IAEA nuclear fusion

PAPER

Real-time capable modeling of neutral beam injection on NSTX-U using neural networks

IOP Publishing | International Atomic Energy Agency Nuclear Fusion

Nucl. Fusion **59** (2019) 056008 (15pp) https://doi.org/10.1088/1741-4326/ab0762

Real-time capable modeling of neutral beam injection on NSTX-U using neural networks

M.D. Boyer, S. Kaye and K. Erickson

Princeton Plasma Physics Laboratory, Princeton NJ, United States of America

E-mail: mboyer@pppl.gov

Received 30 November 2018, revised 11 February 2019
Accepted for publication 15 February 2019
Published 22 March 2019

Abstract

A new model of heating, current drive, torque and other effects of neutral beam injection on NSTX-U that uses neural networks has been developed. The model has been trained and tested on the results of the Monte Carlo code NUBEAM for the database of experimental discharges taken during the first operational campaign of NSTX-U. By projecting flux surface quantities onto empirically derived basis functions, the model is able to efficiently and accurately reproduce the behavior of both scalars, like the total neutron rate and shine through, and profiles, like beam current drive and heating. The model has been tested on the NSTX-U real-time computer, demonstrating a rapid execution time orders of magnitude faster than the Monte Carlo code that is well suited for the iterative calculations needed to interpret experimental results, optimization during scenario development activities, and real-time plasma control applications. Simulation results of a proposed design for a nonlinear observer that embeds the neural network calculations to estimate the poloidal flux profile evolution, as well as Z_{eff} and fast ion diffusivity, are presented.

Keywords: neutral beam modeling, plasma control, current profile control, plasma profile observer design

(Some figures may appear in colour only in the online journal)

1. Introduction

The spatial distributions of electron and ion temperature and density, momentum, and current density are among the most critical factors in determining the stability and performance tokamak plasmas. The evolution of these profiles is governed by coupled nonlinear parabolic partial differential equations and must be carefully tailored to achieve desired operating regimes. External actuators can be used to manipulate the boundary conditions, source terms, (e.g. heating, current drive, momentum and particle injection from neutral beams or radio frequency waves), or the transport coefficients. Since first principles modeling of transport phenomena is computationally challenging, reduced theory-based models are typically used for analysis of experimental results and predictions for device design and experiment planning in predictive transport codes such as TRANSP [1, 2], CRONOS [3], and ASTRA [4]. These models are still, however, computationally expensive, making their use for optimization of operating scenarios challenging, their use for between-shots predictive modeling impractical, and their use in real-time feedback control algorithms impossible. Recent work has demonstrated the benefits of using light-weight control-oriented physics-based transport models for scenario optimization [5–8] and control [9–12] of plasma profiles. The use of semi-empirical models as replacements for the most computationally intensive parts of the profile evolution model enables the rapid execution times required for use in real-time. However, empirical models and scaling laws are limited to particular operating regimes, motivating efforts to enable high-fidelity results to be obtained with reduced computational cost. Neural networks have recently been applied as a means to rapidly calculate transport fluxes based on the results of first principles calculations [13–15]. This work continues along these lines by developing a real-time capable high-fidelity model of neutral beam injection on the National Spherical Torus eXperiment Upgrade (NSTX-U) [16] based on the results of the TRANSP neutral beam module NUBEAM [17, 18]. The real-time capable model is referred to as NubeamNet to distinguish it from the Monte Carlo code the model is based on.

NSTX-U, which completed its first plasma operation campaign in 2016 [19–21], aims to span between the previous class of spherical torus devices, like NSTX [22] or the Mega-Ampere Spherical Tokamak (MAST) [23], and future facilities planned to study plasma-material interaction [24], nuclear components [25], and demonstration of fusion power production [26, 27]. NSTX-U looks to build upon the results of NSTX [28] to improve the physics understanding of several key issues for future devices, including the scaling of electron transport with field and current [29–32], the physics of fast particles [33–38], and the achievement and sustainment of non-inductive, high-β scenarios [39–44]. One of the primary components of the upgrade project was the replacement of the 'center stack' (containing the inner-leg of the toroidal field (TF) coils, the Ohmic heating (OH) solenoid, and some divertor coils) with one capable of reaching much higher fields and providing more Ohmic flux for longer discharges. The second major upgrade was the addition of a second neutral beam injector (NBI), aimed more tangentially, which significantly increases the auxiliary heating power and neutral beam current drive, and adds flexibility in shaping the spatial deposition of these quantities in the plasma. The upgrade increases the TF capability from 0.55 T to 1.0 T, the maximum plasma current from 1.3 MA to 2.0 MA, and enables full-field discharge durations of 5 s.

With the goal of taking full advantage of the new device capabilities, much progress has been made on the development and simulation testing of new algorithms for advanced control applications (e.g. [45–50]), including current and rotation profile. The model-based profile control algorithm designs rely on empirical scalings for beam heating, current drive, and momentum injection, and it is anticipated that the availability of a fast but high-fidelity neural network model of NUBEAM will enable more accurate real-time calculations, improving profile estimation through better dynamic observers, and model predictive control performance through more accurate predictions. Furthermore, a neural network model of NUBEAM will provide a means to compare the model predicted neutron production rate to measured values. This comparison enables real-time monitoring of fast ion losses which could potentially be a useful signal in disruption prediction algorithms, as indicated in [51].

1.1. Organization

The paper is organized as follows: in section 2, the generation of the dataset used for model development and the selection of features is described. In section 3, the selection of model topology is described, and predictions of the validation dataset and timing in the NSTX-U real-time system are demonstrated. Section 4 shows prediction results on testing data not used for training or topology selection, while section 5 discusses an approach to determining when model predictions can be considered reliable. Section 6 shows simulation results of a proposed design for a nonlinear observer that embeds the neural network calculations to estimate the poloidal flux profile evolution, as well as Z_{eff} and fast ion diffusivity. Section 7 includes discussion and plans for future work.

2. Dataset and feature development

The motivation for the neural network model developed in this work is to closely approximate the results of the NUBEAM code quickly enough to enable use in real-time control applications, between shots analysis, and scenario optimization on NSTX-U. To this end, rather than attempting to create a model that accurately reproduces the entire predictive range of NUBEAM, which would require generating a comprehensive dataset encompassing the complete physically possible range of all of the inputs to NUBEAM, we focus on a subset of inputs with ranges defined by the operating space of the NSTX-U's first campaign in 2016. While it is theoretically possible to generate a single neural network model applicable to many or even all devices, beam geometries, and plasma conditions, it is more practical to develop dedicated models for specific machines or even specific plasma scenarios due to the smaller datasets required for training, and the resulting models are likely to require fewer neurons, making evaluation times better suited to real-time applications.

To generate the dataset, the interpretive TRANSP runs that are automatically run between NSTX-U shots were resubmitted with increased NUBEAM fidelity (5 ms time steps and 10 000 particles). With these settings and running with 16 parallel processes, the wall time for the calculation of each time step is roughly 5 s. For each shot, a scan was conducted for key parameters, including Z_{eff} and edge neutral density. In these scans, the anomalous fast ion diffusivity was considered to have the form

$$D_{\text{f}}(\hat{\rho}) = D_{\text{f},1} + (D_{\text{f},0} - D_{\text{f},1})(1 - \hat{\rho}^{\alpha_{\text{f}}})^{\beta_{\text{f}}}$$

and runs either used classical ($D_{\text{f},1} = D_{\text{f},0} = 0$), flat spatial profiles ($D_{\text{f},1} = D_{\text{f},0} = D_{\text{f,mag}}$), or peaked profiles ($D_{\text{f}_1} = D_{\text{f,mag}}$, $D_{\text{f},0} = 0$, $\alpha_{\text{f}} = 2$, $\beta_{\text{f}} = 4$). The value $D_{\text{f,mag}}$ was selected from a uniform distribution between 1 and 50 000 cm^2 s^{-1} for each run.

Randomized beam on-off modulations were applied, constrained by the minimum on and off times of the NSTX-U neutral beams (10 ms). Beam voltages were randomly selected from between 60 keV and 110 keV. Since the NSTX-U beams are not operated away from the optimal beam perveance, the selection of voltage determines the beam power, as well as the energy fractions.

A set of roughly 1000 runs based on approximately 250 shots were used to create a database of nearly 100 000 time samples. Eighty percent of the shots in the dataset were randomly assigned to be used for model training, ten percent were assigned to validation, and the final ten percent were reserved for testing. No NUBEAM results from the discharges assigned to the testing dataset were used to train models, while validation data was used to assess accuracy and generalization during hyper parameter tuning. Inputs to the model were chosen to be: beam powers, edge neutral density, Z_{eff}, electron temperature and density profiles, q profile, fast ion diffusivity,

Table 1. Symbols and descriptions of the inputs and outputs of the developed reduced beam model.

Inputs		Outputs	
Symbol	Name	Symbol	Name
Z_{eff}	Effective charge	s_{neutron}	Total neutron rate
$n_{0,\text{out}}$	Edge neutral density	P_{shine}	Shine-through power
R_0	Major radius	P_{cx}	Charge-exchange loss power
κ	Elongation	P_{orb}	Orbit loss power
I_{p}	Plasma current	$P_{\text{b,e}}$	Beam heating (electrons) profile
a	Minor radius	$P_{\text{b,i}}$	Beam heating (ions) profile
$B_{\phi,\text{v}}R$	Vacuum toroidal field	$T_{\text{b,e}}$	Beam torque (electrons) profile
δ_{u}	Upper triangularity	$T_{\text{b,i}}$	Beam torque (ions) profile
δ_{l}	Lower triangularity	n_{b}	Beam ion density profile
P_{inj}	Injected power for each beam	j_{b}	Beam current drive profile
n_{e}	Electron density profile	p_{fast}	Fast ion pressure
T_{e}	Electron temperature profile		
q	Safety factor profile		
D_{f}	Anom. fast ion diffusivity profile		

major radius, minor radius, elongation, upper and lower triangularity, plasma current, and toroidal magnetic field. The outputs to be predicted by the model were chosen to be the total neutron rate, shine through, charge-exchange and orbit loss, and profiles of beam heating to ions/electrons, beam current drive and torque, fast ion pressure, and beam ion density. A table of inputs and outputs and corresponding symbols used in the text and figures are provided in table 1.

