Real-time machine learning modeling of pressure and density profiles on NSTX and NSTX-U

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

1 Carnegie Mellon University

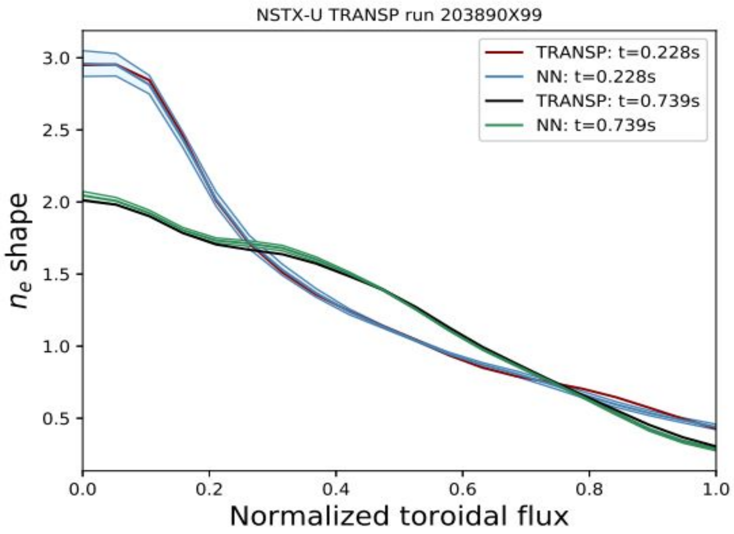
2 PPPL

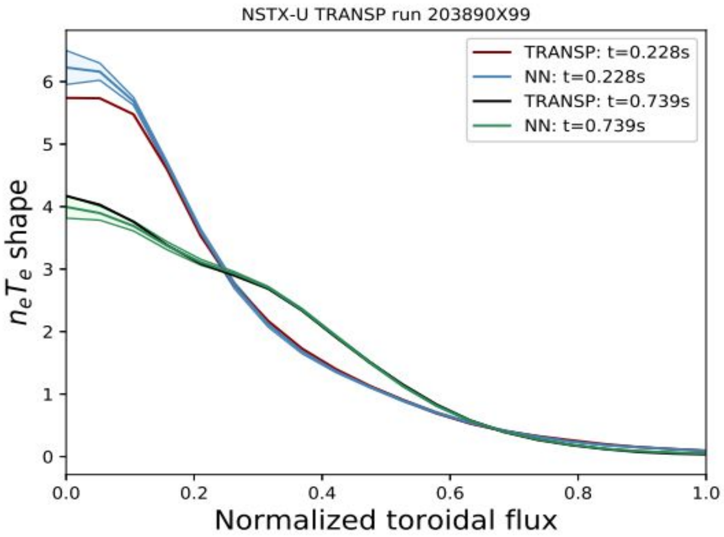
Overview

- Need real-time profile data for control systems
  - Cannot be measured directly
  - Existing code can predict profiles, but too slowly
- Explore the use of neural networks for this (trained/tested on NSTX-U and NSTX data)
- Determine optimal model parameters
- Test model on data outside of training space
  - How far in the future the model can predict
  - How well the model can predict the next shot given all past shots
- Explore methods of establishing confidence in a given prediction

NSTX-U results

- Successful, but NSTX-U dataset is small & does not explore full parameter space
- Want to validate the approach larger dataset





