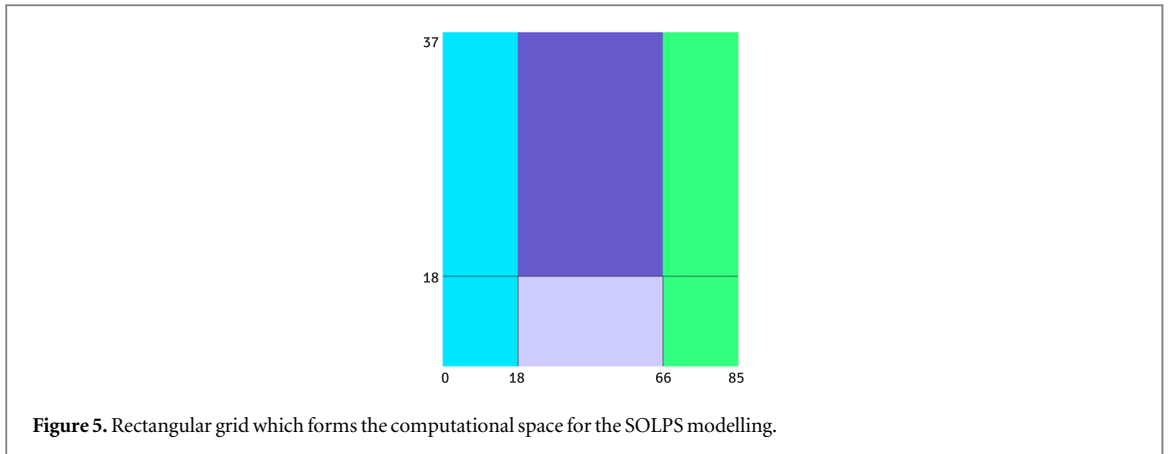
Figure 5. Rectangular grid which forms the computational space for the SOLPS modelling.

3.3. Case setup

The SOLPS simulations were performed for a case designed on an actual JET (Joint European Torus) experiment under the shot number 89241 [17]. Our case assumes toroidal symmetry within the tokamak reducing our focus only onto the poloidal cross-section. Our primary focus was to model the edge plasma and neutral behaviour when a steady state is slightly perturbed. Thus, the case was pre-run until it reaches the steady state regime. The edge in this case was characterised by three entities: Deuterium ion, it's neutral component and the electrons. Toroidal magnetic field was set at 2.2 Tesla. With a q_{95} of 3.4 and plasma current of 2.0 MA. A combination of neutral beam injection (NBI), ion cyclotron resonance heating (ICRH) and ohmic heating (OH) contributed to the total power provided to the plasma. The decision to restrict ourselves to the neighbourhood of the steady state regime was taken to ensure that the proof of concept can be built with minimal data, thus reducing the simulations and hence the time needed to build the synthetic dataset. The simulation space was setup such that the plasma was already in the H-mode [18].

3.4. Data generation

In order to create an adequately diverse dataset, the predefined case (already in steady state) was run under different physical conditions giving us data that is indicative of perturbations around the steady state. We tune three parameters that have significant influence on the plasma and neutral profiles to create these perturbed scenarios:

- Heating power—P_{in} (MW)

- Puffing rate—rate of change of plasma density (s^{-1})

- Pump intensity—percentage change of neutral density (%)

The heating power P_{in} entering the simulation grid is given by $P_{in} = NBI + ICRH + OH - Radiation$. So P_{in} incorporates all the input heating power as well as an estimated radiated power loss. The heating power P_{in} for the simulation domain is defined as a boundary condition for the SOLPS code and is specified at the edge of the grid lying in the plasma core.

The dataset was built from SOLPS in two phases. In Phase 1, the input grids that characterise the plasma and neutral state at the initial point in time are built. Phase 2, the input grids obtained in Phase 1 are simulated further to develop the output grids.

3.4.1. Phase 1

The predetermined case setup is simulated from $t = 0s$ to $t = 0.234$ s up until the simulation arrives at steady state. The simulation arrives at steady state with the parameters as indicated in table below:

Heating power	4.080 MW
Puffing rate	$10^{21} s^{-1}$
Pump intensity	94%.

The steady state configuration was treated as the input setting for further SOLPS simulations under different combinations of our chosen parameters. The parameters were discretised within a domain range and all possible

Figure 6. Phases of data generation. In Phase 1, the setup is perturbed from steady state 1 (S_1) and the edge characteristics are evolved under the new conditions. In Phase 2, the perturbed states are pushed towards a new steady state (S_2) under the same physical conditions, but they are cut-off before reaching it. The output of Phase 1 simulations at T_1 would form the initial/input grids and output of Phase 2 simulations cut off at time T_2 would form the final/output grids.

combinations of those discretised values were simulated. The discretisation of the parameter range is seen in the table below:

Parameter	Min.	Max.	Iteration	Number of values
Heating power	3.0 MW	8.5 MW	+0.5 MW	12
Puffing rate	$10^{17}\,\mathrm{s}^{-1}$	$10^{21}\,\mathrm{s}^{-1}$	×5.0	9
Pump intensity	48%	98%	+5%	11

Total number of combinations $= 12 \times 9 \times 11 = 1188$.

