Hybrid Neural Networks in Nuclear Fusion Transport Modelling

Master Thesis

Faculty of Physics, LMU

Author:

Daniel SCHAEFER

Supervisors:

Prof. Dr. Frank JENKO

Prof. Dr. Hartmut ZOHM

Dr. Jonathan CITRIN

Karel VAN DE PLASSCHE

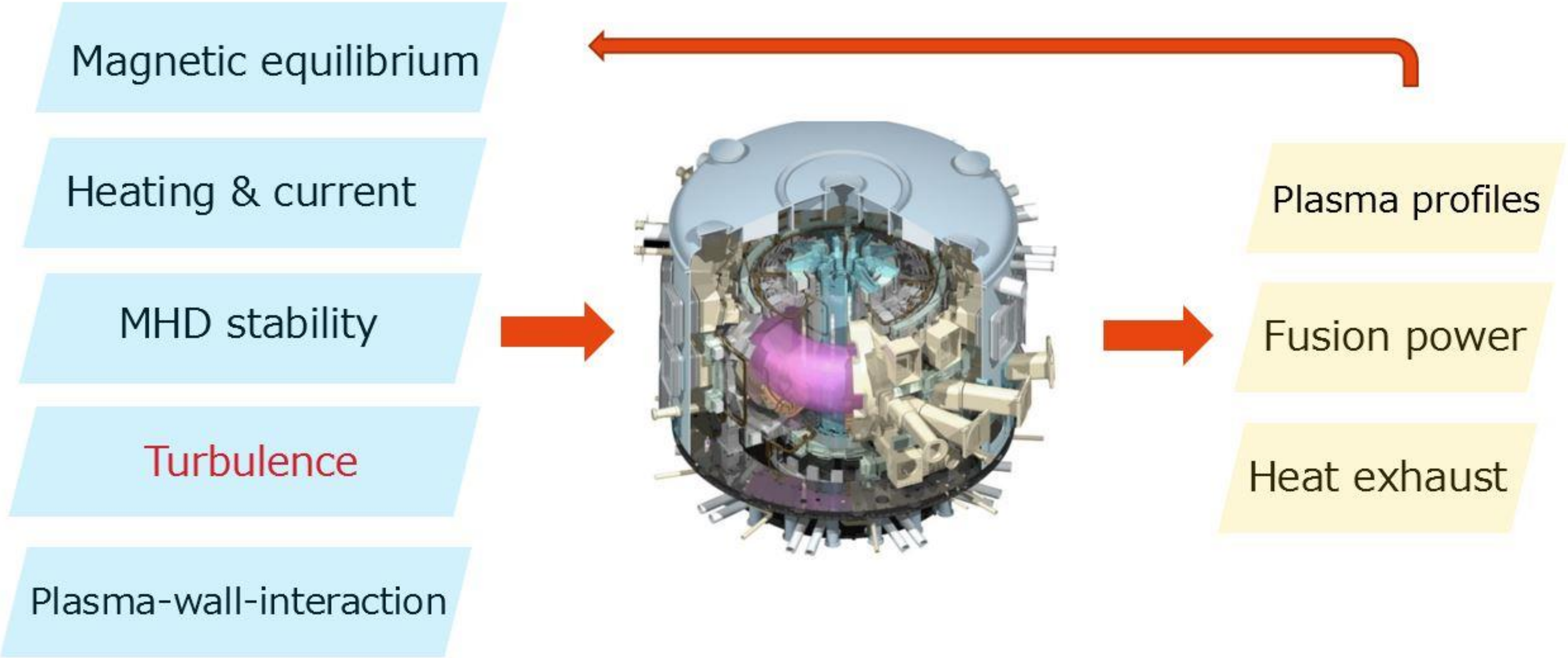
IPP Max-Planck-Institut für Plasmaphysik

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

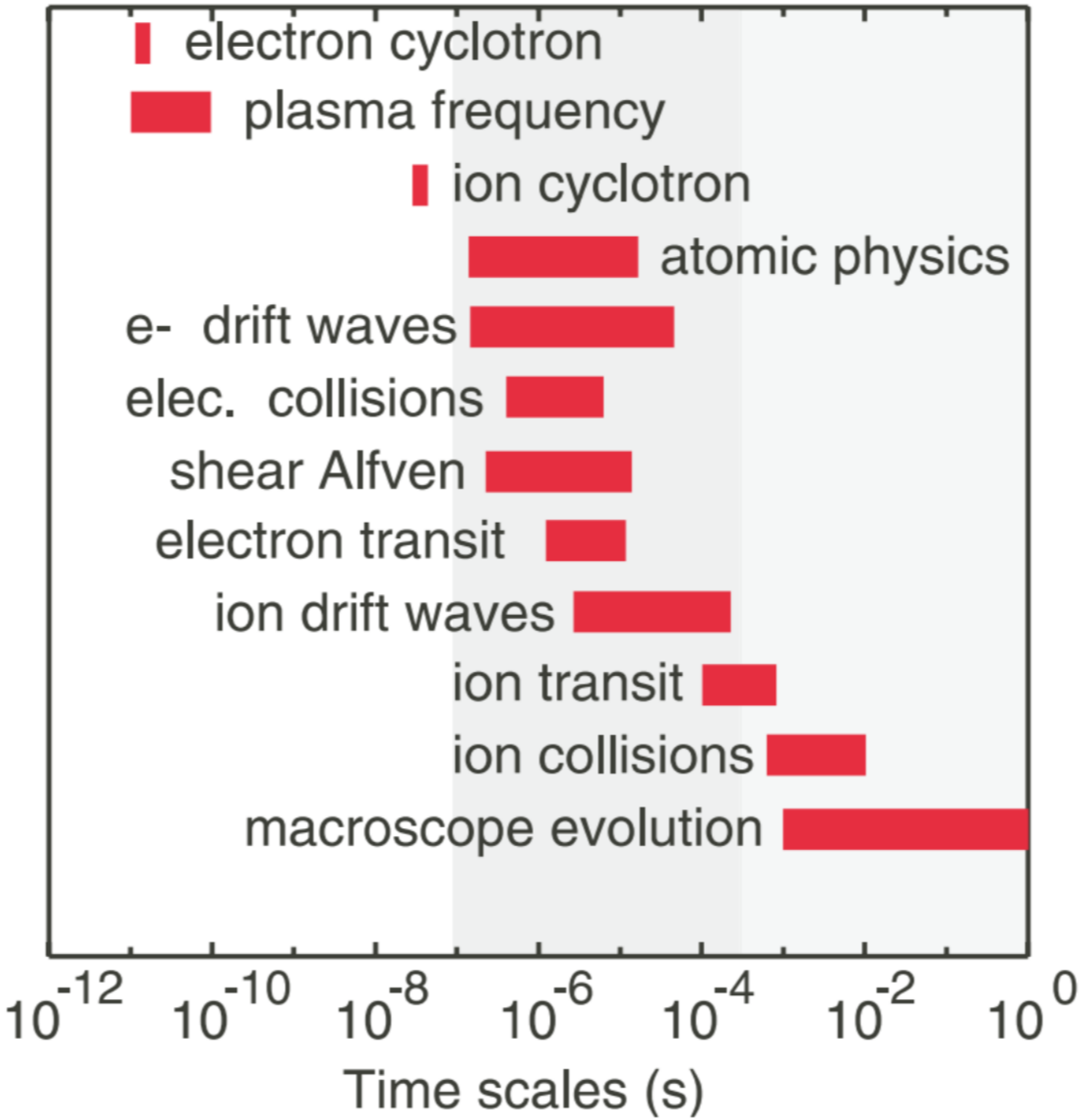
Outline

- The Big Picture
- Previous State of the Art
- Thesis Objective
- Methodology
- Results
- Discussion
- Conclusion & Outlook

The Big Picture: Integrated Modelling



CREDIT: Courtesy of J. Citrin (DIFFER), image by ITER (www.iter.org)



CREDIT: T. Rognlien, Plasma Physics & Controlled Fusion (2005)

Comparison of model execution speeds

Calculation of tokamak radial profile of turbulent transport coefficients:

Model type	CPU time [s]
Nonlinear global (adiabatic electrons)	10^{10}
High-fidelity Gyrokinetic	10^{8}
Reduced Model	10^{2}
Neural Network	10^{-3}
Requirements for real-time control	$\mathcal{O}(10^{-3})$

Previous Work - QuaLiKiz

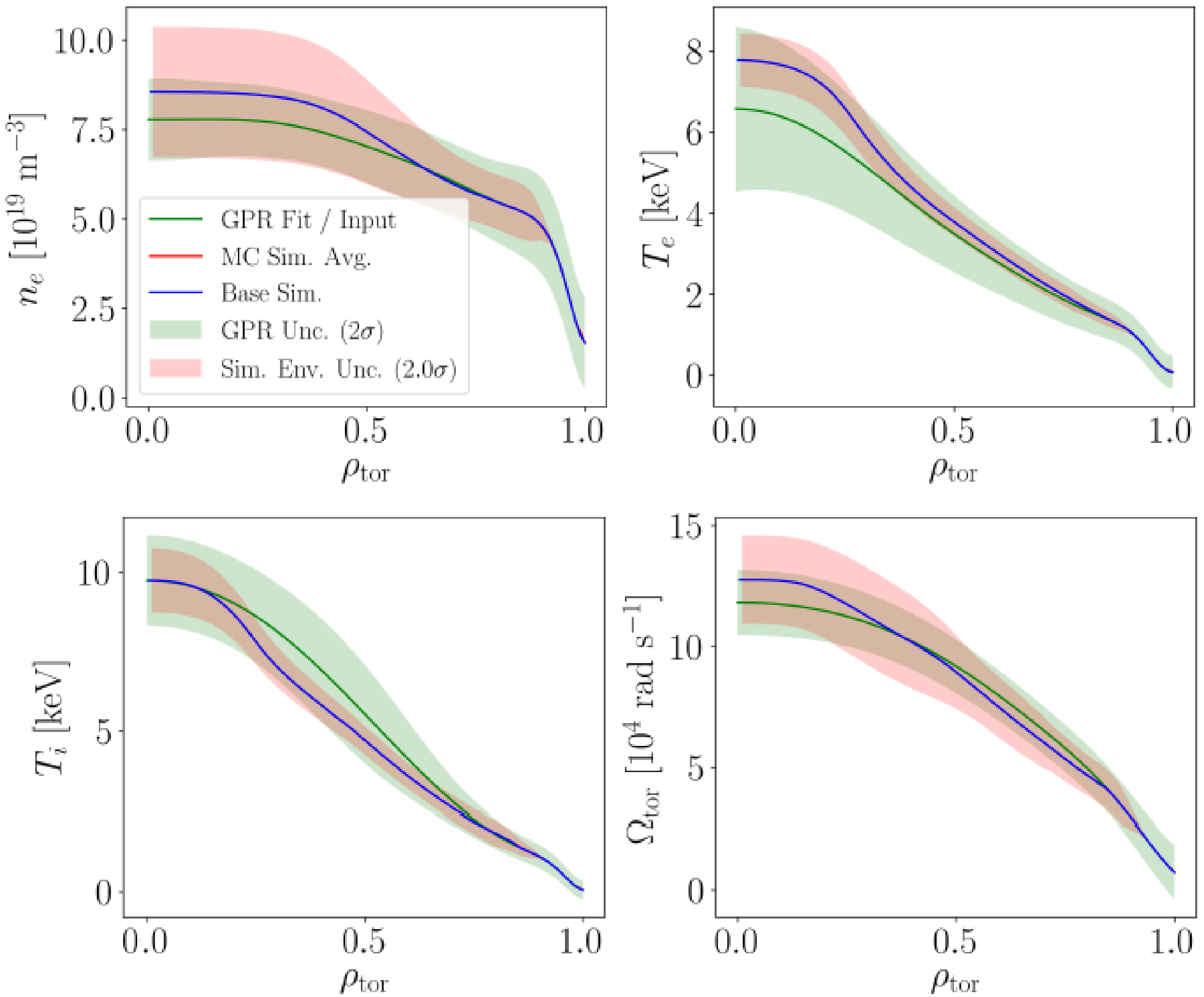
- **Quasi-linear gyrokinetic turbulent transport solver for tokamaks[1]**

