Prediction of electron density and pressure profile shapes on NSTX-U using neural networks

M. D. Boyer[1], **J. Chadwick**[2]

[1]Princeton Plasma Physics Laboratory, Princeton, NJ, USA
[2]Carnegie Mellon University, Pittsburgh, PA, USA

E-mail: mboyer@pppl.gov

February 13, 2021

Abstract. A new model for prediction of electron density and pressure profile shapes on NSTX and NSTX-U has been developed using neural networks. The model has been trained and tested on measured profiles from experimental discharges during the first operational campaign of NSTX-U. By projecting profiles onto empirically derived basis functions, the model is able to efficiently and accurately reproduce profile shapes. In order to project the performance of the model to upcoming NSTX-U operations, a large database of profiles from the operation of NSTX is used to test performance as a function of available data. The rapid execution time of the model is well suited to the planned applications, including optimization during scenario development activities, and real-time plasma control. A potential application of the model to real-time profile estimation is demonstrated.

1. Introduction

The spatial distributions of critical plasma parameters, like electron and ion temperature and density, momentum, and current density, determine the stability and performance of tokamak plasmas, and must therefore be carefully controlled in experiments and future reactors. 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 characteristics. Since first principles modeling of tokamak transport phenomena is computationally challenging, reduced theory-based models are typically used for analysis of experimental results and predictive modeling. These reduced models are used by integrated modeling codes, like TRANSP [1, 2], CRONOS [3], and ASTRA [4]. These codes are steadily improving in physics fidelity, however, they are still very computationally expensive, making them inappropriate for use in real-time control algorithms and making their use in numerical optimization of scenarios time consuming [5]. Light-weight control-oriented physics-based transport models have recently shown promise for scenario optimization[6, 7, 8, 9] and control [10, 11, 12, 13, 14, 15] of plasma profiles. The use of machine learning accelerated 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. Neural networks have recently been applied as a means to rapidly calculate neutral beam source terms [16] and transport fluxes based on the results of first principles calculations [17, 18, 19, 20]. The presented work builds upon these approaches to develop a predictive model of electron density and temperature profiles for NSTX-U.

The NSTX-U spherical torus is a low-aspect-ratio tokamak [21, 22], which enables a more compact device that makes efficient use of magnetic fields for confinement. Experiments on NSTX-U aim to build on the results of NSTX [23] to explore how aspect ratio affects the scaling of confinement quality with power, magnetic field, and plasma current [24] and guide the design of future pilot plants [25]. NSTX-U aims to study key elements of burning plasma physics, which will be critical to ITER and future reactors, including control of non-inductive scenarios [26], fast ion instabilities [27], plasma boundary physics, heat load management [28, 29], as well as stability [30, 31] and disruption prediction [32].

Experiments on NSTX indicated that ion heat transport was well described by neoclassical theory [33, 24]. However, models for electron heat transport, external fuelling, impurity sources, and particle transport are not as well validated. Therefore, rather than developing an accelerated model based on first-principles or reduced models, we consider the development of an empirical neural network model of electron temperature and particle densities. The multi-point Thomson scattering diagnostic on NSTX-U [34, 35, 36] operates at 60Hz, generating up to several hundred snapshots of experimental profiles for each discharge (NSTX-U maximum pulse length is expected to be 5s). This provides a large database suitable for training such a neural network.

The model developed in this work is aimed at real-time estimation, forecasting, and control applications, as well as between-shots actuator planning. Priority is therefore placed on creating a model that executes very quickly, is easy to implement in a real-time environment, and uses data that can be measured or estimated in real-time. The model is also intended for use in accelerated predictive modeling for actuator planning, so use of inputs that can be predicted or controlled in experiments is also favored. With this in mind, inputs that may be hypothesized as being important to profile evolution modeling, like detailed distributions of heating and fueling sources, or wall conditions, are purposely avoided. To help decouple the training of the model from the details of sources, the model is trained to predict the shape of the electron density and pressure profiles, rather than the absolute profile. The volume-averages of density and pressure are considered as inputs to be estimated, measured, or predicted with zero-dimensional particle and energy confinement models. The output dimensionality is reduced through the use of principal component analysis, and a fully-connected neural network architecture is used. An ensemble of models are trained on different subsets of the data, to improved prediction accuracy and provide and estimate of uncertainty.

The paper is organized as follows: Section 2 describes the model development for NSTX-U, including the dataset development and reduction, topology selection, uncertainty estimation, and test results. Section 3 explores the application of the proposed approach to a larger dataset from years of NSTX operation, including the effects of data availability and the feasibility of online learning. An initial application to the between shots analysis of power balance and current drive fractions is described in 4. A discussion of results and future plans is provided in Section 5.

2. Model development for NSTX-U

2.1. Dataset

A database of electron density and temperature profiles was formed from the automatic between shots interpretive TRANSP runs completed during NSTX-U operation. The TRANSP interpretive analysis maps the multi-point Thomson scattering diagnostic spline fits to magnetics-only between shots EFIT equilibrium [37] and stores relevant data on a common time and spatial grid (square root of normalized toroidal flux), providing a consistent dataset for model training. The database consists of 281 shots and 30663 samples. Each sample in the dataset also includes 9 scalars used as input to the neural network: minor radius, major radius, elongation, plasma current, vacuum toroidal field, upper triangularity, and lower triangularity. Symbols for inputs and outputs are provided in table 1. Shots in the dataset were randomly assigned to the training, validation, and testing datasets with probability 0.8, 0.1, and 0.1, respectively. Separation by shot, rather than sample, was performed since the near-stationarity of the flattop of discharges leads to similarity among neighboring time points. Randomly assigning individual samples, rather than shots, would have potentially led to significant overestimation of the generalizability of the trained models. The training dataset was used to optimize model parameters, the validation dataset was used to optimize model topology, and the testing data was reserved to assess how well the model generalizes.

2.2. 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 a neural network to predict each grid point, the profiles were first projected onto a set of empirically derived basis functions. The basis functions for each quantity were chosen by applying principal component analysis to the training dataset and keeping only the most significant modes. Aside from reducing the training dataset size and training time, the use of a reduced basis set helps the neural network model produce spatially smooth output profiles and avoids over-fitting of noise in the profile dataset. Figure 1 shows the explained variance ratio (fraction of the variance in the dataset explained by each mode) for each of the profiles. Nearly all of the variance is explained by a small number of modes (note the logarithmic scale). In the results that follow, modes with an explained variance ratio greater than 1×10^{-3} are kept. This leads to keeping 7 modes for the pressure profile shape and 9 for the density profile shape.

Inputs	
Symbol	**Name**
R_0	Major radius
κ	Elongation
I_p	Plasma current
a	Minor radius
$B_{\phi,v}R$	Vacuum toroidal field
δ_u	Upper triangularity
δ_l	Lower triangularity
$\langle n_e \rangle$	Volume-averaged electron density
$\langle n_e T_e \rangle$	Volume-averaged electron pressure
Outputs	
$n_e/\langle n_e \rangle$	Electron density profile shape
$n_e T_e/\langle n_e T_e \rangle$	Electron pressure profile shape

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

Figure 2 shows the mean profile and first few modes for the electron density and pressure profile shapes. For these profiles, the 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 profiles from shot 204001 in figure 3, illustrating the accuracy with which the profiles can be reconstructed with a reduced number of modes.

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. 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.

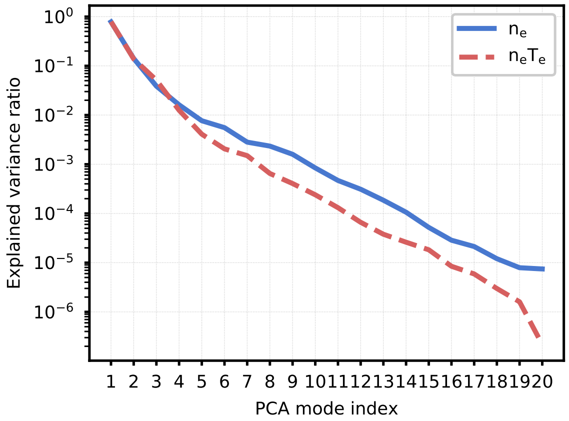


Figure 1: Relative explained variance of principal components for the NSTX-U training data.

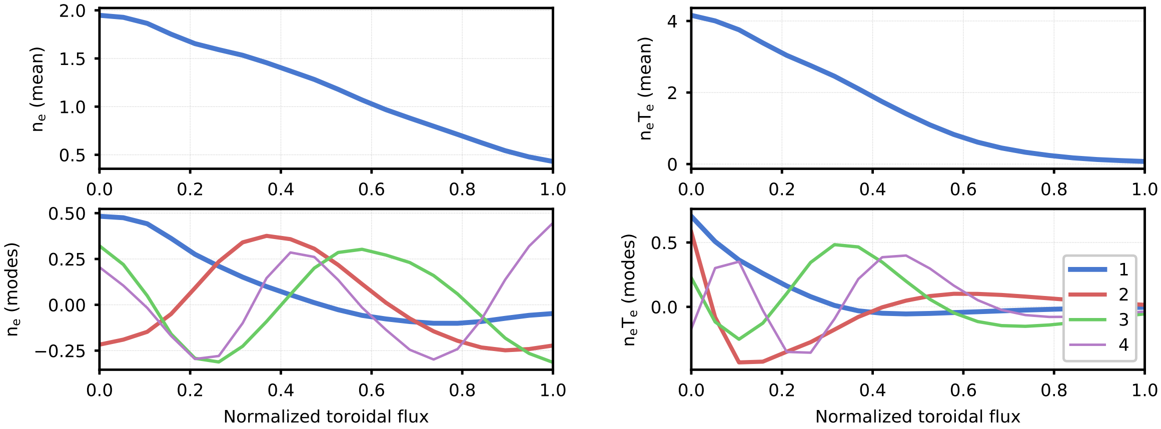


Figure 2: Mean and first 4 modes for NSTX-U profiles: (a) electron density (b) electron pressure.