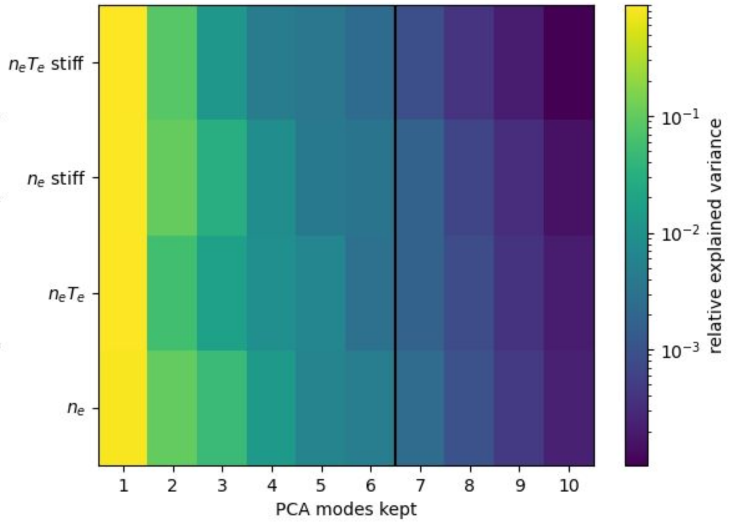
Inputs, outputs, and model topology

Inputs		Outputs	
Symbol	Name	Symbol	Name
R_0	Major radius	n_e	Electron density profile
a	Minor radius	n_e (stiff)	Stiff electron density profile
κ	Elongation	$n_e T_e$	Electron pressure profile
I_p	Plasma current	$n_e T_e$ (stiff)	Stiff electron pressure profile
$B_{\phi,v}R$	Vacuum toroidal field		
σ_u	Upper triangularity		
σ_l	Lower triangularity		
$n_{e,avg}$	Volume average electron density		
$n_e T_{e,avg}$	Volume average electron pressure		

- Hidden layers:
  - 3 fully connected layers of 100 nodes each
- 3-model ensemble
  - 3 models trained on overlapping subsets of training data
  - Gives a mean prediction and a standard deviation

Dataset and preprocessing

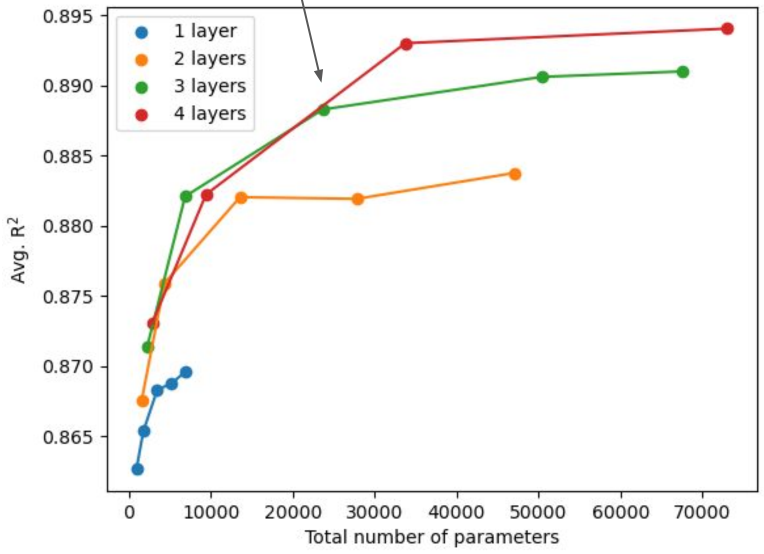
- Dataset: NSTX TRANSP A01 runs from 2004-2011
- 1837 shots in total
- Total of ~995,000 time slices
- 89 measurements per slice
  - 9 real-time scalars
  - 4 profiles, each calculated at 20 radial points
- To reduce dimensionality, used principal component analysis (PCA) to project profile data onto a reduced number of modes (6) before training



Network architecture choice

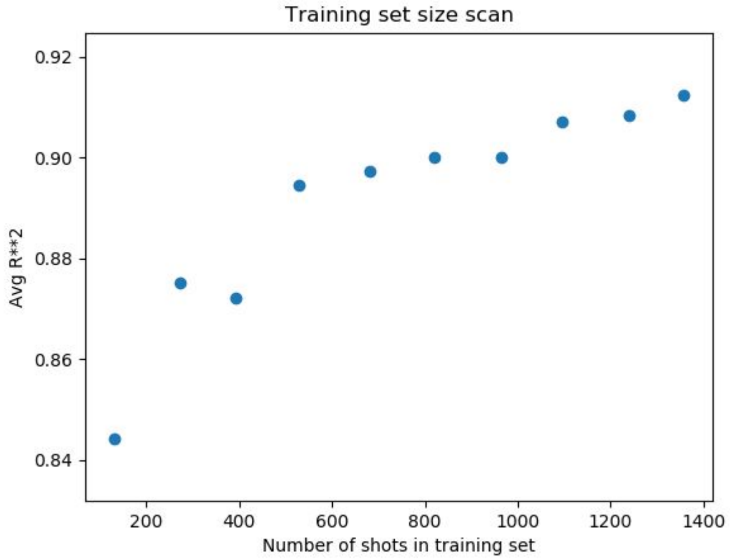
- Scan of models with repeated layers of varying numbers of nodes
- Diminishing returns after ~20,000 params
- Selected architecture consisting of 3 layers of 100 nodes each (23,633 parameters) to balance accuracy and complexity