  - computes:
    - turbulent heat fluxes
    - particle (including impurity) fluxes and D+V transport coefficients
    - angular momentum fluxes

  - coupled in integrated modelling platforms such as CRONOS and JETTO. Validated against experiments, e.g. A. Ho Nucl. Fusion 2019

SOURCE: [1] www.QuaLiKiz.com, Bourdelle PPCF 2016, Citrin PPCF 2017

Previous Work - QuaLiKiz



SOURCE: JETTO-QuaLiKiz simulation of JET baseline #92436. A. Ho Nucl Fusion 2019

Training Data

Inputs:

Quantity	Range	# points
Wavenumber (ion + electron scale) [$k_\theta\rho_s$]	0.1 – 36	18
Ion temperature gradient [R/L_{T_i}]	1 – 14	12
Electron temperature gradient [R/L_{T_e}]	0 – 14	12
Density gradient [R/L_n]	-5 – 6	12
Magnetic pitch angle [q]	0.66 – 15	10
Magnetic pitch angle shear [$\hat{s}$]	-1 – 5	10
Normalized radius [r/R]	0.03 – 0.33	8
Temperature ratio [T_i/T_e]	0.25 – 2.5	7
Collisionaliy [ν^*]	10^{-5} – 1	6
Impurity content [Z_{eff}]	1 – 3	5
Total	~1.3 MCPUh	3.10^8

Outputs:

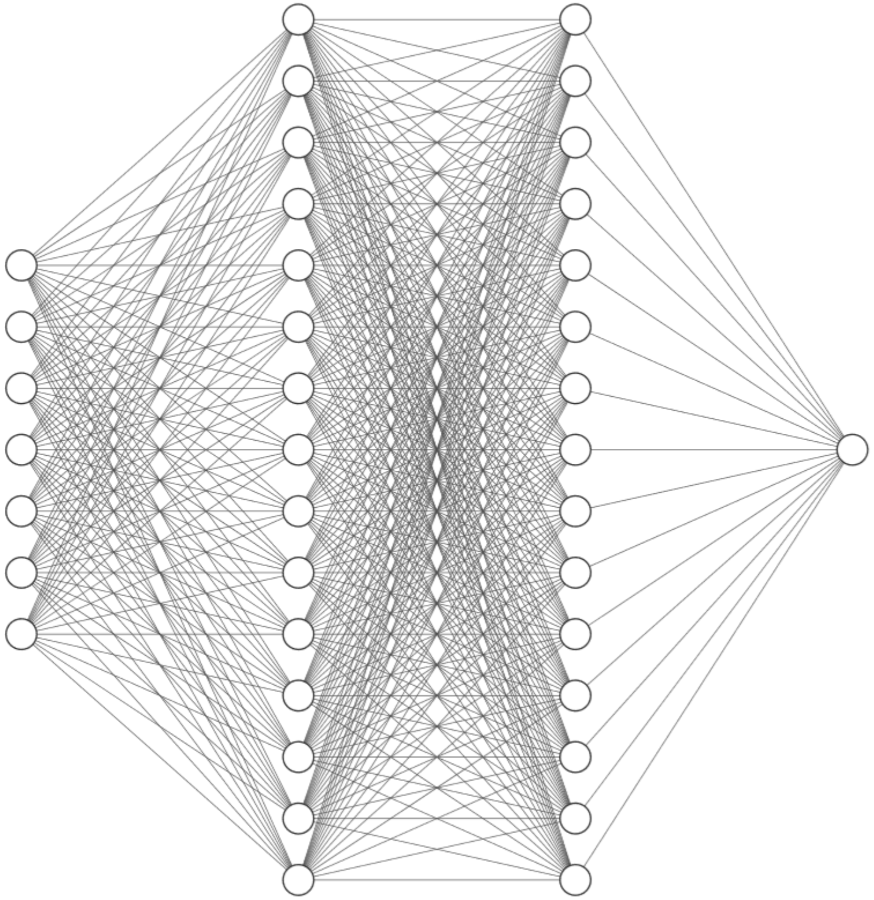
- ion heat flux
- electron heat flux
- electron particle fluxes
- ion particle fluxes
- diffusivity
- convective terms

CREDIT: K.L. van de Plassche, J. Citrin (DIFFER)

Previous Work

- Fully connected feed forward neural networks were used to perform a regression of this training data

- Typically 2-3 layers of 30-128 nodes each



CREDIT: [1] Citrin 2015 NF, Felici 2018 NF
[2] K.L. van de Plassche: M.Sc. thesis 2017, 2018 EPS, 2018 TTF (to be submitted to Nucl. Fusion)

Previous Work

Neural network regression of pre-generated QuaLiKiz calculation database for fast surrogate modelling

- 4D (Input) Proof-of-concept[1]
  - Concept validation - faster-than-realtime simulation for ITER
- 10D Extension[2]
  - Achieved a good neural network fit for QuaLiKiz heat and particle transport
  - <1 *ms* predictions for profiles of transport coefficients
  - Integrated in real-time capable tokamak simulation suite RAPTOR and JETTO integrated modelling suite

CREDIT: [1] Citrin 2015 NF, Felici 2018 NF
[2] K.L. van de Plassche: M.Sc. thesis 2017, 2018 EPS, 2018 TTF (to be submitted to Nucl. Fusion)

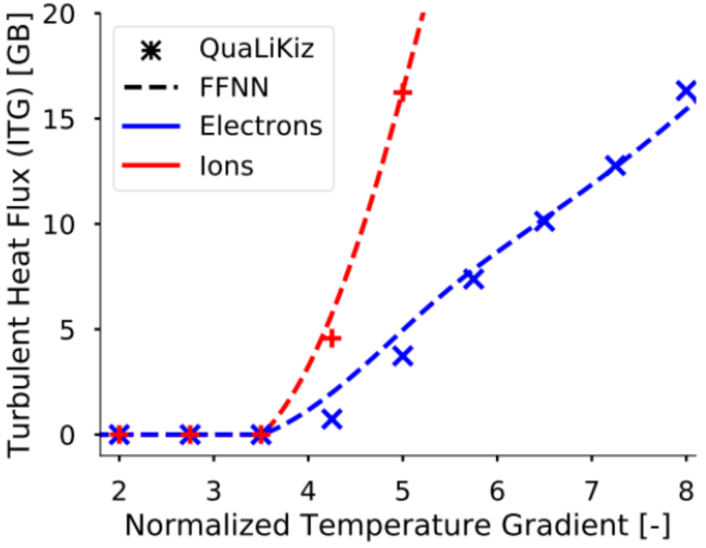
Previous Work

- Cost Function

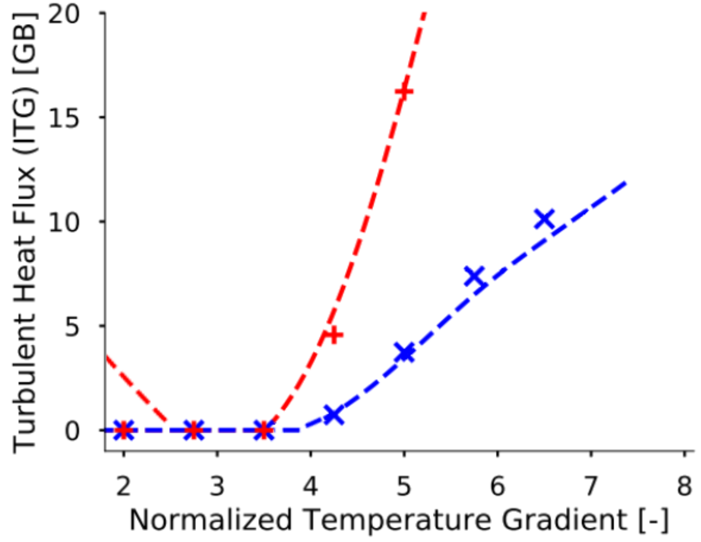
  $C = C_{good} + \lambda_{regu} C_{regu} + \lambda_{stable} C_{stable}$

- Sharp instability thresholds
  - Only include unstable points in "goodness" part of Cost Function

$$C_{good} = \begin{cases} \frac{1}{n}\sum_{i=1}^{n}(QLK_i - NN_i)^2, & \text{if } QLK_i \neq 0 \\ 0, & \text{if } QLK_i = 0 \end{cases}$$



Optimized



Non-Optimized

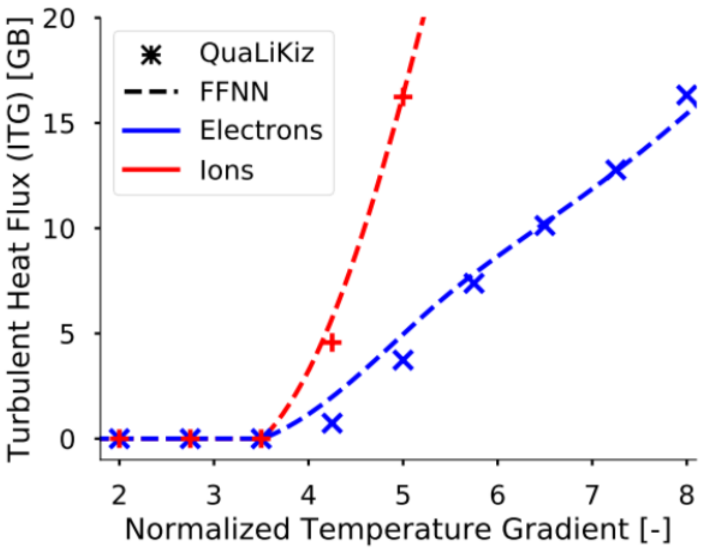
SOURCE: K.L. van de Plassche (DIFFER)

Previous Work

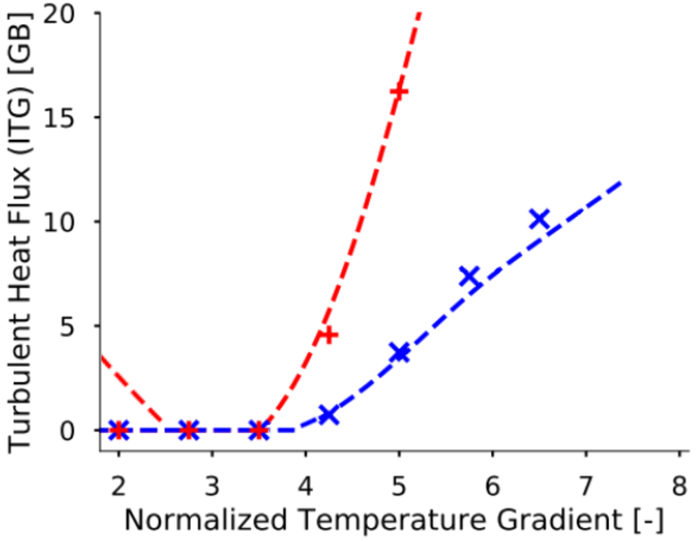
- Matching thresholds for all transport channels
- train on $q_{i,ITG}$, $q_{e,ITG}/q_{i,ITG}$, $\Gamma_{e,ITG}/q_{i,ITG}$ etc. and multiply the output of the networks
  - Clip negative heat-flux to zero

- No spurious positive flux in stable region
  - Punish positive predictions with extra Cost Function term

$$C_{stab} = \begin{cases} 0, & \text{if } QLK_i \neq 0 \\ \frac{1}{n}\sum_{i=1}^{n}\left(NN_i - c_{stab}\right), & \text{if } QLK_i = 0 \end{cases}$$