2.1. Reduction of profile data

Radially varying quantities are represented in TRANSP on a discrete grid of points in the normalized toroidal flux coordinate $\hat{\rho}$, typically using between 20 and 60 points. Rather than fitting to this data directly, the radially varying quantities were first projected onto a set of basis functions. The basis functions for each quantity were chosen by applying principal component analysis to the dataset and keeping only the most significant modes, typically around 4. Aside from reducing the training dataset size and training time, the use of a reduced basis set helps the neural network model produces spatially smooth output profiles and avoids overfitting of noise in the profile dataset (either measurement noise in the case of inputs or noise due to the Monte Carlo scheme used in NUBEAM in the case of outputs). Figure 1 shows a heat map of the relative amount of the variance in the dataset explained by each mode for each of the profiles. The highlighted mode is the first one with a relative explained variance of less than 0.01. Only this mode and those with higher explained variance were kept in the reduced basis set. Figure 2 shows examples of the mean profile and modes for the safety factor profile and beam current drive. For these profiles, the retained modes exhibit low spatial frequency variations with increasing frequency variations in the higher (less significant) mode numbers. Example profiles reconstructed from the reduced set of modes are compared to the original j_{b} and $P_{\text{b,i}}$ profiles (from TRANSP run 204991S28) in figure 3. In addition, comparisons of all data points from the TRANSP run for the p_{fast} and $T_{\text{b,i}}$ profiles are shown in figure 4. Both figures illustrate the accuracy with which the profiles can be reconstructed with just a few modes.

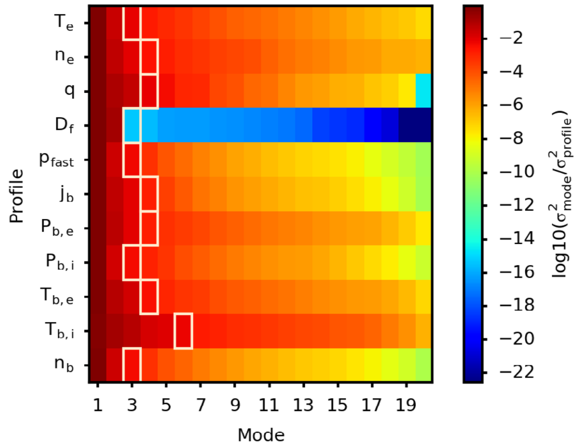


Figure 1. Relative explained variance of the modes for each profile showing that nearly all of the variance is explained by a small number of modes (note the log scale). The last mode kept in the model for each profile is highlighted in white (this mode and those to the left of it are kept in the model).

2.2. Beam slowing down time effects

Due to the slowing down time of fast ions, the various effects of a beam on the plasma depend on the time history of the beam and the discharge. Therefore, it cannot be expected that a model trained only on instantaneous values of the inputs should accurately predict the output behavior (unless the dataset is only made up of steady-state results). While many approaches could be taken to include time history effects in the model, including recurrent neural networks, the simple but evidently effective approach taken here is to augment the inputs of the model with a set of causal low-pass filtered versions of the individual beam powers. To account for the potential range of slowing-down times possible at different plasma conditions, the beam powers are filtered with time constants 0.02 s, 0.05 s, and 0.1 s. A simple first-order filter described by

$$\dot{x} = \frac{u - x}{\tau_{\text{LP}}} \tag{1}$$

is used, where u is the instantaneous input (beam power), x is the filtered output, and τ_{LP} is the filter time constant. An example of the expanded input signals used for beam line 1 A for run 204991S28 are showing in figure 5.

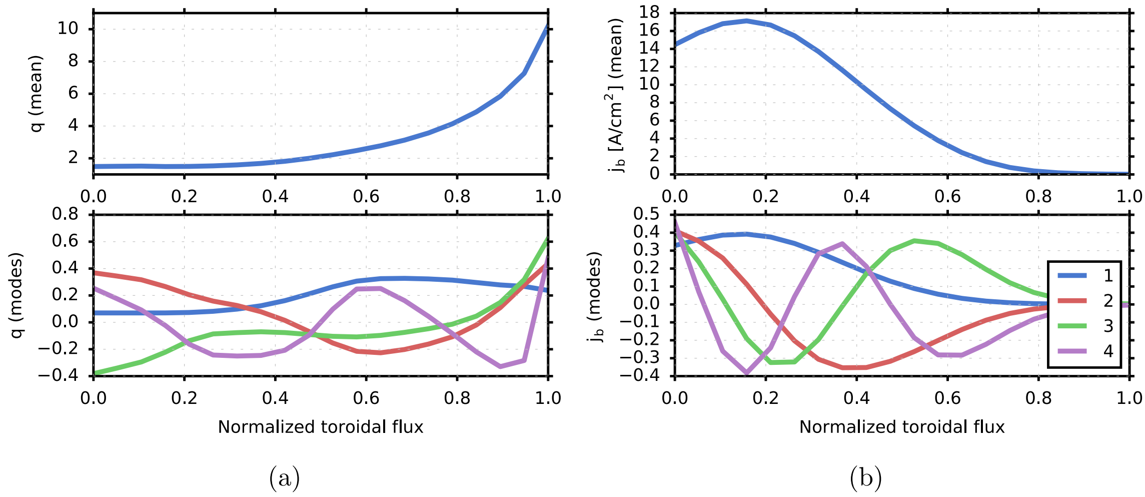


Figure 2. Examples of mean and modes for profiles: (*a*) safety factor profile (*b*) beam current drive.

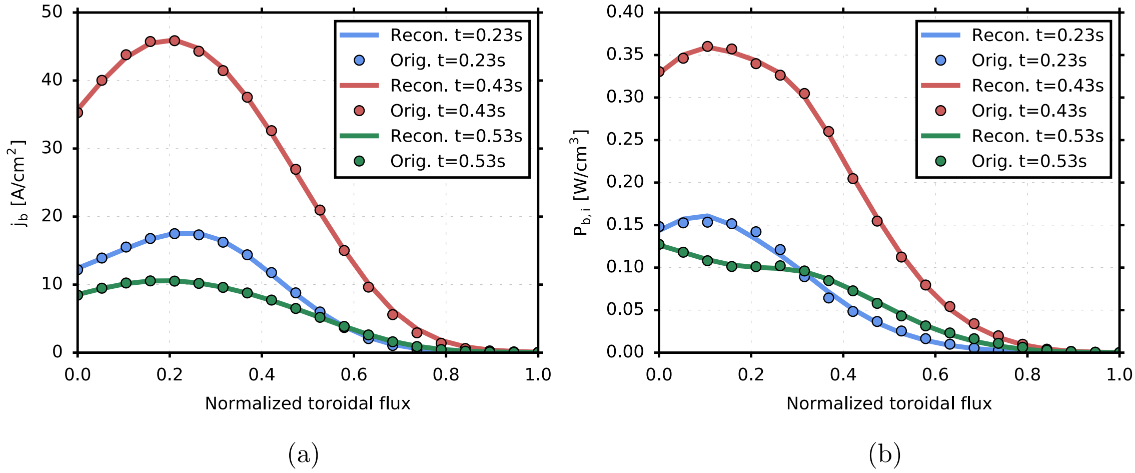


Figure 3. Examples of profiles reconstructed from reduced number of modes (solid line) and original TRANSP data (circles) at various times in run 204991S28.

2.3. Standardization

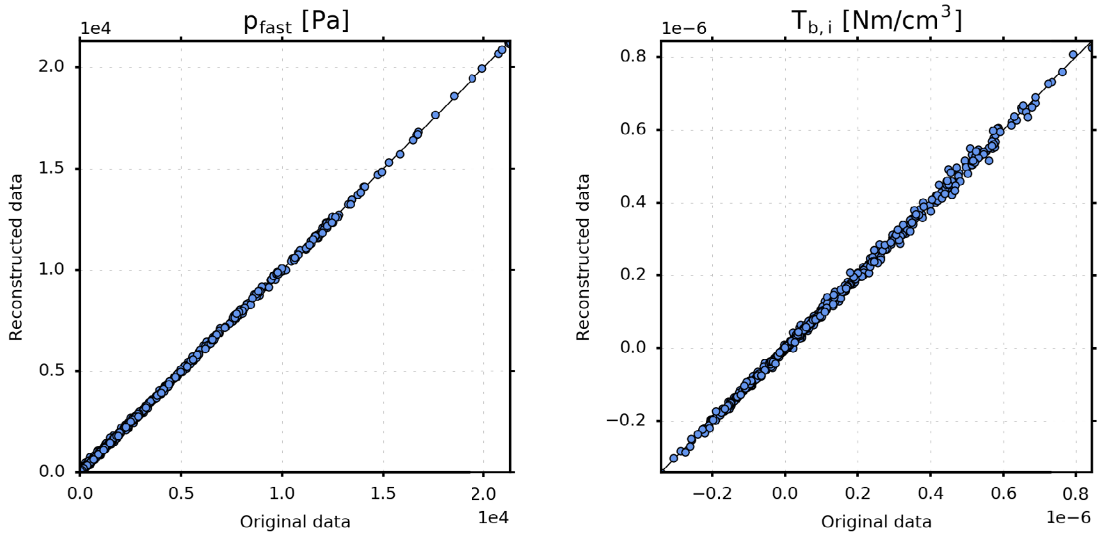
Because the neural network training results are sensitive to the magnitude of the variance of the features, the input and output features were all independently standardized to zero mean and unit variance prior to training. Likewise, when using the trained model for prediction, the input features are first formed by projection of profile data and expansion of beam power inputs, then standardized. The inverse transformation is applied to the predictions of the model prior to projecting the predicted output profile modes onto the significant modes to produce predicted spatial profiles.

3. Model topology selection

A fully connected neural network topology was chosen for the models developed in this work. To provide improved estimates and a sense of the uncertainty of the estimated values, an ensemble of 5 models was trained, each using a randomly selected subset of the training dataset and all using the same neural-network topology. The output of the ensemble is taken to be the average output of the models, and the standard-deviation of the mean of the model predictions is used to provide an estimate of the uncertainty of the predicted output. A diagram of the complete model, including the projection, filtering, and standardization procedures described in the previous section, is shown in figure 6.

The choice of the number of hidden layers and hidden-layer nodes was selected through scoring how well models generalized to the shots in the validation dataset in a grid scan of the hyperparameters. The procedure was also used to select the weight on the L_2-norm of the model coefficients, a regularization term in the loss function minimized during training of the neural networks[1]. As an example, regression plots comparing estimated and predicted total neutron rates for the training and validation sets for two different model topologies are shown

[1] The loss function is given by $f_{\text{loss}} = \frac{1}{2}||\hat{y} - y||_2^2 + \frac{\alpha}{2}||w||_2^2$ where $\hat{y}$ is the set of model prediction, y is the set of true values and w is the set of neuron weights.



(a)

(b)

Figure 4. Comparison of actual profile data to the reconstructed profile data using a reduced number of modes for (*a*) fast ion pressure and (*b*) torque to ions for TRANSP run 204991S28.

