Fast integrated modelling using a neural network surrogate model for turbulent transport

Karel van de Plassche[1,2]

J. Citrin[1], F. J. Casson[3], F. Felici[4], A. Ho[2],
F. Koechl[3], S. Van Mulders[4], O. Sauter[4], and JET Contributors*

[1] *DIFFER, PO Box 6336, 5600 HH Eindhoven, The Netherlands*
[2] *Eindhoven University of Technology, Eindhoven, The Netherlands*
[3] *CCFE, Culham Science Centre, OX14 3DB, Abingdon, UK*
[4] *EPFL-SPC, CH-1015 Lausanne, Switzerland*

* *See the author list of* 'Overview of JET results for optimising ITER operation' *by J. Mailloux et al.to be published in Nuclear Fusion Special issue: Overview and Summary Papers from the 28th Fusion Energy Conference (Nice, France, 10-15 May 2021)*

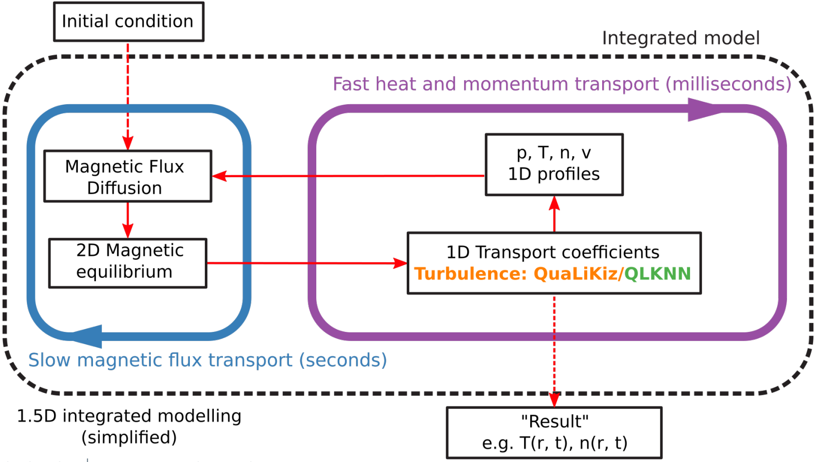
November 30th, 2021

Section A

Surrogates in integrated modelling (introducing QLKNN)

1. Surrogates in integrated modelling (introducing QLKNN)
2. Application of PoP QLKNN to ITER cases
3. Extensions to QLKNN model
4. Wrap-up

Integrated modelling primer



QuaLiKiz: fast and accurate reduced turbulence model

- Reduced quasi-linear gyrokinetic code
- Multiple examples of agreement with experiments[1,2,3,4,5]
- 6 orders of magnitude faster than nonlinear calculations
  - Still in agreement with nonlinear flux due to saturation rule[6,7]
- 10 CPU seconds to calculate turbulent fluxes at a single radial position
- QuaLiKiz[8] + JINTRAC: $\mathcal{O}(24)$ *hrs* per second of JET

Open source: see http://qualikiz.com for more information

[1] C. Bourdelle PPCF 2016
[2] S. Breton NF 2018
[3] O. Linder NF 2019
[4] A. Ho NF 2019
[5] F. Casson NF 2020
[6] A. Casati NF 2009
[7] J. Citrin PoP 2012
[8] C. Stephens JPP 2021

Faster modelling by replacing QuaLiKiz ...

Local dimensionless plasma parameters

R/L_{T_e}
R/L_{T_i}
R/L_n
q
$\hat{s}$
r/R
T_i/T_e
ν^*
Z_{eff}
(γ_E)
$\vdots$

→ 10^1 CPUs*

QuaLiKiz →

Heat flux $q_{i,e}$
Particle flux $\Gamma_{i,e}$
Momentum flux Π_i

* Depends on "physics". It's 1 (no rotation) to 4 (rotation) seconds per radial point per wavenumber

... by fast Neural Network

Local dimensionless plasma parameters

R/L_{T_e}
R/L_{T_i}
R/L_n
q
$\hat{s}$
r/R
T_i/T_e
ν^*
Z_{eff}
(γ_E)