Optimized



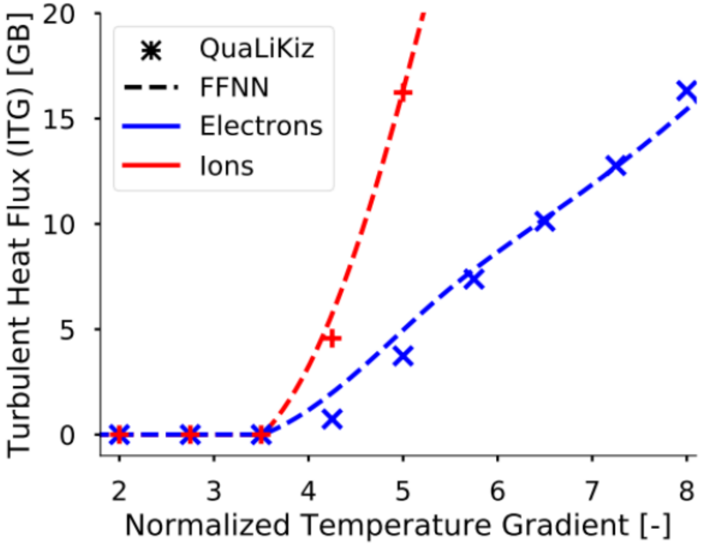
Non-Optimized

SOURCE: K.L. van de Plassche (DIFFER)

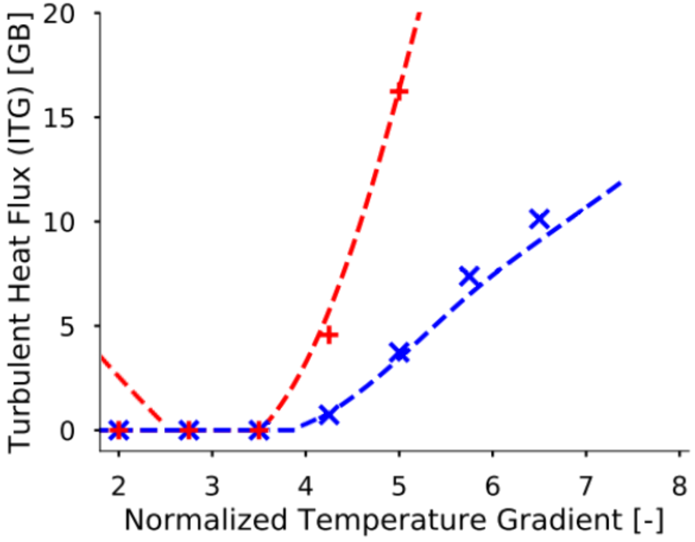
Previous Work

- Enforce Smoothness
  - Punish model complexity using L2 Cost Function

$$C_{regu} = \sum^{k} w_i^2$$



Optimized



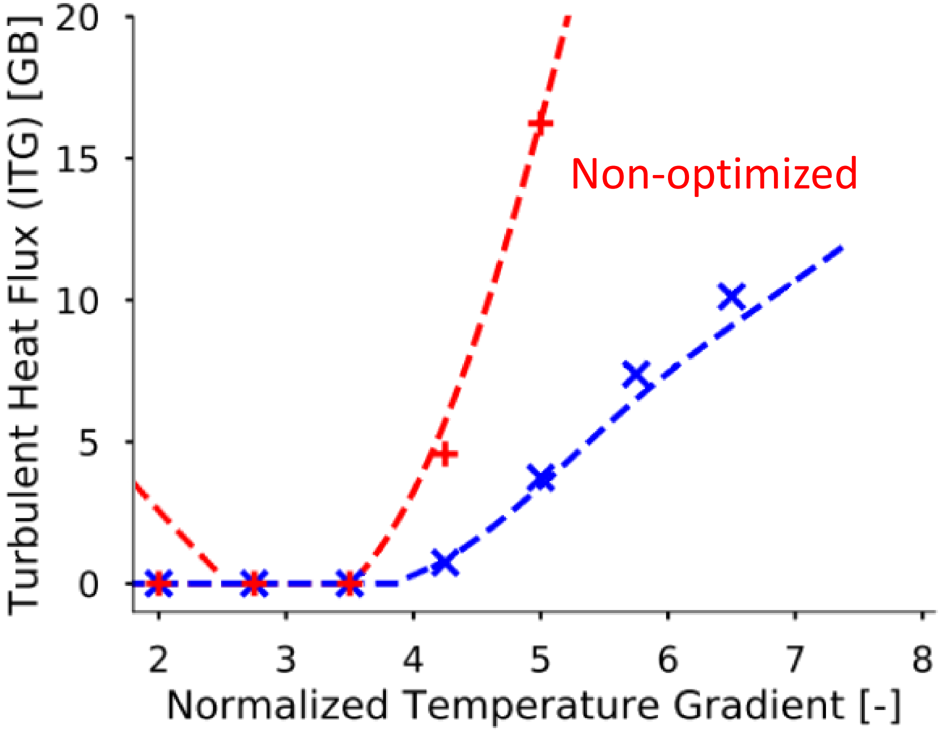
Non-Optimized

SOURCE: K.L. van de Plassche (DIFFER)

Thesis Objective/Research Problem

Eliminate "Pop-back" via alternative approach

- solve pop-back problem in a more elegant manner using prior physical knowledge of system (Critical Gradient Model)



<u>Advantages:</u>

- Avoid direct regression
- (not yet tested) very smooth outputs
    - important for Newton solvers for implicit (fast) PDE solution of the transport equations

Strategy

- Remove additional Cost function terms, custom pre-/post-processing etc.
- Replace pure Universal Function Approximator (fully-connected neural network/black-box) approach with the ability to insert prior domain knowledge.

- **<u>Non-global</u>** hybridization technique needed.

- Previously none were available.

Reduced Dataset used

Inputs:

Quantity	Range	# points
Wavenumber (ion + electron scale) [$k_\theta\rho_s$]	0.1 – 36	18
Ion temperature gradient [R/L_{T_i}]	1 – 14	12
Electron temperature gradient [R/L_{T_e}]	0 – 14	12
Density gradient [R/L_n]	-5 – 6	12
Magnetic pitch angle [q]	0.66 – 15	10
Magnetic pitch angle shear [$\hat{s}$]	-1 – 5	10
Normalized radius [r/R]	0.03 – 0.33	8
Temperature ratio [T_i/T_e]	0.25 – 2.5	7
Collisionaliy [ν^*]	10^{-5} – 1	6
Impurity content [Z_{eff}]	1 – 3	5
Total	~1.3 MCPUh	3.10^8

Output:

- Electron Heat Flux (ETG Mode)

Extracting Domain Knowledge

- Problem type: regression.
- Software exists (**QuaLiKiz**) that performs simulations.
- There are 7 inputs.
  - $(I_1, \ldots, I_7) \in Inputs$

- I_7 (**ATE**) has additional physical significance.
  - Will be injected separately from rest.

- There is 1 output.

Extracting Domain Knowledge

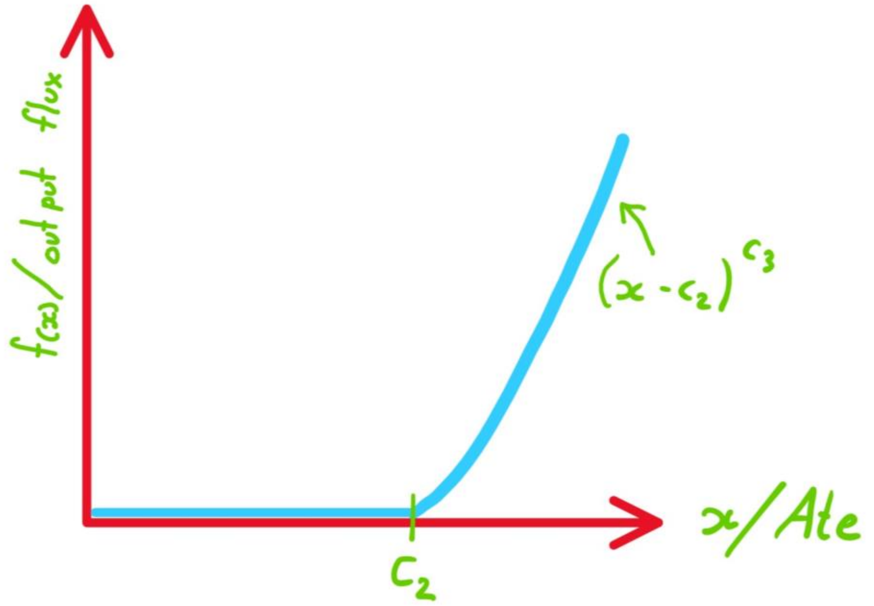
- Expected relationship is known: Critical Gradient Model

$$f(x) = c_1 \cdot \theta(x - c_2) \cdot (x - c_2)^{c_3}$$

- θ is the Heaviside function

$$\theta(x-a) = \begin{cases} 0 & \text{if } x < a \\ 1 & \text{if } x > a \end{cases} \qquad (x \geqslant 0)$$

- x = **ATE** (electron heat flux Wm^{-2} in the electron temperature gradient (ETG) mode)
- c_1 = amplitude factor
- c_2 = threshold where the output flux stops being non-zero
- c_3 = polynomial coefficient describing behaviour of non-zero output flux components



Critical Gradient Model

- $I_1, \ldots, I_7 \;\in\; \mathbb{R}^{I_1, \ldots, I_7}$ (Input space)

- $c_1 = g(I_1, \ldots, I_6)$
- $c_2 = h(I_1, \ldots, I_6)$
- $c_3 = i(I_1, \ldots, I_6)$
- $x = I_7$ **(ATE)**
- $f(x) = f^*(I_1, \ldots, I_7) \;\in\; \mathbb{R}^1$ (Output space)

$$f(x) = c_1 \cdot \theta(x - c_2) \cdot (x - c_2)^{c_3}$$

Aim

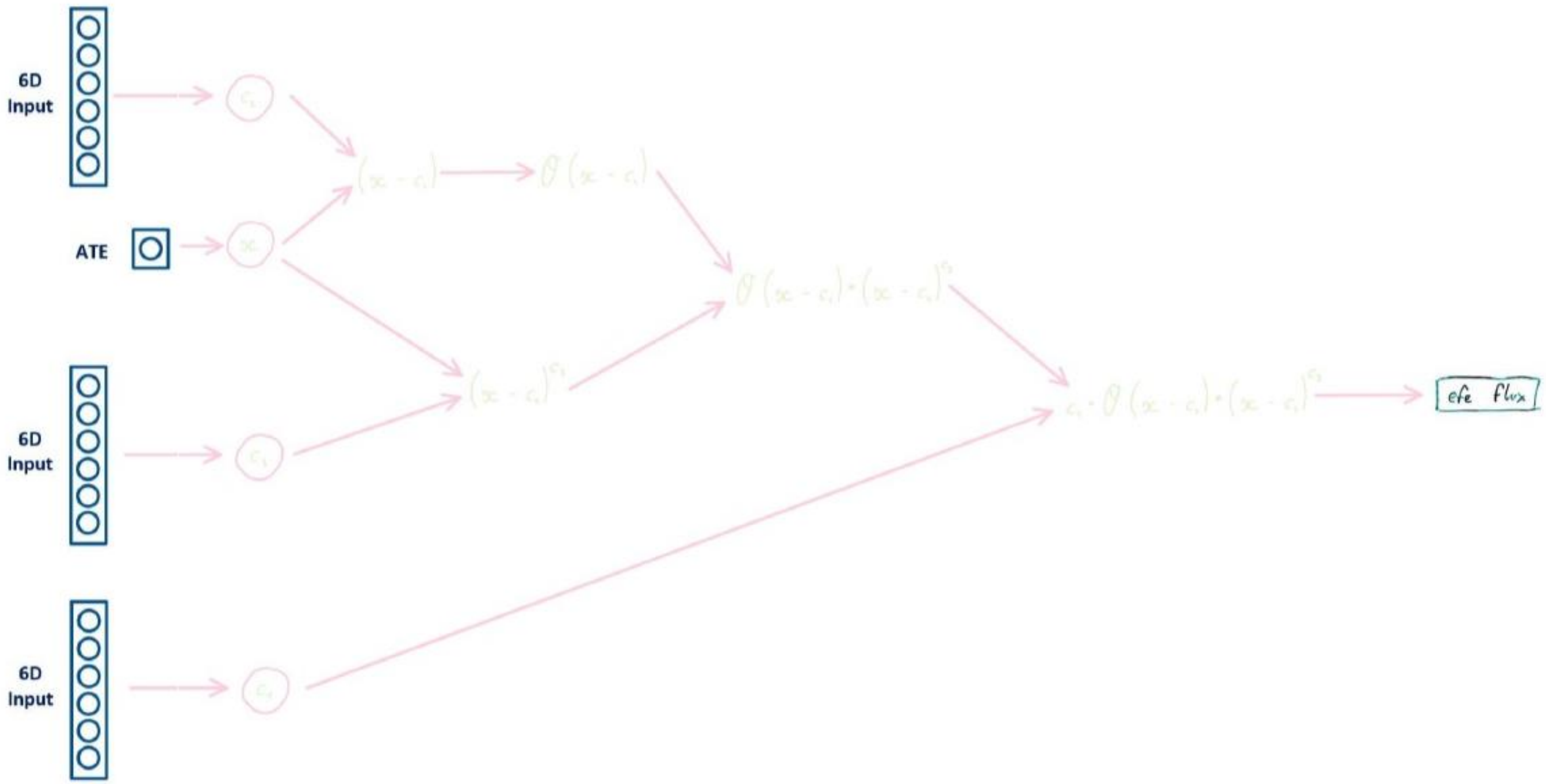
- Numerically find approximation of $f(x) \in \mathbb{R}^{I_1, \dots, I_7}$