Chosen model

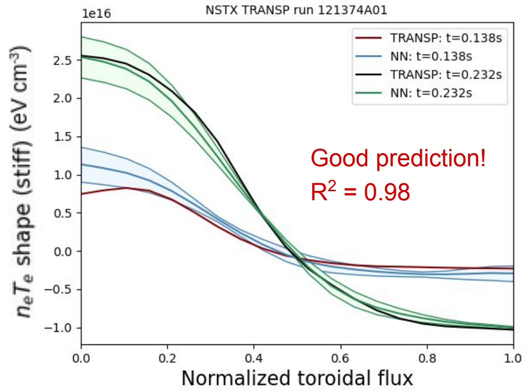


Training set size

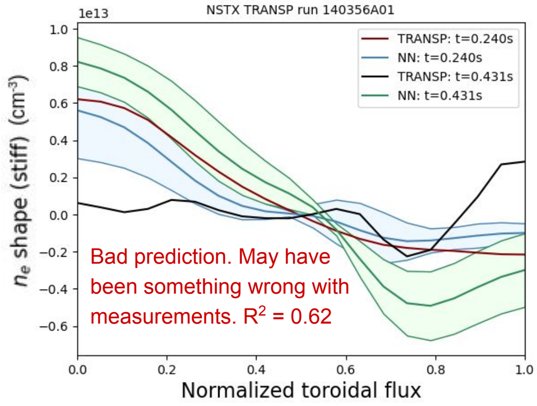
- Trained on random subsets of training data and tested on full test set
- Results: Continually improves with more data, but improvement slows after ~600-700 shots
  - More data is always good, but can get reasonable results without all of it



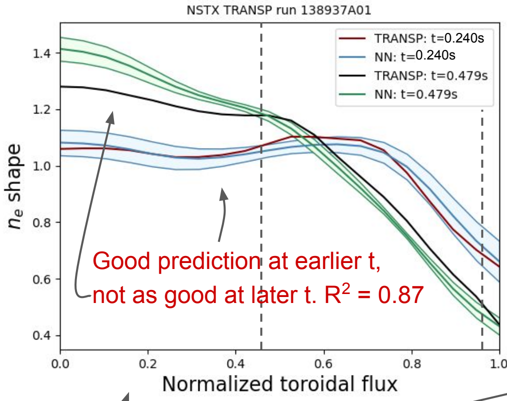
Results: profile examples



Good prediction!
$R^2 = 0.98$

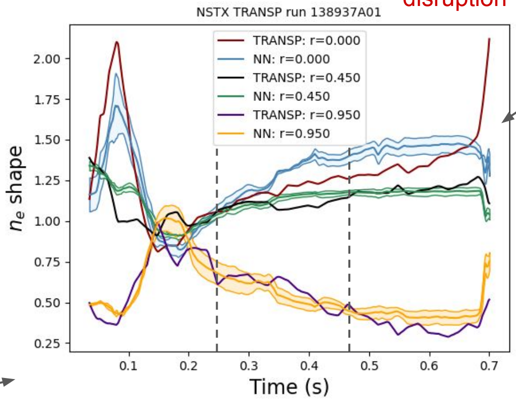


Bad prediction. May have been something wrong with measurements. $R^2 = 0.62$



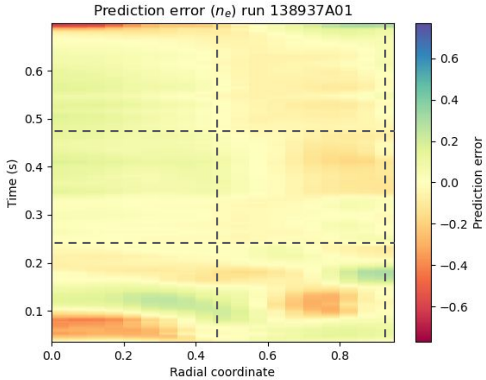
Good prediction at earlier t, not as good at later t. $R^2 = 0.87$