10^{-2} – 10^{-5} CPUs

Neural network(s)

In GyroBohm units

Heat flux $q_{i,e}$
Particle flux $\Gamma_{i,e}$
Jacobians $\partial q_{i,e}/\partial\left(R/L_{T_e}, ..., \gamma_E\right)$

Dataset of 300 million QuaLiKiz points has been generated

Dataset spans wide core-relevant regime, and is freely available on Zenodo: doi.org/10.5281/zenodo.3497065, online visualization:

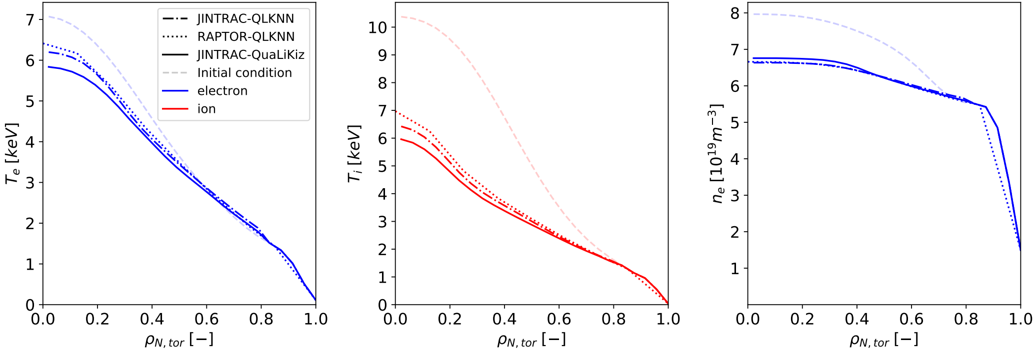
dataslicer.qualikiz.com

variable	# points	min	max
$k_\theta \rho_s \leq 2$	10	0.1	2
$k_\theta \rho_s > 2$	8	3.5	36
R/L_{T_e}	12	0	14
R/L_{T_i}	12	0	14
R/L_n	12	-5	6
q	10	0.66	15
$\hat{s}$	10	-1	5
r/R	8	0.03	0.33
T_i/T_e	7	0.25	2.5
ν^*	6	1×10^{-5}	1
Z_{eff}	5	1	3
Total flux calculations	3×10^8	$\approx$ 1.3 MCPUh	

Benchmark: good match between QLKNN and QuaLiKiz

Simplified physics case

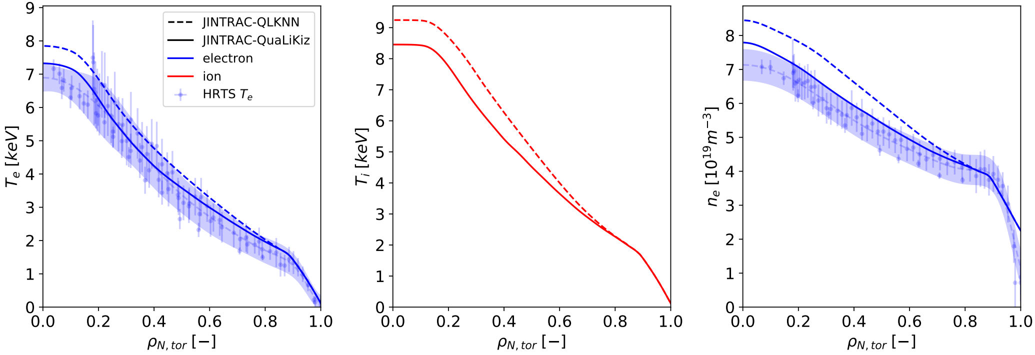
Based on high performance baseline JET 92436 [A. Ho *et al.* NF 2019]



JINTRAC-RAPTOR-QLKNN benchmark at t=22.75s [K.L. van de Plassche et al. PoP 2020]

No full QuaLiKiz surrogate, but most important features are captured

Full physics case, based on high performance hybrid JET 92398 [Casson IAEA 2018]
Parameters not yet included in this QLKNN (e.g. α, $R/L_{n_{i,imp}}$) can play a role; still good match!