Each simulation was run for 1000 iterations with a time step of $\Delta t = 10^{-6}$ s, progressing the simulation by 1 ms. The transport timescale associated with that of the plasma and the neutrals at the edge are about 1 ms. Thus, by progressing the simulation by that time factor ensures that we are able to create effective data points that represent the multi-physics behaviour when perturbed away from the steady state as shown in figure 6.
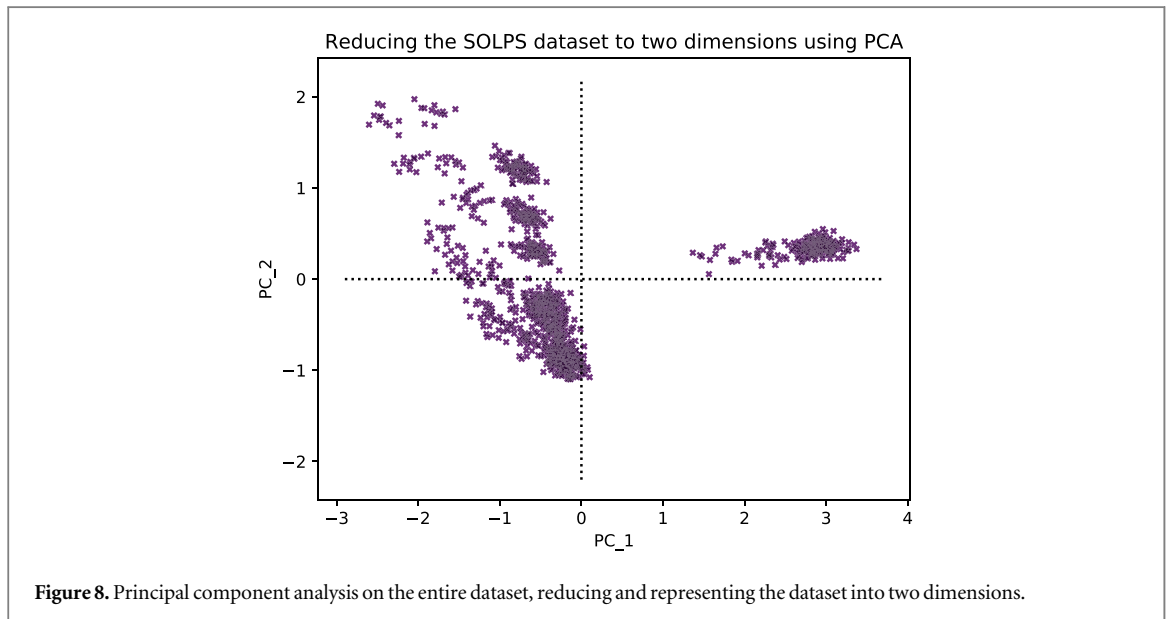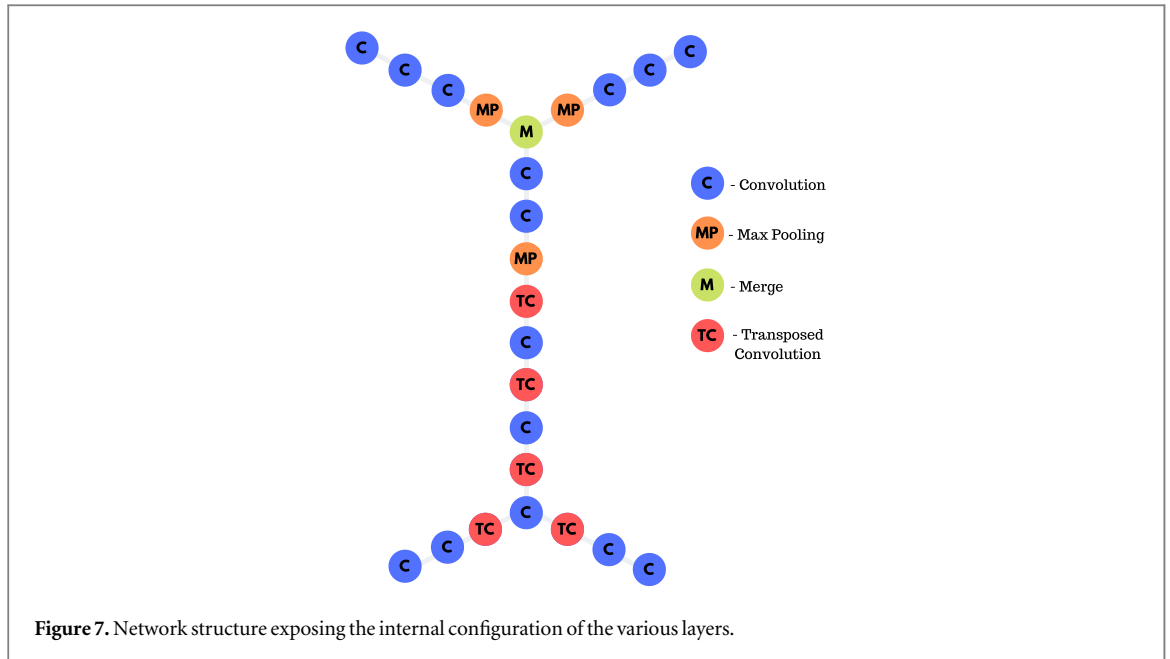
3.4.2. Phase 2
The grids generated in Phase 1 are used as inputs to build the output grids. Each of the 1188 simulated cases were run further, but this time under the same set of parameters. The heating power, puffing rate, and pump intensity would be fixed to the values described in the subsequent table and ran for a total of 2000 time iterations. With a time step of $\Delta t = 10^{-6}$ s, the output grids is 2 ms ahead of the initial grids, with SOLPS accommodating the changes introduced by the new values

$$
\begin{array}{r|c}
\text{Heating power} & 5.000 \text{ MW} \\
\text{Puffing rate} & 10^{18}\mathrm{s}^{-1} \\
\text{Pump intensity} & 94\% \\
\end{array}
$$

In Phase 1, the case having brought up to steady state at 0.234 s is perturbed by changing one or more of the three parameters mentioned in the paper and run for 0.1 ms up to time 0.235 s. In Phase II, the perturbed cases are all set with the same parameters and then run for 2 ms up to 0.237 s. In both phase I and Phase 2, you must notice that the simulation time is below the confinement time of 0.1 s thereby not arriving at steady state but creating data instances that reflect perturbations around steady state.