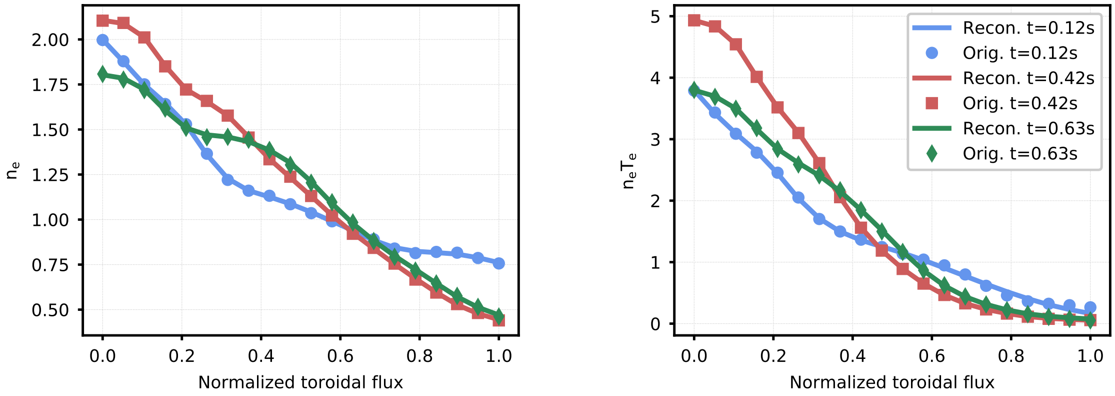


Figure 3: Examples of NSTX-U profiles reconstructed from reduced number of modes (solid line) and original data (circles) at various times for shot 204001.

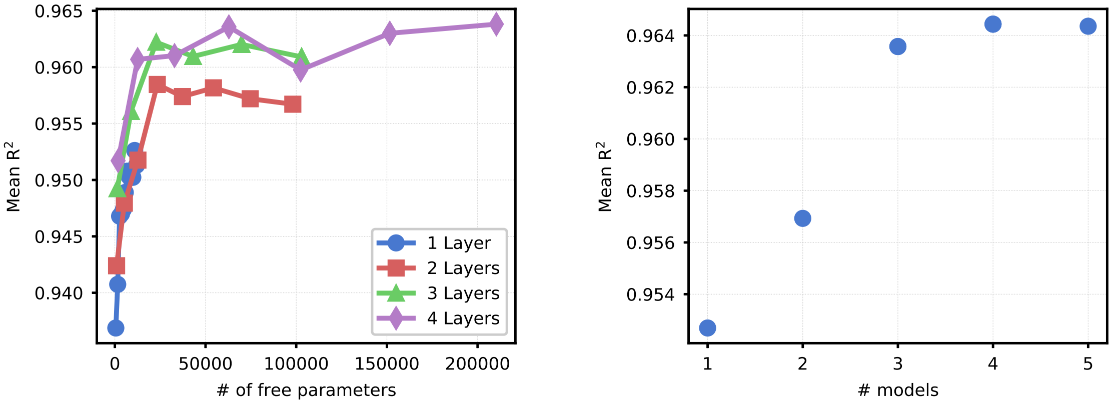


Figure 4: (left) Model prediction accuracy (on the NSTX-U validation dataset) across potential architectures. Each model consisted of a varying number of hidden layers, each with the same number of nodes per layer. An ensemble of 3 models was used for each case. (right) Prediction accuracy (on the NSTX-U validation dataset) as a function of number of models in the ensemble (for model with 4 layers of 140 nodes each).

2.4. 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 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. To compare the performance of different models, the coefficient of determination between predicted and expected (experimental) profiles was chosen. The optimal number of hidden layers and nodes was selected by comparing model performance on the validation dataset in a scan of potential architectures. As shown in Figure 4 (left), the model consisting of 4 hidden layers of 140 nodes each was optimal. Training was done with *Scikit-learn*[38] using the adam optimizer, batch size of 200, learning rate of 0.001, early stopping with a threshold of 0.0001 and 10 iteration tolerance. The size of the ensemble to use was determined by comparing the prediction accuracy on the validation dataset of ensembles of different size using the same topology (4 layers of 140 nodes each) as shown in Figure 4 (right). There is evidently little improvement beyond 3 models in the ensemble.

2.5. Model uncertainty and input sensitivity

For practical applications of the proposed model, it is useful to not only have an accurate prediction of the profile shapes, but also a well calibrated indication of the uncertainty of that prediction. Training ensembles of models with different initial weights and using different subsets of the training data can be used to not only decrease bias, but also provide an indication of uncertainty. In [16], the standard deviation

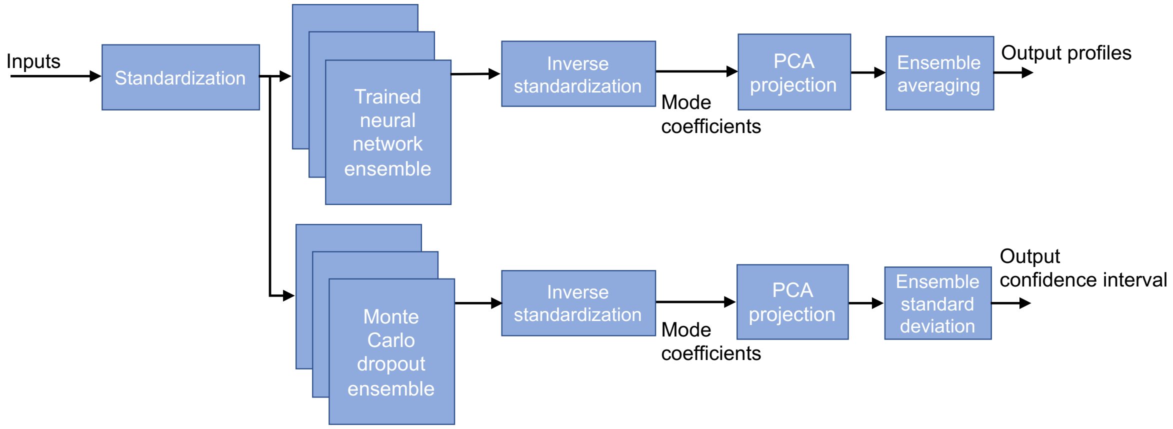


Figure 5: Overview of the proposed model including ensemble averaging and uncertainty estimation.

of ensemble predictions was used as an uncertainty indicator and was observed to qualitatively capture the prediction confidence. It was found that the likelihood of standard deviations observed in the training data could be used to detect when test data fell outside of the range of training data. For the model proposed in this work, however, it was observed that the ensemble standard deviation often significantly over estimates the prediction confidence. The left column of Figure 6 shows validation set prediction errors normalized by the standard deviation of the ensemble of predictions. Comparing to a reference normal distribution, the probability of large normalized errors significantly exceeds the expected probability.

As an alternative to using the standard deviation of the ensemble of trained models, a Monte Carlo drop out approach was tested [39, 40]. In this approach, an ensemble of perturbed models is generated after training an ensemble of models. For each model in the dropout ensemble (10 models were used in the results shown), a model from an ensemble of trained models was selected. Based on performance on the validation dataset, a dropout rate of 0.2 was selected to be applied to each hidden layer (i.e., the output of each node had a 20% chance of being forced to 0) and a dropout rate of 0.3 was applied to the input layer. The mean of the original trained ensemble of 3 models was taken as the prediction, while the standard deviation of the dropout ensemble is used as the model uncertainty estimate. An overview of the complete model including uncertainty estimation is shown in Figure 5. It was found that, although using the adam optimizer and early stopping resulted in more accurate ensemble predictions, the estimation of uncertainty was better calibrated when the Monte Carlo drop out models were drawn from an ensemble trained using stochastic gradient descent without early stopping.

The histogram of prediction errors for the validation data scaled by this uncertainty estimate (Figure 6 middle column) is much closer to the normal distribution than the original estimate. The histogram of prediction errors for the testing data set scaled by the Monte Carlo dropout ensemble standard deviation shows a nearly identical result to

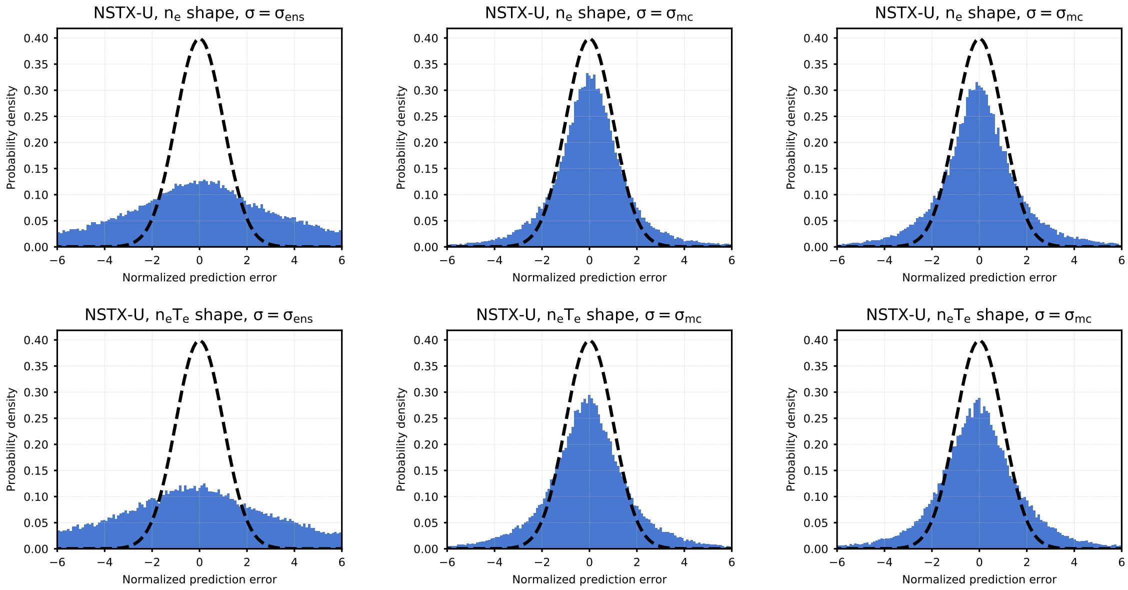


Figure 6: Probability density of prediction errors (normalized by uncertainty estimate) for n_e (top) and n_eT_e (bottom) profile shapes. The left column shows results calculated on the validation dataset with the standard deviation of the ensemble of trained model predictions as the uncertainty estimate. The middle column shows results calculated on the validation dataset with the standard deviation of the Monte Carlo dropout ensemble predictions as the uncertainty estimate. The right column shows the results for the Monte Carlo case evaluated on the test dataset. A reference normal distribution (mean=0, standard deviation=1) is shown as a black dashed line.

the validation data, indicating that the result is robust. If a better calibrated uncertainty estimate is needed, the hyperparameters (number of models and drop out rates) could be more rigorously tuned using the training and validation datasets. Furthermore, an intrinsic uncertainty and a scale factor on the ensemble standard deviations could be fit. Uncertainty quantification in neural network models is an active area of research and other approaches could be considered, for example, uncertainty estimates could be included as an output of a model by replacing the usual mean squared error metric with a metric weighting both prediction error and uncertainty calibration [41].