Bad predictions at disruption



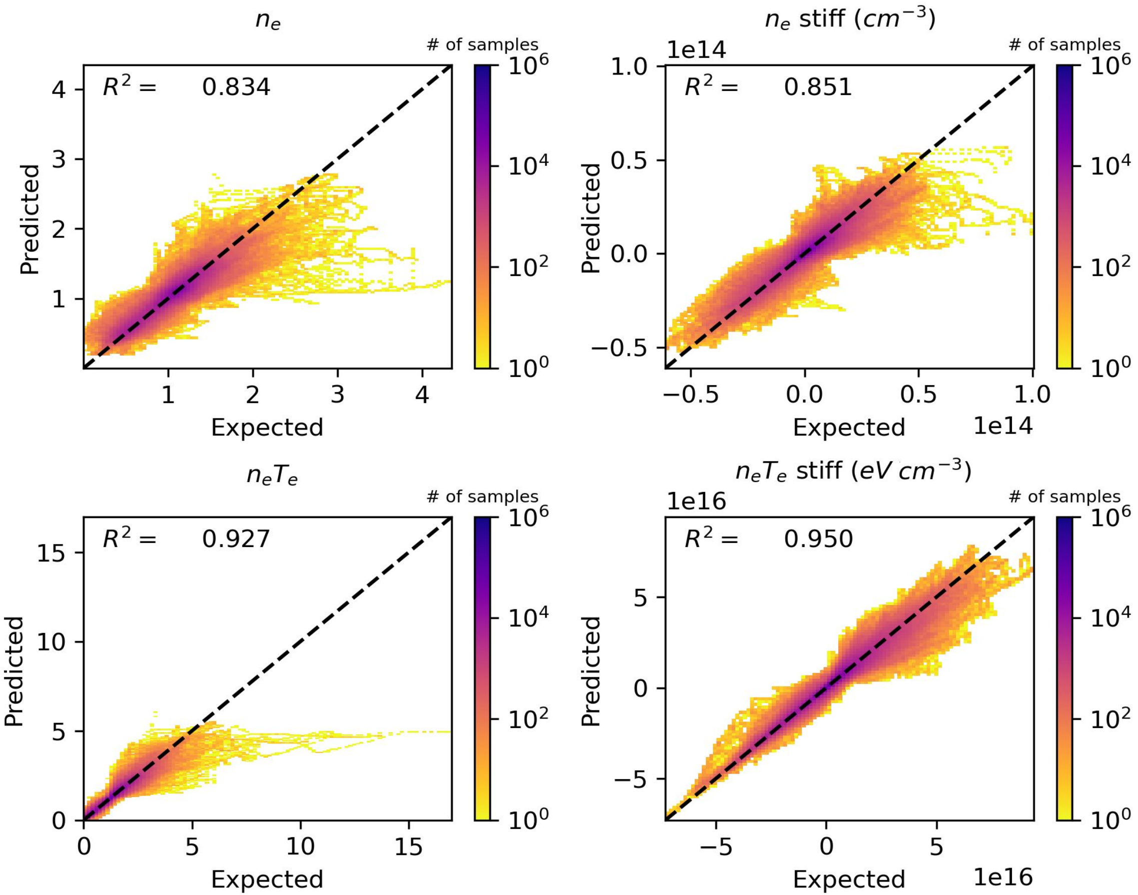
Radial, time-varying, and 2D prediction error plots for shot 138937 n_e