The output of Phase 1 simulations form the input and those of the Phase 2 simulations the output upon which we build our FCN model. This provides us with a labelled dataset upon which we can perform input–output mapping.
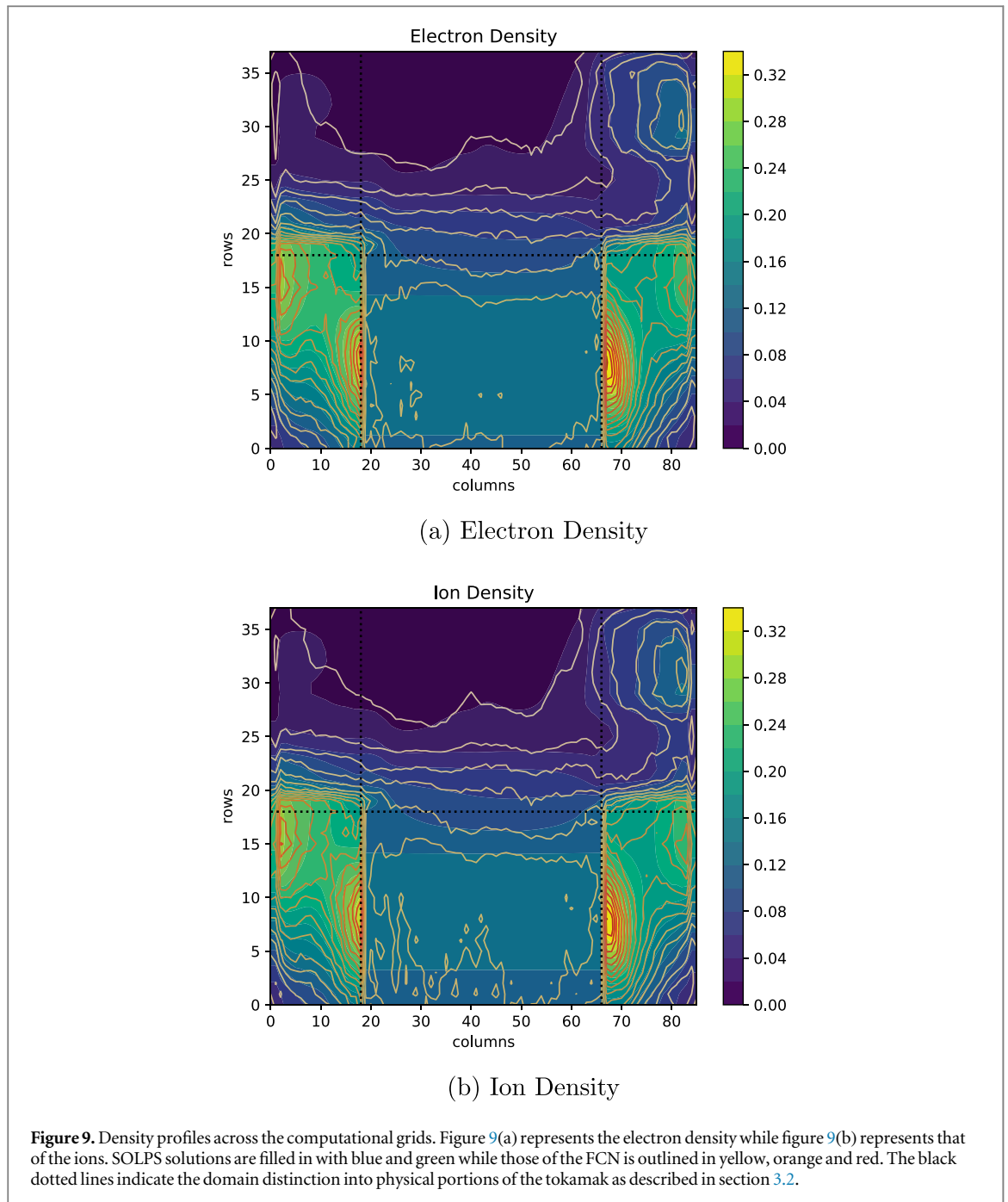
Figure 7. Network structure exposing the internal configuration of the various layers.

Figure 8. Principal component analysis on the entire dataset, reducing and representing the dataset into two dimensions.

4. Network design and training

The network was designed to engage in a routine of dimensionality reduction, that can effectively capture the interdependencies shown by the distributions of physical parameters in space [19]. Due to the spatial nature and relationships of the grids we utilised convolutions to perform dimensionality reduction. Convolutions can effectively capture the general trends found across the grid while reducing the complexity of the data [20]. We observed that while performing dimensionality reductions on the grids it was paramount to maintain the aspect ratio of the grid (refer to supplementary section 12.1.2 online at stacks.iop.org/MLST/1/015006/mmedia and references [25–29] therein). We had randomly sampled the weights and biases to from a uniform distribution between 0 and 1. To effectively model the complex nonlinearities within the plasma and neutral behaviour it was necessary to employ a network of sufficient depth [21]. It was observed that the performance of the network was optimum at 20 hidden layers with nonlinear capabilities. However, we were discouraged from building phenomenally deep networks as they displayed tendencies of strongly over-fitting to the training data, and gave us poor generalisation (refer supplementary section 12.1.1).

Since the task was to perform image mapping, we had to ensure that the dimensions of the outputs matched with those of the inputs. While we employed dimensionality reduction via convolutions in the initial stages of the network, we had to employ a series of transposed convolutions deeper within the network to obtain the