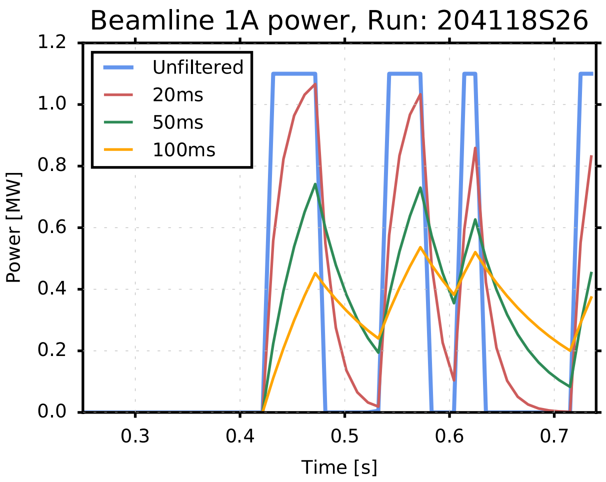


Figure 5. Example of expanded set of input signals used to incorporate time history of beam powers, including instantaneous (unfiltered) power and filtered values from run 204991S28. Values for beam line 1 A are shown.

in figure 7, while comparisons for the fast ion pressure profile are shown in figure 8. The results show that the fit has little to no degradation when moving from training data to validation data (not used for training model coefficients) for either topology. For the results shown, the higher complexity model is a better fit to both the training data and validation data. The mean of the R^2[2] for all of the output variables is plotted in figure 9(*a*) as a function of number of nodes per layer, with different colored markers for each number of layers considered. Evidently, multiple hidden layers is a significant improvement for all numbers of nodes tested. Two and three hidden layers converge to very similar values beyond 150 nodes per layer. While the four layer model out performs the others for small numbers of nodes, it is surpassed by the two and three layer models beyond 50 nodes per layer, indicating the model is too complex and results in overfittting of the training data. The results are plotted in figure 9(*b*) as a function of the number of free parameters in each model. The dependence of R^2 on model topology for three selected outputs (shine through, beam driven current, and charge-exchange loss, respectively) is shown in figure 10. It is clear that each output has different dependence on topology, with some continuing to show improvement, albeit small, at 300 nodes per layer (shine-through), and some reaching a maximum around 125–150 nodes per layer (charge-exchange losses). The dependence on the number of layers also varies, with some (shine through and beam-driven current) showing slight improvement with multiple layers, and others (charge-exchange losses) shown a significant improvement with multiple layers, mostly independent of the number of nodes per layer.

To demonstrate the real-time applicability of the model, the main calculations required by the model (profile reduction, normalization, and neural network evaluation) were implemented in the NSTX-U real-time computer. A scan of model topology was tested using the real-time implementation to assess the scaling of calculation time with model complexity as well as cycle to cycle variations in timing. Results, shown in figure 11, show that models with complexity near that required to optimize the model fit can be run within the typical 200 μs cycle time of the NSTX-U control system, e.g. 3 layers of 100 nodes each. Times are evidently around 10^4–10^5 times faster than the NUBEAM calculations in the database. Cycle to cycle variation was found to be less than 2 μs. Recent advances in real-time PCIe-based internode communication in the NSTX-U control system [52] will enable offloading calculations to a dedicated computer with enough

[2] The coefficient of determination R^2 is calculated as $R^2 = 1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \bar{y})^2}$, where y is a vector of observations, $\bar{y}$ is the mean observation, and f is a vector of predicted values.

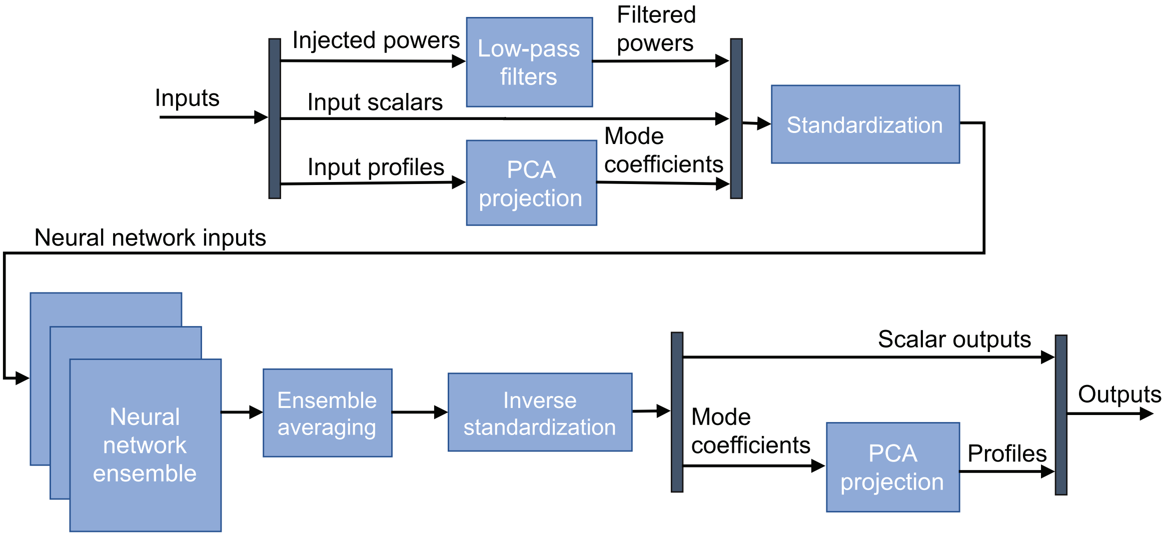


Figure 6. Diagram of the real-time capable beam model, including the processing described in section 2 and the neural network ensemble described in section 3.

cores to simultaneously calculate the models for uncertainty quantification as well as calculation of the sensitivity of outputs to changes in inputs needed by real-time control and optimization algorithms.

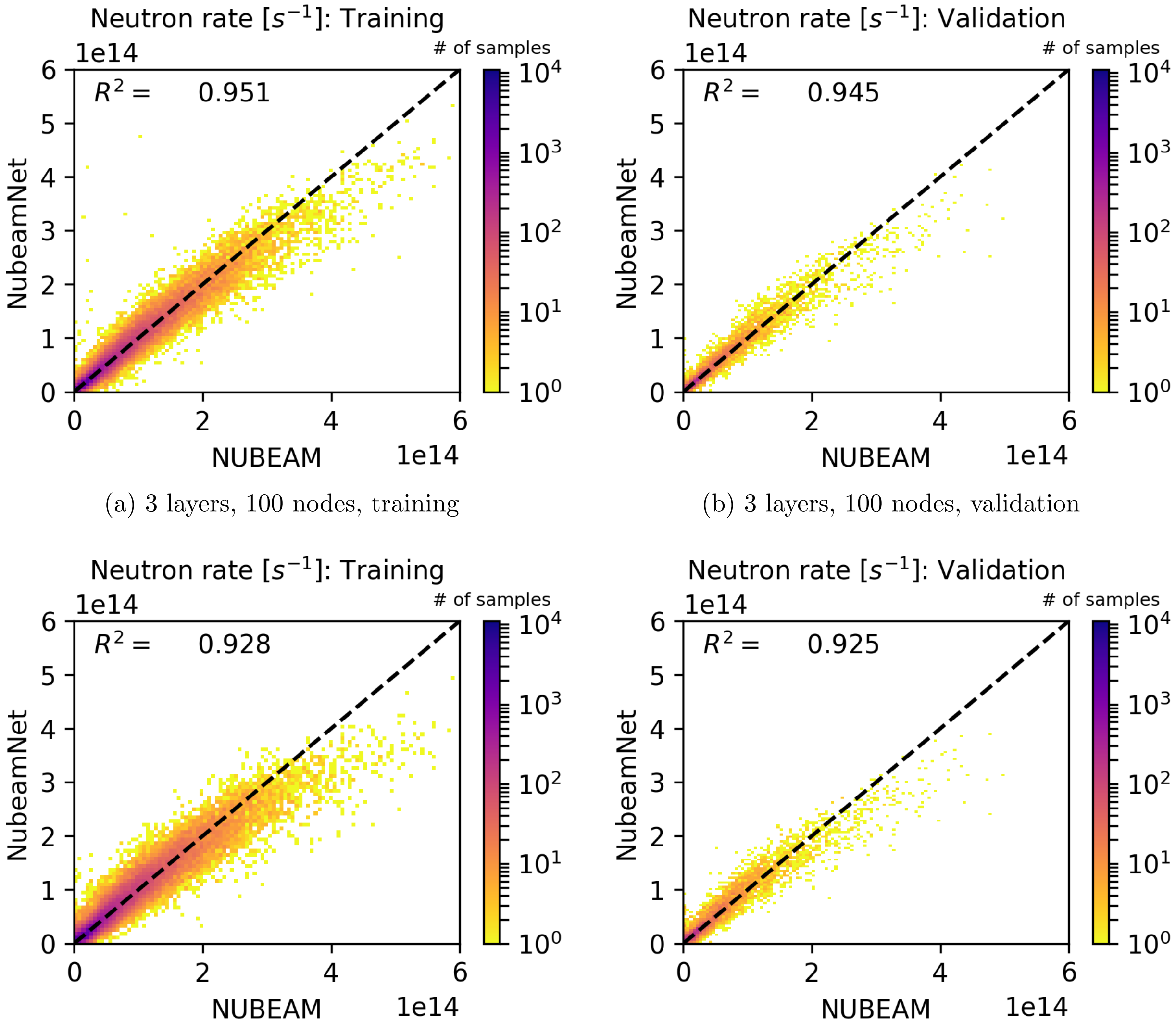
4. Predictions of testing dataset

After studying the topology with the validation dataset, the model using 3 layers and 100 nodes per layer was selected and used to predict the output for the inputs in the testing dataset. Figure 12 shows time traces of the NubeamNet prediction, with the shaded region representing ± one standard deviation of the mean of the model ensemble, compared to the NUBEAM calculations for run 204738S33. Depicted are (*a*) the beam current drive at $\hat{\rho} = 0.421$, (*b*) fast ion pressure at $\hat{\rho} = 0.053$, (*c*) total neutron rate, and (*d*) beam heating of electrons at $\hat{\rho} = 0.421$, where $\hat{\rho}$ is the normalized toroidal flux coordinate. Figure 13 compares the profiles predicted by NubeamNet (shaded region represents ±one standard deviation of the mean of the model ensemble) to the NUBEAM calculated profiles for run 204738S33 at several times during the shot: (*a*) beam current drive, (*b*) fast ion pressure, (*c*) heating to ions, and (*d*) torque to electrons. The results show strong agreement throughout the shot, even in the presence of beam modulations. The expected NUBEAM results are typically within one standard deviation of the NubeamNet ensemble mean. Despite using a reduced set of profile modes, the predictions are able to accurately reproduce the shapes of profiles. For example the model is able to capture the fact that the beam current drive and electron torque are peaked off-axis at $t = 0.438$ s and move toward the magnetic axis in subsequent times shown. Finally, regression plots for the testing dataset are shown for the neutron rate and fast ion pressure in figure 14. Comparing to the results shown in figures 7 and 8, it is evident that the model generalizes well to the data that was not used in training or topology selection.