To assess the sensitivity of the predictions to the input features, a series of predictions were made by setting the input layer weight to zero for one input feature at a time. The prediction quality with each feature removed is compared to the full model in Figure 7. Though this approach ignores correlations among input features, it provides a useful coarse estimate of input sensitivity. Evidently the predictions are most sensitive to the volume averaged electron density and pressure. The next most important inputs are the plasma current and elongation, while the impact of removing any one of the other features is much smaller.

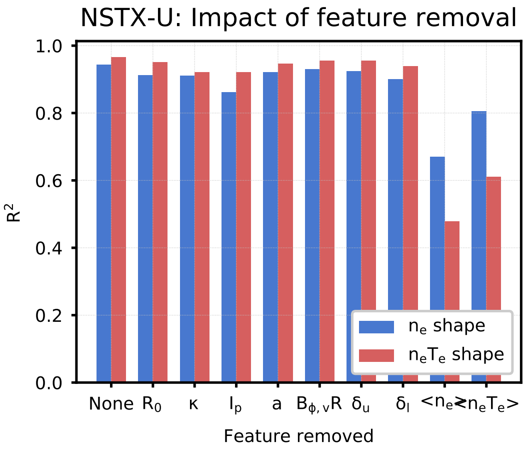
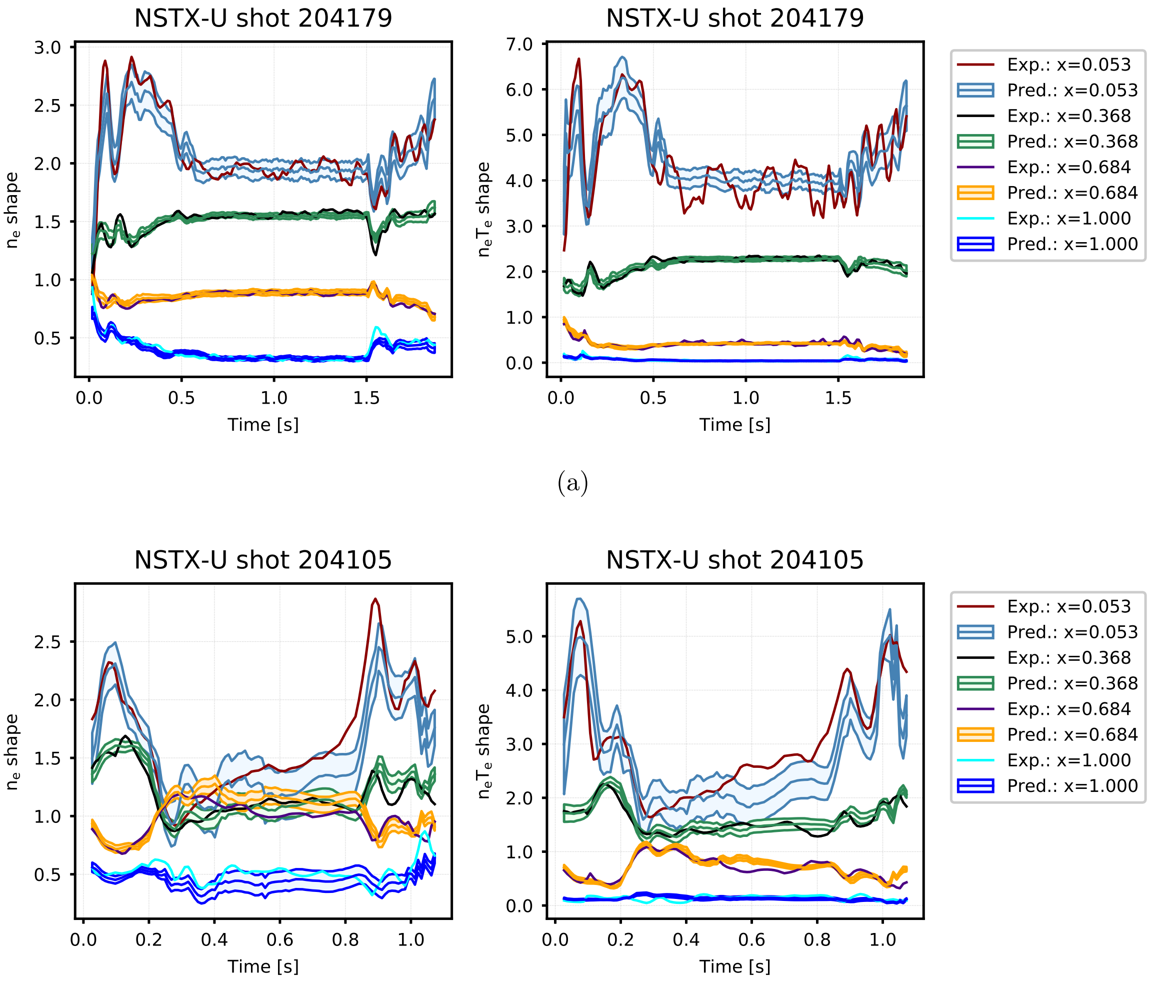


Figure 7: Comparison of prediction quality on the NSTX-U validation dataset with one feature removed from the model with 4 layers of 140 nodes each.

2.6. Predictions of testing dataset

After studying the topology with the validation dataset, the model using 4 layers, 140 nodes per layer, and an ensemble size of 3 was selected and used to predict the output for the inputs in the testing dataset. Figure 8 shows time traces of the model prediction, with the shaded region representing +/- one standard deviation of the Monte Carlo dropout ensemble, compared to the experimental profiles for two shots in the test dataset. Values are shown at $\hat{\rho}$=0.053, 0.368, 0.684, and 1.00. The top plots show results for shot 204179 while the bottom plots show results for shot 204105. These shots were chosen as they demonstrate a variety of profile shapes and time behaviors.

The results generally agree with the experimental values, though there is better matching in 204179 than 204105. It is apparent that the standard deviation is generally larger in the shot with worse predictions, 204105. In this shots, the model seems to struggle most with values very near the magnetic axis and very near the edge, but reproduces values in between very well. It is possible that the core and edge prediction errors are simply a result of having a small training dataset (one campaign of operation) that did not cover the scenario in this particular shot thoroughly. The errors could also be a result of core modes, such as saw teeth, or edge modes (ELMs) transiently modifying profile shapes within the datasets, or sensitivity to detailed source information or wall conditions. Models with an expanded input feature set will be considered in future work to capture the impact of effects like these. Most importantly, the model evidently predicts the changes in profile peaking throughout the discharges. Figure 9 compares the profiles predicted by the model (shaded region represents +/- one standard deviation of the mean of the Monte Carlo dropout ensemble) to the experimental profiles. The top plots show results at t =0.270s and t =0.997s during shot 204179 while the bottom plots show results at t =0.097s and t =0.249s during shot 204105. Despite using a reduced set of profile modes, the predictions are able to accurately reproduce the shapes of profiles.



(a)

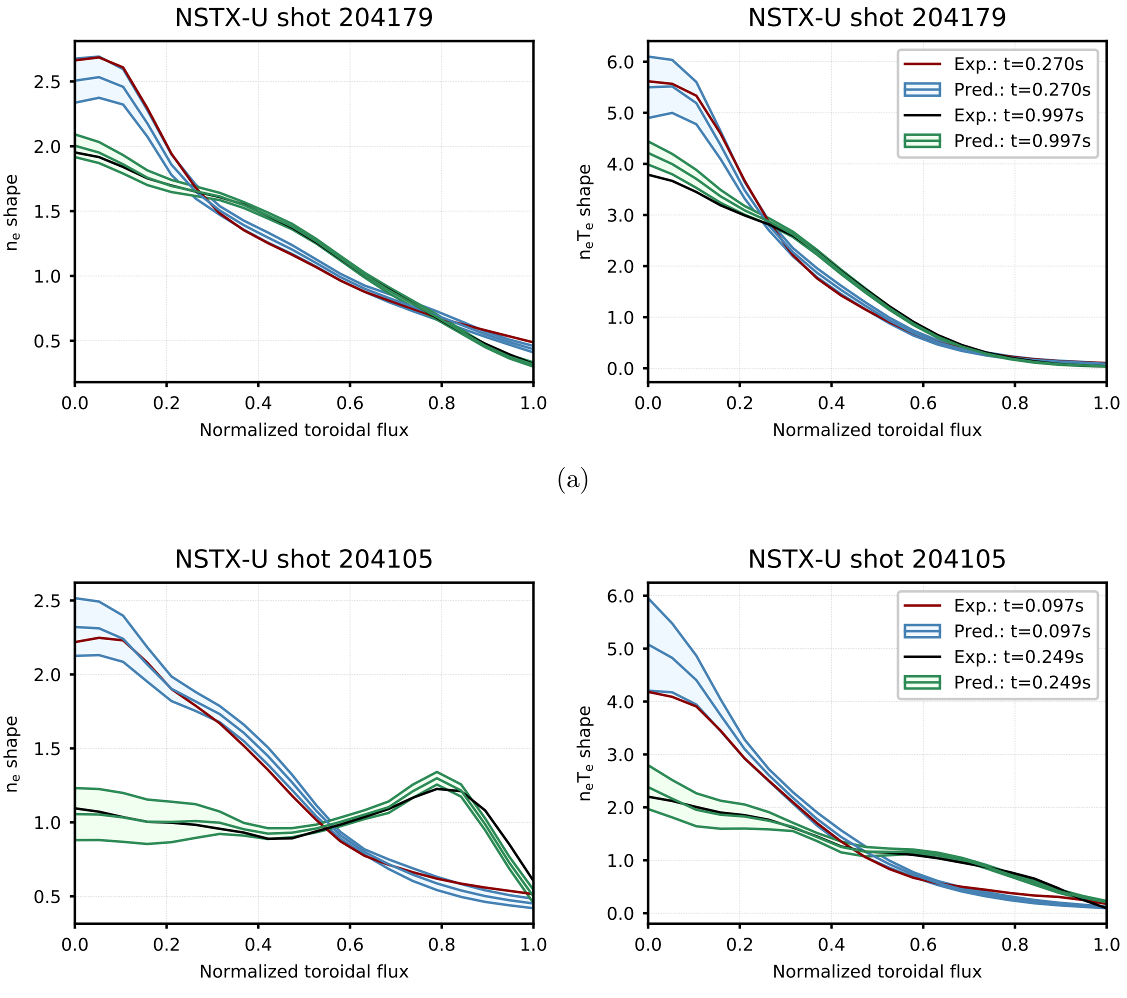
(b)

Figure 8: Time traces comparing model prediction (shaded region represents +/- one standard deviation of the Monte Carlo dropout ensemble) to the expected values at four radial locations for (a) 204179 and (b) 204105. Both shots were in the testing dataset (not used in training or topology selection).