(a) Electron Density

(b) Ion Density

Figure 9. Density profiles across the computational grids. Figure 9(a) represents the electron density while figure 9(b) represents that of the ions. SOLPS solutions are filled in with blue and green while those of the FCN is outlined in yellow, orange and red. The black dotted lines indicate the domain distinction into physical portions of the tokamak as described in section 3.2.
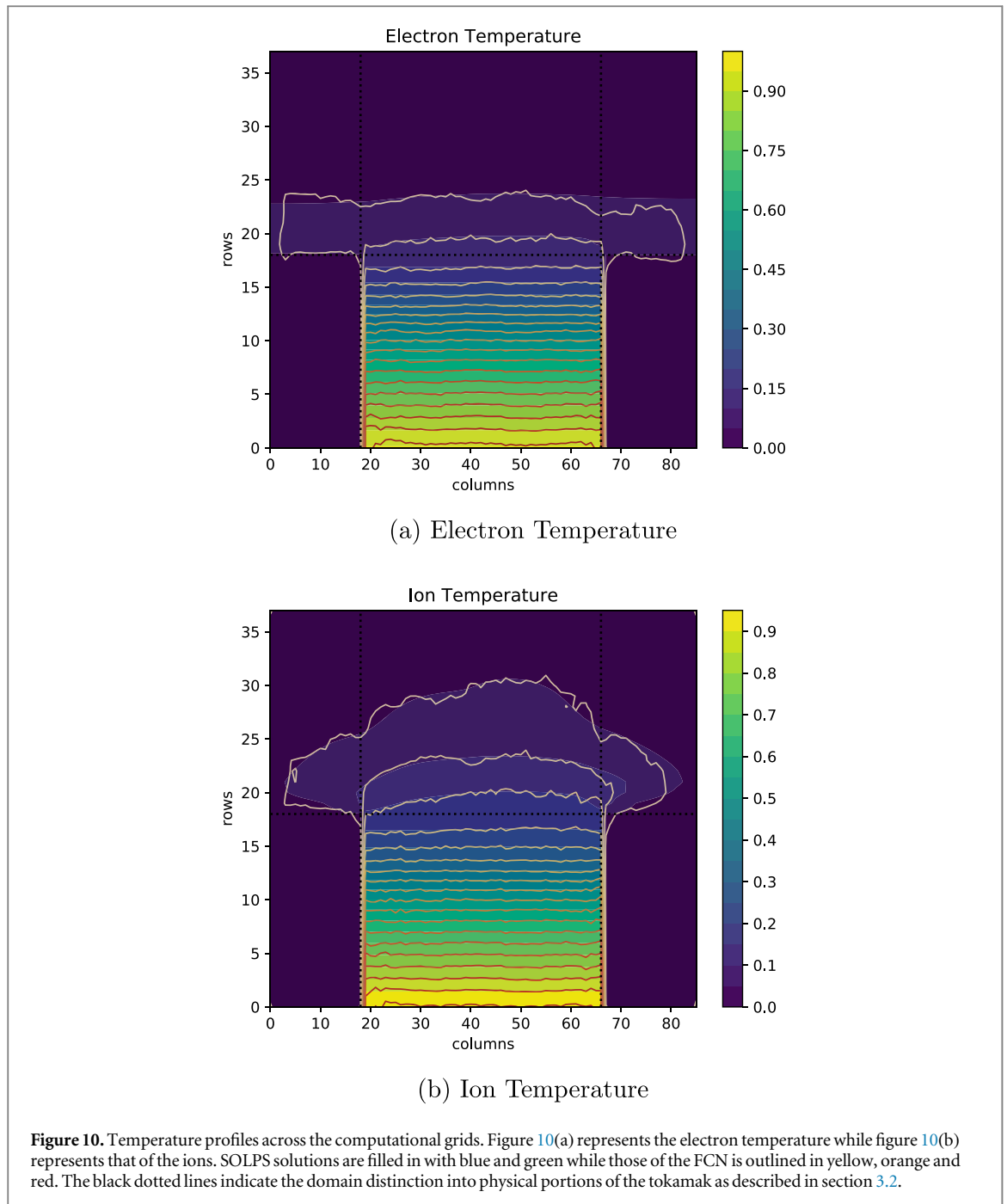
required dimensionality. The figure 7 shows the internal layout of our network. We employ 13 convolutional layers, 5 transposed convolutional layers, and 3 pooling layers to perform the modelling. Physical profiles are heavily neighbourhood dependent, and employing convolutional layers over dense layers ensured that this local connectivity could be captured. In total the network consisted of 2 289 639 trainable parameters.

The plasma input branch is fed with 5 rectangular grids of dimension 38×86, and the neutrals input consists of 2 rectangular grids of the same dimensions. Forking the inputs and outputs has been found to pick up the more salient features of the physics governing the different behaviours of plasma and neutrals. The method also expresses higher accuracy when compared to network designs with a single input with the plasma and neutral grids entangled together throughout the network.

The inspiration for the network design was taken from the architetcure of the SOLPS code itself. As seen in figure 3, within the code suite, the plasma and neutral evolutions are not solved simultaneously but rather sequentially. By employing forked inputs and bifurcated outputs, we follow a similar information structure as SOLPS, with the plasma and neutral behaviour mapped separetely with information exchange between them. This architecture also ensures that there are dedicated layers to pick up the nonlinear behaviours found within the plasma evolution and that of the neutrals, introducing more robustness to the network in modelling two different nonlinear behaviours.

