Applying neural networks for tokamak plasma turbulence predictions

A. Ho[1], J. Citrin[1], C. Bourdelle[2], Y. Camenen[3], F. Felici[4], M. Maslov[5], K.L. van de Plassche[1,4], H. Weisen[6] and JET Contributors*

EUROfusion Consortium, JET, Culham Science Centre, Abingdon, UK

[1] *DIFFER - Dutch Institute for Fundamental Energy Research, Eindhoven, The Netherlands*

[2] *CEA, IRFM, Saint Paul Lez Durance, France*

[3] *IIFS/PIIM, CNRS - Université de Provence, Marseille, France*

[4] *Eindhoven University of Technology, Eindhoven, The Netherlands*

[5] *EURATOM-CCFE Fusion Association, Culham Science Centre, Abingdon, UK*

[6] *Swiss Plasma Center, EPFL, Lausanne, Switzerland*

* *See the author list of Overview of the JET results in support to ITER by X. Litaudon et al. accepted for publication in Nuclear Fusion*

The objective of this study is to train a neural network to emulate turbulent transport predictions of benchmarked gyrokinetic simulation codes [1]. One limitation in this modelling approach is to balance the two conflicting constraints of maximizing the training set size and minimizing the compuational cost of generating it, as gyrokinetic codes are expensive to execute [2]. In order to overcome this, it was decided that the simulation inputs can be derived from previous experimental measurements, such as those of JET.

The gyrokinetic code to be used in this study is QuaLiKiz [3], a quasi-linear gyrokinetic code. The inputs of interest in this simulation code are typically the densities and temperatures of the various particle species in the plasma and their gradients, as they drive the growth of linear instabilities in the plasma. Other required parameters include the safety factor, magnetic shear, toroidal flow velocity, effective ion charge, magnetic pressure, and flux surface shape parameters. Due to the requirement for multiple gradient quantities, the experimental profiles must be determined by using a fitting procedure in order to extract them. Afterwards, all relevant data for the simulation input and output validation are saved within a profile database.

For this application, a Gaussian process regression (GPR) technique was chosen as the fitting procedure, since it provides an error estimate on the fit derivative as well. These error estimations provide a more rigourous definition of the parameter range that needs to be covered by the gyrokinetic simulations while simultaneously adhering to scenarios described by the experimental data. A brief overview of the GPR is provided, limited in notation to the one-dimensional case as it is sufficient for the purposes of kinetic profile fitting.

The GPR technique is similar to that of a least-squares regression, except that it makes use of the so-called *kernel trick* in order to employ an infinite set of basis functions in the fit. By using this trick, the predicted values of the model can be expressed solely in terms of a *kernel*,

denoted as $k(x_1,x_2)$, as follows [4] [5]:

$$\begin{aligned} Y_* &= K(X_*,X)\left[K(X,X)+R(X,X)\right]^{-1} Y \\ \sigma^2_{Y_*} &= K(X_*,X_*)+R(X_*,X_*)-K(X_*,X)\left[K(X,X)+R(X,X)\right]^{-1} K(X,X_*) \\ R(x_1,x_2) &= r(x_1)\,r(x_2)\,\delta(x_1-x_2) \end{aligned} \tag{1}$$

where (X,Y) are the discrete set of data points input into the GPR, (X_*,Y_*) are the discrete set of data points to be predicted by the GPR, K represents the kernel evaluated at a set of discrete points, $r(x)$ represents the standard deviation of the data, and δ is the Kronecker delta. Note that $r(x)$ is found using a GPR on the data set, (X,Σ_Y), where Σ_Y represents the standard deviation of the Y-values [5].

Additionally, provided that the derivatives of the kernel can be calculated, the derivatives of the fit can also be predicted directly from the data using the following equations [6]:

$$\begin{aligned} Y'_* &= \frac{\partial K(X_*,X)}{\partial X_*}\left[K(X,X)+R(X,X)\right]^{-1} Y \\ \sigma^2_{Y_*} &= \frac{\partial^2 K(X_{*,1},X_{*,2})}{\partial X_{*,1}\,\partial X_{*,2}}+\frac{\partial^2 R(X_{*,1},X_{*,2})}{\partial X_{*,1}\,\partial X_{*,2}}-\frac{\partial K(X_*,X)}{\partial X_*}\left[K(X,X)+R(X,X)\right]^{-1}\frac{\partial K(X,X_*)}{\partial X_*} \end{aligned} \tag{2}$$

It should be noted that a numerical approximation of the derivative suffices if an analytical form of the kernel derivatives do not exist but this comes with a cost in computation time, depending on the accuracy of the numerical method used.