- Whilst also optimizing/finding simultaneously for
  - $c_1 = g(I_1, \dots, I_6) \in \mathbb{R}^{I_1, \dots, I_6}$
  - $c_2 = h(I_1, \dots, I_6) \in \mathbb{R}^{I_1, \dots, I_6}$
  - $c_3 = i(I_1, \dots, I_6) \in \mathbb{R}^{I_1, \dots, I_6}$

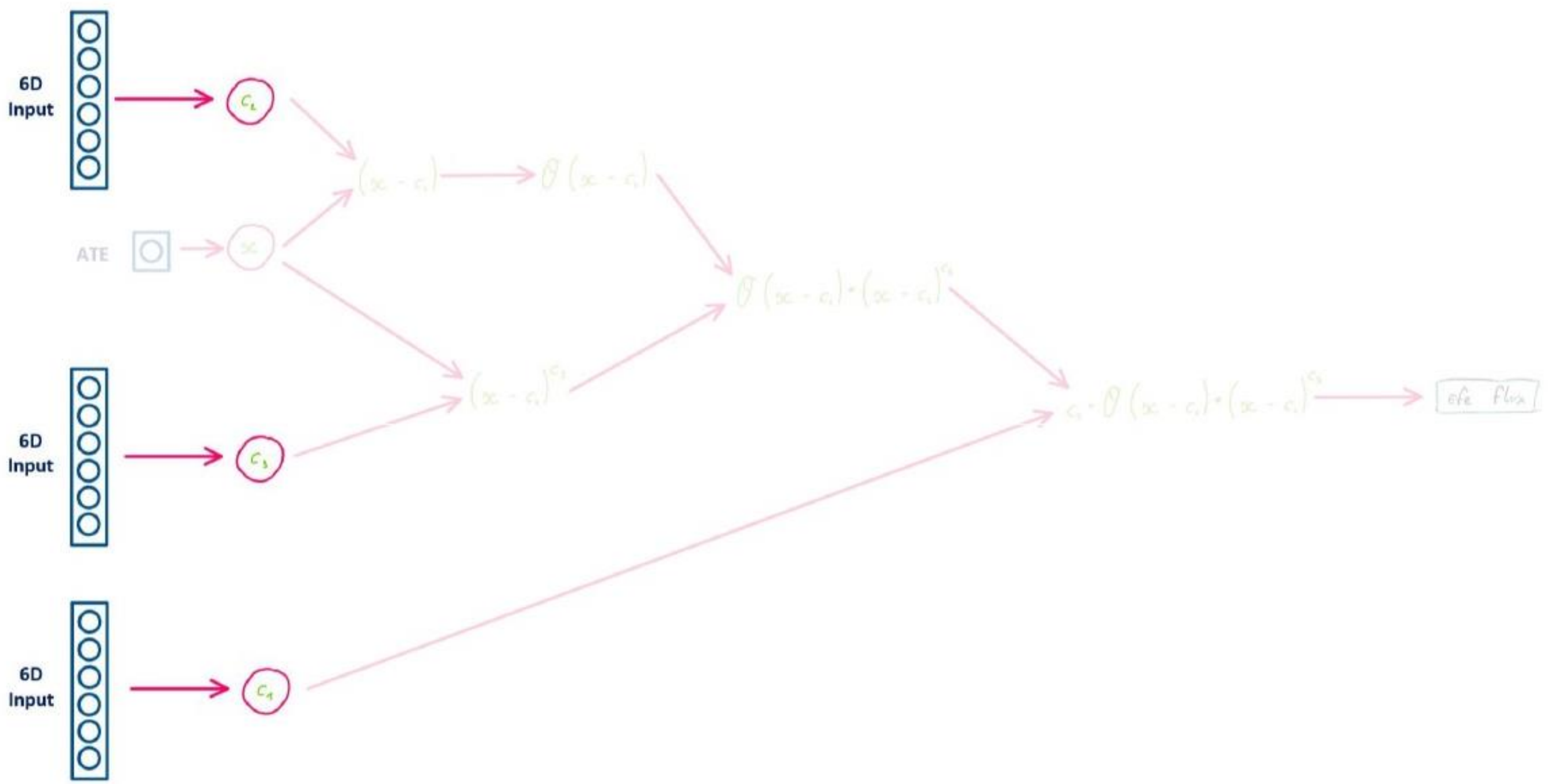
- Whilst performing multi-variable optimization