(a) Electron Temperature

(b) Ion Temperature

Figure 10. Temperature profiles across the computational grids. Figure 10(a) represents the electron temperature while figure 10(b) represents that of the ions. SOLPS solutions are filled in with blue and green while those of the FCN is outlined in yellow, orange and red. The black dotted lines indicate the domain distinction into physical portions of the tokamak as described in section 3.2.
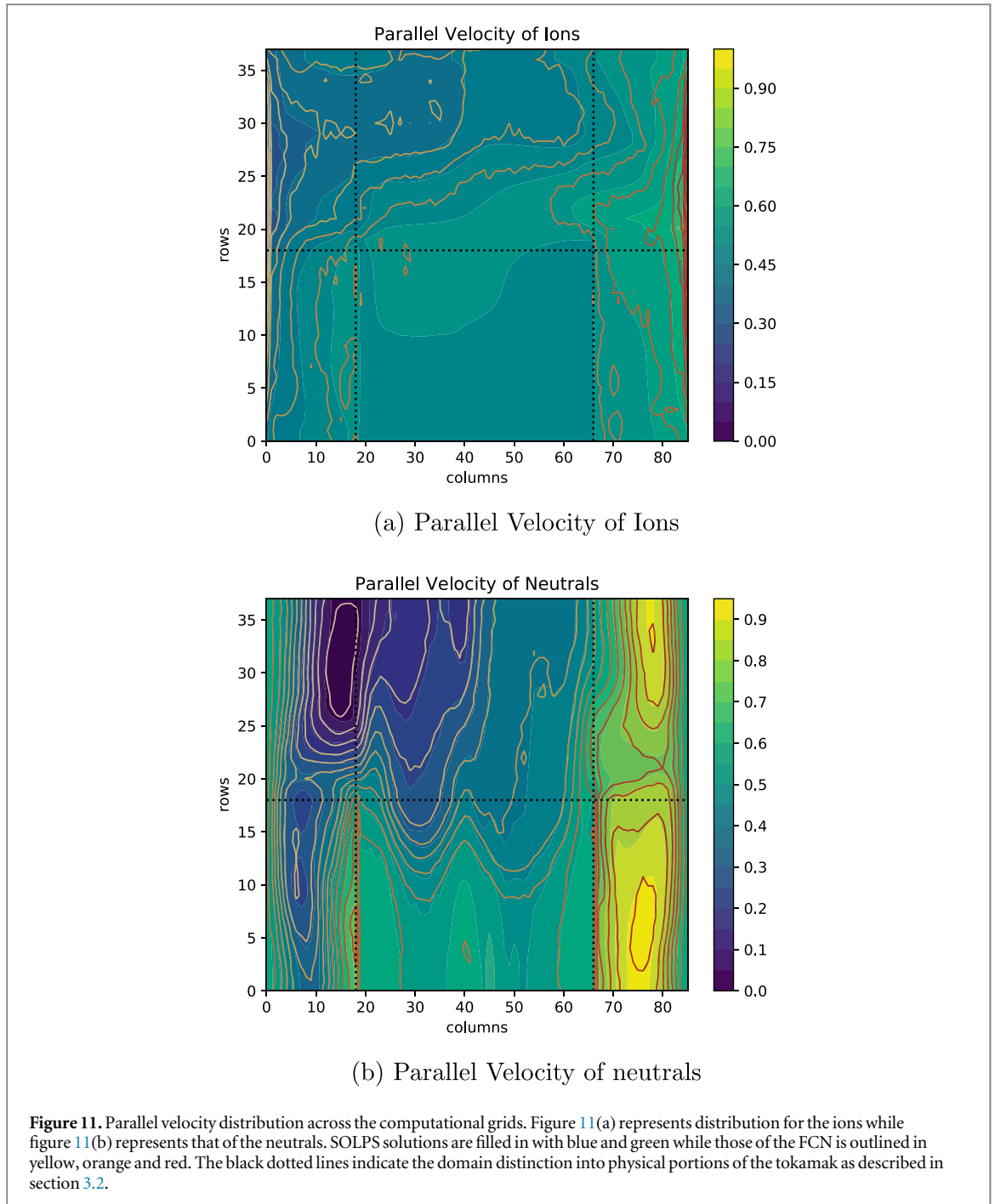
We designed, built and tested the network using the Keras API while employing tensorflow at the backend [22]. Out of the 1188 data samples as described in section 3.4, we found 22 of the SOLPS simulations to be corrupt, leaving us with 1166 data samples to train and test the model with. The grids characterising each physical parameter were normalised between 0 and 1 by taking the maximum and minimum values for each parameter across the entire dataset. The hyperparameter search showed us that the network performed comfortably optimal with the Adadelta optimizer and with mean squared error as the loss function. 20% of the entire data samples were sampled randomly and taken to be the test set leaving the rest to be split up for the training and validation. Trained over 100 iterations with a batch size of 10 samples, we observed a mean squared error of 9.2×10^{-05} across the test dataset.

5. Results

5.1. PCA
To understand the adequate diversity within the dataset and its impact on the trained network, we employed a principal component analysis across the entire dataset. By utilising a SVD approach we reduced the

(a) Parallel Velocity of Ions

(b) Parallel Velocity of neutrals

Figure 11. Parallel velocity distribution across the computational grids. Figure 11(a) represents distribution for the ions while figure 11(b) represents that of the neutrals. SOLPS solutions are filled in with blue and green while those of the FCN is outlined in yellow, orange and red. The black dotted lines indicate the domain distinction into physical portions of the tokamak as described in section 3.2.
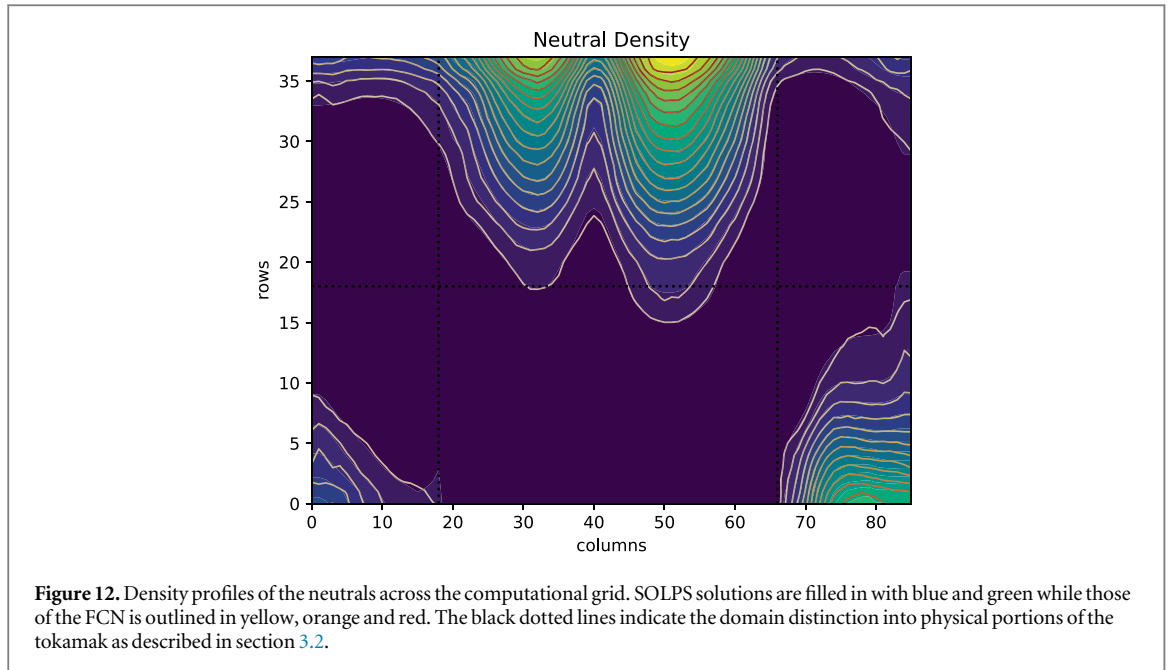
dimensionality of the dataset to two components. The principal components of the difference between the final and initial states were taken to characterise the variation within the dataset. The analysis showed that the majority of the dataset is clustered onto the bottom left quadrant as shown in figure 8. Both the test and train datasets were randomly sampled across the dataset making it inclusive of the points taken from all the quadrants. To demonstrate the ability of our FCN, the results we have shown are for a point sampled from the cluster in the top right quadrant. For a detailed breakedown of the PCA, refer to the supplementary section 11.