For example the model is able to capture the significant flattening of the density profile in 204105 between $t = 0.097$s and $t = 0.249$s. Finally, regression plots for the entire testing dataset are shown for electron density and pressure in Figure 10. The results show that the model generalizes well to data not used in training or topology selection.

3. Validation of modeling approach using NSTX data

While there is only one run campaign of data available for NSTX-U, there are many years of data available from operation of NSTX. We use this large dataset to validate the modeling approach. In particular, we are interested in exploring how model performance changes based on the amount of data available for training, especially as new data is



(a)

(b)

Figure 9: Comparison of profiles predicted by the model (shaded region represents +/- one standard deviation of the Monte Carlo dropout ensemble) to the expected profiles for (a) 204179 and (b) 204105. Both shots were in the testing dataset (not used in training or topology selection).

acquired during operations.

3.1. Dataset

The same input and output variables described in Table 1 were selected for the NSTX case. The dataset consists of TRANSP interpretive analysis runs on 1837 NSTX shots from 2004-2011. In total, there are approximately 995,000 time slices in the dataset.

3.2. Reduction of profile data

Principal component analysis was applied to the NSTX training dataset to reduce the dataset size, as described in 2.2. Figure 11 shows the relative amount of the variance

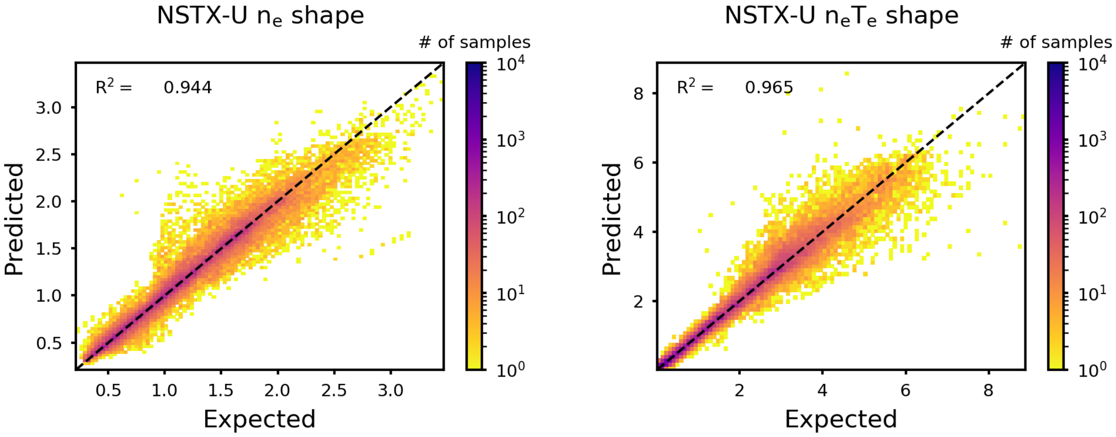


Figure 10: Log-scale histograms of regression results for the testing data set for electron density and pressure profile shapes (includes all radial locations). Results shown for 4 layers, 140 nodes.

in the dataset explained by each mode for each of the profiles. As in the NSTX-U case, most of the variance is explained by only a few modes. The electron density profile data variance on NSTX is evidently described by fewer modes than on NSTX-U. The same criterion of keeping only modes that describe greater than 0.1% of the variance of the dataset was applied, resulting in 8 modes for n_eT_e and 9 modes for n_e. Figure 12 shows examples of the mean profile and modes for the electron density and pressure profile shapes. The mean density profile is less peaked than on NSTX-U, while the density modes have qualitatively similar shapes. The pressure profile is also more peaked on NSTX-U, while the pressure modes exhibit more variation in the edge on NSTX. This is likely due to the relatively low fraction of H-mode discharges in the NSTX-U dataset.

3.3. Training approaches

3.3.1. Randomized data The shots in the entire NSTX dataset were randomly assigned to either the training, validation, or test sets. To compare with the NSTX-U results, an ensemble of 3 models with 4 layers and 140 nodes per layer was trained. Regression plots for the predictions made for the test dataset are shown in Figure 13. The accuracy of pressure profile prediction is similar to the NSTX-U case, while the density prediction accuracy is decreased. To study the learning curve for the problem, the shots in the training set were randomly split into five batches. Five models with four layers of 140 nodes each were then trained, each using a different number of batches of data for training. The performance of these models as a function of the number of training samples is shown in Figure 14, showing that performance improves rapidly at first and more slowly beyond 0.4×10^6 samples. As performance is still improving with all data points included, it is possible that the n_e profile shape predictions could be improved if more data were available. It is also possible that the input features considered,

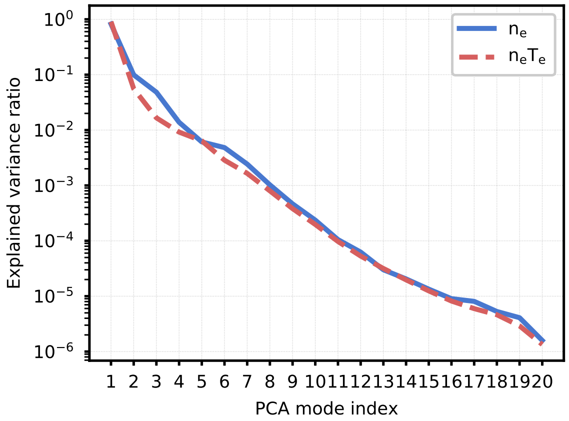


Figure 11: Relative explained variance of principal components for the NSTX training data. Nearly all of the variance is explained by a small number of modes (note log scale). For this model, the first 7 modes were kept for the pressure profile shape and the first 8 kept for the density profile shape.

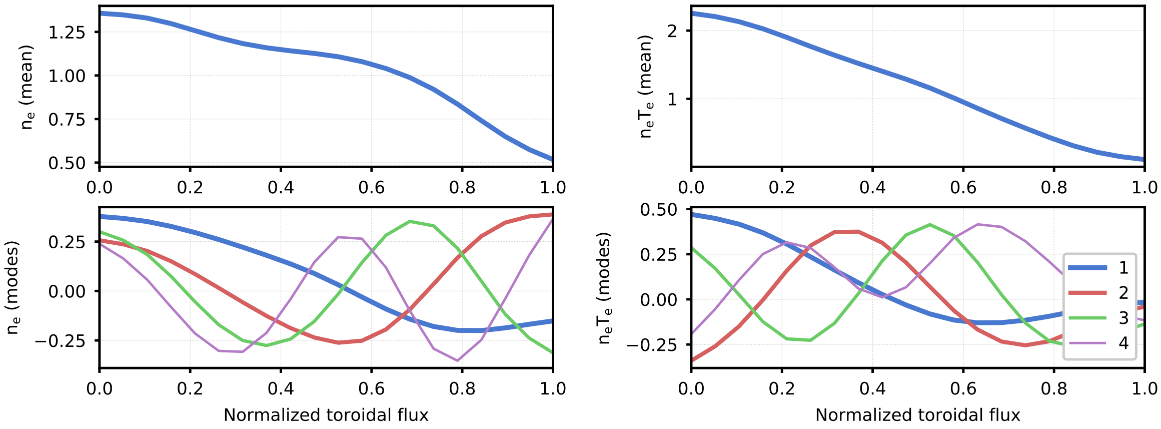


Figure 12: Mean and first 4 modes for NSTX profiles: (a) electron density (b) electron pressure.

which worked well for NSTX-U, do not provide enough information to predict the entire NSTX dataset accurately. For example, NSTX had several modifications during years of operation, and intermittently studied the use of lithium wall conditioning, which could result in variations in profile behavior throughout the dataset.

The assessment of input sensitivity performed in Section 2 was repeated for the model trained on NSTX data. The results, shown in Figure 15 are generally similar, showing the strongest sensitivity to the volume averages of electron density and temperature. Compared to NSTX-U, there is more of an impact when removing some of the other features. This is likely a result of having a wider range of plasma conditions in the larger NSTX dataset.

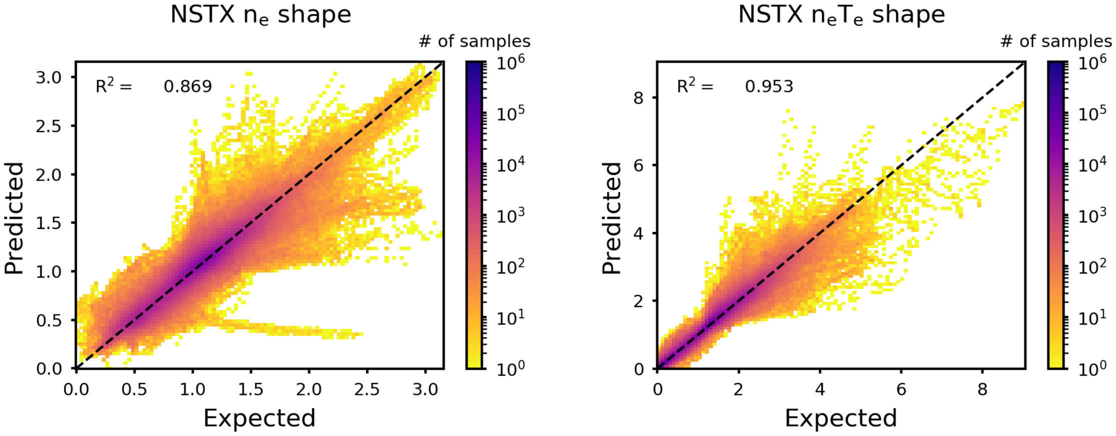


Figure 13: Log-scale histograms of regression results for the NSTX testing data set for electron density and pressure profile shapes (includes all radial locations). Results shown for 4 layers, 140 nodes.

