

## Near-realtime tokamak scenario simulation with neural networks

K.L. van de Plassche<sup>1</sup>, J. Citrin<sup>1</sup>, C. Bourdelle<sup>2</sup>, Y. Camenen<sup>3</sup>, F. J. Casson<sup>4</sup>,

V. I. Dagnelie<sup>1,5</sup>, F. Felici<sup>6</sup>, A. Ho<sup>1</sup>, and JET Contributors\*

<sup>1</sup>*DIFFER, PO Box 6336, 5600 HH Eindhoven, The Netherlands*

<sup>2</sup>*CEA, IRFM, F-13108 Saint-Paul-lez-Durance, France*

<sup>3</sup>*CNRS, Aix-Marseille Univ., PIIM UMR7345, Marseille, France*

<sup>4</sup>*CCFE, Culham Science Centre, Abingdon, UK*

<sup>5</sup>*ITP, Utrecht University, Princetonplein 5, 3584 CC Utrecht, The Netherlands*

<sup>6</sup>*EPFL-SPC, CH-1015, Lausanne, Switzerland*

\*See the author list of E. Joffrin et al. accepted for publication in Nuclear Fusion Special issue 2019, <https://doi.org/10.1088/1741-4326/ab2276>

### Introduction

Accurate prediction of turbulent transport fluxes is essential for interpretation of current-day fusion experiments, designing future devices, and optimization of plasma scenarios. Turbulent transport in the plasma core is well-described by quasilinear theory in extensive parameter regimes, and is the basis of reduced transport models. Application of quasilinear transport models, integrated within a tokamak simulation suite, are used to predict temperature, density, and rotation profiles in fusion devices. Such an integrated model, with extensive validation on JET plasmas in particular, is the QuaLiKiz quasilinear gyrokinetic transport model [1, 2], within the JINTRAC tokamak simulation suite [3]. This combination is able to evolve plasma profiles over a JET discharge timescale within a few days walltime, parallelized over 16 cores.

While fast enough for increasing our understanding of current experiments, and extrapolating to future scenarios and devices, this workflow is too slow for high throughput demanding applications, such as systematic scenario optimization, large-scale model validation, and control-oriented modeling. The bottleneck in these simulations are usually the turbulent flux predictions. To accelerate the modelling while minimizing the sacrifice of model accuracy, we apply feed-forward neural networks (FFNNs) as a surrogate model. Once trained, the FFNN can reproduce the underlying reduced model within tens of microseconds.

### Training data generation

Neural networks are universal approximators and hence a powerful tool for regression [4]. In this work we apply neural networks to a supervised regression problem, in which we reproduce the input-output mapping of the QuaLiKiz code. We constrain the input space dimensionality to a subset known to have a large impact on turbulent transport. These input dimensions include the ion temperature gradient ( $R/L_{T_i}$ ), electron temperature gradient ( $R/L_{T_e}$ ), density gradient ( $R/L_n$ ), ion-electron temperature ratio ( $T_i/T_e$ ), safety factor ( $q$ ), magnetic shear ( $\hat{s}$ ), local inverse aspect ratio ( $r/R$ ), collisionality ( $\nu^*$ ), and effective charge ( $Z_{eff}$ ), with a carbon impurity and deuterium main ion. This significantly extends a previous proof-of-principle 4D neural network QuaLiKiz regression [5].

A database consisting of  $3 \times 10^8$  QuaLiKiz input-flux relations was generated with HPC resources, using 1.3 MCPUh. The database spans ion scales ( $k_\theta \rho_s \leq 2$ ) and electron scales ( $k_\theta \rho_s > 2$ ) and contains contributions to transport fluxes  $q$ ,  $\Gamma$ ,  $D$ , and  $V$  per species and per mode (ETG/ITG/TEM). The input space was chosen as a rectangular, non-uniform 9-dimensional grid. The bounds and spacing of the grid covers dimensionless parameter regimes typically encountered in the core of standard aspect-ratio present-day tokamaks, and future devices such as ITER and DEMO. See Table 1.

### Physics-based neural network training

Regularized neural networks provide a smooth regression of supplied training data. It does not assume any features of the underlying mapping. Physics-informed features can be directly implemented into the training methodology to significantly improve the fidelity of the surrogate transport model. For our application, the desired features are sharp flux discontinuities at critical (temperature) gradients of the underlying instabilities, as well as an identical critical gradient for all transport channels driven by a single (TEM/ITG) instability.

### Training targets

To avoid unphysical results in integrated modelling, we forced identical critical thresholds for all transport channels by a careful choice of training targets. These were separated into a 'leading flux' (ion heat flux for ITG and electron heat flux for TEM/ETG) and, then the ratio of the transport flux of interest, e.g.  $q_e/q_i$  for ITG electron heat flux. The ratio-predicting neural network is then multiplied with the leading flux, forcing the prediction of the flux of interest to be at exactly the same spot as the leading flux network.