5.2. Output comparison—qualitative

The effectiveness of our FCN approach is best understood visually. To this end, we have over-imposed contour maps of the neural network outputs on top of the actual outputs (obtained through SOLPS). The SOLPS outputs are represented in blue and green filled contour plots while the FCN solutions are outlined with the contour lines ranging from yellow to red.

Glancing across from figures 9 to 12, it can be seen that there is a good correlation and fit between the SOLPS solutions and the FCN output. There are some discrepancies especially seen in the figures 9(a), (b) and 11(a).

Figure 12. Density profiles of the neutrals across the computational grid. SOLPS solutions are filled in with blue and green while those of the FCN is outlined in yellow, orange and red. The black dotted lines indicate the domain distinction into physical portions of the tokamak as described in section 3.2.

This is attributed to the larger variation of these parameters within the training dataset leading to the intentional build up of under-fits to ensure generalisation.

5.3. Output comparison—quantitative

The figures 9–12 gives an overview of how well the specified FCN structure can model evolution of multi-physics models as laid out by SOLPS, but gives limited information on the quantitative performance of the model. To shed some light on that we have constructed the set of plots given under figures 13 and 14.

Figures 13(a) and (b) show the SOLPS solution of the electron density ecvolution placed alongside that of the FCN solution for that parameter in a particular case. The values have been normalised to unity across the dataset and the colorbars indicate the magnitudes within this normalisation scale for the utilised case. Figure 13(c) represents the squared deviation of the FCN solution from that obtained using SOLPS.

