11 November 2020

EUROfusion

Pedestal Neural Network (PENN) for ETS WIMAS-2 Sprint: Core-Edge, part 2 - progress and results

Andreas Gillgren, Dimitriy Yadykin, Pär Strand
Chalmers University of Technology

andreas.gillgren@chalmers.se

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

Outline

- Goals
- PENN in ETS: Test cases setup
    - Test 1: Comparison when PENN is active and not active
    - Test 2: Rhotor scan
    - Test 3: Comparison of PENN in ETS 6 and ETS 5
- To be done: realistic test case for JET and AUG
- How to deal with ion density in PENN?
- Requests

Backup

- What is PENN?
- Upgrading PENN (preliminary results)
    - Predicting entire edge profiles
    - Quantification of prediction uncertainty

Goals

- Run ETS 6 with or without PENN and compare (DONE)
- Test different positions of the pedestal position (DONE)
  - We used to set rho_tor_norm = 0.95 before, but database suggests 0.98 (at least for temperature)
- Verification of ETS 5 and ETS 6 (DONE)
- Realistic test case for JET and AUG (NOT DONE)
- Find proper way to handle ion densities (NOT DONE)

PENN in ETS: Test cases setup

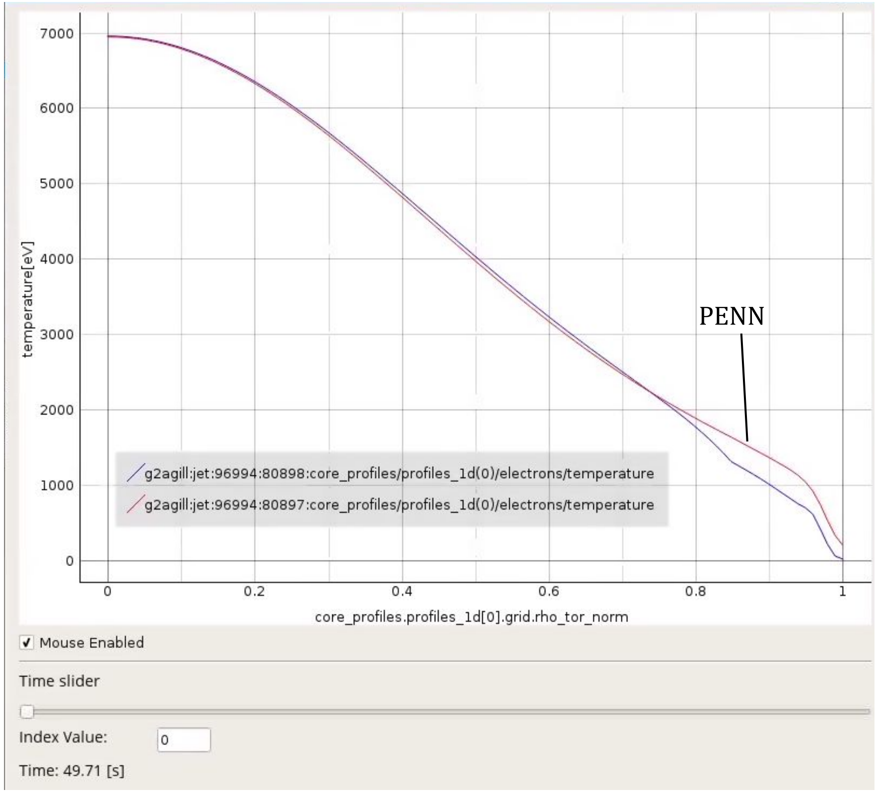
Settings for the following test cases:

- Input: g2mporad/jet/96994/808
- tstart=49.7s, tend=51.7s, dt=0.01s
- plasma composition: e, D,H
- execution: predictive Te, TH, TD; static nD,nH; from quasineutrality ne
- transport: TCIAnalytic: 1.0 m^2/s radially constant for Te, TH, TD
- sources: Gaussian with Power distribution as: 14 MW for e, 14 MW for D, 4 MW for H (32 MW in total)
- Boundary Condition position: 0.85 (normalized rhotor)

Parameter files in ETSwf/param/jet/
param_training_jet_96944_80897_predict_te_ti_penn.txt
param_training_jet_96944_80898_predict_te_ti.txt