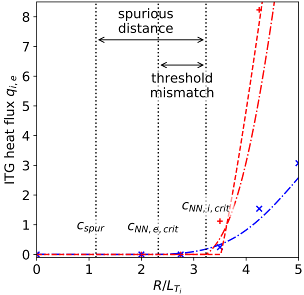
JINTRAC-QLKNN benchmark at t=22.75s [K.L. van de Plassche et al. PoP 2020]

Essential: Capture physics in the surrogate model

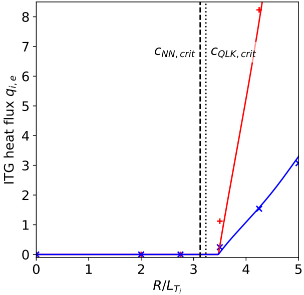
Same threshold for all transport channels ($q_{i,e}, \Gamma_e$) essential

- Global regression measures less important than local features
- Global RMS error weak indicator of surrogate model quality

Essential for integrated modelling: Include this in surrogate model.



Bad fit

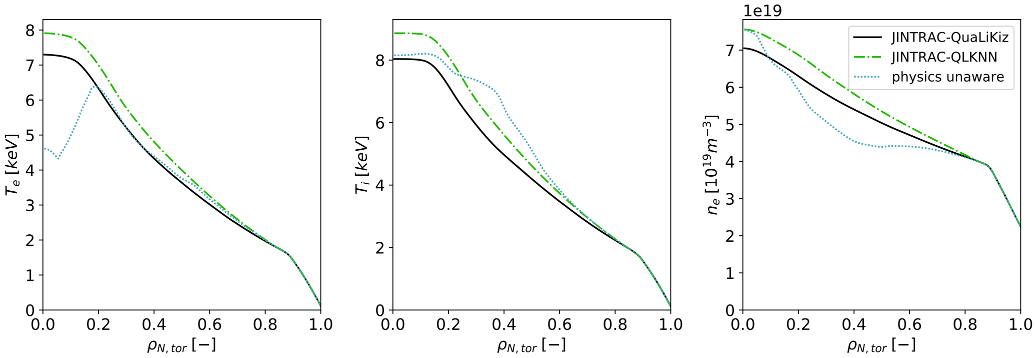


Good fit

Example of matching ITG temperature gradient critical thresholds using customised cost function and network structure (left)

Physics-unaware: Less quality for the same rms!

Full physics case, based on high performance hybrid JET 92398 [Casson IAEA 2018]
NO special cost function, NO special train targets, *same RMS of fit!*



JINTRAC-QLKNN-hyper-10D vs a network trained on the full fluxes at t=22.75s [K.L. van de Plassche et al. PoP 2020]

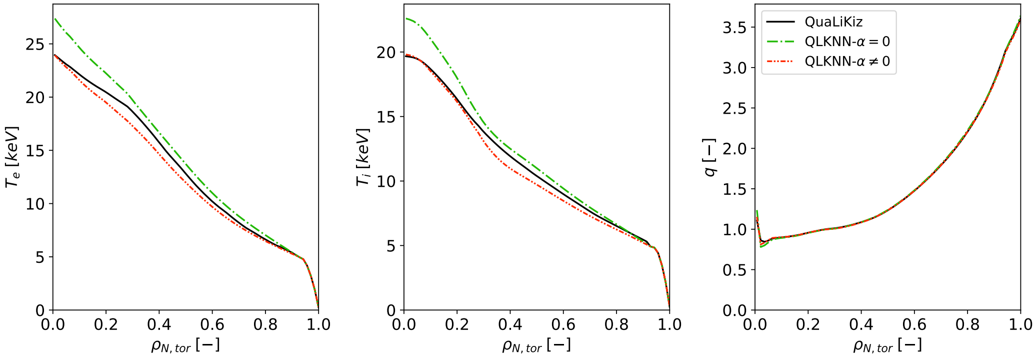
Section B

Application of PoP QLKNN to ITER cases

1. Surrogates in integrated modelling (introducing QLKNN)
2. Application of PoP QLKNN to ITER cases
3. Extensions to QLKNN model
4. Wrap-up