5. Ensemble uncertainty as an indicator of extrapolation

A potential weakness of the presented modeling approach is the possibility for large errors if the neural network is presented with inputs that are outside of the training region, since neural networks models are not able to extrapolate. Furthermore, because the input space is high-dimensional, it is not trivial to determine whether a particular input should be considered to be inside the training region. The potential impact of extrapolation errors is application specific, so the appropriate approach to mitigating this issue will depend on the future uses of the model. Nonetheless, we present one possible heuristic solution for detected extrapolation that takes advantage of the ensemble approach used in this work.

First, the normalized cumulative sum of the distribution of ensemble uncertainties for the predictions on the training data is calculated, and a likelihood p_{std} is specified. From this, a threshold value $t_{\mathrm{std},i}$ for the standard deviation of each ensemble output i is determined such that the standard deviation of ensemble predictions of i is expected to be less than $t_{\mathrm{std},i}$ with a likelihood of p_{std}. Figure 15(*a*) shows the cumulative sum of standard deviations for the neutron rate prediction for the training set, along with lines indicating $p_{\mathrm{std}} = 0.8$ and $t_{\mathrm{std,neutron}}$. The same value of p_{std} was used to calculate thresholds for the other outputs. For each of the predictions on the training set, n_{above} is calculated as the number of outputs for which the standard deviation exceeds the output's threshold $t_{\mathrm{std},i}$. The normalized cumulative sum of the distribution n_{above} is then calculated for the training set and the threshold t_n such that n_{above} is expected to be less than t_n with a likelihood of



(a) 3 layers, 100 nodes, training

(b) 3 layers, 100 nodes, validation

(c) 1 layer, 50 nodes, training

(d) 1 layer, 50 nodes, validation

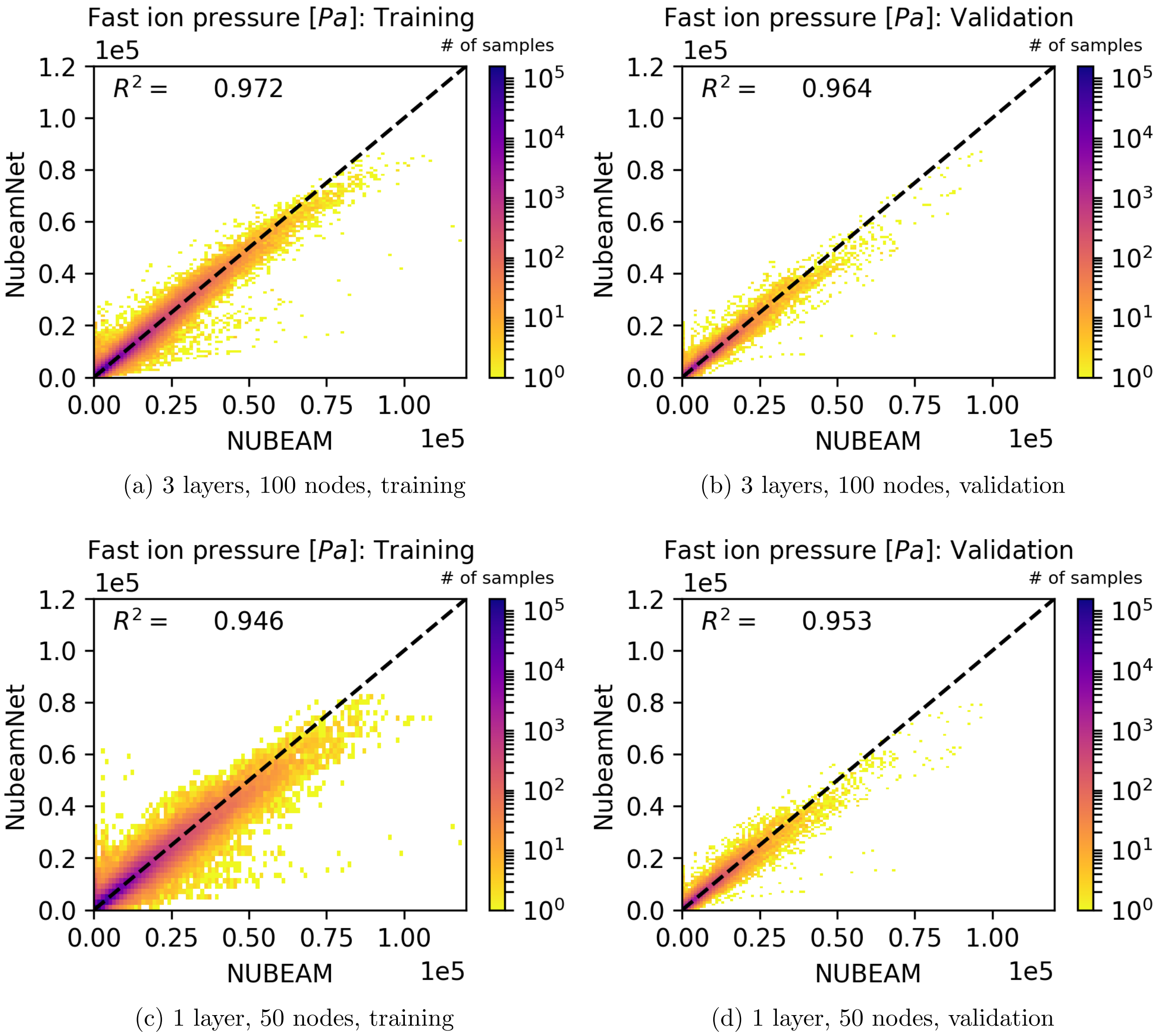
Figure 7. Log-scale histograms of regression results for total neutron rate prediction. The top row shows results for 3 layers, 100 nodes for the (*a*) training data and (*b*) validation data, while the bottom row shows results for 1 layer, 50 nodes for the (*c*) training data and (*d*) validation data.

p_n is determined. The cumulative sum of n_{above} is shown in figure 15(*b*), along with lines for $p_n = 0.95$ and t_n.

Ensemble prediction standard deviations for any given sample can then be compared with the values t_{std} to determine n_{above}, which can in turn be compared to t_n. Predictions for which $n_{\text{above}} > t_n$ can be considered unreliable. As a test, a series of predictions was made based on the inputs from TRANSP run 204738S33. To test whether the approach can detect extrapolation, a scan was done in which the elongation input was changed from 1.0 to 3.0. Figure 16 shows the histogram of elongation in the training set overlaid with the fraction of samples in the test found to be unreliable as a function of elongation. The fraction of unreliable samples is lowest for elongations that occurred frequently in the training set, increases for elongations that occurred less often in the training set and rises toward one outside of the interval that was trained on. As mentioned above, the appropriate response to finding an unreliable prediction will depend on the specific application of the model. For example, in the case of offline applications, if inputs in a region of interest are found to produce unreliable predictions, the training dataset could be expanded through additional NUBEAM calculations to include nearby points. The model could then be trained/updated to improve its reliability in the region of interest.

6. Example application: real-time current profile observer with parameter estimation

For interpretive analysis of experimental results with TRANSP, the fast ion diffusivity used in NUBEAM is typically adjusted in order to match the predicted and measured neutron rates. This procedure is most typically done by assuming a fast ion diffusivity profile shape and scanning the



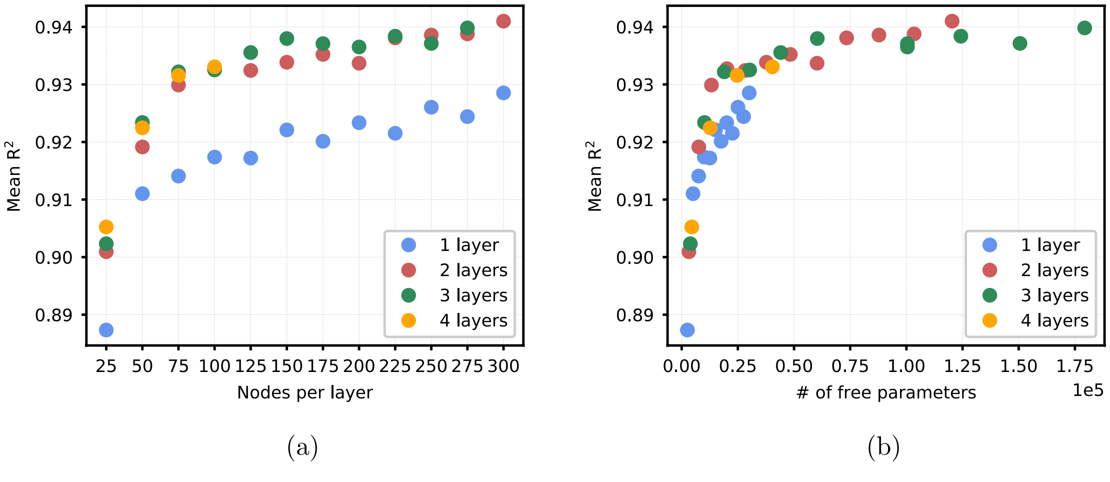
(a) 3 layers, 100 nodes, training

(b) 3 layers, 100 nodes, validation

(c) 1 layer, 50 nodes, training

(d) 1 layer, 50 nodes, validation

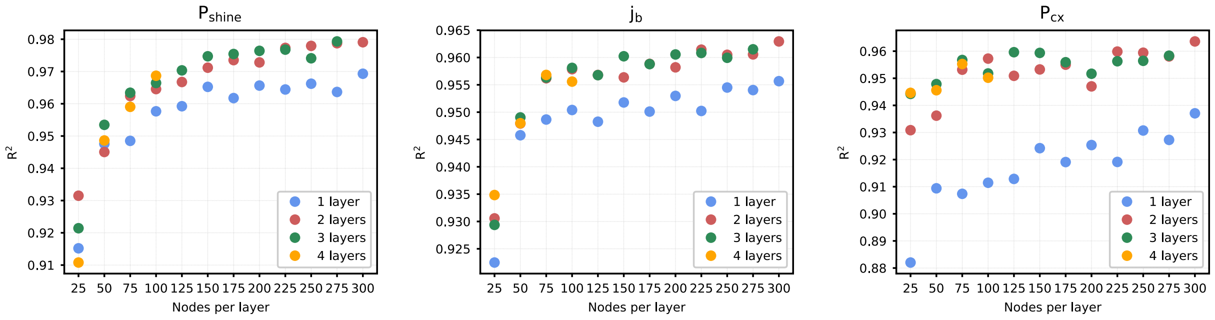
Figure 8. Log-scale histograms of regression results for fast ion pressure profile (includes all radial locations). The top row shows results for 3 layers, 100 nodes for the (*a*) training data and (*b*) validation data, while the bottom row shows results for 1 layer, 50 nodes for the (*c*) training data and (*d*) validation data.