Prediction generally good except for very early and very late times

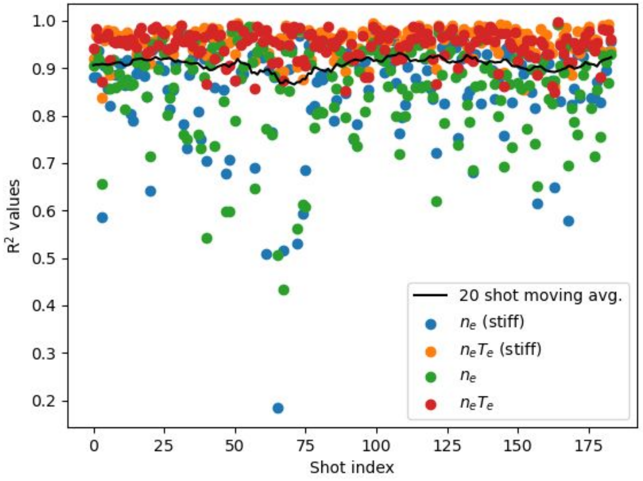


Results: regression plots

- Electron pressure prediction generally much better than density prediction
- Stiff profile prediction slightly better than scaled profile prediction

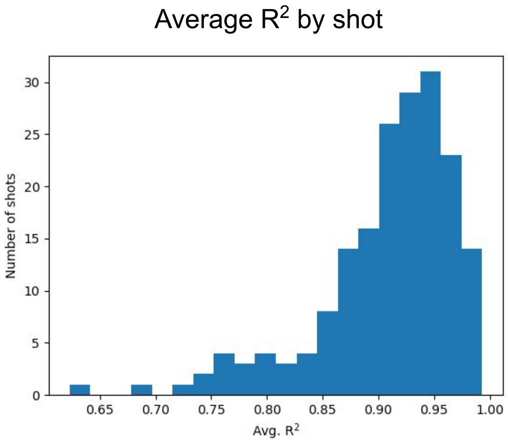


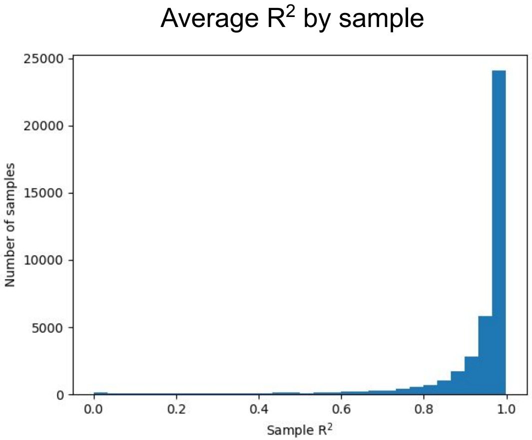
Results: R^2 values by shot



- n_e values generally harder to predict
- Overall average is relatively constant across the dataset