Within the GPR framework, the free weight parameters are replaced with the *hyperparameters* of the chosen kernel, denoted as a set using θ. The optimal value for these hyperparameters can be obtained by maximizing the *log-marginal-likelihood*, given as follows [4]:

$$\log p(Y|X) = -\frac{1}{2}Y^T L^{-1} Y - \frac{1}{2}\log\left|L^{-1}\right| - \frac{n}{2}\log 2\pi, \qquad L = K(X,X,\theta)+R(X,X) \tag{3}$$

where n is the number of data points to be fit. By maximizing this value, the chosen model is the most probable match to the given data points but offers no insight into how well the model corresponds to the underlying physics. Note that this is only one way of optimizing the hyperparameters and that other methods provide models that have different relations to the data.

In the case of profile fitting, X represents the radial coordinate taken from the magnetic equilibrium calculated by EFIT and Y represents the corresponding measured value. The program first tries fitting using the *rational quadratic (RQ) kernel*, which can be expressed as follows [4]:

$$k(x,x') = \sigma^2\left(1+\frac{(x-x')^2}{2\alpha l^2}\right)^{-\alpha} \tag{4}$$

where the hyperparameters are given by $\theta=\{\sigma,\alpha,l\}$. This kernel enforces smoothness when the l-hyperparameter is sufficiently large. However, it was noticed that this kernel consistently

fails to model the plasma pedestal when it is present. Thus, when a large jump is detected within $\rho > 0.8$, the *Gibbs kernel with a Gaussian length-scale function* is applied to the data instead. This kernel can be expressed mathematically as such [4]:

$$k(x,x') = \sigma^2 \left(\frac{2l(x)\,l(x')}{l^2(x)+l^2(x')} \right)^{\frac{1}{2}} \exp\left(\frac{(x-x')^2}{l^2(x)+l^2(x')} \right), \quad l(x) = l_0 - l_1 \exp\left(\frac{(x-\mu)^2}{2\sigma_l^2} \right) \quad (5)$$

where the hyperparameters are given by $\theta = \{\sigma, l_0, l_1, \mu, \sigma_l\}$. This kernel acts similarly to the RQ kernel, except with a localized variability in the correlation lengths defined by $l(x)$. The Gaussian shape of the l-function effectively allows the fit to vary significantly in a localized region around μ, specifically chosen to be near the pedestal in this application. Samples of the output from this profile fitting procedure are shown in Figure 1.

Using this tool, the profiles from a selection of $\sim$ 12000 time windows over 2000 JET discharges have been extracted, from which gyrokinetic input parameters can be sampled. Figure 2 provides some insight into the parameter space covered by the preliminary profile database. The unusually high density of points near the zeroes of the axes, shown in Figure 2, along with the extremal outliers indicate that the automated GPR fitting procedure still experiences some failures particularly in the ramp-up and ramp-down phases. However, it can be seen that the tool is robust enough to handle the majority of JET discharges.

Overall, an autonomous tool was successfully developed to extract data from the JET PPF system and apply the Gaussian process regression procedure to measurement data in order to determine the kinetic profiles and their derivatives. The capability to do this on-demand at relatively low computational cost in unprecedented. However, the statistical rigour of the derivative errors must still be formalized. In parallel to this, the workflow of generating gyrokinetic simulation inputs from the profile database is under development. This is intended to facilitate the execution of a large number of simulations on high-performance computing clusters, in order to generate the required training set for the neural network. The final neural network model will be integrated into the RAPTOR fast tokamak simulator [7] for validation with experimental data.