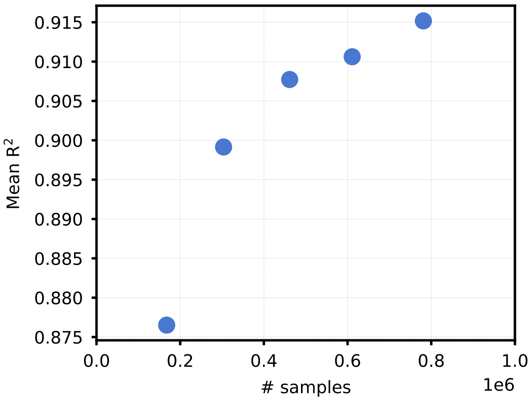


Figure 14: Model prediction accuracy (on NSTX validation dataset) as a function of number of training samples for a model with 4 layers of 140 nodes each.

3.3.2. Online learning The learning curve could be interpreted as an indication that many years of future NSTX-U operation might be required before a fully converged model is possible. In practice, however, data will not be acquired randomly, instead it will only be available from the part of the possible operating space that has been explored so far. Practically, tokamak operation typically involves development of a few experimental scenarios which are then studied in detail through small perturbations. A model that is frequently retrained on available data may quickly learn to accurately predict the behavior of profiles in commonly/recently studied scenarios. As major upgrades to the device are made or new regions of the operating space are explored, the model may not perform as well initially. As long as the learning rate is fast enough, the model could quickly adapt and remain useful for control applications. Furthermore,

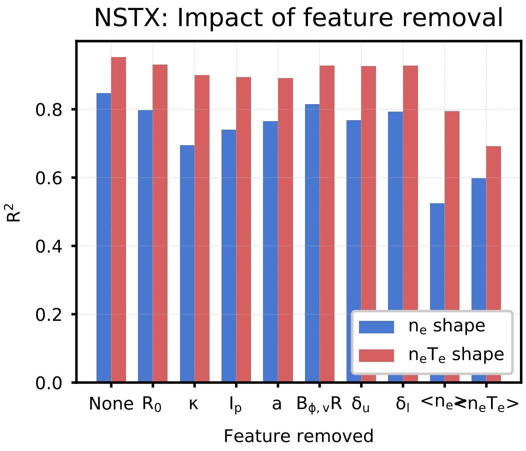


Figure 15: Comparison of prediction quality on the NSTX validation dataset with one feature removed from the model with 4 layers of 140 nodes each.

the uncertainty of the model could potentially be used to guide the choice of plasma parameters to explore in experiments.

To study how the prediction quality of models trained on only available data evolves over the course of many run campaigns, a series of models were trained on the NSTX dataset. Starting with the 50th shot in the training dataset, for every 10th shot, a model was trained on only the training dataset available at the time of the shot. To assess the performance of the models, the validation dataset was evaluated for each model. The sum of squared errors was calculated for each shot in the validation set and divided by the number of samples in the shot. We take the root of this value as the root-mean-squared-error (RMSE) for the shot. For each of the trained models, the average of the RMSE for 5 shots prior to the reference shot used to train the model is plotted in Figure 16 (left). For each reference shot, the RMSE averaged over the same 5 previous validation shots is also plotted for a model trained on the entire NSTX training database. All models used 4 layers of 140 nodes each. Validation scores are quite similar for both models, indicating that the model can quickly learn to explain the available dataset as it is expanded. The distributions of the results are plotted as shaded regions in 16 (right). In these plots, white dots indicate medians, thick lines indicate the interval between first and third quartiles, and thin lines represent the interval between lower and upper adjacent values. The distributions show that the results of the model trained on the available data are only slightly worse than the model trained on all data.

In order to assess the prediction quality on new samples in an online learning environment, all of the shots in the validation dataset were evaluated using the latest available model, i.e., the model in the series of models described in the previous paragraph that was trained based on the highest reference shot number less than the validation shot being evaluated. Figure 17 (left) compares the validation scores for shots evaluated this way with those evaluated using the model trained on all shots. The distribution of results is compared in 17 (right), showing that the median prediction error increases when predicting based on available data and the worst predictions, while rare, are significantly worse than for the model trained on all data.

For a qualitative sense of the impact of this increase in prediction error, time traces for a shot with a prediction error near the median of errors for models trained on available data are shown in Figure 18. The top row shows results for the model trained on all data, while the bottom shows results for the model trained only with available data. In this particular shot, the prediction quality of both models are similar. Although they do not predict the edge density very well throughout the shot or the core pressure between $t = 0.6$ and $t = 0.9$, the predictions still qualitatively describe the changes in shape of the profiles over time. Similarly, traces for a shot with a prediction error near the third quartile are shown in Figure 19. While prediction results with the model based on available data are generally not as accurate as the model trained on all data, the model still qualitatively predicts the change in profile peaking for much of the discharge. The uncertainty estimate associated with the incorrect predictions made by the model trained on available data do not seem to reflect the increased prediction error. This motivates future research into methods for detecting out-of-distribution inputs that were not reflected in the training dataset. Nonetheless, considering 75% of shots in the validation dataset had smaller prediction error than this shot, the model trained on available data appears to perform well enough to be a useful tool for real-time control and scenario planning applications.

This study assumed models were only updated for every 10th shot in the training dataset, and the dataset was generated from manual TRANSP analysis runs, which were not performed on every NSTX shot. TRANSP analysis runs are now automatically run immediately following each NSTX-U discharge, such that it should be possible to generate updated models more frequently. This is expected to improve the predictions of the online learning models. While training on the entire NSTX dataset currently takes on the order of an hour, the code used for training does not yet take advantage of the possibility of parallelizing the training of ensemble members and does not use GPU acceleration. It is anticipated that with these changes, training models between shots will be feasible (the NSTX-U shot cycle is 15-20 minutes). If retraining a complete model becomes infeasible as the dataset grows, approaches such as incremental learning or transfer learning could be used to perform lightweight updates between shots.

4. Example application: profile observer

Real-time estimation of the electron density and pressure is required for feedback control of these profiles, while estimation and forecasting can provide important information for disruption predictors, other control algorithms, and equilibrium reconstructions. Real-time measurements are planned to be available on NSTX-U [42], however, the real-time Thomson system will have limited spatial and temporal resolution (8 channels at 17ms sample time) compared with the offline system.

Based on the execution time of a similar neural network model studied in [16],

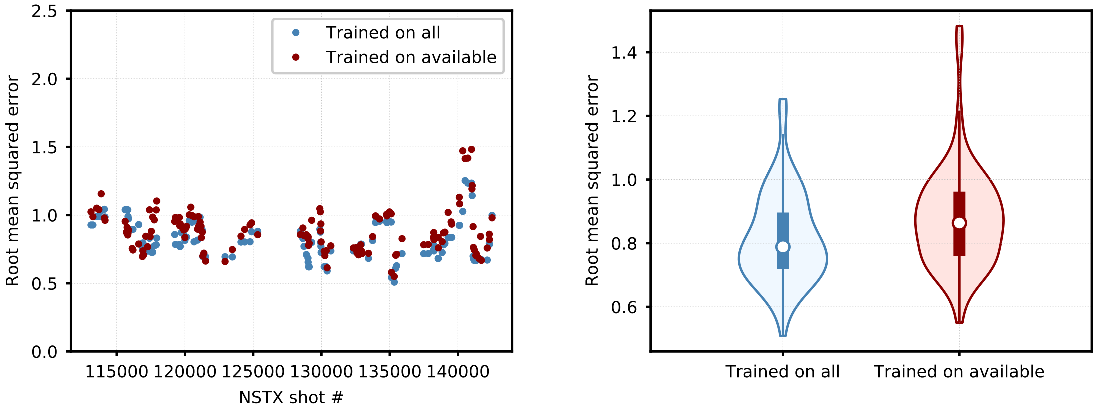


Figure 16: (left) Model prediction errors averaged over the five NSTX validation shots preceding the online learning model reference shots. Results for the model trained on all NSTX training data are compared with the results for the online learning models trained only on available data the time of the model reference shot. (right) Distribution of results for both cases (shaded regions). In these plots, white dots indicate medians, thick lines indicate the interval between first and third quartiles, and thin lines represent the interval between lower and upper adjacent values.