(a)

(b)

Figure 9. Mean R^2 across all output variables as a function of number of nodes per layer (*a*) and number of free parameters (*b*).



(a) (b) (c)

Figure 10. R^2 as a function of number of nodes per layer for (*a*) shine through, (*b*) beam-driven current, and (*c*) charge exchange losses.

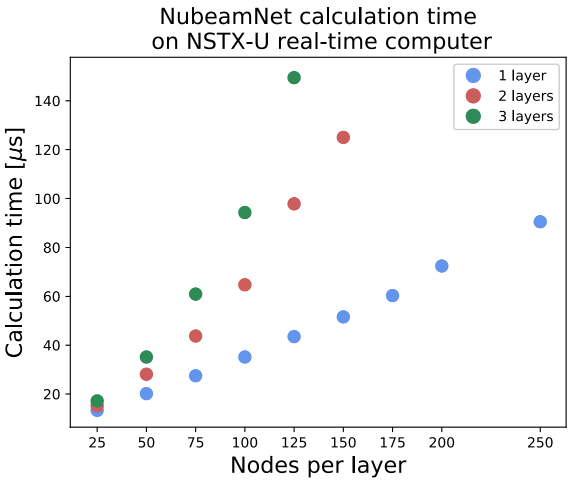


Figure 11. Calculation time as a function of model complexity tested on the NSTX-U real-time control computer.

magnitude in a series of simulations to find the best match. This is time consuming and becomes challenging when the diffusivity varies in time. To improve upon this procedure, a scheme for changing the fast ion diffusivity magnitude automatically during the run was recently added to TRANSP. The scheme is a proportional-integral-derivative control law treating the fast ion diffusivity profile magnitude as an actuator to decrease the neutron rate matching error. TRANSP has options to apply a similar approach to modify $Z_{\rm eff}$ to match the measured current profile evolution. These schemes provide a means for estimating plasma parameters that are uncertain or unmeasured from available measurements and models, however, they rely on running TRANSP and NUBEAM and are therefore not suitable for real-time use. One potential application of the neural network developed in this work is in the development of dynamic observers to enable this estimation to be performed in real-time. While additional reduced models are required to develop an observer for all of the profiles and parameters tracked in TRANSP, we present simulation results of a preliminary design here. The NubeamNet neural net is combined with a reduced model of magnetic diffusion to design an estimator for the current profile, effective charge, and anomalous fast ion diffusivity. For the purposes of this demonstration, we consider a model like the one used for current profile control design for DIII-D in [9, 10]. The poloidal flux diffusion equation is given by

$$\frac{\partial\psi}{\partial t} = \frac{\eta(T_{\rm e})}{\mu_0\rho_{\rm b}^2\hat{F}^2}\frac{1}{\hat\rho}\frac{\partial}{\partial\hat\rho}\left(\hat\rho\hat{F}\hat{G}\hat{H}\frac{\partial\psi}{\partial\hat\rho}\right) + R_0\hat{H}\eta(T_{\rm e})\frac{\langle\bar{j}_{\rm NI}\cdot\bar{B}\rangle}{B_{\phi,0}} \qquad (2)$$

with boundary conditions given by

$$\left.\frac{\partial\psi}{\partial\hat\rho}\right|_{\hat\rho=0} = 0, \qquad (3)$$

$$\left.\frac{\partial\psi}{\partial\hat\rho}\right|_{\hat\rho=1} = -\frac{\mu_0}{2\pi}\frac{R_0}{\hat{G}\Big|_{\hat\rho=1}\hat{H}\Big|_{\hat\rho=1}}I(t). \qquad (4)$$

We assume the bootstrap current is negligible or can be approximated as a time-independent profile. The terms $\hat\rho_{\rm b}$, $\hat{F}$, $\hat{G}$, and $\hat{H}$ are equilibrium-dependent geometric quantities, which are defined in detail in [53] and are assumed to be known and time-independent for the purposes of this work. It is assumed that the electron temperature profile is known and time-independent, and that the resistivity is given by

$$\eta(\hat\rho) = Z_{\rm eff}\frac{k_\eta(\hat\rho)}{T_{\rm e}(\hat\rho)}^{3/2}. \qquad (5)$$

The proposed approach will be extended in future work to include additional reduced models for the resistivity, bootstrap current, and geometric quantities in order to relax the assumptions made here.

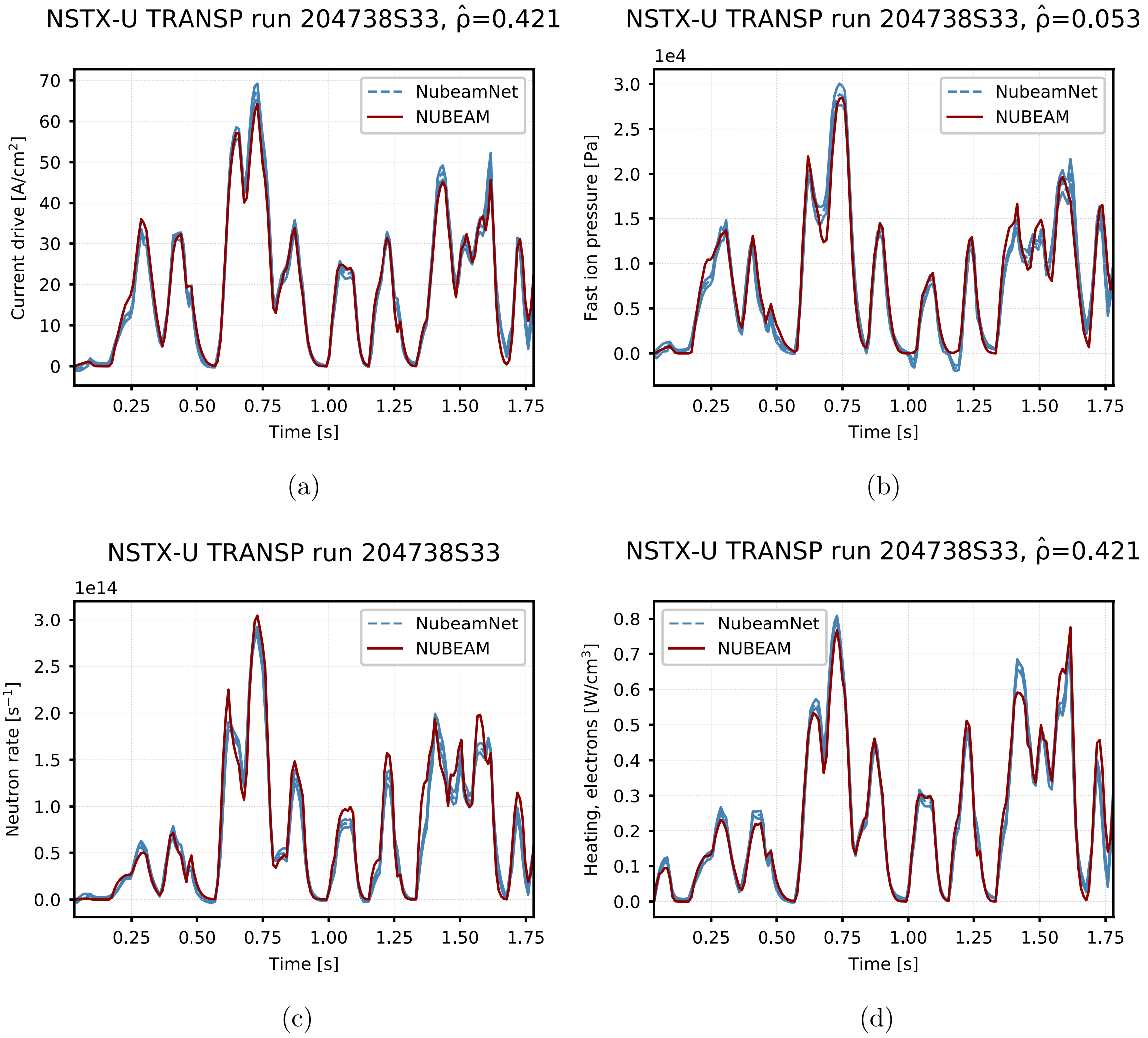
The NubeamNet model is used to calculate the beam current drive component of $\frac{\langle\bar{j}_{\rm NI}\cdot\bar{B}\rangle}{B_{\phi,0}}$ as well as the total neutron rate.

For observer design, this model is discretized in time and space:

$$x_k = f(x_{k-1}, u_k) + w_k \qquad (6)$$

$$z_k = h(x_{k-1}, u_k) + v_k \qquad (7)$$

where $x_k \in \mathcal{R}^{n_g}$ is the vector of state estimates (n_g is the number of points in the spatial discretization of the magnetic diffusion equation. The left and right boundary points are not included in x_k as they are prescribed. The vector includes the two estimated parameters $Z_{\rm eff}$ and $D_{\rm f}$). $z_k \in \mathcal{R}^{n_m}$ is a vector of



(a)

(b)

(c)

(d)

Figure 12. Time traces comparing NubeamNet prediction (dashed line, shaded region represents ± one standard deviation of the model ensemble) to the NUBEAM calculation (solid line) for run 204738S33. (*a*) beam current drive at $\hat{\rho} = 0.421$, (*b*) fast ion pressure at $\hat{\rho} = 0.053$, (*c*) total neutron rate, and (*d*) beam heating of electrons at $\hat{\rho} = 0.421$.

n_m measurements, $w_k \in \mathcal{R}^{n_g}$ and $v_k \in \mathcal{R}^{n_m}$ are the process and measurement noise, assumed to be zero mean Gaussian white noise with covariances Q_k and R_k respectively. f and h are nonlinear system and output functions, respectively.

It is assumed that the boundary poloidal flux is known from magnetic measurements using a real-time equilibrium reconstruction code, along with the inputs to the neural network. The plasma current and total neutron rate are assumed to be measured in real-time. In-domain measurements of the poloidal flux gradient are also considered (results with and without in-domain measurements are compared).