Test case 1: Comparison when PENN is active and not active

Initial profiles

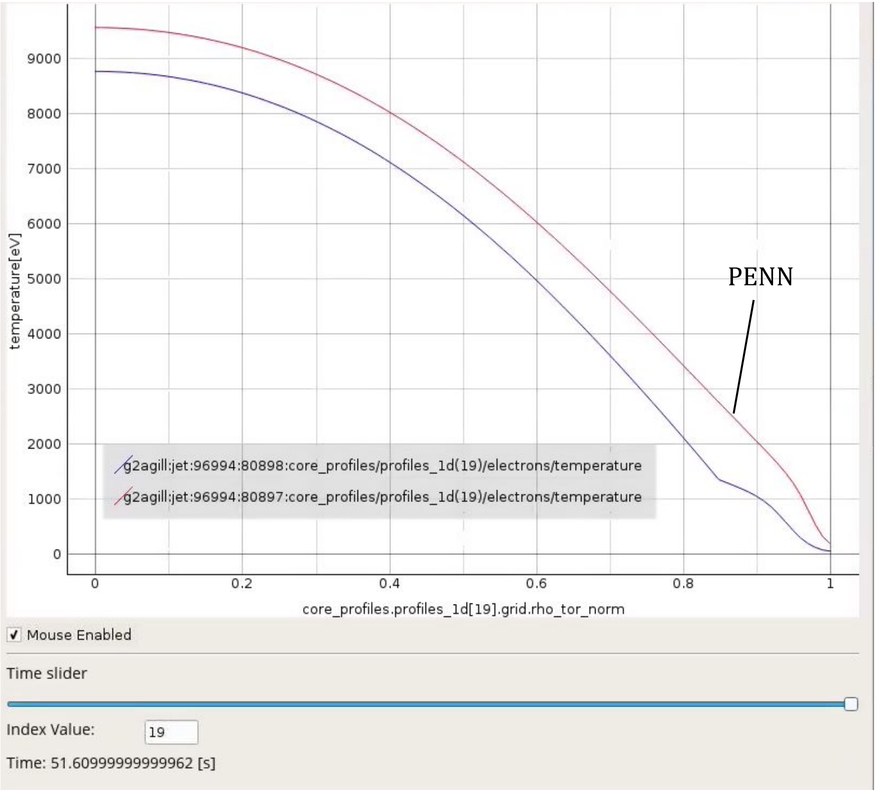


☑ Mouse Enabled

Index Value: 0

Time: 49.71 [s]

2 seconds after evolution



☑ Mouse Enabled

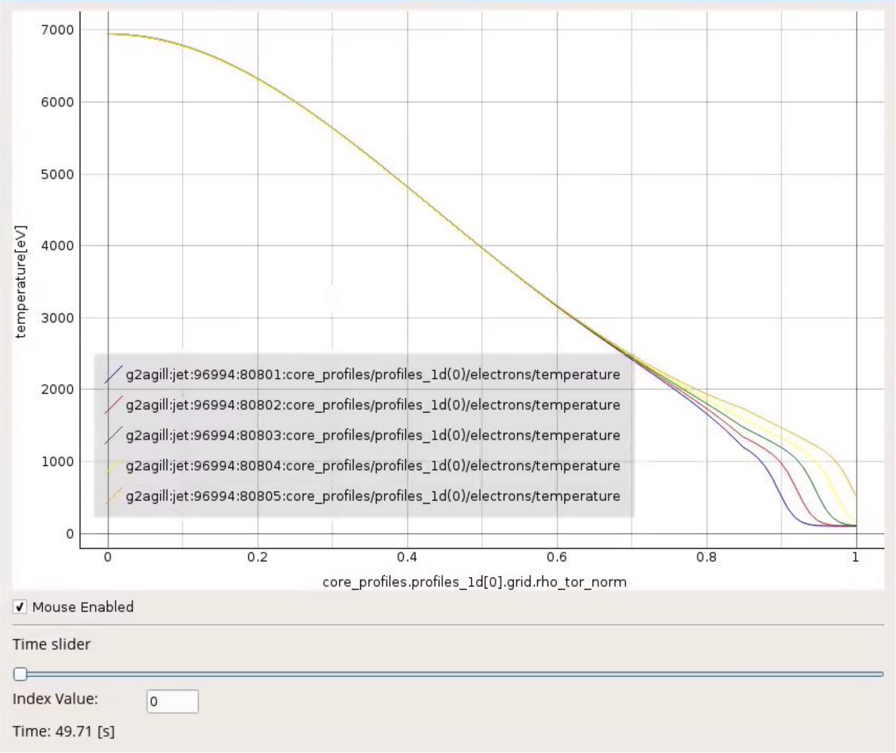
Index Value: 19

Time: 51.60999999999962 [s]

Test case 2: Rhotor scan

PENN is used for all 5 cases, but with different pedestal positions: 0.9, 0.925, 0.95, 0.975, 1.0