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 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

References

[1] J. Citrin et al., Nuclear Fusion **55**, 9 (2015)
[2] K.L. van de Plassche, P2.182, this conference
[3] C. Bourdelle et al., Plasma Physics and Controlled Fusion **58**, 1 (2016)
[4] C.E. Rasmussen and C.K.I. Williams, *Gaussian Processes for Machine Learning*, (2006)
[5] K. Kersting, *Most Likely Heteroscedastic Gaussian Process Regression*, (2007)
[6] A. McHutchon, *Differentiating Gaussian Processes*, (2013)
[7] F. Felici and O. Sauter, Plasma Physics and Controlled Fusion **54**, 2 (2012)

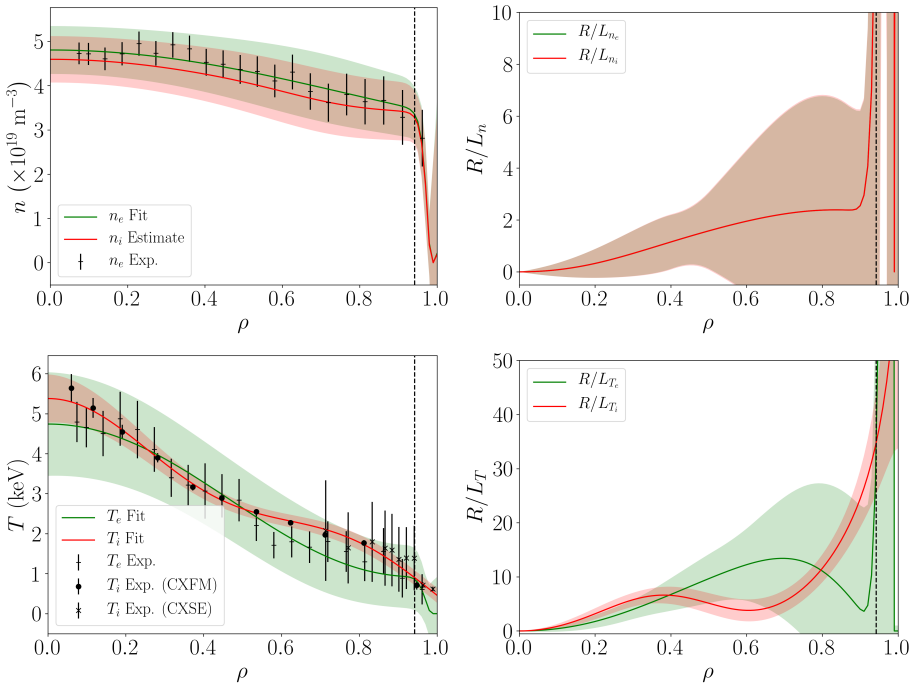

Figure 1: Profiles for JET #70274. Upper left: Electron (green line) and ion (red line) densities with experimental data (black points), showing good fitting of pedestal shoulder for use as the boundary condition for core transport modelling, n_i estimated using Z_{eff}. Upper right: Normalized electron, ion, and main impurity (blue) density gradient lengths. Lower left: Electron and ion temperatures with experimental data. Lower right: Normalized electron and ion temperature gradient lengths.

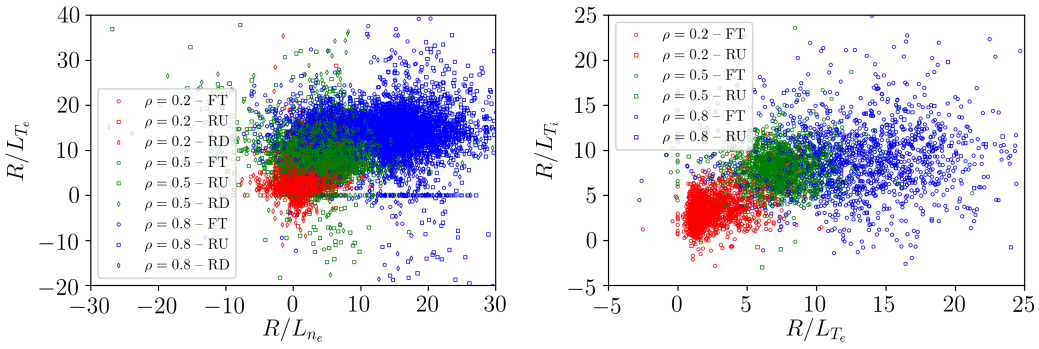

Figure 2: Left: Normalized electron temperature gradient length against normalized electron density gradient length at three different radial positions (red, green, blue) and during ramp-up (RU – squares), flat-top (FT – circles), and ramp-down (RD – diamonds) phases. Right: Normalized ion temperature gradient length against normalized electron temperature gradient length at three different radial positions (red, green, blue) and during ramp-up (RU – squares) and flat-top (FT – circles) phases.