A typical discrete-time extended Kalman filter is used. The predict and update equations are written as:

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_k) \tag{8}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k \tag{9}$$

$$\tilde{y}_k = z_k - h(\hat{x}_{k|k-1}, u_k) \tag{10}$$

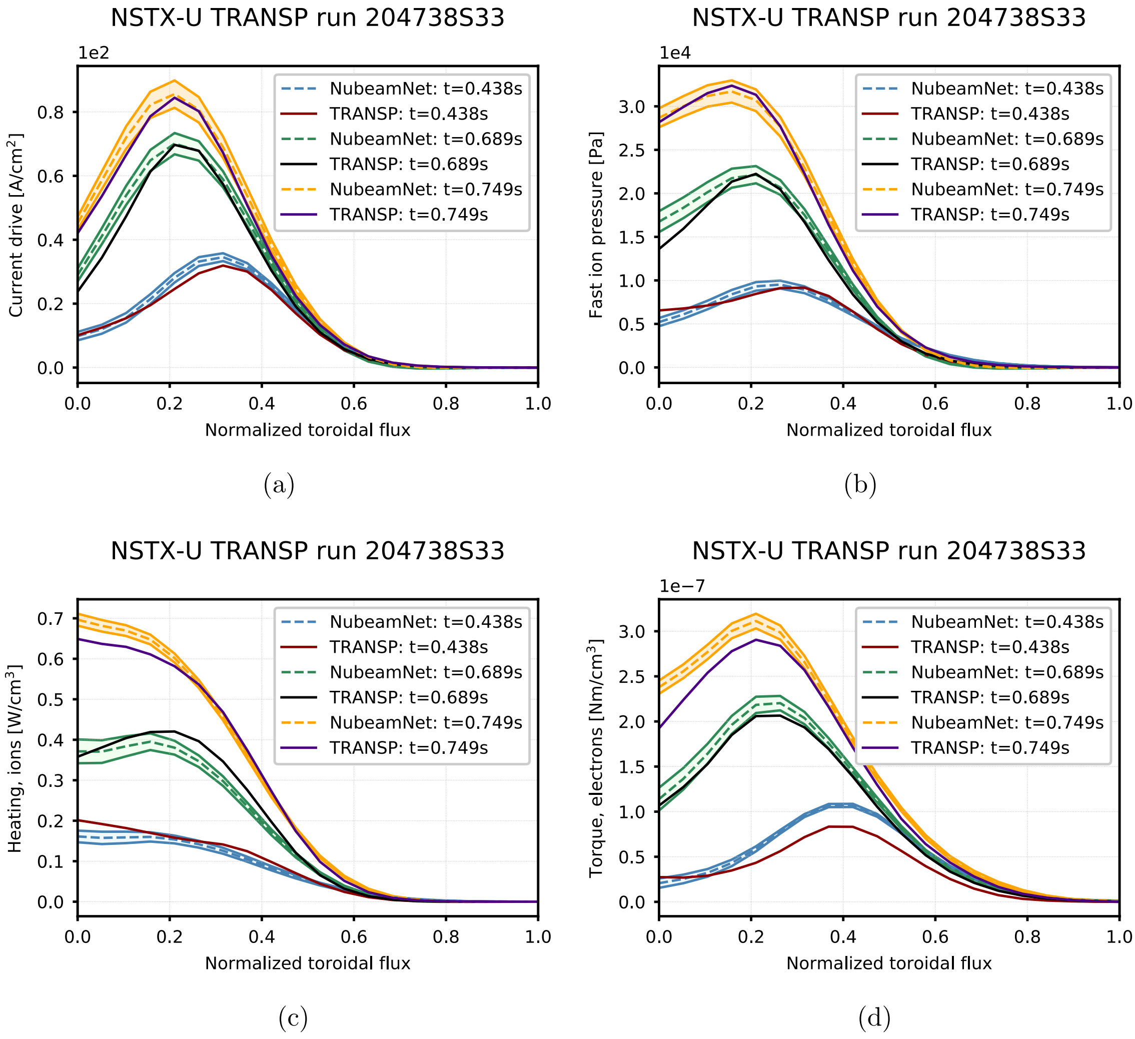
$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \tag{11}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \tag{12}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + \mathbf{K}_k \tilde{y}_k \tag{13}$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \tag{14}$$

where $\mathbf{F}_k$ and $\mathbf{H}_k$ are Jacobian matrices formed by linearizing the system and output functions around the current state estimates. These matrices are formed numerically based on evaluation of the model equations (including the neural network for NUBEAM). This step requires many evaluations of the neural network $(2n_g + 1)$, however, the calculations can be trivially parallelized for each element of the state vector if enough real-time CPUs are available. Furthermore, it is likely that the linearization does not need to be performed at every update time. These implementation details will be addressed in future work. The matrices $\mathbf{Q}_k$ and $\mathbf{R}_k$ represent the process



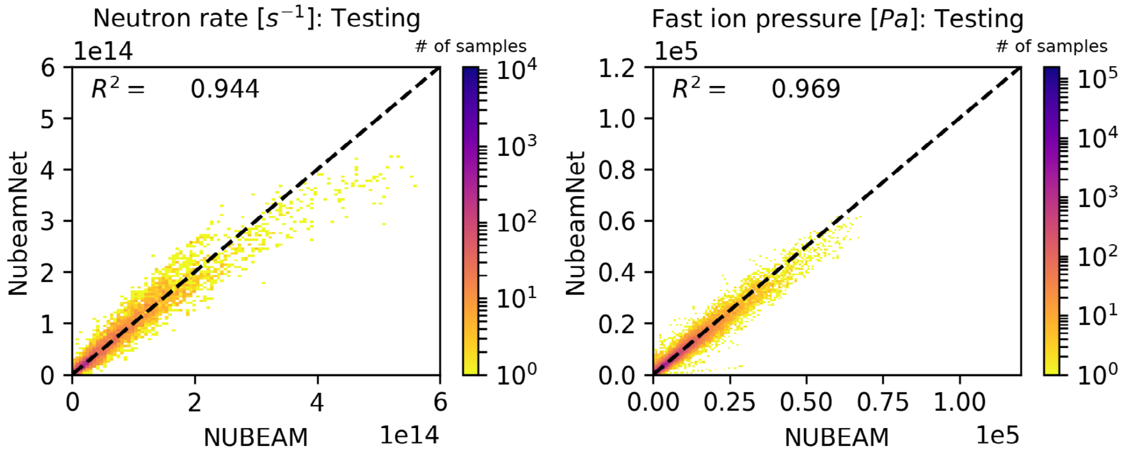
(a)

(b)

(c)

(d)

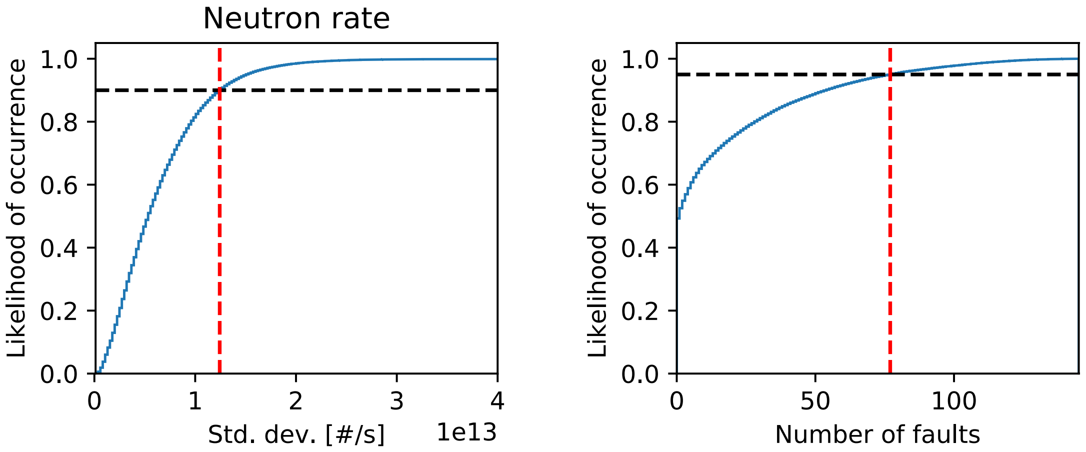
Figure 13. Comparison of profiles predicted by NubeamNet (dashed line, shaded region represents ± one standard deviation of the model ensemble) to the NUBEAM calculated profiles (solid line) for run 204738S33 at several times during the shot. (*a*) Beam current drive, (*b*) fast ion pressure, (*c*) heating to ions, and (*d*) torque to electrons.



(a) 3 layers, 100 nodes, training

(b) 3 layers, 100 nodes, validation

Figure 14. Log-scale histograms of regression results for the testing data set for neutron rate and fast ion pressure profile (includes all radial locations). Results shown for 3 layers, 100 nodes.



(a) (b)

Figure 15. (*a*) Likelihood of occurrence for values of the neutron rate standard deviation. The value of p_{std} is shown in black while the corresponding threshold $t_{\text{std},s_{\text{neutron}}}$ is shown in red. (*b*) Likelihood of occurrence for the number of faults (outputs that exceed $t_{\text{std},i}$) per sample. The value of p_n is shown in black while the corresponding threshold t_n is shown in red.

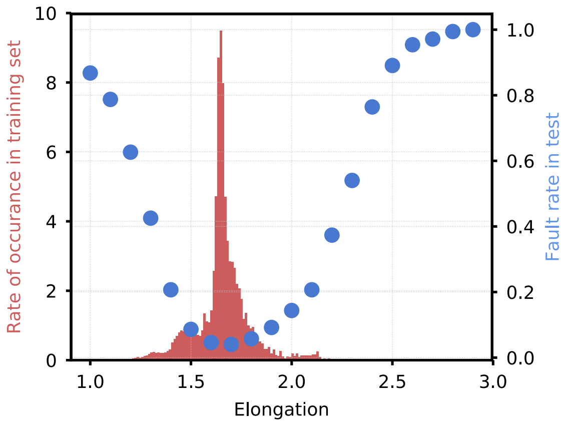
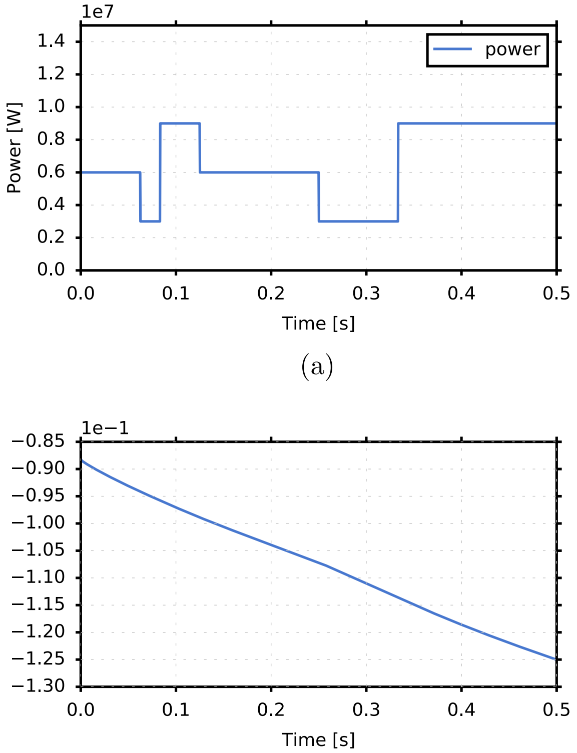


Figure 16. Ensemble uncertainty as an indicator of extrapolation: histogram of elongation values in the training dataset (red bars) overlaid with the fraction of samples with $n_{\text{above}} > t_n$ for different values of elongation (with other inputs taken from test run 204738S33).

and measurement noise covariances, respectively, while $\mathbf{P}_k$ is the state estimation error covariance and $\mathbf{K}_k$ is the Kalman gain used to update state predictions based on measurements.