Initial profiles

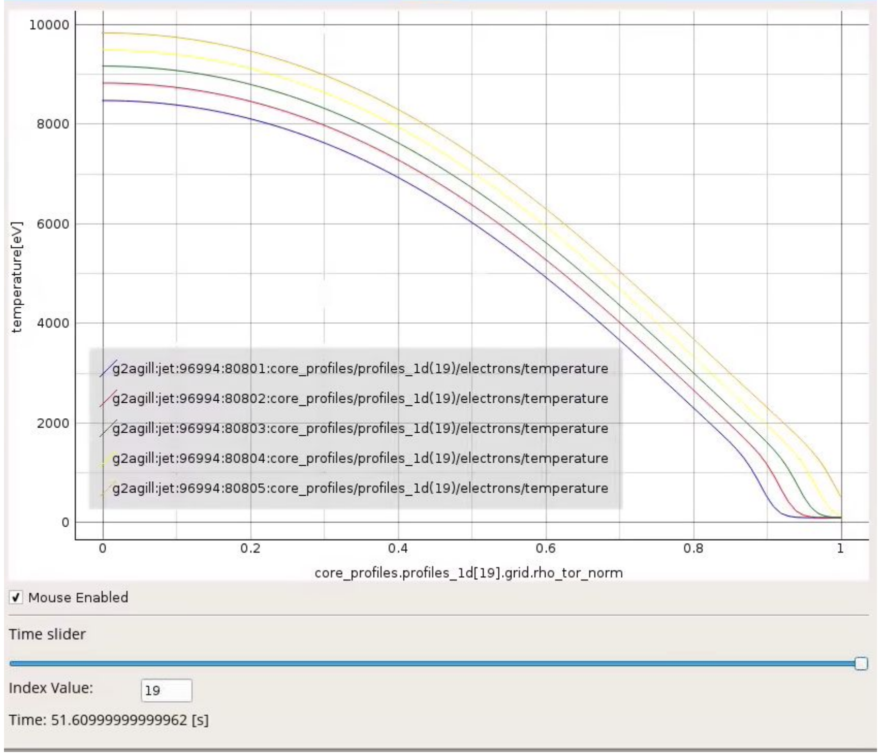


☑ Mouse Enabled

Index Value: 0

Time: 49.71 [s]

2 seconds after evolution

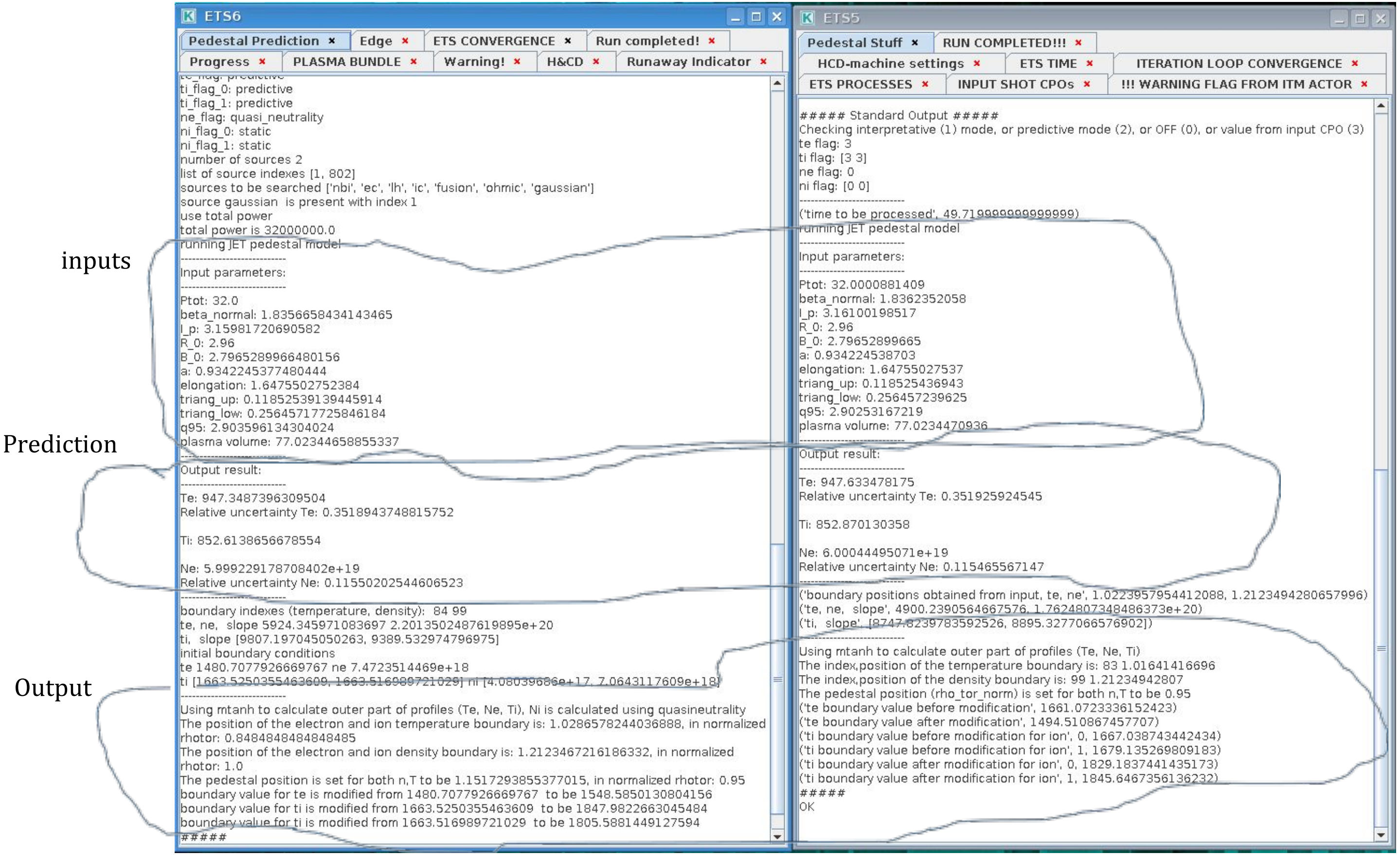


☑ Mouse Enabled

Index Value: 19

Time: 51.60999999999962 [s]