New regime: ITER baseline core modelling

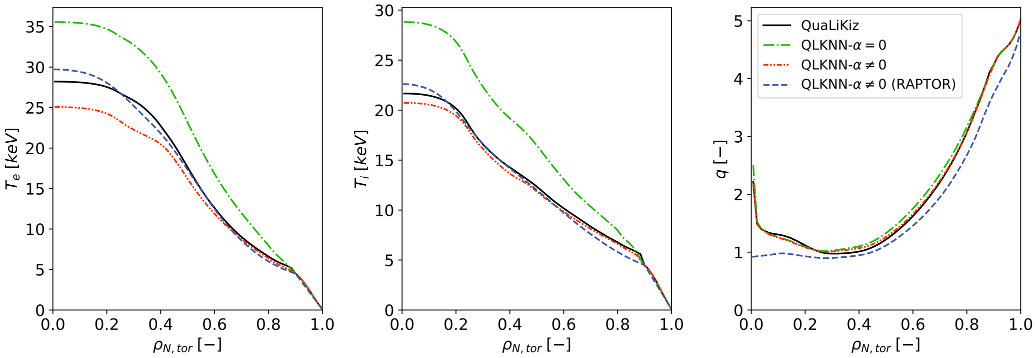
Very encouraging results, QLKNN performs similarly as for JET simulations in PoP2020
Based on [F. Koechl *et al.* IAEA 2018, P. Mantica *et al.* PPCF 2020], pedestal constrained by EPED.



QLKNN-hyper-α = 0 on ITER hybrid with $\alpha \neq 0$ rule-of-thumb $\hat{s}_{eff} = \hat{s} - {}^{\alpha}/_{2}$. $Q \approx 9$

New regime: ITER hybrid

Reference run for WIP optimization exercise. α rule important for hybrid scenario due to large $\hat{s}$ and large α. Based on [J. Citrin *et al.* NF 2010]. Final version in [S. Van Mulders *et al.* NF 2021]



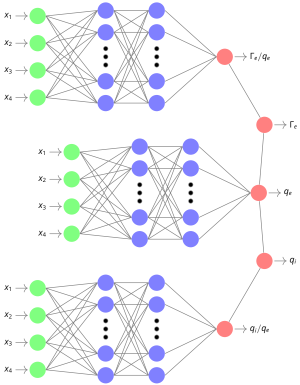
QLKNN-hyper-α = 0 on ITER hybrid with $\alpha \neq 0$ rule-of-thumb $\hat{s}_{eff} = \hat{s} - \alpha/2$ in RAPTOR optimalization [S. Van Mulders et al.]

Section C

Extensions to QLKNN model

1. Surrogates in integrated modelling (introducing QLKNN)
2. Application of PoP QLKNN to ITER cases
3. Extensions to QLKNN model
4. Wrap-up

Include physics in network structure: QLKNN-hyper



General unconstrained 'combined NN'

For ITG general input $\mathbf{x}_g$ and special input x_s

$$\mathbf{x}_g \in \{R/L_{T_e}, R/L_n, q, ..., Z_{eff}\}$$
$$x_s \in \{R/L_{T_i}\}$$

QLKNN-hyper

$$q_e(\mathbf{x}_g, x_s) = NN_{q_e/q_i}(\mathbf{x}_g, x_s) * NN_{q_i}(\mathbf{x}_g, x_s)$$
$$q_i(\mathbf{x}_g, x_s) = NN_{q_i}(\mathbf{x}_g, x_s)$$
$$\Gamma_e(\mathbf{x}_g, x_s) = NN_{\Gamma_e/q_i}(\mathbf{x}_g, x_s) * NN_{q_i}(\mathbf{x}_g, x_s)$$