### Cost function

A neural network is a series of nested nonlinear functions (e.g. a sigmoid or ReLU) linked with weights and biases which are free parameters. Training a neural network means optimizing the weights and biases of the network to minimize a cost function  $C$ , which typically compares for each set of inputs, the neural network output to desired targets - in our case the QuaLiKiz input-output mapping. Typically the cost function consists of a measure of goodness-of-fit, and a regularizing term. For our training we only apply the

Table 1: 9D hyperrectangle bounds and number of points of the QuaLiKiz neural network training set.

variable	# points	min	max
$k_\theta \rho_s$	18	0.1	36
$-\frac{R}{T_e} \frac{dT_e}{dr}$	12	0	14
$-\frac{R}{T_i} \frac{dT_i}{dr}$	12	0	14
$-\frac{R}{n} \frac{dn}{dr}$	12	-5	6
$q_x$	10	0.66	15
$\hat{s}$	10	-1	5
$r/R$	8	0.03	0.33
$\frac{T_i}{T_e}$	7	0.25	2.5
$v^*$	6	$1 \times 10^{-5}$	1
$Z_{eff}$	5	1	3
Total	$3 \times 10^8$	$\approx 1.3$ MCPUh	

measure of goodness-of-fit on the turbulent unstable fluxes, i.e. when the QuaLiKiz flux sample  $QLK_i \neq 0$ . This provides sharp transitions at thresholds. To then avoid any possible FFNN extrapolation to non-zero fluxes in the stable region, we add an additional cost  $C_{stab}$  for samples predicted to be stable by QuaLiKiz. Any negative heat fluxes (and associated particle fluxes) predicted by the network are then clipped to zero in the transport code implementation. The free parameters  $\lambda_{regu}$ ,  $\lambda_{stab}$ , and  $C_{stab}$ , as well as other hyperparameters like network topology, are then optimized using a simple grid search. To test generalization, the dataset is split in a test set of 5% never seen during training, and a validation set of 5% used during training to avoid overfitting on training data. The remainder is used as training set. So, for each network prediction  $NN_i$  we have for all  $n$  samples and  $k$  weights:

$$C = \begin{cases} \frac{1}{n} \sum_{i=1}^n (QLK_i - NN_i)^2 + \lambda_{regu} \sum_{i=1}^k w_i^2, & \text{if } QLK_i \neq 0 \\ \frac{\lambda_{stab}}{n} \left( \sum_{i=1}^n NN_i - c_{stab} \right) + \lambda_{regu} \sum_{i=1}^k w_i^2, & \text{if } QLK_i = 0 \end{cases} \quad (1)$$

### Rotation rule

To save computation time, the dataset was ran without rotation. We add rotation in postprocessing using a rule based on new linear GENE scans[6] of  $q$ ,  $\varepsilon$ , and  $\hat{s}$ . The rule scales all ion-scale fluxes with  $f_{rot}(q, \hat{s}, \varepsilon)$ [6] and rotationless maximum growth rate  $\gamma_0$  which is predicted by a NN based on the HPC-generated QuaLiKiz database. The rule includes both  $E \times B$  stabilisation and PVG destabilisation effects.

$$f_{rotrule} = c_1 q + c_2 \hat{s} + c_3 / \varepsilon - c_4 \quad (2)$$

$$f_{rot} = \max(1 + f_{rotrule} \gamma_E / \gamma_0, 0) \quad (3)$$

$$x_{i/e, ITG/TEM} = f_{rot} * x_{i/e, ITG/TEM} \quad (4)$$

### Application in transport codes

The QuaLiKiz neural networks have been implemented in JINTRAC and RAPTOR[7, 8] as the QLKNN10D transport model. First we compare the full QuaLiKiz versus QLKNN10D on the high performance baseline JET shot #92436[9] in JINTRAC, shown in figure 1. The was run from 10s to 12s. Comparison with experimental data is given by 1-sigma bounds of a Gaussian Process fit. Correspondence between full-QuaLiKiz and QLKNN10D is close. The key point is that the QLKNN10D run took only 0.5 CPUh on two cores, while the QuaLiKiz baseline took 112 CPUh on 16 cores, where in the QLKNN10D case the transport model itself was no longer the bottleneck. Remaining disagreements between full-QuaLiKiz and QLKNN10D are due to the reduced feature space in the neural network training (e.g. impurity content), and to the different way of treated rotation impact.