Training stage of the FCN model involved minimising the mean squared error, the entity that highlights the deviation of the model's output from that of the SOLPS output. However in order to bear in mind the bias-variance tradeoff, we had chosen to optimise the model until the mean squared error was around 1×10^{-4}. Allowing the model to have an intuitive understanding of the physics at the same time not hampering much of it's generalisation capabilities. Constricting the model to this loss threshold provided the model to operate with a margin of error of ± 0.01.
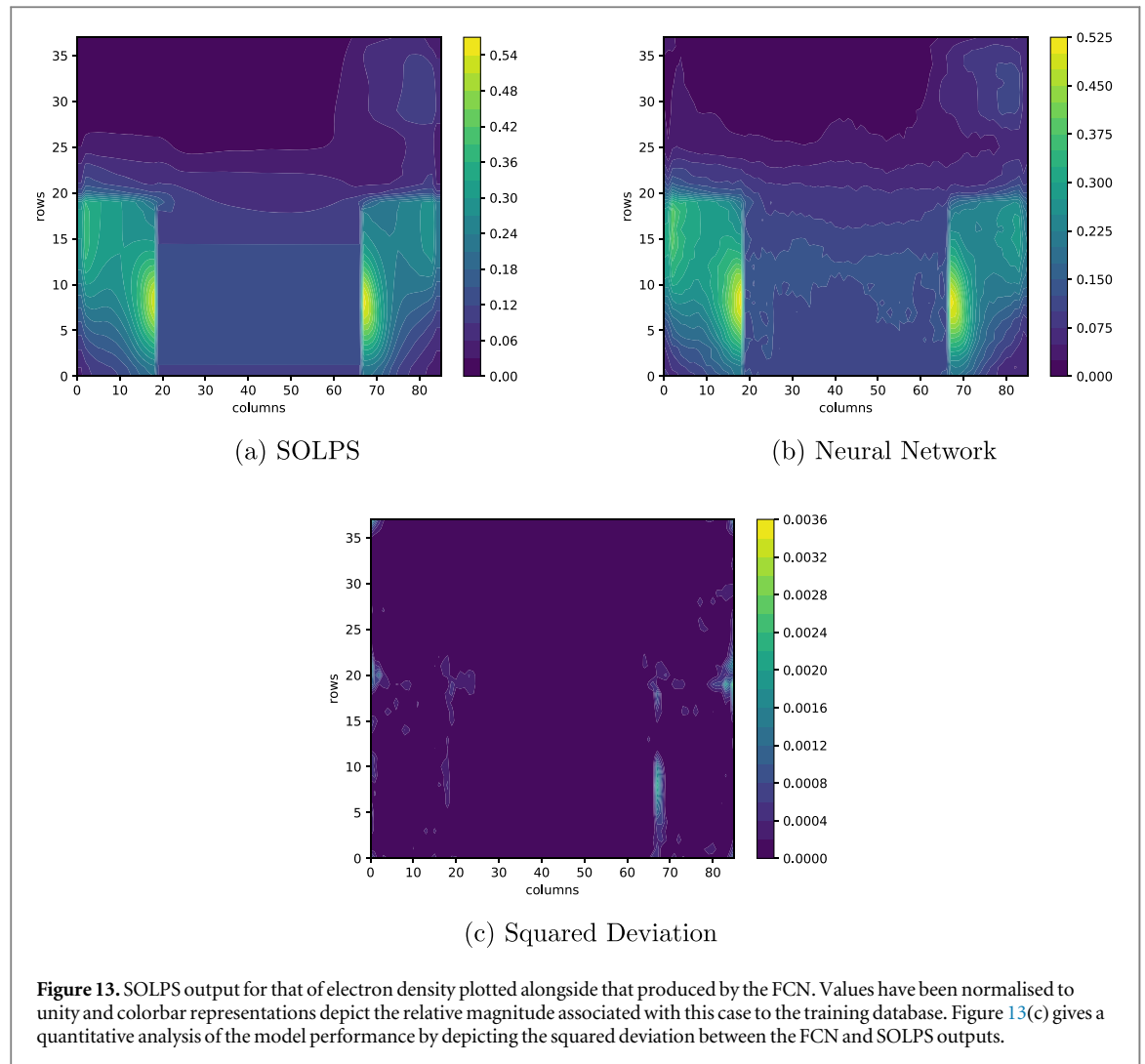
The performance of the model bearing in mind this margin of error can be illustrated using line plots. Figure 14 shows the variation of the neutral density, electron density and the electron temperature along the radial axis on the outer mideplane. From the graphs it can be seen that the FCN model is capable of modelling the evolution of the plasma and neutral behaviour within the intended margin of error.
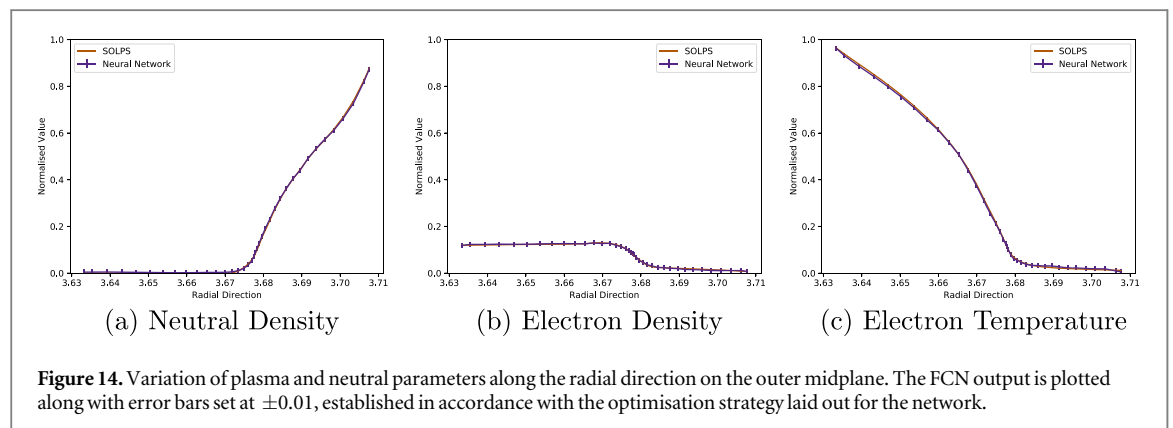
5.4. Time gain

SOLPS simulations were implemented on the Marconi cluster. Each simulation of SOLPS was run on 2×18 cores of Intel Xeon E5-2697 with each core clocking 2.3 GHz [23]. On the Marconi framework each cluster that advanced the simulation by 2 milliseconds took somewhere between 2 and 3 h (depending on the case). In contrast, the FCN took 117 min to train on a CPU arrangement of 16 cores of Intel Xeon E5-2640 v3 clocking 2.6 GHz, on a considerably less powerful machine. However, once trained the FCN could arrive at the solution in under 0.033 s on the same framework. Thus, our novel FCN approach reduces the time required to model the edge plasma and neutral evolution by more than 5 orders of magnitude while not compromising vastly on the accuracy of the output.

5.5. Limitations

Neural networks when employed for regression purposes, often comes with an issue of determinacy, mainly due to the nature of its approximateness. The higher the dimensionality of the output, the larger this issue grows. This sense of indeterminacy is inducted into the model to avoid over-fitting and produce model robustness [24]. Thus, the issue of determinacy, being a function of the dimensionality of the output is depended on 3268 values

(a) SOLPS

(b) Neural Network

(c) Squared Deviation

Figure 13. SOLPS output for that of electron density plotted alongside that produced by the FCN. Values have been normalised to unity and colorbar representations depict the relative magnitude associated with this case to the training database. Figure 13(c) gives a quantitative analysis of the model performance by depicting the squared deviation between the FCN and SOLPS outputs.

in this case. However, what we lose in precision for our model is made up by generalisation. Our models shows

(a) Neutral Density

(b) Electron Density

(c) Electron Temperature

Figure 14. Variation of plasma and neutral parameters along the radial direction on the outer midplane. The FCN output is plotted along with error bars set at ± 0.01, established in accordance with the optimisation strategy laid out for the network.

significant capability to approximate the solution and get us in the neighbourhood of the accurate output. The other limitation our approach faces is that in order to build and employ our model, we are still dependent on data generated using the SOLPS framework. Our trained model will only work within the confines of the case defined in section 3.3. With a more diverse and comprehensive database, this approach could still be used to build an approximate solver for the edge physics. While developing this approach, we could not ignore the irony that to avoid the time lag that SOLPS brings forth by using our FCN, we had to first build the data by entering into this time sink, that we are trying to avoid in the first place.

6. Conclusion

This work demonstrates a successful application of neural networks as function approximators in modelling complex nonlinear phenomena as seen in the multi-physics representations of high energy plasma physics. We show that an optimized configuration of neural networks can efficiently map the evolution of the entire plasma and neutrals at the edge of a tokamak at all points across the simulation grid. Mapping at this scale, accounting for the intricacies of multiple correlated dense nonlinear systems is performed for the first time to the best of our knowledge.