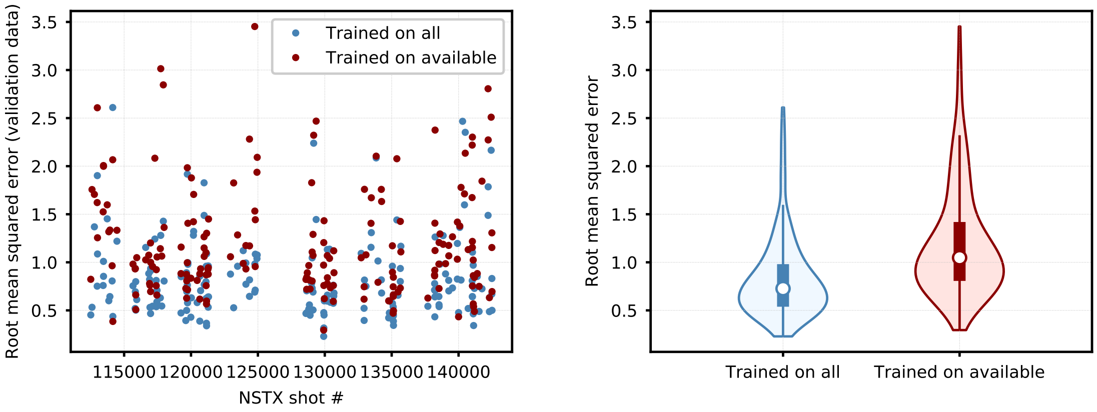
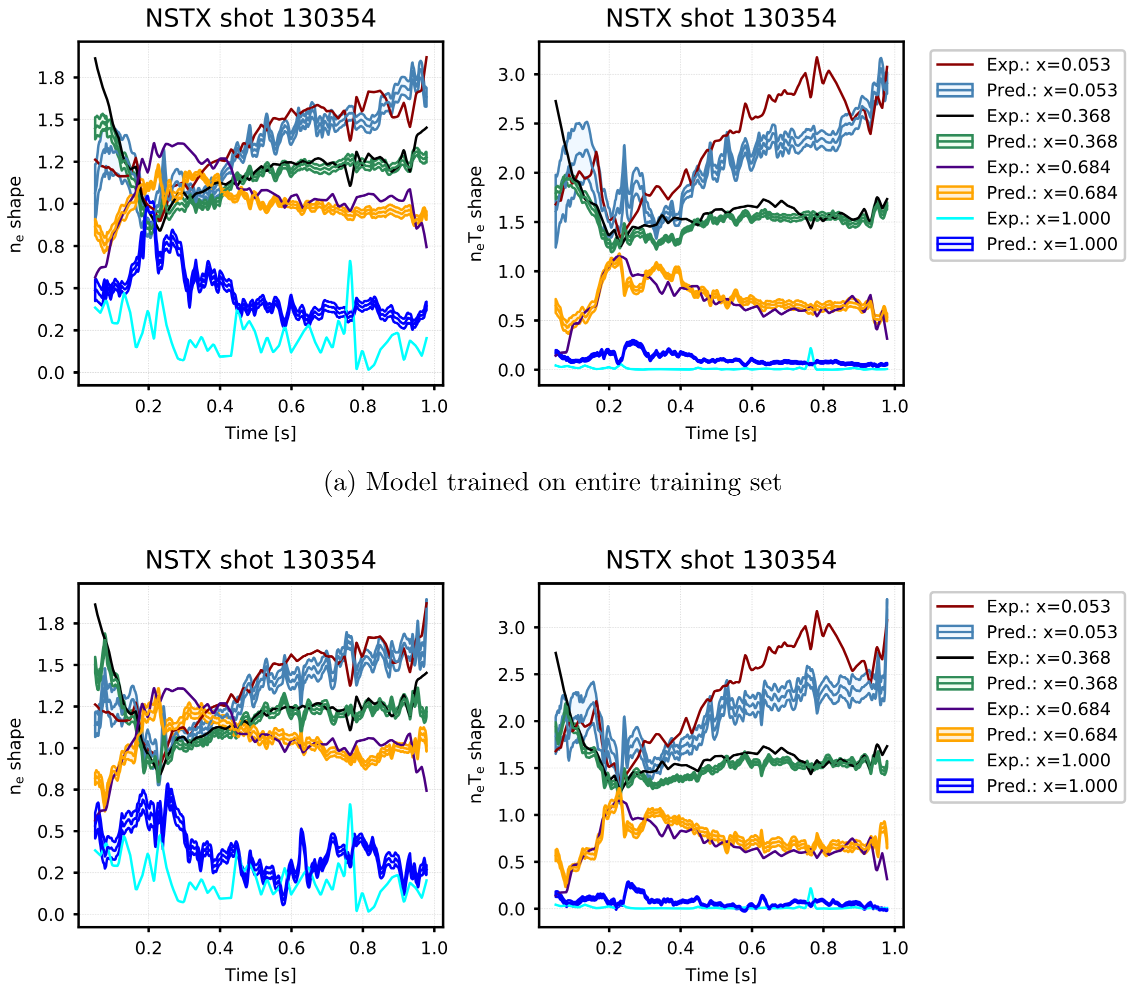


Figure 17: (left) Model prediction errors for NSTX validation shots using the latest available online learning model compared with the model trained on all NSTX training shots. (right) Distribution of results for both cases (shaded regions). In these plots, white dots indicate medians, thick lines indicate the interval between first and third quartiles, and thin lines represent the interval between lower and upper adjacent values.

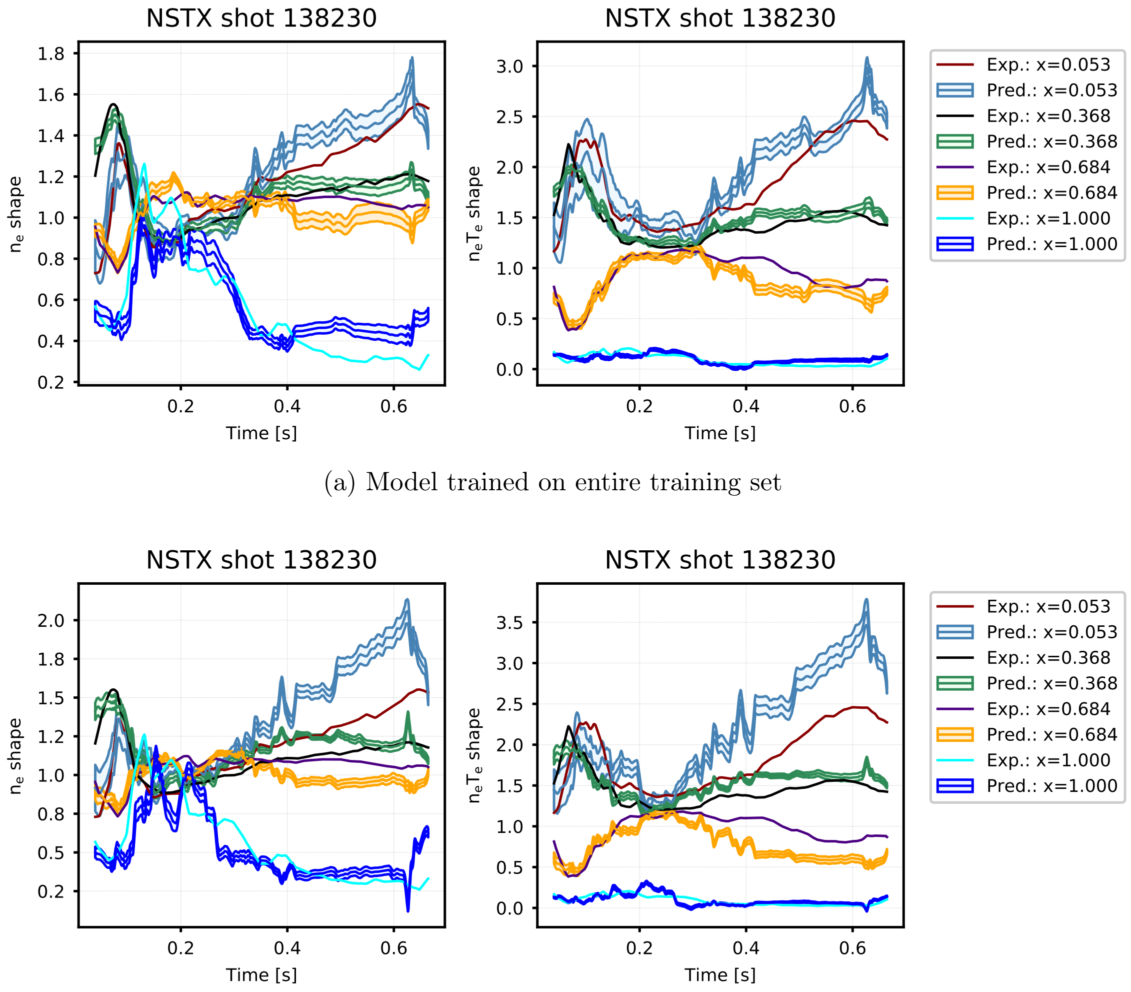


(a) Model trained on entire training set

(b) Latest available online learning model

Figure 18: Time traces comparing model prediction (shaded region represents +/- one standard deviation of the Monte Carlo dropout ensemble) to the expected values at four radial locations for (a) a model trained on all training set shots and (b) the latest available online learning model. Results shown for shot 130354, which had an online learning model RMSE near the median value.

the proposed model is expected to have evaluation times less than 0.5ms. NSTX-U's real-time equilibrium reconstruction, required for providing estimates of plasma boundary shape parameters as input to the neural network, executes in the range of 1-5ms depending on the choice of current distribution models [43]. The volume averaged electron density and pressure can be calculated based on the reconstructed equilibrium and Thomson scattering results. The limiting factor on the update time is therefore the 17ms Thomson scattering sample time. Since the neural network model is trained based on offline measurements, it will capture a much higher spatial resolution than the real-time measurements. Updating the neural network at a higher sample rate will be made possible by estimating the volume averaged electron density and pressure



(a) Model trained on entire training set

(b) Latest available online learning model

Figure 19: Time traces comparing model prediction (shaded region represents +/- one standard deviation of the Monte Carlo dropout ensemble) to the expected values at four radial locations for (a) a model trained on all training set shots and (b) the latest available online learning model. Results shown for shot 138230, which had an online learning model RMSE near the 3rd quartile value.

using confinement models and/or faster diagnostics, e.g., the FIReTIP real-time electron density measurement [44].

In this section, we demonstrate a proof-of-concept Kalman filter for online correction of fast, high spatial resolution profile model predictions (n_g radial points) with low resolution real-time profile measurements (n_{meas} measurement points). For the demonstration, fast estimates of volume averaged electron density and pressure are assumed to be available. We define the model predicted shapes of the two profiles (density and pressure) as $z_{pred} \in \mathbb{R}^{2n_g \times 1}$ and the actual profiles as $z_{exp} \in \mathbb{R}^{2n_g \times 1}$. The residual $\tilde{z} = z_{pred} - z_{exp}$ will be approximately modeled as

$$\tilde{z} = \mathbf{C}_{pca}\hat{x}, \qquad (1)$$

where $\mathbf{C}_{pca} \in \mathbb{R}^{2n_{pca,z} \times 2n_g}$ is a block diagonal matrix formed from the first $n_{pca,z}$ principal components of the density and pressure profiles. The vector $\hat{x}$ represents the to-be-estimated principal component weights. Given the diagnostic response matrix $\mathbf{C}_z \in \mathbb{R}^{n_{meas} \times 2n_g}$, the real-time measurements of profile shape are given by $y_{meas} = \mathbf{C}_z z_{exp}$ while the model predicted measurements are $y_{pred} = \mathbf{C}_z z_{pred}$.