Finally, we show a simulation in JINTRAC of the high performance JET scenario subject to DT extrapolation in upcoming campaigns, #92398[10]. The QuaLiKiz simulation - run with predictive particles but interpretive momentum - took 5CPUh on 16 cores, while the QLKNN9D run took 420 CPUs on two cores. We note again the excellent agreement between the models in Figure 2.

## Conclusion

We have trained physics-based neural networks as turbulent transport models. We have shown the application of such a neural network as a surrogate transport model within JINTRAC to predict the temperature and density evolution of JET fusion plasmas, in excellent agreement with the original QuaLiKiz model, yet up to four orders of magnitude faster. This allows us to simulate JET plasmas at a speed that is unprecedented for first-principle based transport simulations, opening up new avenues for tokamak scenario optimization and realtime control applications.

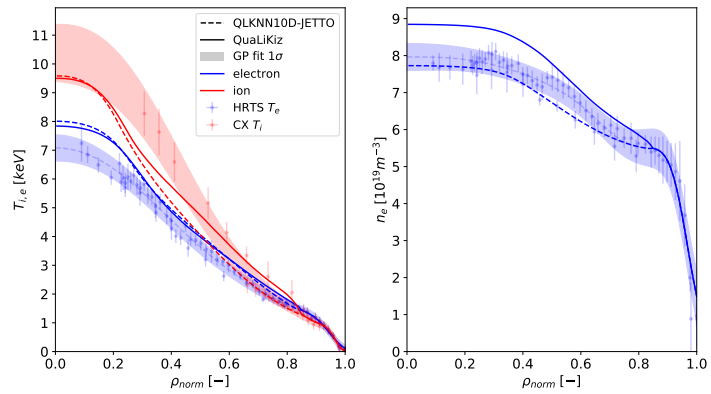


Figure 1: *JINTRAC* simulations with *QLKNN10D* or full-*QuaLiKiz* as the turbulence transport model of JET #92436 flat-top after 2 seconds of simulation. For settings of the *JINTRAC-QuaLiKiz* run, see Ref. [9]. Relative root mean square  $\sqrt{\frac{\sum_{i=0}^n (QLK_i - N\hat{N}_i)^2}{\sum_{i=0}^n QLK_i^2}}$  of  $T_e = 2\%$ ,  $T_i = 11\%$ ,  $n_e = 12\%$  for  $0.212 < \rho < 0.85$ .

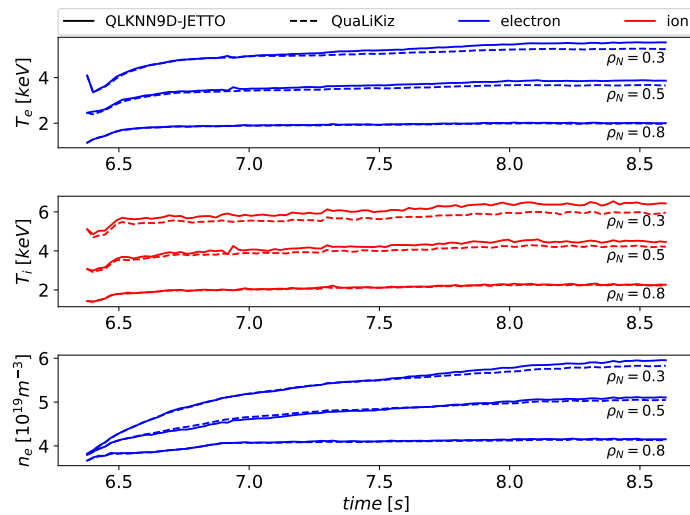


Figure 2: A time-dependent *JINTRAC-QLKNN9D* (no rotation) simulation of JET #92398. Note the density buildup that is very well captured by *QLKNN*. RRMS of  $T_e = 4\%$ ,  $T_i = 5\%$ ,  $n_e = 1\%$  at  $\rho = 0.5$

[1] S. Breton *et al.* 2018 Nucl. Fusion **58** 096003 [2] J. Citrin *et al.* 2017 PPCF **59** 064010 [3] M. Romanelli *et al.* 2014 Plasma and Fusion research **9** 3403023 [4] S. Haykin 1998 Neural Networks: A Comprehensive Foundation, ISBN 0132733501 [5] J. Citrin *et al.* 2015 Nucl. Fusion **55** 092001 [6] V.I. Dagnelie *et al.* 2017, MSc thesis, Utrecht University [7] F. Felici *et al.* 2012 PPCF **54** 025002 [8] F. Felici *et al.* 2018 Nucl. Fusion **58** 096006 [9] A. Ho *et al.* 2019 Nucl. Fusion **59** 056007 [10] F.J. Casson *et al.* to be submitted to Nucl. Fusion

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 and from the RCUK [grant number EP/P012450/1]. The views and opinions expressed herein do not necessarily reflect those of the European Commission.