New approach needed

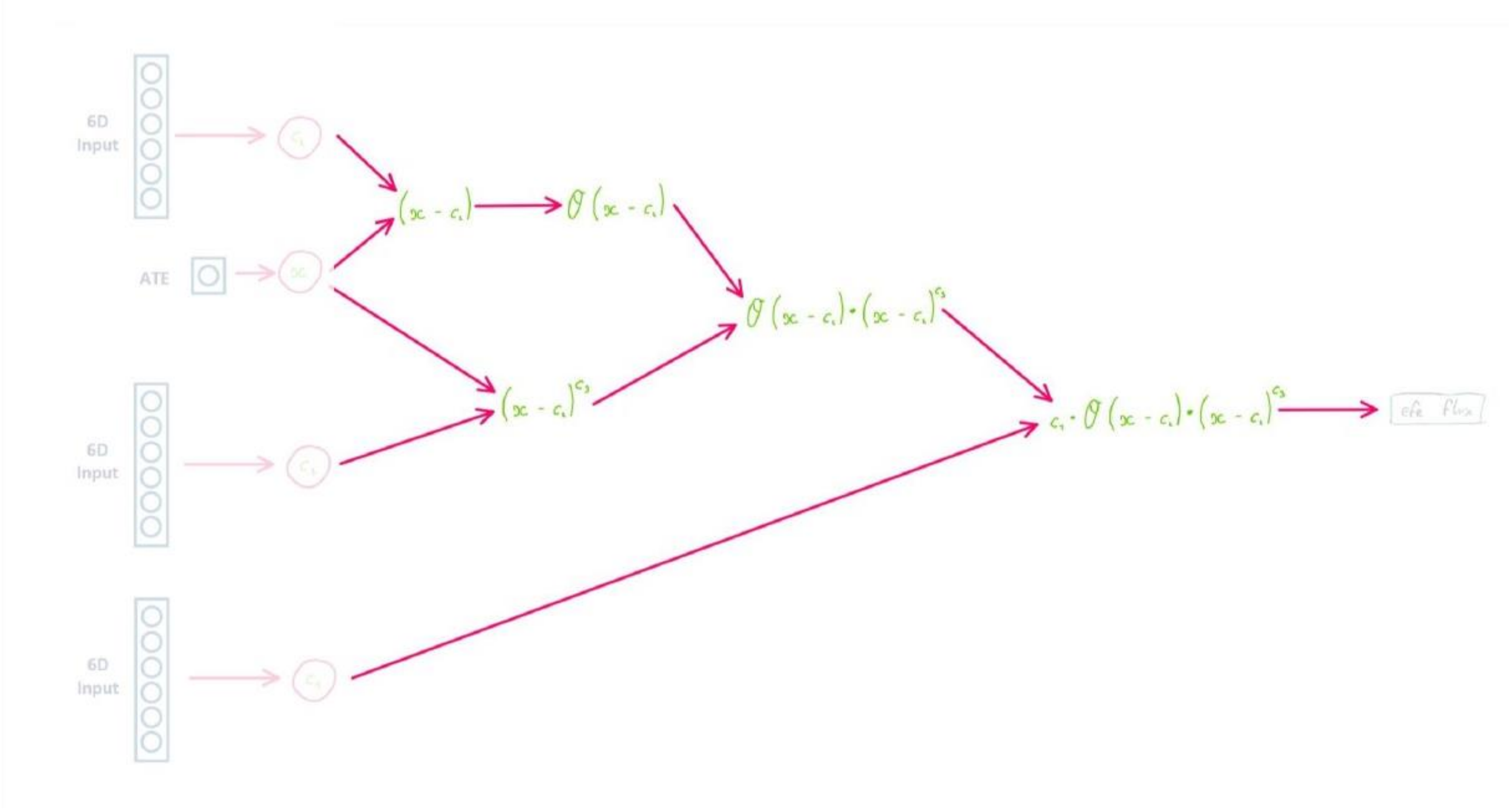
Physics/Black-Box Hybrid Architecture



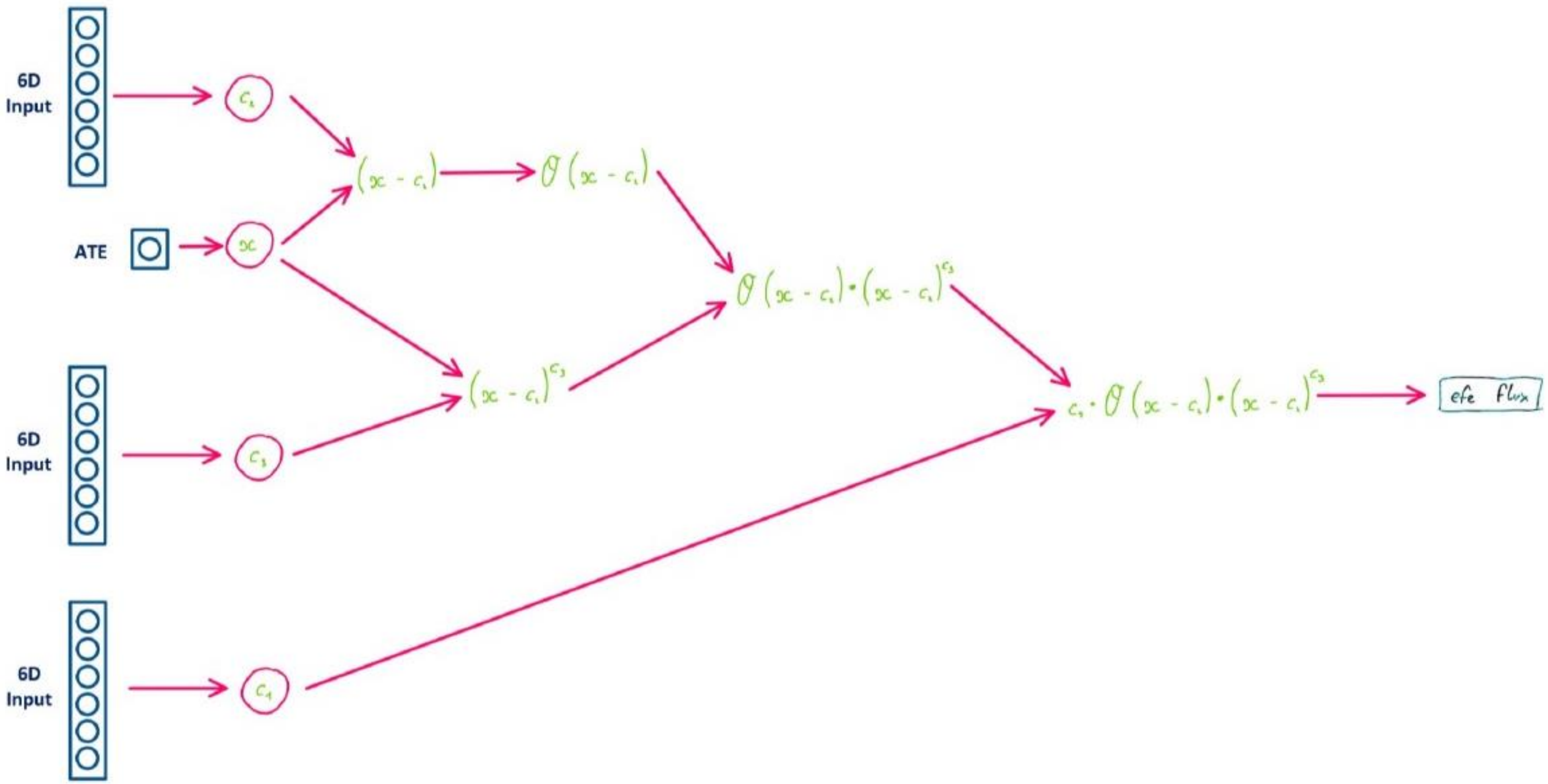
Physics/Black-Box Hybrid Architecture



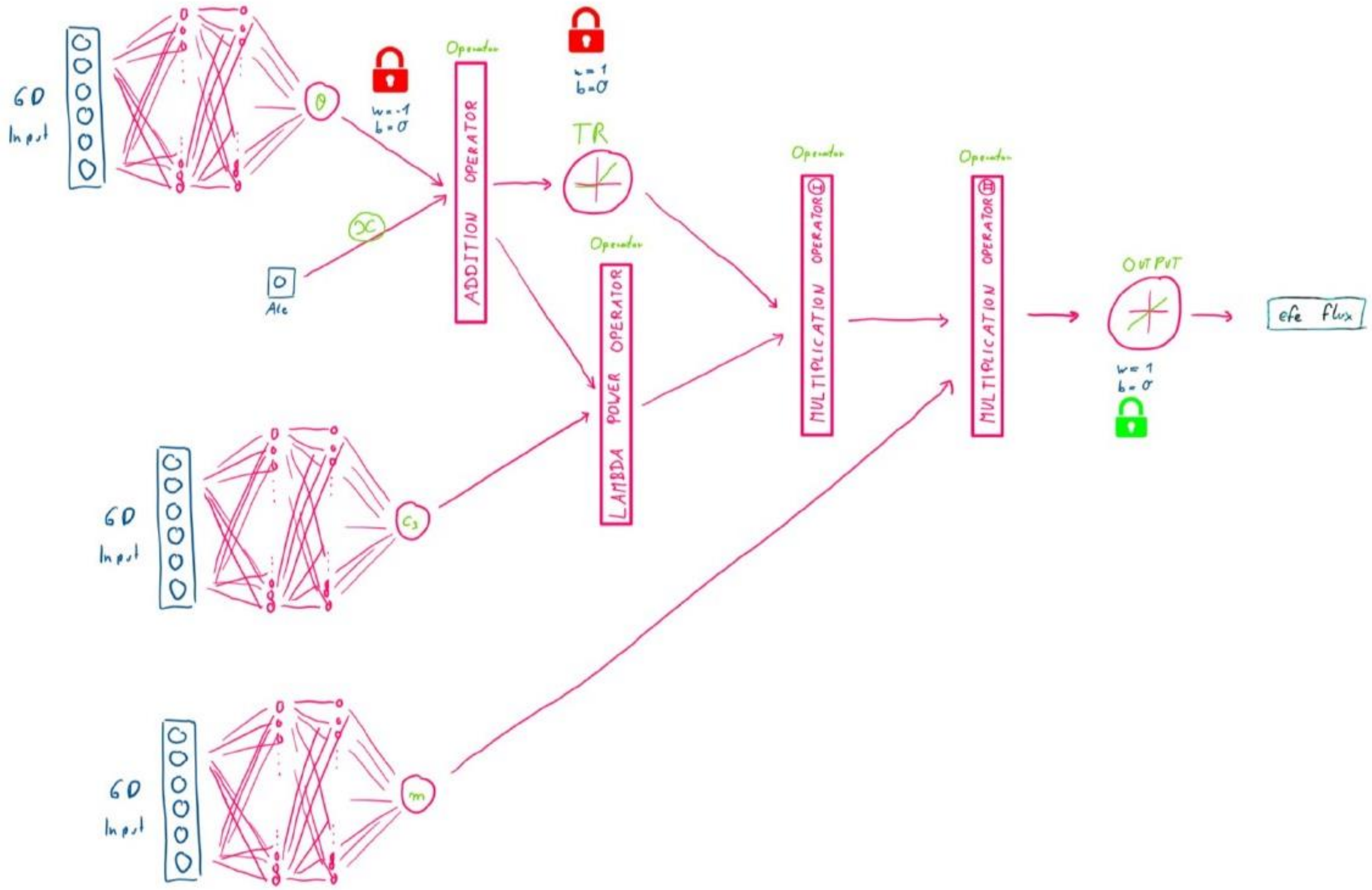
Physics/Black-Box Hybrid Architecture

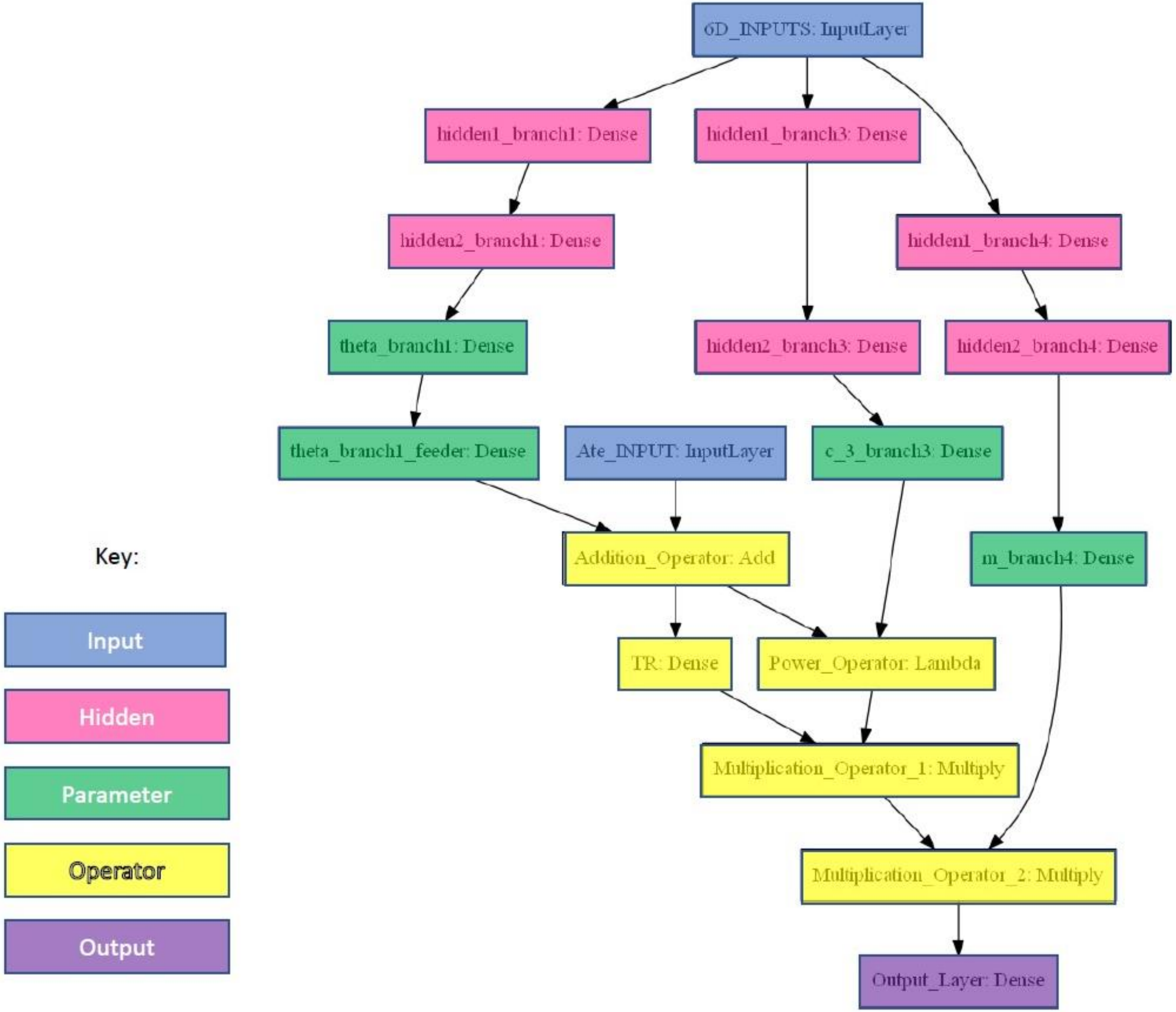


Physics/Black-Box Hybrid Architecture



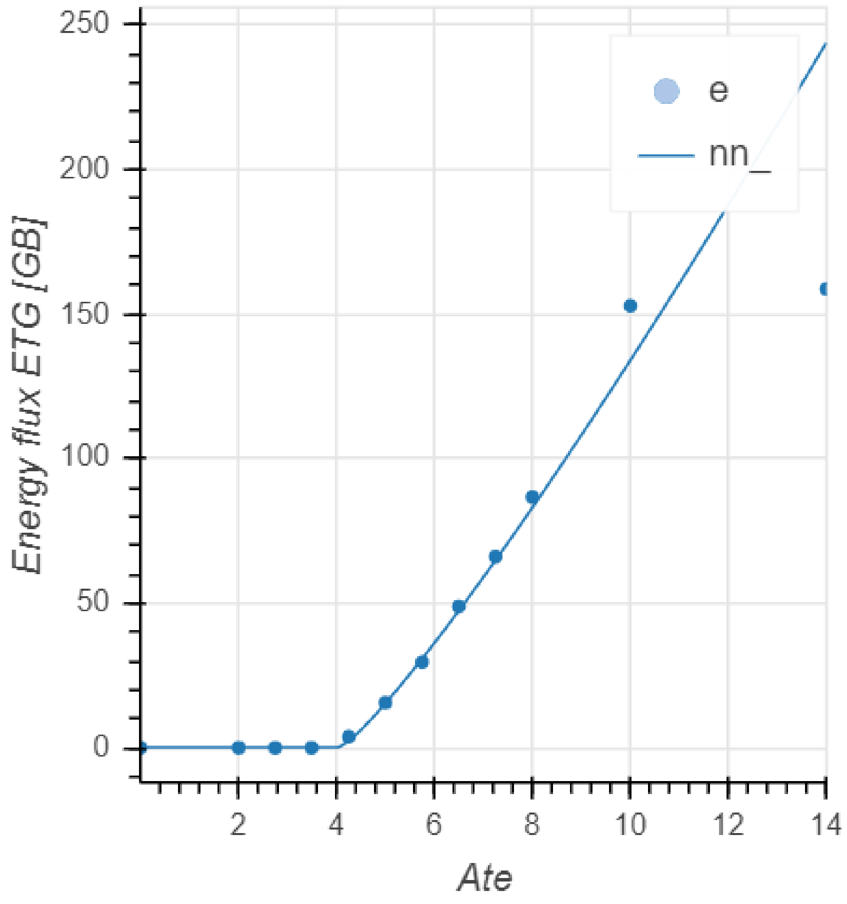
Physics/Black-Box Hybrid Architecture

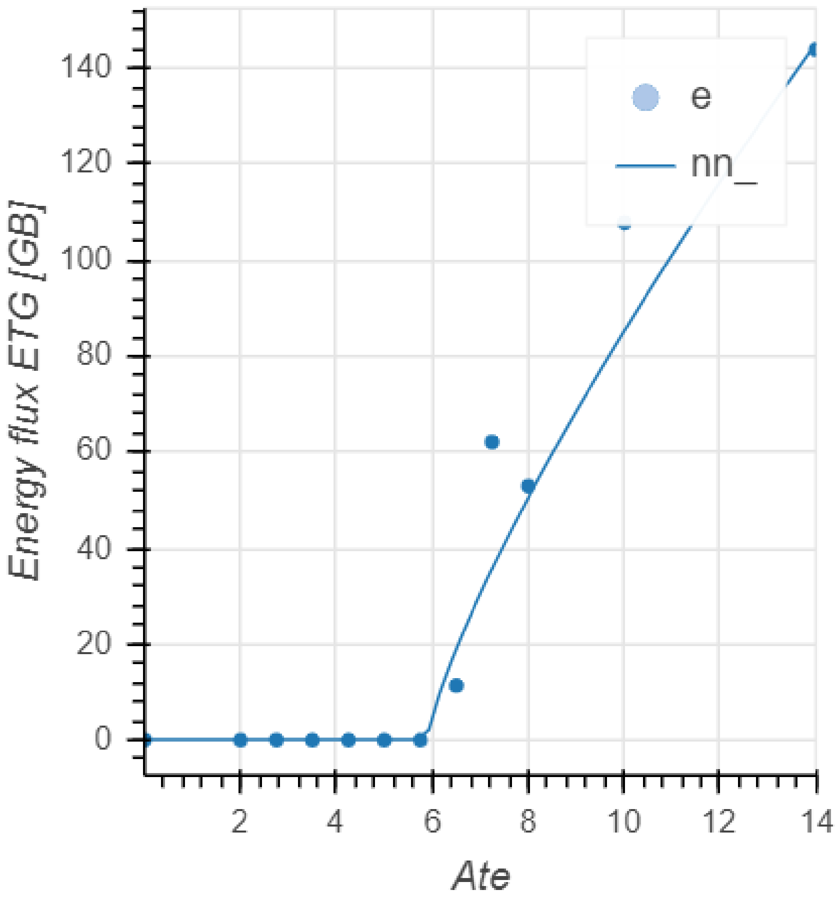




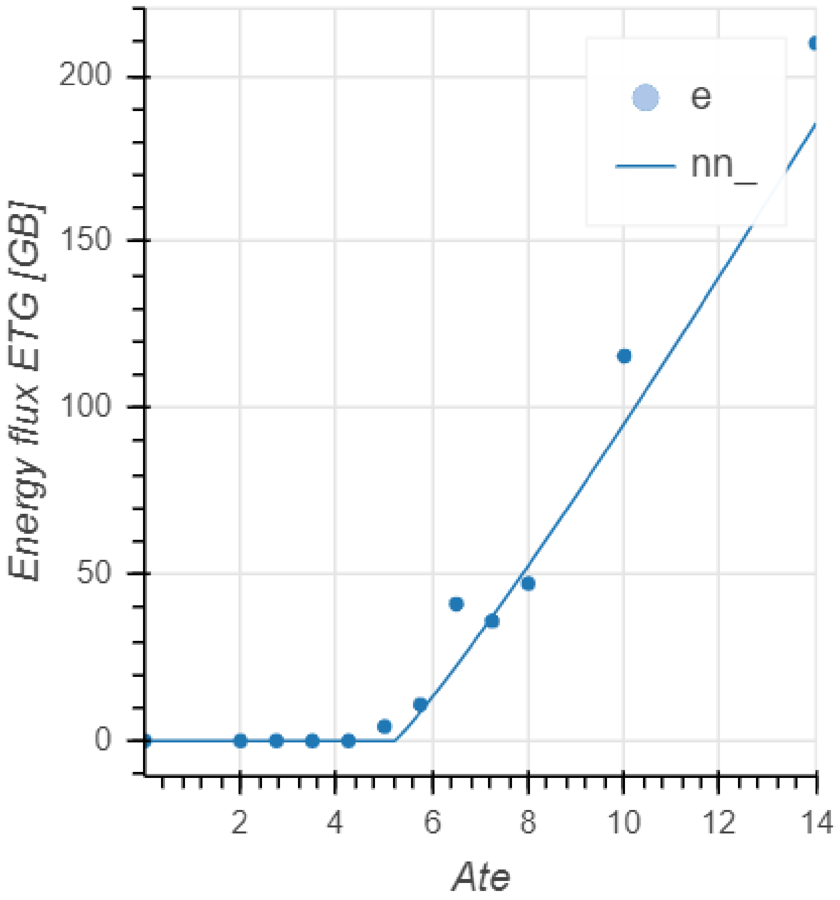
6D_INPUTS: InputLayer
hidden1_branch1: Dense
hidden1_branch3: Dense
hidden2_branch1: Dense
hidden1_branch4: Dense
theta_branch1: Dense
hidden2_branch3: Dense
hidden2_branch4: Dense
theta_branch1_feeder: Dense
Ate_INPUT: InputLayer
c_3_branch3: Dense
Key:
Addition_Operator: Add
m_branch4: Dense
Input
TR: Dense
Power_Operator: Lambda
Hidden
Multiplication_Operator_1: Multiply
Parameter
Operator
Multiplication_Operator_2: Multiply