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

$$\hat{x}_{k|k-1} = \mathbf{A}_k \hat{x}_{k-1|k-1} \tag{2}$$

$$\mathbf{P}_{k|k-1} = \mathbf{A}_k \mathbf{P}_{k-1|k-1} \mathbf{A}_k^T + \mathbf{Q}_k \tag{3}$$

$$\tilde{y}_k = y_{meas,k} - \mathbf{C}_k \hat{x}_{k|k-1} \tag{4}$$

$$\mathbf{S}_k = \mathbf{C}_k \mathbf{P}_{k|k-1} \mathbf{C}_k^T + \mathbf{R}_k \tag{5}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{C}_k^T \mathbf{S}_k^{-1} \tag{6}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + \mathbf{K}_k \tilde{y}_k \tag{7}$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k) \mathbf{P}_{k|k-1} \tag{8}$$

where A_k is the system transition matrix and $\mathbf{C}_k = \mathbf{C}_z \mathbf{C}_{pca}$ is the output matrix. The matrices $\mathbf{Q}_k$ and $\mathbf{R}_k$ represent the process 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. For cycles in which new measurements are unavailable, equations (4)-(8) are not evaluated. The estimated profile shapes are finally calculated as $z_{est} = z_{pred} + \mathbf{C}_{pca}\hat{x}$.

For results shown, the process covariance was set to $\mathbf{Q}_k 10^{-4} \mathbb{I}^{n_{2n_{pca,z}} \times n_{2n_{pca,z}}}$ and the measurement noise covariance was set to $\mathbf{R}_k = 10^{-5} \mathbb{I}^{n_{meas} \times n_{meas}}$. The system transition matrix was set to $\mathbf{A}_k = 0.98 \mathbb{I}^{n_{2n_{pca,z}} \times n_{2n_{pca,z}}}$. In practice, these values will be tuned to adjust the sensitivity of the estimator, guided by estimates of uncertainty of the neural network model predictions and the noise in the measurements. For this study, a time step of 0.01s is used for the filter, a state size of 8 is used ($n_{pca,z} = 4$) and $n_{meas} = 8$ equally spaced measurements are considered, with observations taken every 3 cycles (0.03s). Figure 20 compares regression plots for the electron density and pressure profiles for the neural network predictions (top) and Kalman filter estimates (bottom) for NSTX-U shot 204105. By incorporating low resolution measurements, the Kalman filter evidently improves the profile predictions. Time traces of the profiles at four different spatial locations are shown in Figure 21. While the neural network prediction (black dashed line) is typically quite close to the expected value (blue solid line), the Kalman filter estimates (red dash-dot line) generally improve upon tracking of the experimental profile evolution. For example, the bias in the edge density (g) is eliminated by the estimator between t =0.4-0.8s, and the bias in the core pressure (b) and (d) is reduced between t =0.5 and 0.9s. Figure 22 shows a comparison of expected (blue solid), predicted (black dashed), and estimated (red dash dot) profiles at $t = 0.689$s (top) and $t = 0.922$s (bottom). Despite only using low resolution measurements, the estimated profiles generally match the expected values and are typically much more

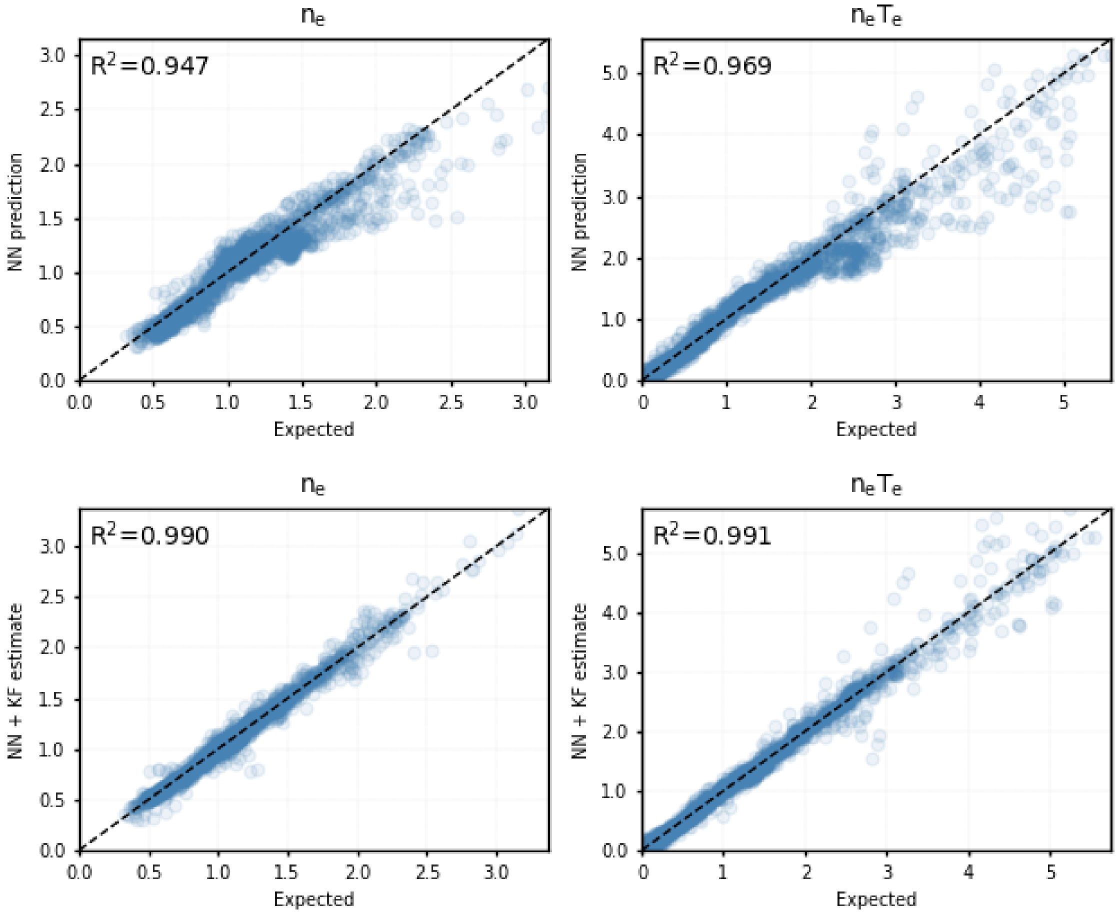


Figure 20: (top) Regression plots for neural network predicted values for NSTX-U shot 204105. (bottom) Regression plots for Kalman filter estimated values.

accurate than the neural network predictions alone. Note that both of these times were chosen to be just before a cycle on which a new measurement was made available to the Kalman filter.

5. Discussion

A neural network model for predicting the shape of electron density and pressure profiles on NSTX-U has been developed. The model was trained on measured profiles from the first NSTX-U campaign. The approach has been validated on the larger NSTX database and the effects of data availability on model performance have been explored. While it takes a large amount of randomly sampled data for performance to converge to a maximum, it is anticipated that in future applications of the model, the weights will be updated as more data becomes available. Tests of incremental training show promising results.

The speed of the resulting model makes it potentially useful for many real-time applications. A design for a Kalman filter is proposed, combining high resolution

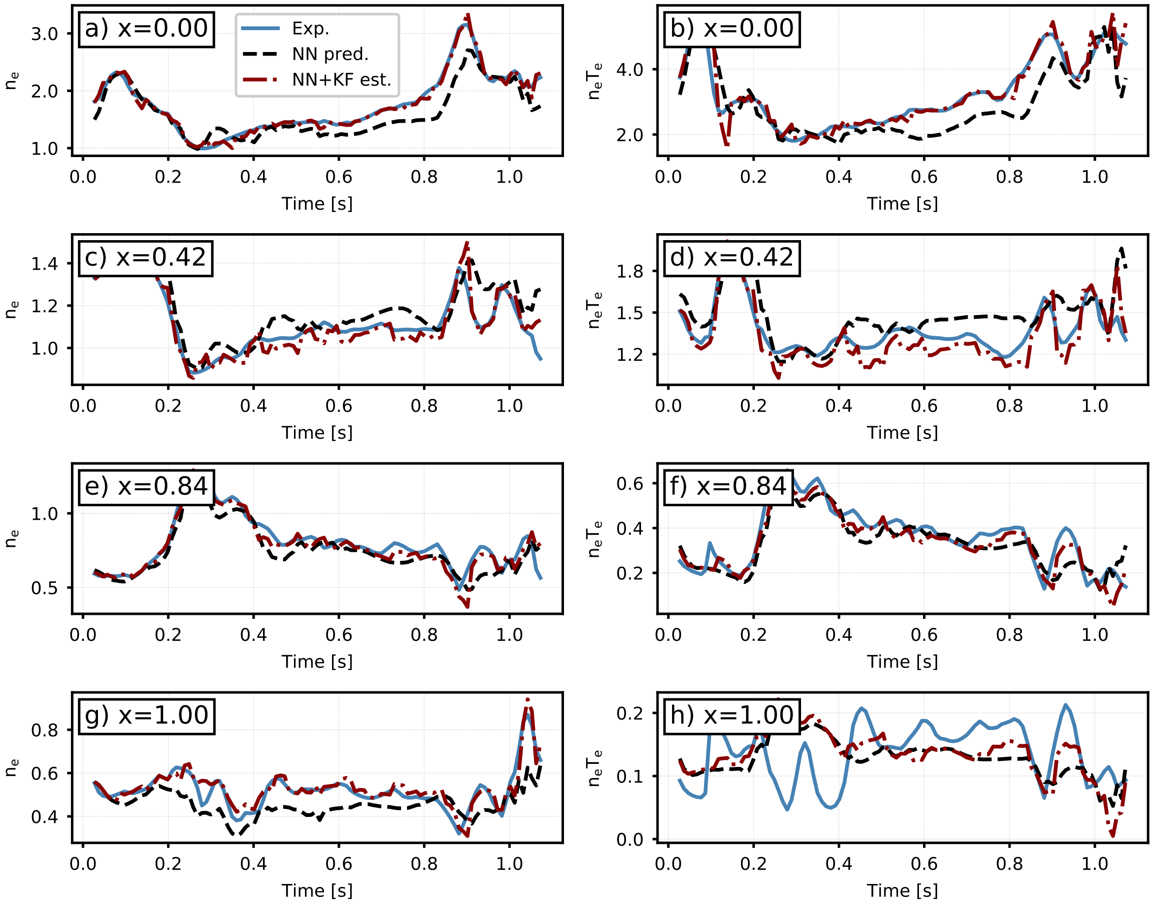


Figure 21: Comparison of time traces of expected (blue solid), predicted (black dashed), and estimated (red dash dot) profiles at several locations for NSTX-U shot 204105.