Test case 3: Comparison of PENN in ETS 6 and ETS 5



```
te_flag: predictive
ti_flag_0: predictive
ti_flag_1: predictive
ne_flag: quasi_neutrality
ni_flag_0: static
ni_flag_1: static
number of sources 2
list of source indexes [1, 802]
sources to be searched ['nbi', 'ec', 'lh', 'ic', 'fusion', 'ohmic', 'gaussian']
source gaussian  is present with index 1
use total power
total power is 32000000.0
running JET pedestal model
--------------------------
Input parameters:
--------------------------
Ptot: 32.0
beta_normal: 1.8356658434143465
I_p: 3.15981720690582
R_0: 2.96
B_0: 2.7965289966480156
a: 0.9342245377480444
elongation: 1.6475502752384
triang_up: 0.11852539139445914
triang_low: 0.25645717725846184
q95: 2.903596134304024
plasma volume: 77.02344658855337
--------------------------
Output result:
--------------------------
Te: 947.3487396309504
Relative uncertainty Te: 0.3518943748815752

Ti: 852.6138656678554

Ne: 5.999229178708402e+19
Relative uncertainty Ne: 0.11550202544606523
--------------------------
boundary indexes (temperature, density):  84 99
te, ne,  slope 5924.345971083697 2.2013502487619895e+20
ti,  slope [9807.197045050263, 9389.532974796975]
initial boundary conditions
te 1480.7077926669767 ne 7.4723514469e+18
ti [1663.5250355463609, 1663.516989721029] ni [4.08039686e+17, 7.0643117609e+18]
--------------------------
Using mtanh to calculate outer part of profiles (Te, Ne, Ti), Ni is calculated using quasineutrality
The position of the electron and ion temperature boundary is: 1.0286578244036888, in normalized
rhotor: 0.8484848484848485
The position of the electron and ion density boundary is: 1.2123467216186332, in normalized
rhotor: 1.0
The pedestal position is set for both n,T to be 1.1517293855377015, in normalized rhotor: 0.95
boundary value for te is modified from 1480.7077926669767  to be 1548.5850130804156
boundary value for ti is modified from 1663.5250355463609  to be 1847.9822663045484
boundary value for ti is modified from 1663.516989721029  to be 1805.5881449127594
#####
```

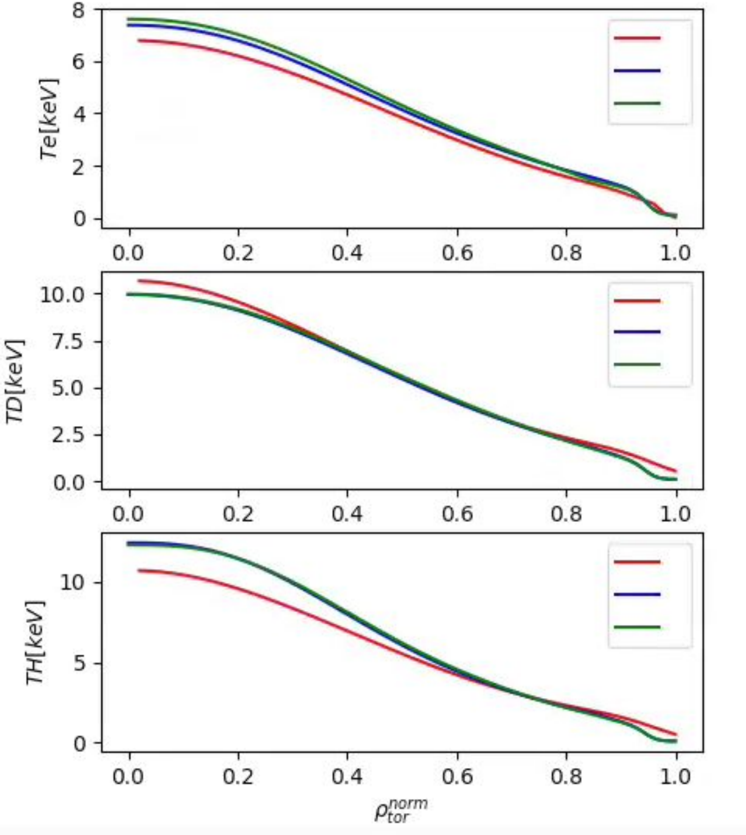
```
##### Standard Output #####
Checking interpretative (1) mode, or predictive mode (2), or OFF (0), or value from input CPO (3)
te flag: 3
ti flag: [3 3]
ne flag: 0
ni flag: [0 0]
--------------------------
('time to be processed', 49.71999999999999)
running JET pedestal model
--------------------------
Input parameters:
--------------------------
Ptot: 32.0000881409
beta_normal: 1.8362352058
I_p: 3.16100198517
R_0: 2.96
B_0: 2.79652899665
a: 0.934224538703
elongation: 1.64755027537
triang_up: 0.118525436943
triang_low: 0.256457239625
q95: 2.90253167219
plasma volume: 77.0234470936
--------------------------
Output result:
--------------------------
Te: 947.633478175
Relative uncertainty Te: 0.351925924545

Ti: 852.870130358

Ne: 6.00044495071e+19
Relative uncertainty Ne: 0.115465567147
--------------------------
('boundary positions obtained from input, te, ne', 1.0223957954412088, 1.2123494280657996)
('te, ne,  slope', 4900.2390564667576, 1.7624807348486373e+20)
('ti,  slope', [8747.8239783592526, 8895.3277066576902])
--------------------------
Using mtanh to calculate outer part of profiles (Te, Ne, Ti)
The index,position of the temperature boundary is: 83 1.01641416696
The index,position of the density boundary is: 99 1.21234942807
The pedestal position (rho_tor_norm) is set for both n,T to be 0.95
('te boundary value before modification', 1661.0723336152423)
('te boundary value after modification', 1494.510867457707)
('ti boundary value before modification for ion', 0, 1667.038743442434)
('ti boundary value before modification for ion', 1, 1679.135269809183)
('ti boundary value after modification for ion', 0, 1829.1837441435173)
('ti boundary value after modification for ion', 1, 1845.6467356136232)
#####
OK
```