- R^2 by shot mostly above 0.85

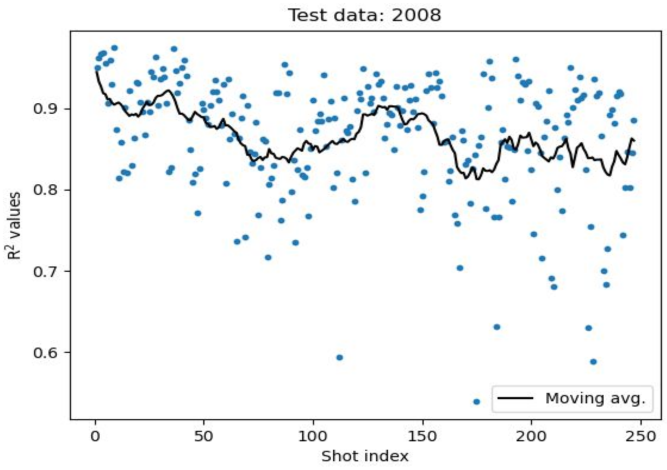


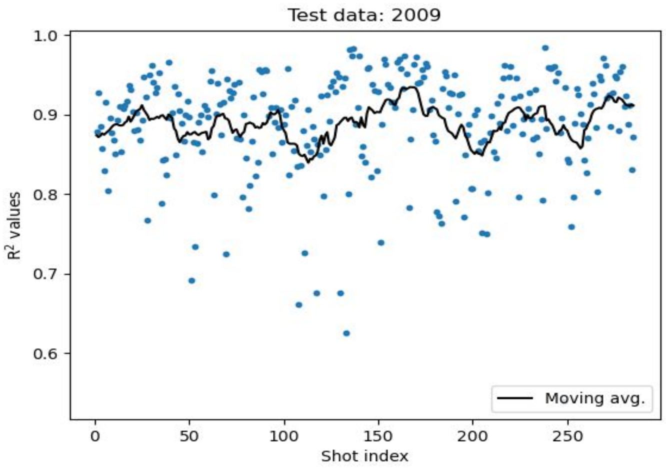


- Can we predict when a given sample will have a low R^2?

Accurate predictions outside of training space

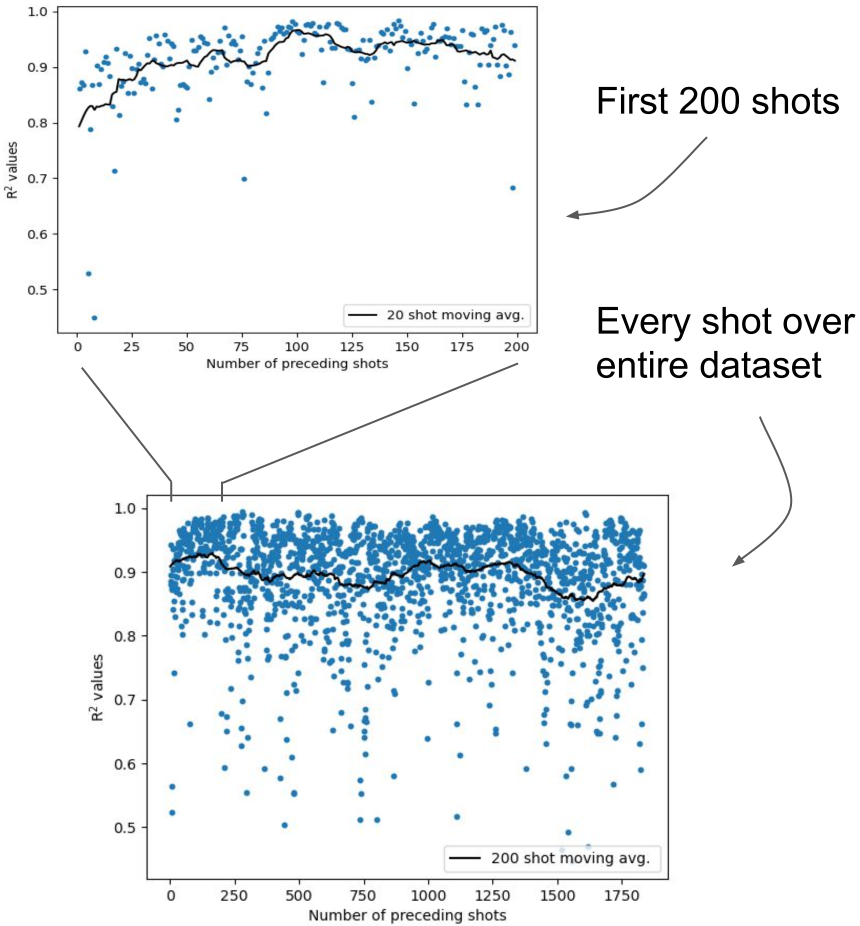
- Trained on all data prior to a certain year, then tested on all data from that year
- Results: relatively consistent performance, even predicting 250 shots into the future