(but potentially biased) neural network predictions with low resolution real-time profile measurements to estimate and forecast the electron density and pressure profile shapes. Promising simulation results motivate further development and exploration of other potential real-time applications. The model could also contribute to accelerated modeling for numerical optimization of experimental scenarios. An initial version of the model was used to study this approach in [45].

Future work will explore additional inputs, including those related to heating sources, fueling, and wall conditions. Alternative model architectures, including convolutional neural networks, and recurrent neural networks will also be studied. Methods for integrating the proposed model with other accelerated models, like the model for beam heating and current drive developed in [16] and reduced current profile evolution models [11, 46] will be explored. Models for other devices will also be developed to study how the approach generalizes. 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. Improved methods of

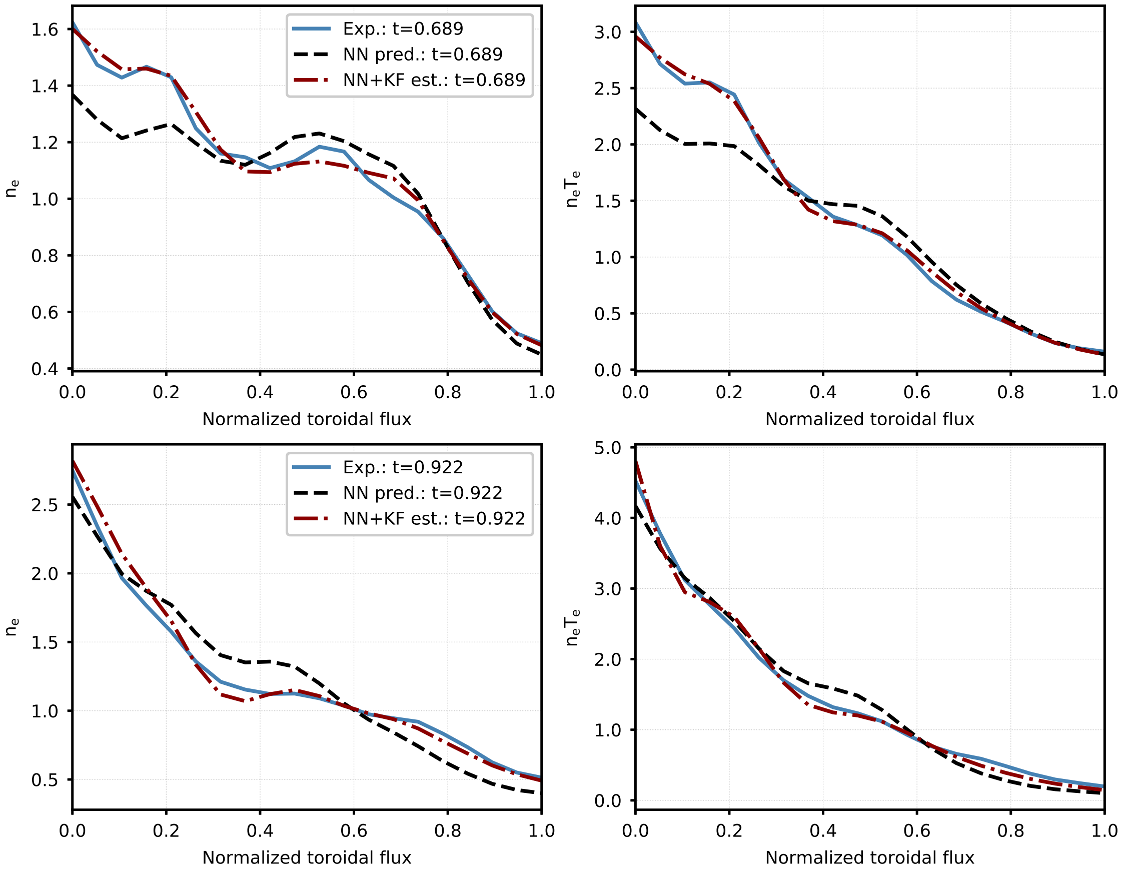


Figure 22: Comparison of expected (blue solid), predicted (black dashed), and estimated (red dash dot) profiles at t =0.689s (top) and t =0.922s (bottom) for NSTX-U shot 204105.

estimating uncertainty and detecting out-of-distribution inputs will also be studied, as this is expected to be important for practical applications of the models.

Acknowledgements

The authors would like to acknowledge Benoit LeBlanc for his work developing and operating the multi-point Thomson scattering diagnostic hardware and processing. This work would not have been possible without the availability of the large database of detailed profile measurements generated by this diagnostic.

The digital data for this paper can be found in: http://arks.princeton.edu/ark:/88435/dsp01j6731612k

This work was supported by the US Department of Energy Grant under contract number DE-AC02-09CH11466. J. Chadwick was supported by funding from the Department of Energy for the Summer Undergraduate Laboratory Internship (SULI) program.

This report was prepared as an account of work sponsored by an agency of the

United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

References

[1] HAWRYLUK, R., An Empirical Approach To Tokamak Transport, in COPPI, B., editor, *Physics of Plasmas Close to Thermonuclear Conditions: Proceedings of the Course Held in Varenna, Italy, 27 August-8 September 1979*, volume 1, pp. 19–46, Varenna, Italy, 1981, Elsevier Ltd.

[2] POLI, F. et al., TRANSP, [Computer Software] doi:10.11578/dc.20180627.4, 2018.

[3] ARTAUD, J. et al., Nuclear Fusion **50** (2010) 043001.

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

[5] WEHNER, W. P. et al., Fusion Engineering and Design **146** (2019) 547.

[6] BARTON, J. et al., Nuclear Fusion **55** (2015) 093005.

[7] FELICI, F. et al., Plasma Physics and Controlled Fusion **54** (2012) 025002.

[8] XU, C. et al., IEEE Transactions on Plasma Science **38** (2010) 163.

[9] OU, Y. et al., Plasma Physics and Controlled Fusion **50** (2008) 115001.

[10] GOUMIRI, I. R. et al., Physics of Plasmas **24** (2017) 056101.

[11] ILHAN, Z. O. et al., Model Predictive Control with Integral Action for the Rotational Transform Profile Tracking in NSTX-U, in *Proc. 2016 IEEE Conference on Control Applications (CCA)*, Buenos Aires, Argentina, 2016, IEEE.

[12] GOUMIRI, I. et al., Nuclear Fusion **56** (2016) 036023.

[13] MALJAARS, E. et al., Nuclear Fusion **55** (2015) 023001.

[14] BOYER, M. D. et al., Plasma Physics and Controlled Fusion **55** (2013) 105007.

[15] BARTON, J. et al., Nuclear Fusion **52** (2012) 123018.

[16] BOYER, M. D. et al., Nuclear Fusion **59** (2019) 056008.

[17] Van De Plassche, K. L. et al., Physics of Plasmas **27** (2020) 022310.

[18] FELICI, F. et al., Nuclear Fusion **58** (2018) 096006.

[19] MENEGHINI, O. et al., Nuclear Fusion **57** (2017) 086034.

[20] CITRIN, J. et al., Nuclear Fusion **55** (2015) 092001.

[21] MENARD, J. E. et al., Nuclear Fusion **57** (2017) 102006.

[22] BATTAGLIA, D. et al., Nuclear Fusion **58** (2018).

[23] KAYE, S. et al., Nuclear Fusion **55** (2015) 104002.

[24] KAYE, S. et al., Nuclear Fusion **47** (2007) 499.

[25] MENARD, J. E. et al., Nuclear Fusion **56** (2016) 106023.

[26] GERHARDT, S. et al., Nuclear Fusion **51** (2011) 073031.

[27] FREDRICKSON, E. et al., Nuclear Fusion **58** (2018) 082022.

[28] VAIL, P. J. et al., Plasma Physics and Controlled Fusion **61** (2019) 035005.

[29] VAIL, P. J. et al., Nuclear Materials and Energy **19** (2019) 516.

[30] BERKERY, J. W. et al., Physics of Plasmas **24** (2017) 056103.

[31] PICCIONE, A. et al., Nuclear Fusion **60** (2020) 046033.

[32] GERHARDT, S. et al., Nuclear Fusion **53** (2013) 063021.

[33] KAYE, S. et al., Physical Review Letters **98** (2007) 175002.

[34] DIALLO, A. et al., Review of Scientific Instruments **83** (2012) 1.

[35] LEBLANC, B. P., Review of Scientific Instruments **79** (2008) 10E737.

[36] LEBLANC, B. P. et al., Review of Scientific Instruments **74** (2003) 1659.

[37] SABBAGH, S. et al., Nuclear Fusion **41** (2001) 1601.

[38] PEDREGOSA, F. et al., Journal of Machine Learning Research **12** (2011) 2825.

[39] GAL, Y. et al., 33rd International Conference on Machine Learning, ICML 2016 **3** (2016) 1651.

[40] ZHU, L. et al., IEEE International Conference on Data Mining Workshops, ICDMW **2017-November** (2017) 103.

[41] LAKSHMINARAYANAN, B. et al., Advances in Neural Information Processing Systems **2017-December** (2017) 6403.

[42] LAGGNER, F. M. et al., Review of Scientific Instruments **90** (2019) 043501.

[43] BOYER, M. et al., Nuclear Fusion **58** (2018) 036016.

[44] SCOTT, E. R. et al., Review of Scientific Instruments **87** (2016) 11E114.

[45] BOYER, M., Toward fusion plasma scenario planning for nstx-u using machine-learning-accelerated models, volume 120 of *Proceedings of Machine Learning Research*, pp. 698–707, The Cloud, 2020, PMLR.

[46] OU, Y. et al., Fusion Engineering and Design **82** (2007) 1153.