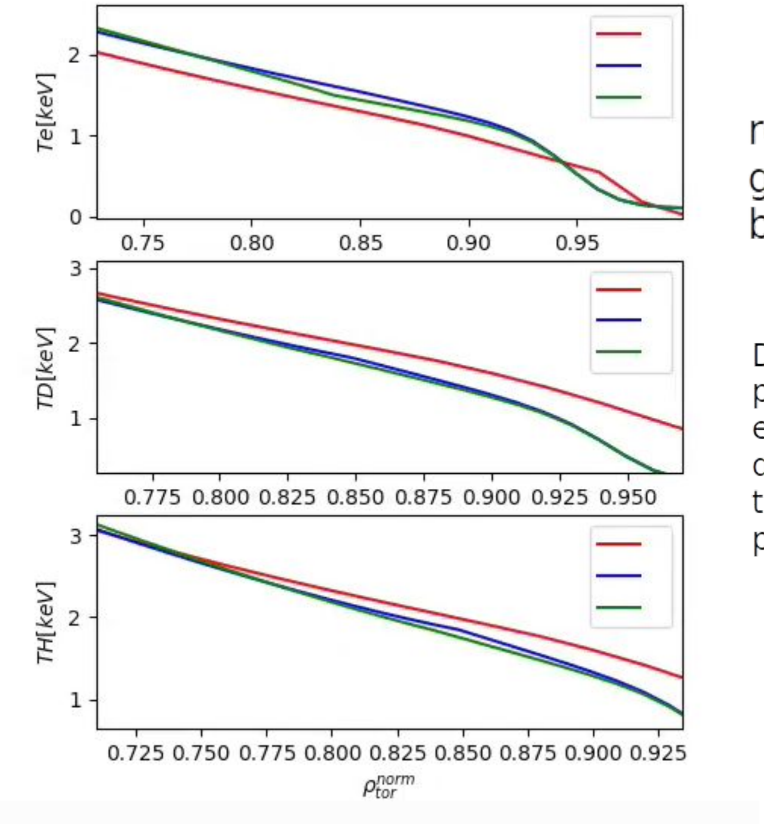
Input parameters are almost identical (fast particles are removed from ETS5 reducing betaN, plasma current is slightly lower for ETS6)

Prediction values are practically identical

Output is slightly different: (boundary position is shifted (index 83 vs 84), rho_tor vector is shifted (one radial point), boundary values are slightly different)

Test case 3: Comparison of PENN in ETS 6 and ETS 5





red - input
green - ETS5
blue - ETS6

Difference in Te profile could be explained by the different shape of the power density profile

To be done: realistic test case for JET and AUG

- JET
  - use the same input (96994)
  - try fully predictive (including density) after ion density will be updated
  - use more sophisticated transport model
  - use “realistic” sources

- AUG:
  - decide on the shot
  - all above as for JET

How to deal with ion density in PENN? (open for discussion)

- We cannot yet predict ion densities in PENN (no ion data in database)
- If we modify the electron edge density, should we adjust the ion edge density in PENN to fulfil QN or leave it untouched to be handled by other modules?
- QN calculations in PENN cannot be exact since we only consider main ions in PENN
    - How would we handle impurities if we had access in PENN?
- What if the ion density is predictive but not the electron density?
    - Could, for instance, use the Ne prediction to calculate Ni at the pedestal top from QN, and then use mtanh (as for Te, and Ti) on Ni and force Ne to fulfil QN?
- Things get more complicated when multiple ion species are considered