6.1. Simulation results

The design is tested against a simulated current profile evolution with magnetic diffusion equation parameters taken from shot 204118 at $t = 0.6$ s. The six beams are set to have equal powers that vary in time. The total power waveform is shown in figure 17. The simulated magnetic diffusion equation uses a value of $I_{\text{p}} = 0.96$ MA as its boundary condition. The evolution of the boundary flux required to maintain this value of total current is shown in figure 17. Time-varying values for Z_{eff} and D_{f} are used, as indicated by the red lines in figures 18(*c*) and (*d*).



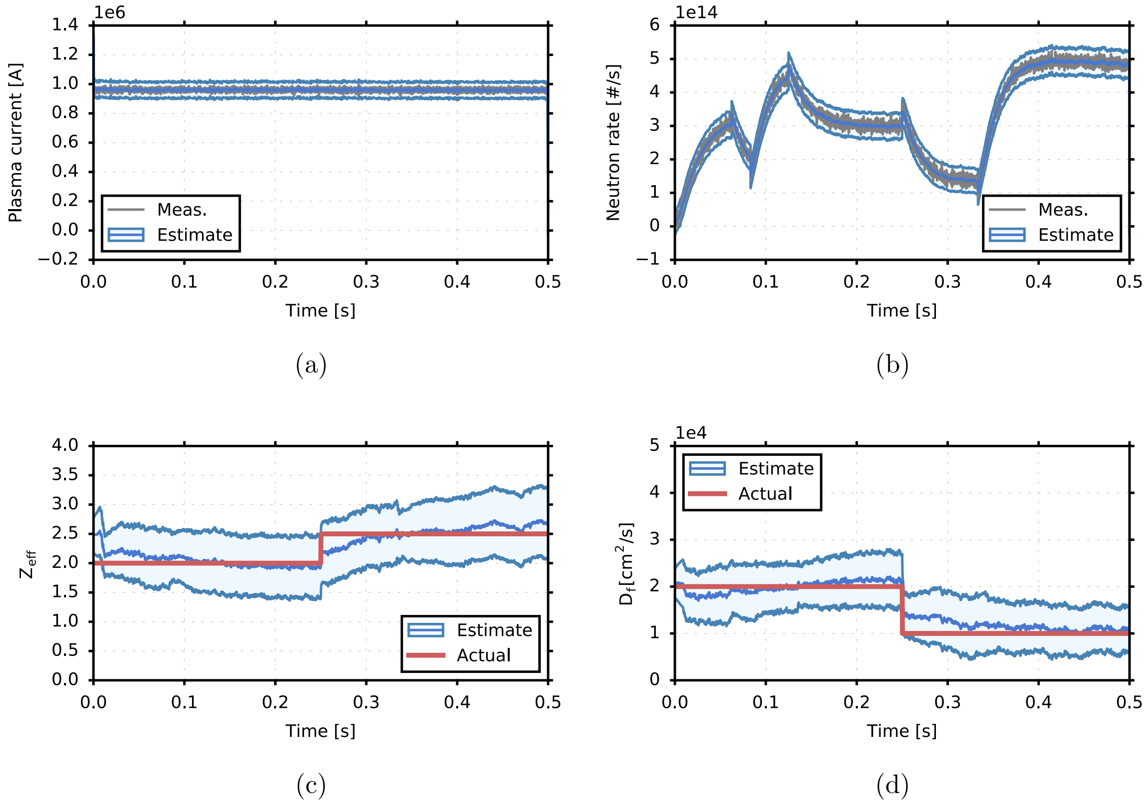
(a)

(b)

Figure 17. Power (*a*) and ψ_{b} (*b*) used in the simulations.

The observer is initialized with a perturbed initial condition $\psi_{0,\text{obs}}(\hat{\rho}) = \psi_{0,\text{sim}}(\hat{\rho}) - 0.004\sin(2\pi\hat{\rho})$. Initial values of $Z_{\text{eff,obs}}$ and $D_{\text{f,obs}}$ are chosen from random Gaussian distributions centered on the initial values used in the simulated model. Measurements are corrupted by additive Gaussian white noise. The observer model parameters are taken to be the same as those used in the simulation model.

Two simulation cases are shown: the first with only measurements of the plasma current, boundary flux, and total



(a) (b)

(c) (d)

Figure 18. Parameter estimation without in-domain state measurements: observer estimates of plasma current and total neutron rate (*a*) and (*b*), compared to noisy measured values (grey). Estimates of $Z_{\rm eff}$ and $D_{\rm f}$ (blue) compared to actual values (red). Shaded regions represent 3 standard deviations of the estimated values.

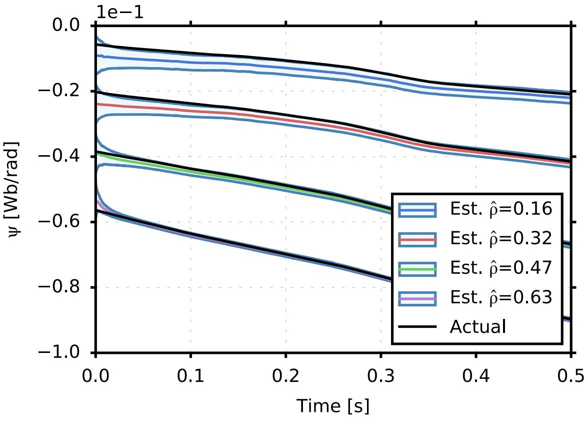
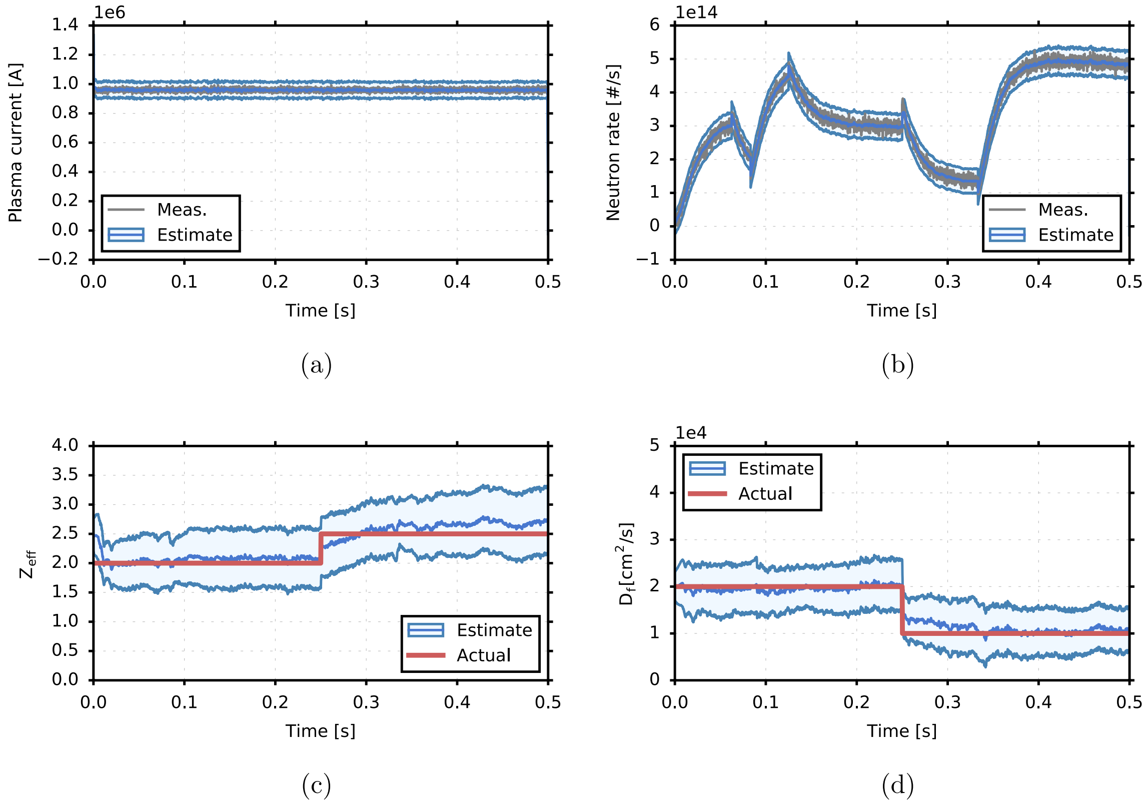


Figure 19. Parameter estimation without in-domain state measurements: observer estimates of poloidal flux at several spatial locations compared to actual (unmeasured) values (black). Shaded regions represent 3 standard deviations of the estimated values.

neutron rate as inputs to the observer, and the second with the addition of four interior measurements of the flux gradient (a proxy for pitch angle measurements).

6.1.1. Parameter estimation without in-domain state measurements. In this test, the observer is tested without in-domain measurements of the poloidal flux or poloidal flux gradient. Figures 18(*a*) and (*b*) compare the measured plasma current and total neutron rate (the simulated values with added Gaussian noise) to the values estimated by the observer. The observer is able to adjust the states (including estimates of $Z_{\rm eff}$ and $D_{\rm f}$ to minimize the output estimation error and providing estimates that are not very sensitive to the measurement noise. The estimates of $Z_{\rm eff}$ and $D_{\rm f}$ are shown in figures 18(*c*) and (*d*) compared with the actual values used in the simulation. It is evident that the estimates converge to the actual values and can track step changes in these parameters. The estimated values of poloidal flux at several in-domain locations are compared with the simulated values in figure 19. Although the estimates converge to the actual values, the estimates closer to the magnetic axis converge more slowly due to the higher temperatures in the core (slower dynamics) and the lack of in-domain measurements in the observer to help drive the estimation error to zero.