7. Future work

Currently the model only performs evolution between two fixed instants of time. We are working to build more comprehensive models that can do more progressive modelling across the time domain. The approximation capabilities of our FCN based approach can be utilised for more complex multi-species plasmas. It may be used in novel numerical methods such as the Parareal Algorithm. The Parareal algorithm employs a time parallelisation approach to better the SOLPS convergence time [15]. We hope that our FCN approach can solve the majority of the evolution within a fraction of the time with a better prediction of the final result as compared to traditional methods.

Acknowledgments

Data availability

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

ORCID iDs

Vignesh Gopakumar ⓘ https://orcid.org/0000-0003-0904-3448

References

[1] Krasheninnikov S I, Pigarov A Y, Knoll D A, LaBombard B, Lipschultz B, Sigmar D J, Soboleva T K, Terry J L and Wising F 1997 Plasma recombination and molecular effects in tokamak divertors and divertor simulators *Phys. Plasmas* **4** 1638–46

[2] Colonna G 2016 Boltzmann and vlasov equations in plasma physics *Plasma Modeling* vol 2053–2563 (Bristol: IOP Publishing) pp 1-1–1-23

[3] Giruzzi G, Garcia J, Artaud J F, Basiuk V, Decker J, Imbeaux F, Peysson Y and Schneider M 2011 Advances on modelling of ITER scenarios: physics and computational challenges *Plasma Phys. Control. Fusion* **53** 124010

[4] Wiesen S *et al* 2015 The new solps-iter code package *J. Nucl. Mater.* **463** 480–4 Plasma-surface Interactions 21

[5] Hornik K, Stinchcombe M and White H 1989 Multilayer feedforward networks are universal approximators *Neural Netw.* **2** 359–66

[6] Attali J-G and Pagès G 1997 Approximations of functions by a multilayer perceptron: a new approach *Neural Netw.* **10** 1069–81

[7] Stanojevic M 2015 Solps manual (http://solps-mdsplus.aug.ipp.mpg.de:8080/solps/Documentation/solps.pdf)

[8] Kukushkin A S, Pacher H D, Kotov V, Pacher G W and Reiter D 2011 Finalizing the iter divertor design: the key role of solps modeling *Fusion Eng. Des.* **86** 2865–73

[9] Coster D P 2011 Detachment physics in solps simulations *Proc. 19th Int. Conf. on Plasma-Surface Interactions in Controlled Fusion: J. Nucl. Mater.* **415** S545–8

[10] Wiesen S, Groth M, Brezinsek S and Wischmeier M 2016 Modelling of plasma-edge and plasma–wall interaction physics at JET with the metallic first-wall *Phys. Scr.* **T167** 014078

[11] Schneider R, Bonnin X, Borrass K, Coster D P, Kastelewicz H, Reiter D, Rozhansky V A and Braams B J Plasma edge physics with b2-eirene *Contrib. Plasma Phys.* **46** 3–191

[12] Braginskii S I 1965 Transport processes in a plasma *Rev. Plasma Phys.* **1** 205

[13] Dekeyser W, Baelmans M, Voskoboynikov S, Rozhansky V, Reiter D, Wiesen S, Kotov V and Börner P 2011 *B2-b2.5 code benchmarking* Jül-4337 Forschungszentrums Jülich

[14] Reiter D, Baelmans M and Börner P 2005 The eirene and b2-eirene codes *Fusion Sci. Technol.* **47** 02

[15] Samaddar D, Coster D P, Bonnin X, Bergmeister C, Havlíčková E, Berry L A, Elwasif W R and Batchelor D B 2017 Temporal parallelization of edge plasma simulations using the parareal algorithm and the solps code *Comput. Phys. Commun.* **221** 19–27

[16] Coster D, Chankin A and Konz C 2005 Solps modelling: inputs and outputs, and the connection between *Max Planck Institute for Plasma Physics* (Garching, Germany: EURATOM Association)

[17] Rebut P-H 2018 The joint european torus (jet) *Eur. Phys. J.* H **43** 459–97

[18] Wagner F *et al* 1984 Development of an edge transport barrier at the h-mode transition of asdex *Phys. Rev. Lett.* **53** 1453–6

[19] Venna J and Kaski S 2007 Nonlinear dimensionality reduction as information retrieval *Proc. Eleventh Int. Conf. on Artificial Intelligence and Statistics. Proc. Machine Learning Research* ed M Meila and X Shen (San Juan, Puerto Rico: PMLR) pp 572–9

[20] Vincent Dumoulin F V 2018 A guide to convolution arithmetic for deep learning stat.arXiv:1603.07285

[21] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V and Rabinovich A 2015 Going deeper with convolutions *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*

[22] Chollet F 2015 Keras: The Python Deep Learning Library (https://keras.io)

[23] Iannone F *et al* 2018 Marconi-fusion: the new high performance computing facility for european nuclear fusion modelling *Fusion Eng. Des.* **129** 354–8

[24] Lawrence S and Giles C L 2000 Overfitting and neural networks: conjugate gradient and backpropagation *Proc. IEEE-INNS-ENNS Int. Joint Conf. on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium* vol 1, pp 114–9

[25] Zeiler M D 2012 ADADELTA: an adaptive learning rate method *CoRR* arXiv:1212.5701

[26] Xu B, Wang N, Chen T and Li M 2015 Empirical evaluation of rectified activations in convolutional network *CoRR* arXiv:1505.00853

[27] Cybenko G 1989 Approximation by superpositions of a sigmoidal function *Math. Control Signals Syst.* **2** 303–14

[28] Zeiler M D and Fergus R 2014 Visualizing and understanding convolutional networks *Computer Vision—ECCV 2014* ed D Fleet *et al* (Cham: Springer International Publishing) pp 818–33

[29] Schmidhuber J 2015 Deep learning in neural networks: an overview *Neural Netw.* **61** 85–117