We must work through the different scenarios and decide on a method before interpreting results of the density profiles. <u>This is of highest priority</u>

QN - quasineutrality

Requests (not implemented yet)

- Ratio between ion and electron temperature to be an array, one element for each ion species
- Different pedestal position for density and temperature
- Optional scaling constant to multiply with neural network predictions:
  - $Te_{height} = \alpha \times NN_{Te}$ (indirectly affects Ti)
  - $Ne_{height} = \beta \times NN_{Ne}$

The scaling constant can be used, for instance, to apply effective mass scalings set by the user: $\alpha = C1 \times M^{C2}$ (or some other definition)

Here, α must be calculated before using it in PENN. For the effective mass, it is important to remember that the model is mostly trained on deuterium dominated plasma: $M \sim 2.0$ (or slightly below 2.0)

Thanks for listening

Questions?

BACKUP

What is PENN?

- A neural network based pedestal prediction tool (trained on JET/AUG pedestal database)
- Able to predict electron pedestal height for temperature and density
- mtanh is used to together with predictions to determine edge profiles
- The extrapolation of mtanh towards the core gives new boundary conditions

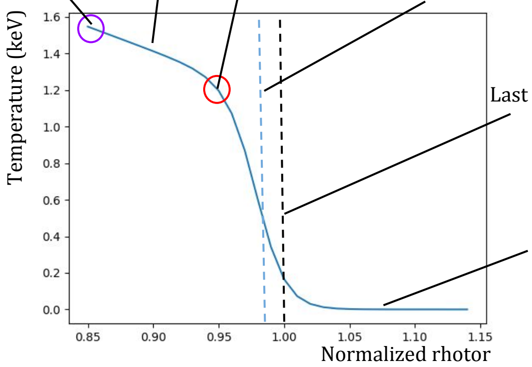
Input parameters

Beta_N (MHD)
I_p (plasma current)
B_0 (toroid field)
R_0 (major radius)
a (minor radius)
Elongation
Upper triangularity
Lower triangularity
P_tot (total power input)
q95
Plasma volume

Neural Networks

Output parameters

Te height
Ne height



Core slope, calculated from previous time step

Boundary condition

PENN prediction

Pedestal position, set by user (0.98 recommended since mean of database)

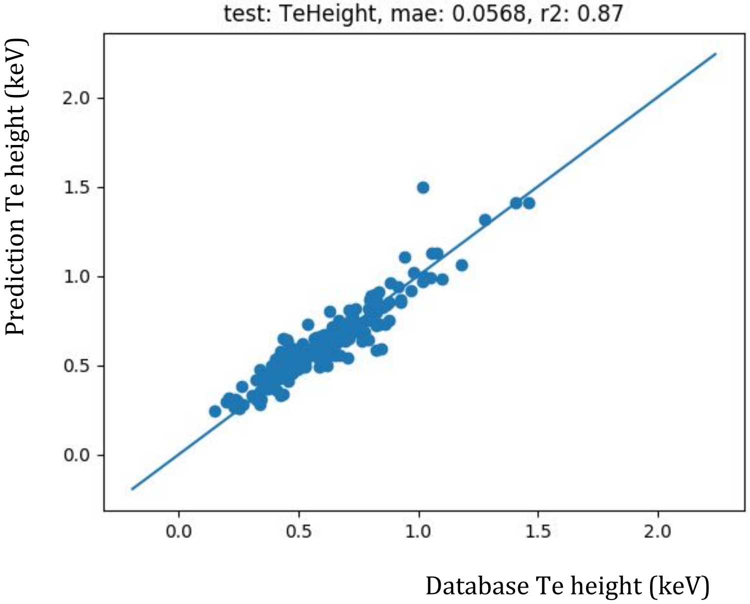
Last closed flux surface

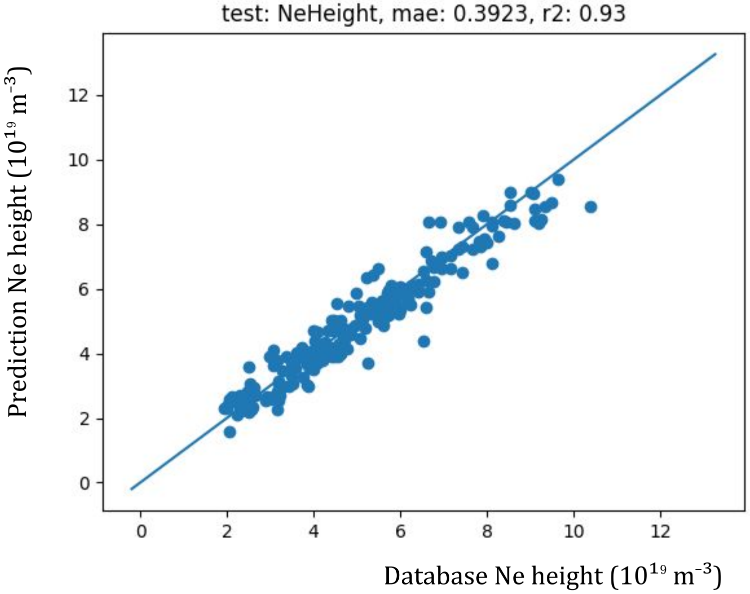
Saturation offset = 100 eV

Pedestal width = 0.045 (mean of database)

What is PENN?