Output
Output_Layer: Dense

Results

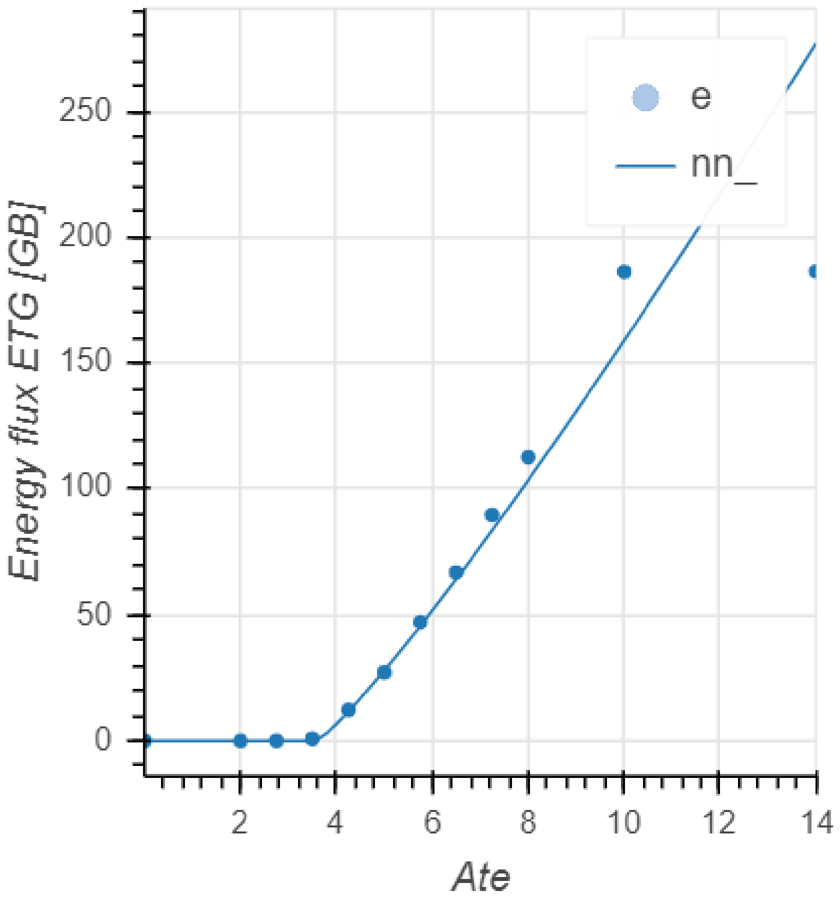

e
nn_
Energy flux ETG [GB]
250
200
150
100
50
0
2
4
6
8
10
12
14
Ate


e
nn_
Energy flux ETG [GB]
140
120
100
80
60
40
20
0
2
4
6
8
10
12
14
Ate

Results

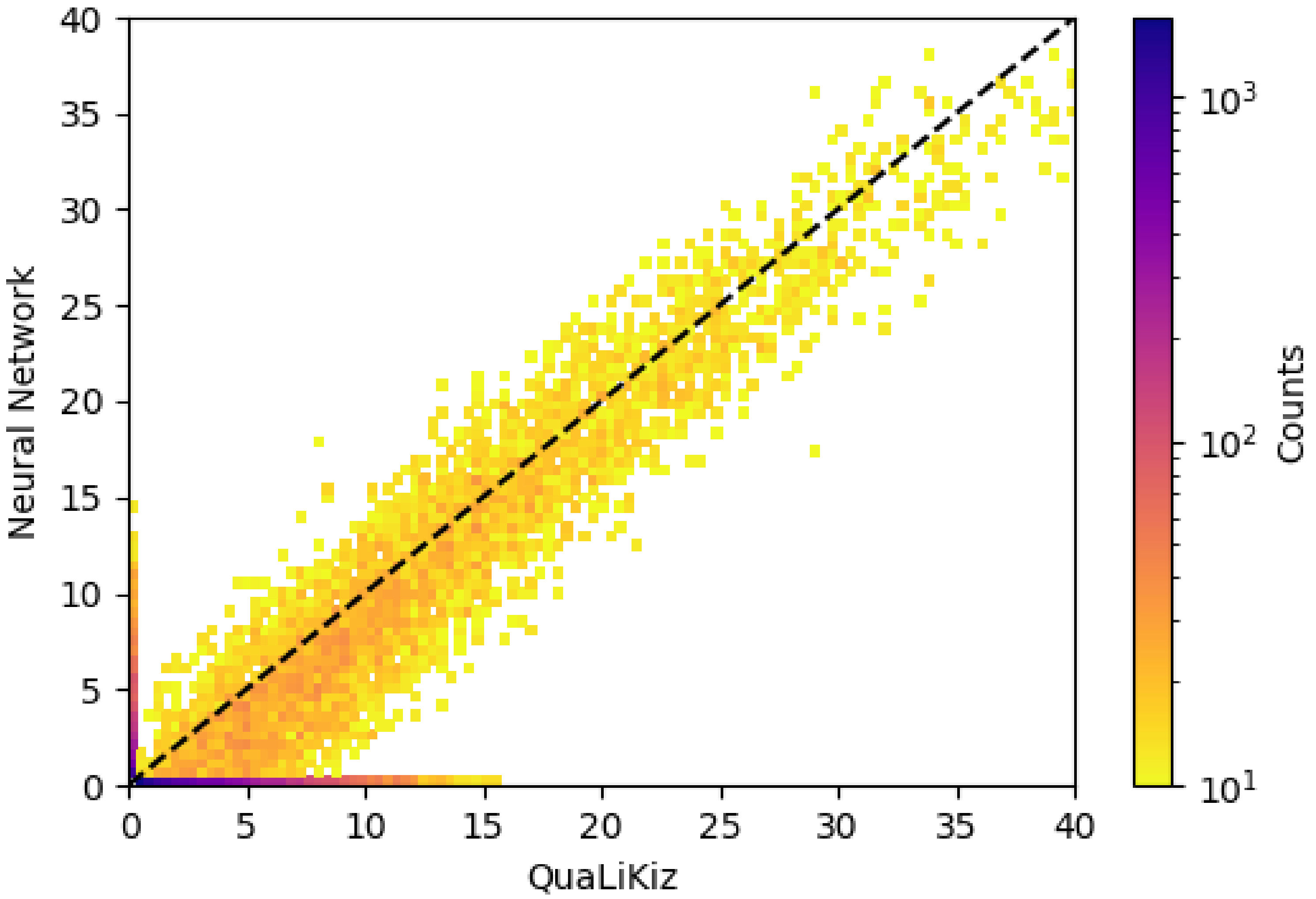


e
nn_
Energy flux ETG [GB]
Ate
0
50
100
150
200
2
4
6
8
10
12
14



e
nn_
Energy flux ETG [GB]
Ate
0
50
100
150
200
250
2
4
6
8
10
12
14

Results



Neural Network
QuaLiKiz
Counts
10^3
10^2
10^1
0
5
10
15
20
25
30
35
40

Results

METRIC	QuaLiKiz-LF	QuaLiKiz-NN
Relative mis-prediction of instability critical threshold	0.054	0.057
Absolute mis-prediction of ETG critical threshold [R/L_{Te}]	0.357	0.379
Absolute median R/L_{te} level below R/L_{Te} critical threshold where pop-back occurs	0	4.075
Fraction of R/L_{Te} slices with pop-back occurring	0	0.023
Wobble (related to second derivative and over-fitting) in unstable region	0.019	0.020