QLKNN family: Include relevant physics in different ways

- We want to approximate the full QuaLiKiz model *including isotopes, impurities*
  - Challenge: Needs larger dataset, make sure QuaLiKiz is okay
  - Solution: Extend dataset ⇒ QLKNN-hyper-*11D*
- We want to extend incorporation of known physics constraints
  - Solution: Include physics in network architecture itself: *QLKNN-HornNet* (P. Horn *et al.*, in preparation)
  - Solves challenge: Combing QLKNN-hyper nets compounds errors
  - Solves challenge: QLKNN-hyper training strategy allows for non-physical freedom
- We want to approximate the full QuaLiKiz model
  - Challenge: Hypercube dataset scales poorly with input dimensionality, multiple TCPUh!
  - Solution: Base on experimental data ⇒ *QLKNN-jetexp*, on A. Ho *et al.* PoP 2021
  - New challenge: Need large database of profiles of different machines

QLKNN-hyper-11D: Extend to isotopes and impurities

- Extend dataset with impurity density gradients
- Includes updates to physics and numerics, see QuaLiKiz-2.8.0[1]
- Data generated, NN training pipeline preparation ongoing. ***2TiB*** of compressed netCDF before filtering!

variable	# points	min	max
$k_\theta \rho_s \leq 2$	10	0.1	2
$k_\theta \rho_s > 2$	8	3.5	36
R/L_{T_e}	11	0	14
R/L_{T_i}	11	0	14
R/L_{n_e}	11	-5	5
$R/L_{n_{i,0}}$	12	-15	15
q	9	0.66	10
$\hat{s}$	9	-1	4
r/R	8	0.1	0.9
T_i/T_e	7	0.25	2.5
ν^*	5	0	0.1
n_i/n_e	4	0	0.3
Total flux calculations	2×10^9	$\approx$ 8 MCPUh	

[1] But not the most recent QLK version `https://gitlab.com/qualikiz-group/QuaLiKiz/-/tags/2.8.2`

QLKNN family: Include relevant physics in different ways

- We want to approximate the full QuaLiKiz model *including isotopes, impurities*
  - Challenge: Needs larger dataset, make sure QuaLiKiz is okay
  - Solution: Extend dataset ⇒ QLKNN-hyper-*11D*
- We want to extend incorporation of known physics constraints
  - Solution: Include physics in network architecture itself: *QLKNN-HornNet* (P. Horn *et al.*, in preparation)
  - Solves challenge: Combing QLKNN-hyper nets compounds errors
  - Solves challenge: QLKNN-hyper training strategy allows for non-physical freedom
- We want to approximate the full QuaLiKiz model
  - Challenge: Hypercube dataset scales poorly with input dimensionality, multiple TCPUh!
  - Solution: Base on experimental data ⇒ *QLKNN-jetexp*, on A. Ho *et al.* PoP 2021
  - New challenge: Need large database of profiles of different machines

QLKNN family: Include relevant physics in different ways

- We want to approximate the full QuaLiKiz model *including isotopes, impurities*
  - Challenge: Needs larger dataset, make sure QuaLiKiz is okay
  - Solution: Extend dataset ⇒ QLKNN-hyper-*11D*
- We want to extend incorporation of known physics constraints
  - Solution: Include physics in network architecture itself: *QLKNN-HornNet* (P. Horn *et al.*, in preparation)
  - Solves challenge: Combing QLKNN-hyper nets compounds errors
  - Solves challenge: QLKNN-hyper training strategy allows for non-physical freedom
- We want to approximate the full QuaLiKiz model
  - Challenge: Hypercube dataset scales poorly with input dimensionality, multiple TCPUh!
  - Solution: Base on experimental data ⇒ *QLKNN-jetexp*, on A. Ho *et al.* PoP 2021
  - New challenge: Need large database of profiles of different machines

Include physics in network structure: QLKNN-HornNet

New work by P. Horn, K.L. van de Plassche *et al.*

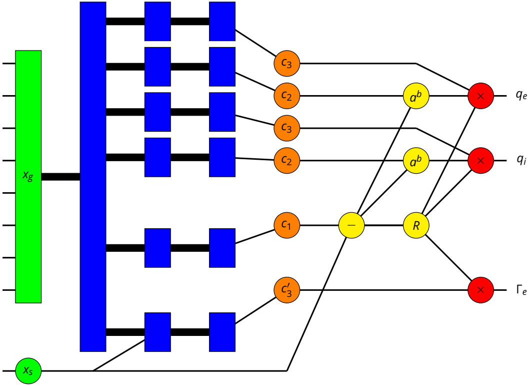
- Inspired by late-fusion techniques in RAPTOR [F. Felici, S. Van Mulders *et al.*]