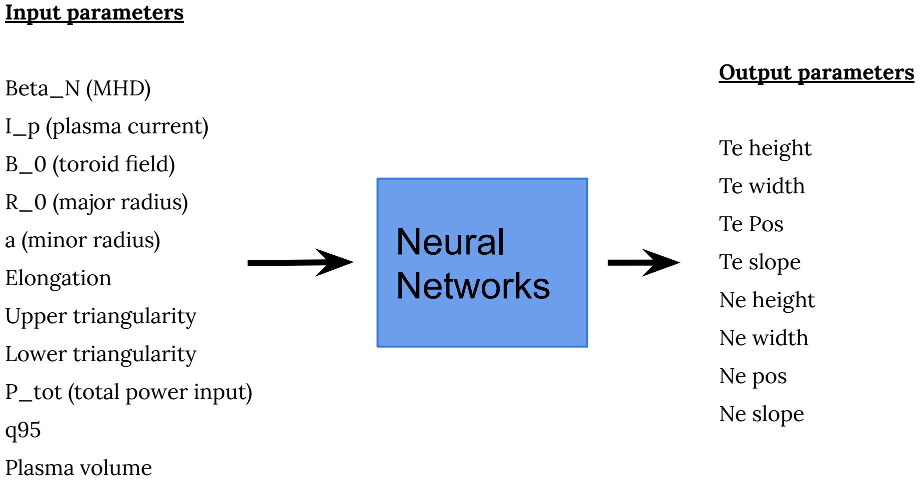
- Prediction accuracy on test set (data not seen during training)





Upgrading PENN (preliminary results)

- Idea: predict all pedestal parameters to avoid potential errors from setting parameters manually
- User will be able to choose PENN v.1 or PENN v.2 (names might be updated)



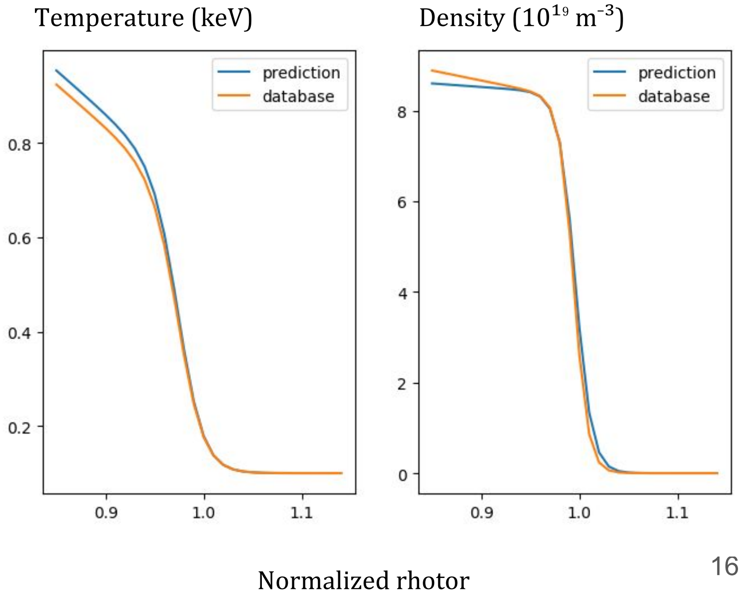
Input parameters

Beta_N (MHD)
I_p (plasma current)
B_0 (toroid field)
R_0 (major radius)
a (minor radius)
Elongation
Upper triangularity
Lower triangularity
P_tot (total power input)
q95
Plasma volume

Neural Networks

Output parameters

Te height
Te width
Te Pos
Te slope
Ne height
Ne width
Ne pos
Ne slope



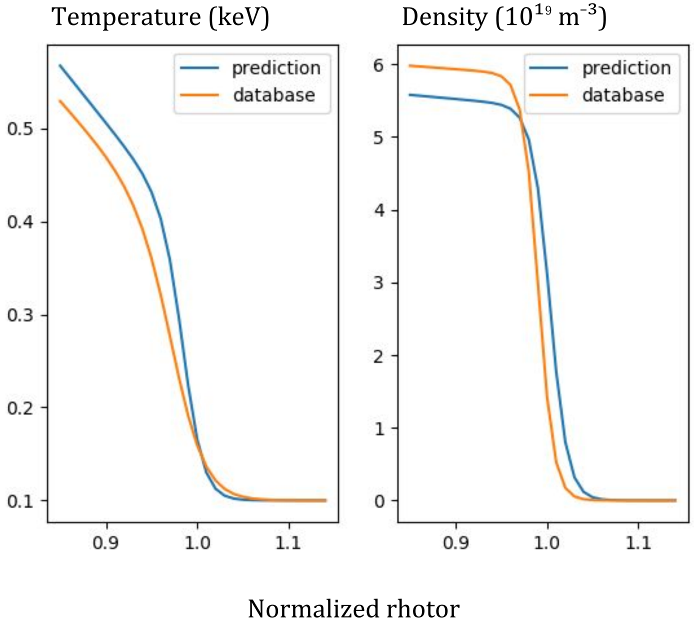
Temperature (keV)