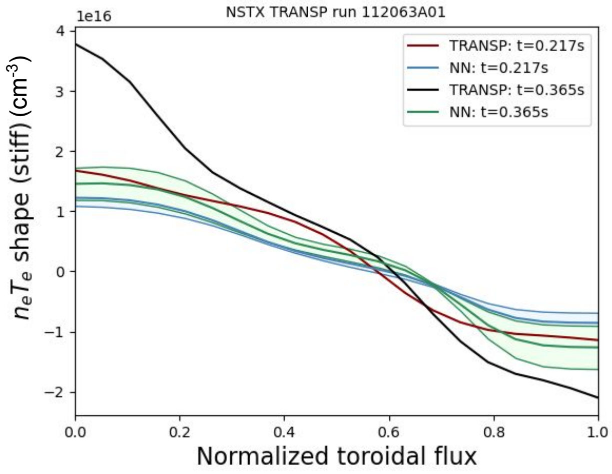
Prediction of next shot

- Method: train on all shots before a certain shot, then test on that shot
    - Simulates the process that would be used in an ongoing experiment
- Results: Good average predictions throughout dataset
    - Most are good, but still a significant number with very low R^2
    - Average drops after ~250 shots; experiments were possibly exploring new areas of operating space



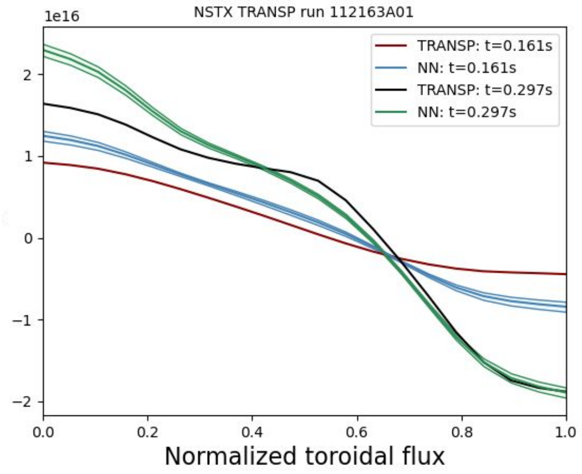
Accurate prediction of next shot: Profile examples

10th shot in set



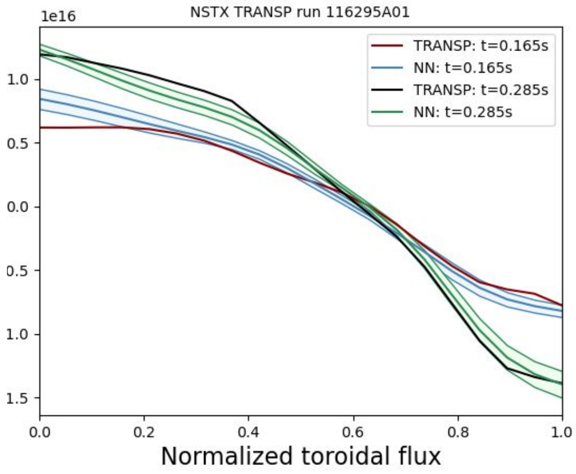
Poor agreement

30th shot in set



Better agreement (usable for real time prediction)

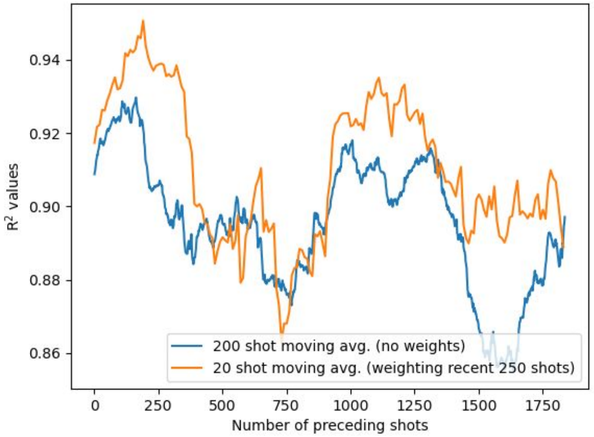
300th shot in set



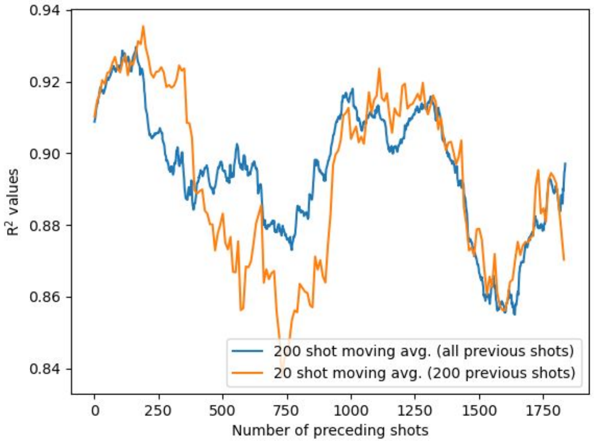
Very good agreement

Improving next-shot prediction

- Possible solution: weight most recent 250 shots 3x as much
- Result: significant improvement, especially around 1500th shot