6.1.2. Parameter estimation with interior measurements. In this test, the observer is tested with in-domain measurements of the poloidal flux gradient $\frac{\partial \psi}{\partial \hat{\rho}}$ at four locations, $\hat{\rho} = [0.05, 0.21, 0.47, 0.74]$. Again, the observer adjusted the states (including estimates of $Z_{\rm eff}$ and $D_{\rm f}$ to minimize the output estimation error, as seen in figures 20(*a*) and (*b*). The estimates of $Z_{\rm eff}$ and $D_{\rm f}$, shown in figures 18(*c*) and (*d*) converge to the actual values used in the simulation. Performance is similar to the case without in-domain measurements. Like the measurements of plasma current and total neutron rate, the output estimation for the in-domain measurements is kept close to zero throughout the simulation, as shown in



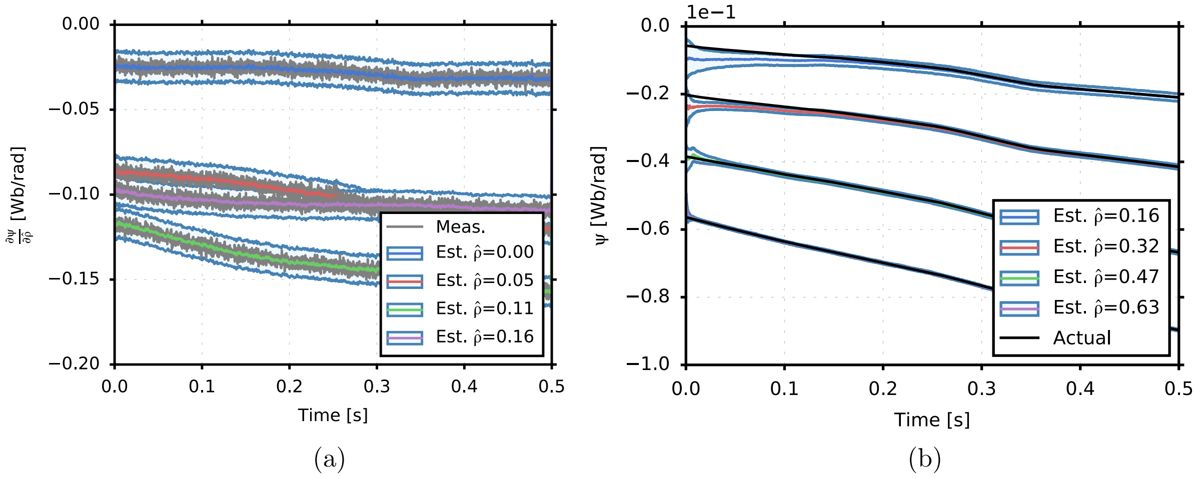
(a)

(b)

(c)

(d)

Figure 20. Parameter estimation with in-domain state measurements: observer estimates of plasma current and total neutron rate (*a*) and (*b*), compared to noisy measured values (grey). Estimates of Z_{eff} and D_{f} (blue) compared to actual values (red). Shaded regions represent 3 standard deviations of the estimated values.



(a)

(b)

Figure 21. Parameter estimation within-domain state measurements: (*a*) observer estimates of poloidal flux gradient at several locations compared to measured values (grey). (*b*) Poloidal flux estimates at several spatial locations compared to actual (unmeasured) values (black). Shaded regions represent 3 standard deviations of the estimated values.

figure 21(*a*). The estimated values of poloidal flux at several in-domain locations, compared with the simulated values in figure 19(*b*) converge to the actual values much faster than in the case without in-domain measurements.

7. Discussion and conclusions

A neural network model for evaluating the beam heating, current drive, torque, and other effects of the NSTX-U neutral beam system on the plasma has been developed. The model was trained on NUBEAM results calculated for the discharges in the first NSTX-U campaign. The speed of the resulting model makes it potentially useful for many real-time applications, as well as optimization for scenario development. Simulation results of a proposed design for a nonlinear observer that embeds the neural network calculations to estimate the poloidal flux profile evolution, as well as Z_{eff} and fast ion diffusivity, are presented.

As an alternative to the approach presented here, a reduced analytical model for rapid calculation of the fast ion distribution, like the one presented in [54] could also be used in these applications. The analytical approach is currently much slower (25 ms versus 150 μs for the neural network approach), and requires simplifying assumptions to be made that are not required in the neural network approach. However, the neural network requires a large dataset and can only make predictions in the region of operating space for which it has been trained, as discussed above. The speed of the neural net makes it ideal for quickly calculating sensitivities or performing iterative calculations. It may be beneficial to combine the approaches, relying on the neural network for applications with the most stringent timing requirements, while using the analytical model to generate training data for the neural network or to provide intermittent real-time validation of the neural network results, e.g. to detect whether or not the inputs to the neural network are outside the training region.

Future work will include developing training sets and neural network models based on predicted discharges to make the model useful for planning future NSTX-U campaigns. Models for other devices will also be developed. Alternative approaches to handling the time-history dependence, including recurrent neural networks, will be explored. The observer design presented in this work will be expanded upon and tested against a database of shots. Following this, the observer will be implemented in the real-time system, and use of the neural network model in feedback control algorithms will be pursued.

Acknowledgments

The authors thank O. Meneghini for helpful discussions and comments on training and ensemble modeling, as well as S. Sabbagh, W. Heidbrink, and D. Liu for encouraging discussions regarding potential applications of the reduced beam modeling approach presented in this paper.

This work was supported by the US Department of Energy Grant under contract number DE-AC02-09CH11466.

Digital data for this paper can be found in: http://arks.princeton.edu/ark:/88435/dsp011v53k0334.

ORCID iDs

M.D. Boyer https://orcid.org/0000-0002-6845-9155
K. Erickson https://orcid.org/0000-0003-1216-8719

References

[1] Poli F., Sachdev J., Breslau J., Gorelenkova M. and Yuan X. 2018 TRANSP (Computer Software) (https://doi.org/10.11578/dc.20180627.4)

[2] Hawryluk R. 1981 An empirical approach to tokamak transport *Physics of Plasmas Close to Thermonuclear Conditions: Proc. of the Course (Varenna, Italy, 27 August–8 September 1979)* vol 1, ed B. Coppi (Varenna: Elsevier) pp 19–46

[3] Artaud J. *et al* 2010 *Nucl. Fusion* **50** 043001

[4] Perverzev G. *et al* 2002 Astra automated system for transport analysis in a tokamak *Technical Report* 5/98 IPP Report

[5] Ou Y. *et al* 2008 *Plasma Phys. Control. Fusion* **50** 115001

[6] Xu C. *et al* 2010 *IEEE Trans. Plasma Sci.* **38** 163

[7] Felici F. and Sauter O. 2012 *Plasma Phys. Control. Fusion* **54** 025002

[8] Barton J. *et al* 2015 *Nucl. Fusion* **55** 093005

[9] Barton J.E. *et al* 2012 *Nucl. Fusion* **52** 123018

[10] Boyer M.D. *et al* 2013 *Plasma Phys. Control. Fusion* **55** 105007

[11] Schuster E. *et al* 2017 *Nucl. Fusion* **57** 116026

[12] Mavkov B. *et al* 2018 *Nucl. Fusion* **58** 056011

[13] Citrin J. *et al* 2015 *Nucl. Fusion* **55** 1

[14] Meneghini O. *et al* 2017 *Nucl. Fusion* **57** 086034

[15] Felici F. *et al* 2018 *Nucl. Fusion* **58** 096006

[16] Menard J. *et al* 2012 *Nucl. Fusion* **52** 083015

[17] Goldston R.J. *et al* 1981 *J. Comput. Phys.* **43** 61

[18] Pankin A., McCune D., Andre R., Bateman G. and Kritz A. 2004 *Comput. Phys. Commun.* **159** 157

[19] Menard J.E. *et al* 2017 *Nucl. Fusion* **57** 102006

[20] Battaglia D. *et al* 2018 *Nucl. Fusion* **58** 046010

[21] Boyer M. *et al* 2018 *Nucl. Fusion* **58** 036016

[22] Ono M. *et al* 2000 *Nucl. Fusion* **40** 557

[23] Sykes A. *et al* 2001 *Nucl. Fusion* **41** 1423

[24] Goldston R. *et al* 2008 *Proc. 22nd Int. Fusion Energy Conf. (Geneva, 2008)* (Vienna: IAEA) (CD-ROM file FT/P3-12 and http://www-naweb.iaea.org/napc/physics/FEC/FEC2008/html/node101.htm#23968)

[25] Stambaugh R.D. *et al* 2010 Candidates for a fusion nuclear science facility (FDF and ST-CTF) P2.110 *37th EPS Conf. on Plasma Physics (Dublin, Ireland)* vol 51

[26] Menard J.E. *et al* 2016 *Nucl. Fusion* **56** 106023

[27] Menard J. *et al* 2011 *Nucl. Fusion* **51** 103014

[28] Kaye S. *et al* 2015 *Nucl. Fusion* **55** 104002

[29] Kaye S. *et al* 2006 *Nucl. Fusion* **46** 848

[30] Kaye S. *et al* 2007 *Phys. Rev. Lett.* **98** 175002

[31] Kaye S. *et al* 2007 *Nucl. Fusion* **47** 499

[32] Valovic M. *et al* 2011 *Nucl. Fusion* **51** 073045

[33] Fredrickson E.D. *et al* 2006 *Phys. Plasmas* **13** 056109

[34] Gryaznevich M. *et al* 2008 *Nucl. Fusion* **48** 084003

[35] Podesta M. *et al* 2009 *Phys. Plasmas* **16** 056104

[36] Fredrickson E.D. *et al* 2009 *Phys. Plasmas* **16** 122505

[37] Fredrickson E.D. *et al* 2017 *Phys. Rev. Lett.* **118** 265001

[38] Fredrickson E. *et al* 2018 *Nucl. Fusion* **58** 082022

[39] Gates D. *et al* 2007 *Nucl. Fusion* **47** 1376

[40] Menard J. *et al* 2007 *Nucl. Fusion* **47** S645

[41] Gates D. *et al* 2009 *Nucl. Fusion* **49** 104016

[42] Gerhardt S. *et al* 2011 *Nucl. Fusion* **51** 073031

[43] Buttery R. *et al* 2004 *Nucl. Fusion* **44** 1027

[44] Chapman I. *et al* 2011 *Nucl. Fusion* **51** 073040

[45] Boyer M. *et al* 2017 *Nucl. Fusion* **57** 066017

[46] Goumiri I.R. *et al* 2017 *Phys. Plasmas* **24** 056101

[47] Ilhan Z.O., Wehner W.P. and Schuster E. 2016 Model predictive control with integral action for the rotational transform profile tracking in NSTX-U *Proc. 2016 IEEE Conf. on Control Applications (Buenos Aires, Argentina)* (IEEE)

[48] Goumiri I. *et al* 2016 *Nucl. Fusion* **56** 036023

[49] Boyer M. *et al* 2015 *Nucl. Fusion* **55** 053033

[50] Vail P.J. *et al* 2019 *Plasma Phys. Control. Fusion* **61** 035005

[51] Gerhardt S. *et al* 2013 *Nucl. Fusion* **53** 063021

[52] Erickson K.G., Boyer M.D. and Higgins D. 2018 *Fusion Eng. Des.* **133** 104

[53] Ou Y. *et al* 2007 *Fusion Eng. Des.* **82** 1153

[54] Weiland M. *et al* 2018 *Nucl. Fusion* **58** 082032