Discussion

Numerical results from **QuaLiKiz-NN-accelerated** model:

- “Pop-back” eliminated (by definition) due to CGM constraint.
- Same RMS achieved as State-Of-The-Art **QuaLiKiz-NN**[1].
- CGM provides automatic high level of regularization (smoothness) – may be advantageous when deployed in RAPTOR[2] plasma simulation due to implicit PDE solver therein, which needs smooth gradients of transport fluxes.
  - To be tested in future work.

CREDIT: [1] van de Plassche TTF 2018 (to be submitted to Nucl. Fusion)
[2] Felici PPCF 2012

Discussion:

Developed Hybridization Technique:

- Method Advantages:
  - Physics/Black-Box hybridization capability
  - Transferability
  - Human Interpretable
  - Parameter Extraction
  - “Dial-A-Uncertainty”

Discussion

Developed Hybridization Technique:

- Method Disadvantages:
  - Loss of Universality

Conclusion

- **QuaLiKiz-NN** works but added extra term to Cost Function to mitigate "Pop-back" issues
- **QuaLiKiz-LF** extends **QuaLiKiz-NN** and solves “Pop-back”
  - Similar/slightly better Critical Threshold Matching
  - Zero Pop-back
  - (Not yet tested) very smooth outputs – important for Newton solvers for implicit (fast) PDE solution of transport equations
- New Method developed:
  - Physics/Black-Box Hybridisation Capability

Outlook

- **QuaLiKiz-LF** was 7D (Inputs) and **ETG** mode only
- Next Steps:
  - Include all modes (**ITG**, **TEM**, **ETG**)
  - Go from 7D to 9D inputs
  - More sensitivity tests
  - Expanded testing tools/processes within Machine Learning pipeline
  - Plug into Integrated Modelling Suite

Acknowledgements

Jonathan Citrin, DIFFER

Karel van de Plassche, DIFFER

Frank Jenko, IPP

Hartmut Zohm, IPP

Sources

Slide 3:

- Courtesy of J. Citrin (DIFFER), image by ITER (www.iter.org)

Slide 4:

- T. D. Rognlien, Understanding of edge plasmas in magnetic fusion energy devices, Plasma Physics and Controlled Fusion, 47(5A):A283, 2005

  http://stacks.iop.org/0741-3335/47/i=5A/a=020

Slide 9:

- K.L. van de Plassche (Transport Task Force 2018)

http://www.psft.eu/ttf2018/repository/ password: ttf_2018

Sources

Slides 11 - 13:

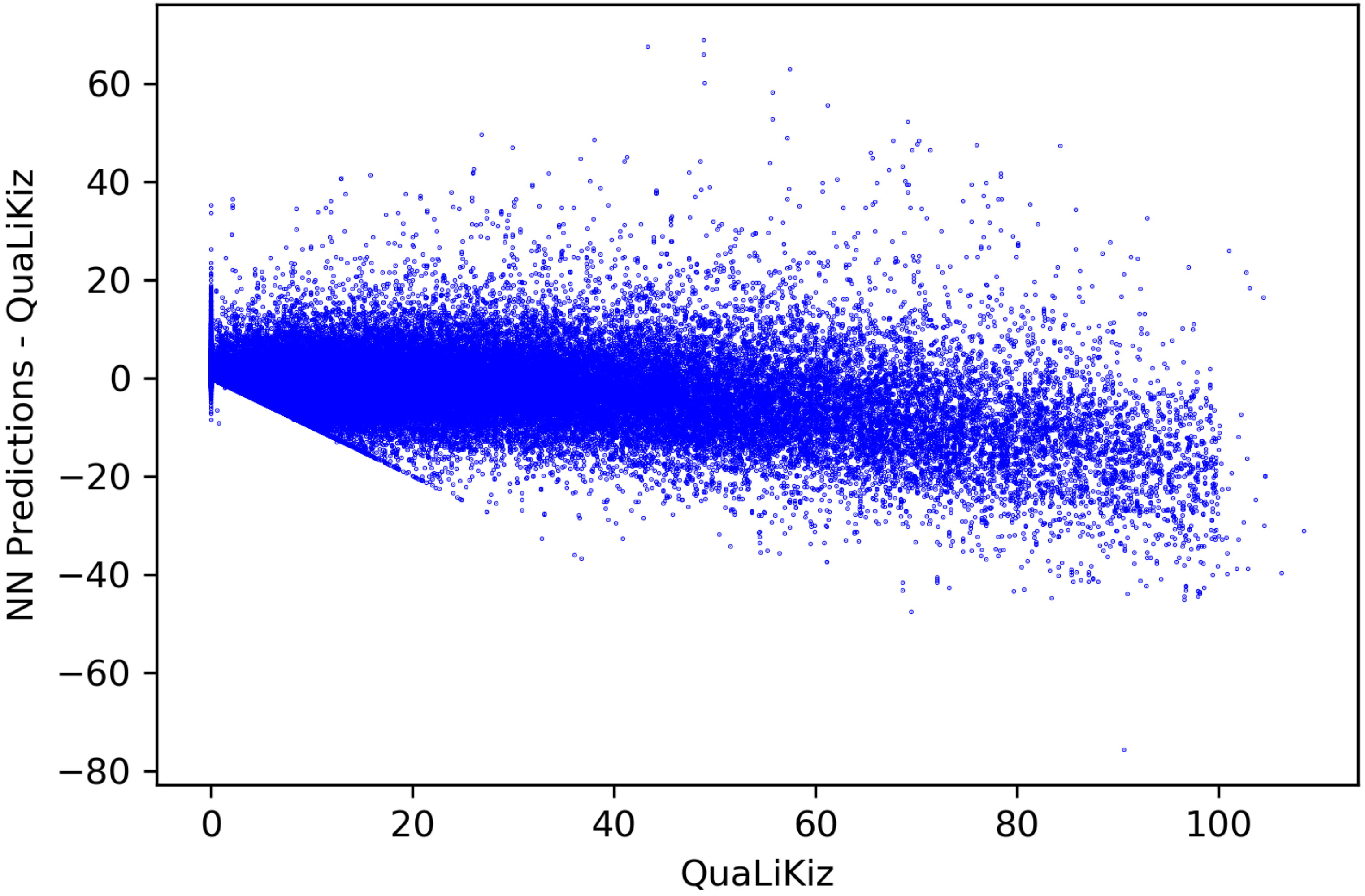
- K.L. van de Plassche et al. TTF 2018 Seville poster

Code used in this thesis can be accesses at:

www.github.com/Dan-Schaefer/IPP-Neural-Networks

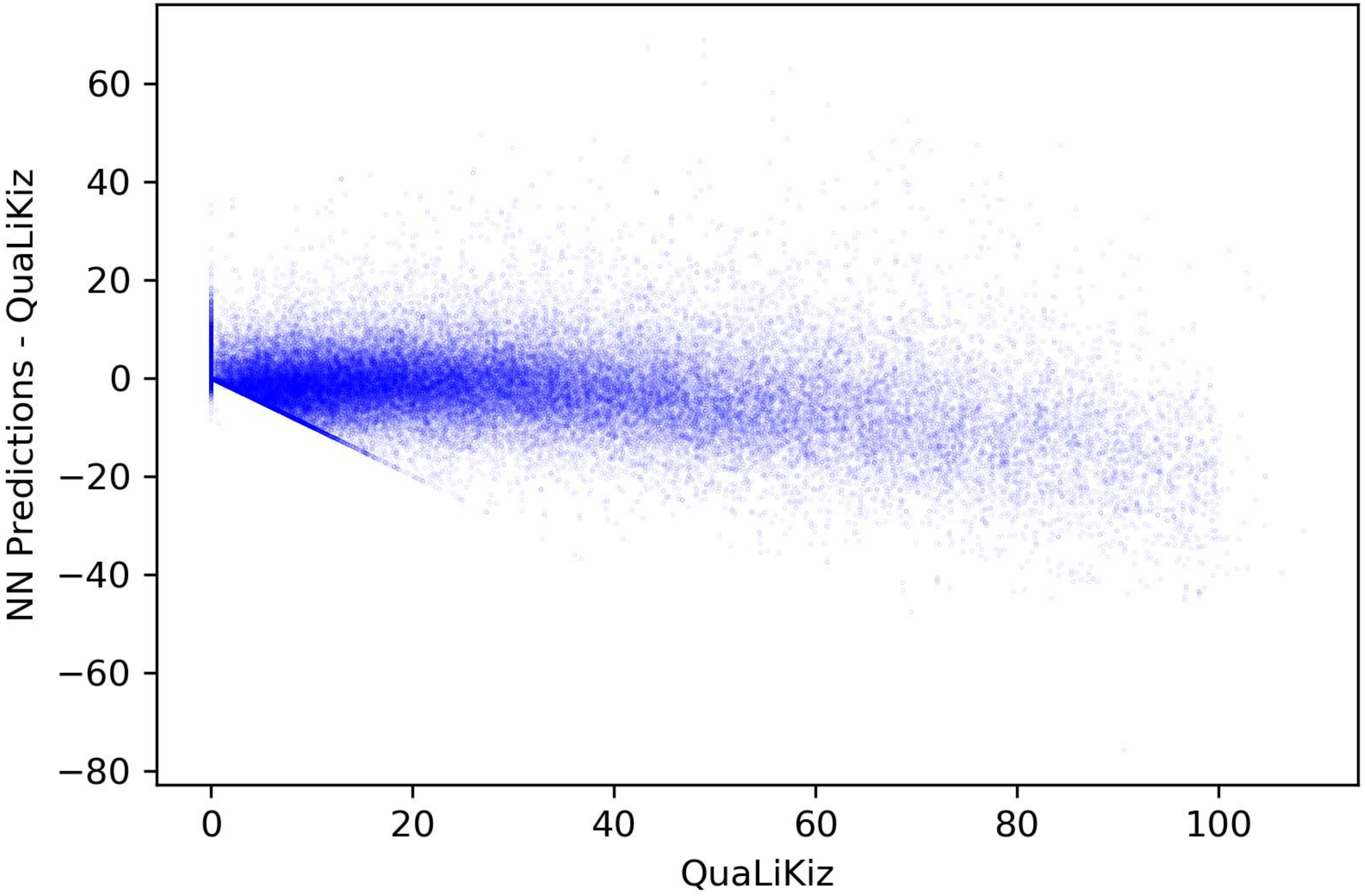
Backup Slides

Results



60
40
20
0
−20
−40
−60
−80
NN Predictions - QuaLiKiz
0
20
40
60
80
100
QuaLiKiz

Results

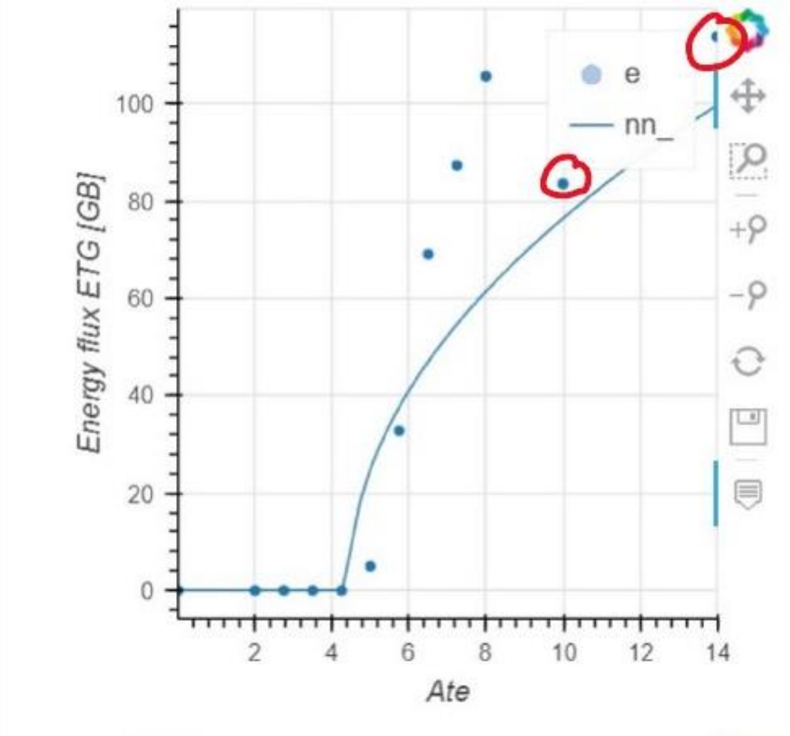


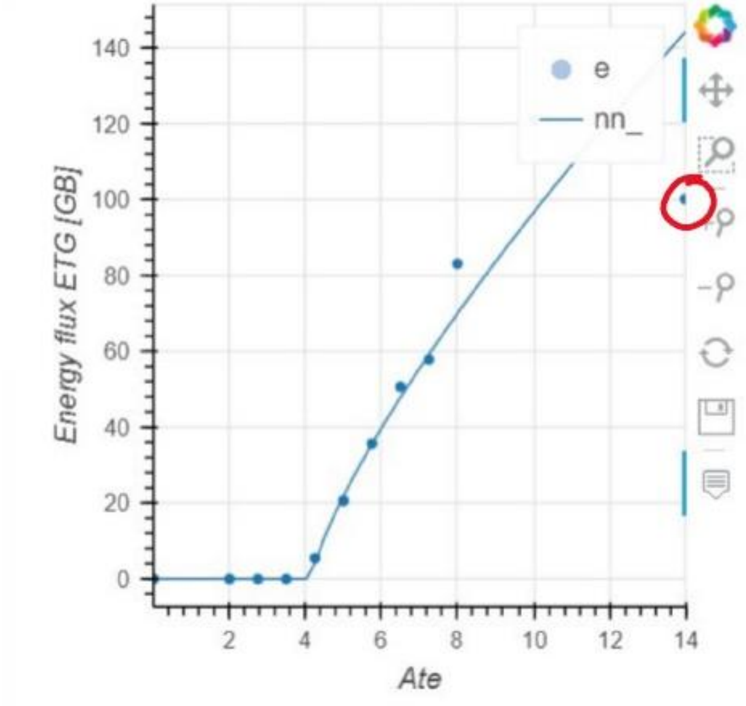
NN Predictions - QuaLiKiz
QuaLiKiz
60
40
20
0
−20
−40
−60
−80
0
20
40
60
80
100

```
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
6D_INPUTS (InputLayer)          (None, 6)            0
__________________________________________________________________________________________________
hidden1_branch1 (Dense)         (None, 30)           210         6D_INPUTS[0][0]
__________________________________________________________________________________________________
hidden2_branch1 (Dense)         (None, 30)           930         hidden1_branch1[0][0]
__________________________________________________________________________________________________
theta_branch1 (Dense)           (None, 1)            31          hidden2_branch1[0][0]
__________________________________________________________________________________________________
hidden1_branch3 (Dense)         (None, 30)           210         6D_INPUTS[0][0]
__________________________________________________________________________________________________
theta_branch1_feeder (Dense)    (None, 1)            2           theta_branch1[0][0]
__________________________________________________________________________________________________
Ate_INPUT (InputLayer)          (None, 1)            0
__________________________________________________________________________________________________
hidden2_branch3 (Dense)         (None, 30)           930         hidden1_branch3[0][0]
__________________________________________________________________________________________________
hidden1_branch4 (Dense)         (None, 30)           210         6D_INPUTS[0][0]
__________________________________________________________________________________________________
Addition_Operator (Add)         (None, 1)            0           theta_branch1_feeder[0][0]
                                                                 Ate_INPUT[0][0]
__________________________________________________________________________________________________
c_3_branch3 (Dense)             (None, 1)            31          hidden2_branch3[0][0]
__________________________________________________________________________________________________
hidden2_branch4 (Dense)         (None, 30)           930         hidden1_branch4[0][0]
__________________________________________________________________________________________________
TR (Dense)                      (None, 1)            2           Addition_Operator[0][0]
__________________________________________________________________________________________________
Power_Operator (Lambda)         (None, 1)            0           Addition_Operator[0][0]
                                                                 c_3_branch3[0][0]
__________________________________________________________________________________________________
m_branch4 (Dense)               (None, 1)            31          hidden2_branch4[0][0]
__________________________________________________________________________________________________
Multiplication_Operator_1 (Mult (None, 1)            0           TR[0][0]
                                                                 Power_Operator[0][0]
__________________________________________________________________________________________________
Multiplication_Operator_2 (Mult (None, 1)            0           m_branch4[0][0]
                                                                 Multiplication_Operator_1[0][0]
__________________________________________________________________________________________________
Output_Layer (Dense)            (None, 1)            2           Multiplication_Operator_2[0][0]
==================================================================================================
Total params: 3,519
Trainable params: 3,515
Non-trainable params: 4
__________________________________________________________________________________________________
None
```

QuaLiKiz – reduced model artefacts

- Due to occasional QuaLiKiz eigenvalue solver failure, the computed outputs in our dataset will not always be correct, i.e. there is a degree of uncertainty with regard to the data used for training.





Previous Work - QuaLiKiz

Approximations:

- Axis-symmetry
- Gyrokinetic
- Adiabatic invariance
- Local
- Quasi-linear approximation
- Electrostatic
- Collisions
- Shifted circle geometry
- Gaussian eigenfunctions
- Strong ballooning
- Strongly passing and strongly trapped
- Small Mach number

SOURCE: www.qualikiz.com

QuaLiKiz – Input and Output Variables

Species (Electron + Ion)

Filename	Full name	Units	Definition
An	Logarithmic density gradient	-	$An_s \equiv -\frac{R_0}{n_s}\frac{\mathrm{d}n_s}{\mathrm{d}r}$
At	Logarithmic temperature gradient	-	$At_s \equiv -\frac{R_0}{T_s}\frac{\mathrm{d}T_s}{\mathrm{d}r}$
T	Temperature	keV	-
anis	Temperature anisotropy	-	$\left(\frac{T_\perp}{T_\parallel}\right)_{LFS}$
danisdr	Radial gradient of temperature anisotropy	-	-

SOURCE: https://github.com/QuaLiKiz-group/QuaLiKiz/wiki/Input-and-output-variables

QuaLiKiz – Input and Output Variables

Electrons

Filename	Full name	Units	Definition
n	Density	$10^{19}m^{-3}$	
typee	Electron type	-	

The electron type can be:

typee	effect
1	Active electrons
2	Adiabatic electrons
3	Passing at ion scales

SOURCE: https://github.com/QuaLiKiz-group/QuaLiKiz/wiki/Input-and-output-variables

QuaLiKiz – Input and Output Variables

Ions

Filename	Full name	Units	Definition
normni	Normalized density	-	$\frac{n_i}{n_e}$
Ai	Ion mass	proton masses	-
Zi	Ion change	e	-
typei	Ion type	-	-

The ion type can be:

typei	effect
1	Active ions
2	Adiabatic ions
3	Tracer (i.e. for heavy impurities). Note: ninorm input is arbitrary for this case
4	Tracer in dispersion relation, but include the impact on Zeff and collisionality. Note: ninorm has to obey quasineutrality in this case

SOURCE: https://github.com/QuaLiKiz-group/QuaLiKiz/wiki/Input-and-output-variables

QuaLiKiz – Input and Output Variables

Geometry

Filename	Full name	Units	Definition
x	Midplane-averaged minor radius of flux surface, normalized by midplane-averaged minor radius of last-closed-flux-surface.	-	$\frac{r}{a} = \frac{r}{R_{\text{min}}} = \frac{r_{\text{out}} + r_{\text{in}}}{r_{\text{out,LCFS}} + r_{\text{in,LCFS}}}$
Ro	Midplane-averaged major radius of flux surface. Can include Shafranov shift.	m	$\frac{R_{\text{out}} + R_{\text{in}}}{2}$
rho	Normalized toroidal flux coordinate. Only used in QuaLiKiz to set the non-rotation region when using `rot_flag=2`	-	-
Bo	Magnetic field at magnetic axis	T	-
R0	Midplane-averaged major radius of last-closed-flux-surface.	m	$\frac{R_{\text{out,LCFS}} + R_{\text{in,LCFS}}}{2}$
Rmin	Midplane-averaged minor radius of last-closed-flux-surface.	m	$\frac{r_{\text{out,LCFS}} + r_{\text{in,LCFS}}}{2}$
q	Safety factor	-	-
smag	Magnetic shear	-	$r\frac{q'}{q}$
alpha	MHD alpha	-	$\alpha_{MHD} \equiv q^2 \sum_s \beta_s \left(\frac{R}{L_{T_s}} + \frac{R}{L_{n_s}}\right)$

SOURCE: https://github.com/QuaLiKiz-group/QuaLiKiz/wiki/Input-and-output-variables

QuaLiKiz – Input and Output Variables

Output - Fluxes

All output files have the structure (flux name)(species [i|e])(mode [|ETG|ITG|TEM])_(style [GB|SI|cm])

Filename	Full name (units refer to SI outputs only)	`phys_meth`	Definition
pf	Particle flux [m^-2 s^-1]	0	Γ_s
ef	Heat flux [W m^-2]	0	q_s
vf	Angular momentum flux [kg s^-2]	0	Π_s
df	Particle diffusivity [m^2/s]	1	D_s
vt	Particle thermodiffusion [m/s]	1	$V_{P,s}^{th}$
vr	Particle rotodiffusion [m/s]	1	$V_{P,s}^{rot}$
vc	Particle pure convective term [m/s]	1	$V_{P,s}^{th}$
chie	Heat conductivity [m^2/s]	2	χ_s
ven	Heat thermodiffusion [m/s]	2	$V_{H,s}^{n}$
ver	Heat rotodiffusion [m/s]	2	$V_{H,s}^{rot}$
vec	Heat pure convective term [m/s]	2	$V_{H,s}^{c}$

SOURCE: https://github.com/QuaLiKiz-group/QuaLiKiz/wiki/Input-and-output-variables