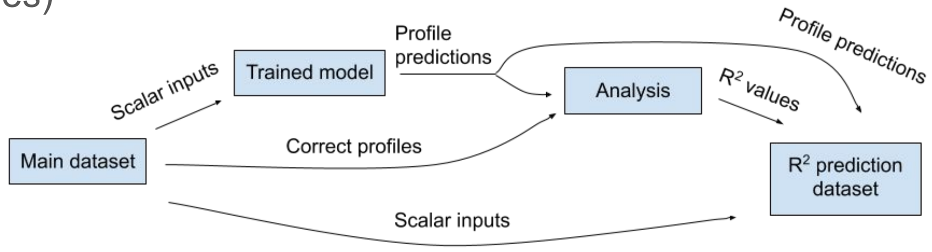
- Possible solution: only train on previous 200 shots
- Result: makes predictions worse around shots 500-1000



Note: these strategies were only tested on every 10th shot in dataset due to high computational cost

Predicting inaccuracies

- Trained a network to predict the main model’s output R^2 values for each time slice in model test set
- Original model test set: 184 shots, ~95k samples
- Trained a new network on results of predicting test set
  - Inputs: original input scalars (9 scalars), predicted profile PCA components (4 * 6 = 24 scalars)
  - Outputs: model prediction R^2 values (4 values: n_e, n_eT_e, n_e stiff, n_eT_e stiff)
  - Training data: 90% of test set results (~85k samples)
  - Test data: remaining 10% (~9.5k samples)
- Evaluated model’s ability to predict when R^2 will fall below a given threshold (e.g. 0.80)



Predicting inaccuracies

Threshold: $R^2 \leq 0.80$

R^2 category	Total expected below threshold	Total predicted below threshold	Num. overlapping	Num. false positives	**False positive rate**	Num. false negatives	**False negative rate**
n_e	1566	2252	1367	885	**0.11**	199	**0.13**
n_e (stiff)	1690	2478	1495	983	**0.11**	195	**0.12**
n_eT_e	50	104	45	59	**0.01**	5	**0.10**
n_eT_e (stiff)	309	484	279	205	**0.02**	30	**0.10**

- Results: helpful, but can be refined

Machine learning library choice

- Switched from scikit-learn to TensorFlow/Keras
  - Used MLPRegressor in sklearn and keras.models.Sequential in TensorFlow
- Benefits of TensorFlow:
  - Many more customization possibilities
    - Can customize loss function (could include physical constraints, e.g. conservation of energy)
    - Can include sample weights
  - Much faster local training out of the box
  - Allows for straightforward training on Princeton GPU cluster
  - Can experiment with recurrent networks in the future

Conclusions

- A neural network is capable of reliably reproducing TRANSP profile predictions for most shots in the dataset
  - Promising for control system applications
- Approach was effective on both NSTX and NSTX-U
- Model predicts electron pressure well, but we still need to improve density predictions
- Model is capable of predicting future shots that are not in the training space
- We have reasonable measures of model confidence

Future work

- Include time history when predicting later times in the same shot
  - Include filtered scalar inputs
  - Include preceding time slice profile prediction as input
  - Recurrent neural network
- Determine scalar inputs needed to improve n_e profile prediction
  - Could help guide future reactor design
- Improve measures of model confidence
  - Improve prediction of R^2 values
  - Find ways to rigorously define the training set parameter space, so we can know when we are outside of it
- Test technique on other machines (DIII-D etc.)

Acknowledgements

This work was made possible by funding from the Department of Energy for the Summer Undergraduate Laboratory Internship (SULI) program. This work is supported by the US DOE Contract No. DE-AC02-09CH11466.