Density (10^{19} m^{-3})

Normalized rhotor

Upgrading PENN (preliminary results)

- Another example



- Note: For time evolutions, it could, for instance, be potentially harmful to get widely different slope predictions between two timesteps

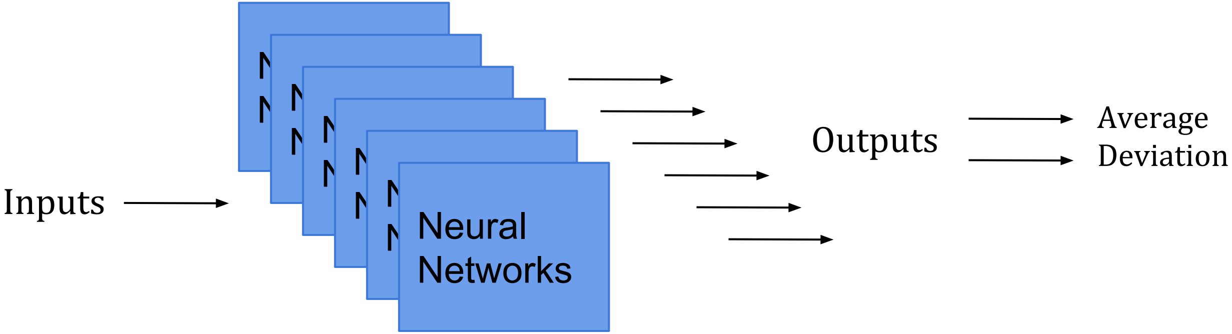
  The PENN v.1 approach to use the core slope of the previous time step might be considered to be more stable

  PENN v.2 predictions are useful for stationary predictions, and potentially the first time step, but we must decide which parameters to evolve in time, and how to handle each pedestal parameter

Quantification of prediction uncertainty

- We use a committee neural network in our predictions
  - By employing several networks that are initiated uniquely, each will end up at different local cost function minimums once they are trained
  - The final prediction becomes the average of the individual predictions of the networks
  - The deviation between the individual predictions can be used to detect uncertainty and extrapolation

- It is easy to spot extrapolation for individual input parameters
- A committee network allows us to detect extrapolation of the relations between the input parameters (even for cases where individual parameters are within training range)



Inputs

Neural Networks

Outputs

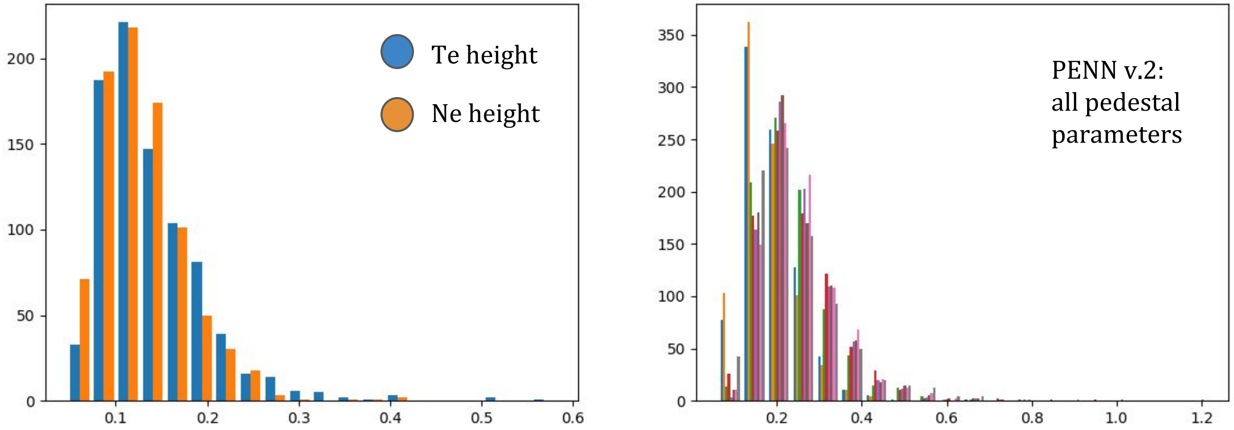
Average
Deviation

Quantification of prediction uncertainty

- Mean value is straightforward to use
- Which degree of deviation is too much?

- During training we normalize input and output data
  - Allows for comparison of standard deviation between different outputs in the normalized space

Histograms of standard deviation in normalized outputs of dataset



Conclusion: For norm standard deviations > 0.6, there is reason suspect extrapolation and to question the reliability of a prediction, especially above deviation > 1.0

We fabricated an example to provoke extrapolation and got a normalized deviation: 1.79 for te, 1.29 for ne