Strategy: Force a Critical Gradient Model *directly into network architecture*

- Pro: Still general, but smooth 1st derivative of IO mapping
- Pro: Simple input/output derivatives
- Pro: Very sharp turbulent threshold

Combining CGM and NNs: The power of general NN tools

NN training techniques can be applied to arbitrary functions. So we can include a Critial Gradient Model as such:



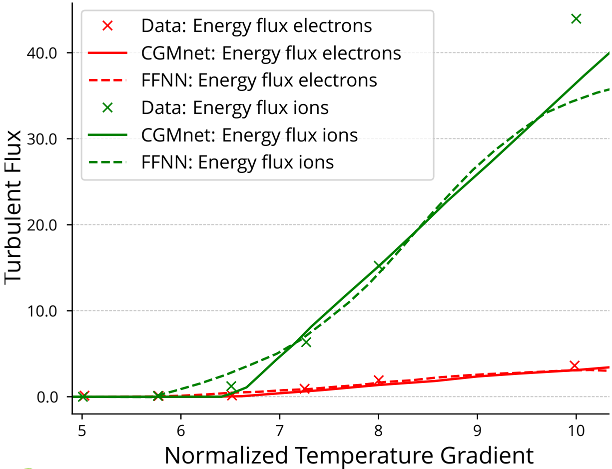
QLKNN-HornNet

$$q_e(\mathbf{x}_g, x_s) = c_{3,e} R(x_s - c_1)(|x_s - c_1|)^{c_{2,e}}$$
$$q_i(\mathbf{x}_g, x_s) = c_{3,i} R(x_s - c_1)(|x_s - c_1|)^{c_{2,i}}$$
$$\Gamma_e = NN(\mathbf{x}_g, x_s) R(x_s - c_1)$$

"slope" $c_{3,[i,e]}(\mathbf{x}_g)$
"threshold location" $c_1(\mathbf{x}_g)$
"bendiness" $c_{2,[i,e]}(\mathbf{x}_g)$
network output $NN(\mathbf{x}_g, x_s)$
Rectifier $R(\cdot)$

Gives a well-constrained neural network model

Standalone QLKNN-HornNet performs well



- Shown in [P. Horn *et al.* MSc. thesis]
- Shows "bendiness" of QLKNN-hyper
- Warning: Bad slice of QLKNN-hyper, good slice of QLKNN-HornNet
- WIP: Implementation in RAPTOR. Paper to be submitted

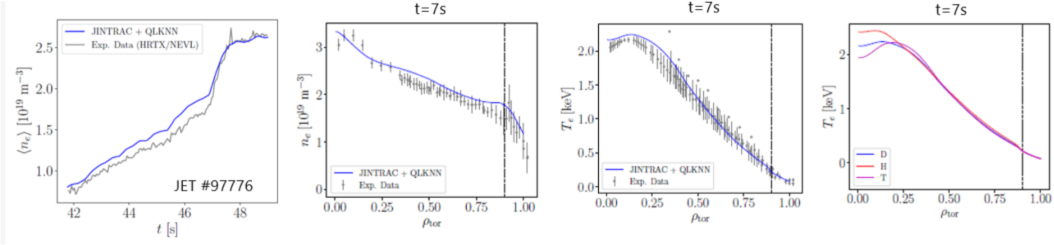
QLKNN family: Include relevant physics in different ways

- We want to approximate the full QuaLiKiz model *including isotopes, impurities*
  - Challenge: Needs larger dataset, make sure QuaLiKiz is okay
  - Solution: Extend dataset $\Rightarrow$ QLKNN-hyper-*11D*
- We want to extend incorporation of known physics constraints
  - Solution: Include physics in network architecture itself: *QLKNN-HornNet* (P. Horn *et al.*, in preparation)
  - Solves challenge: Combing QLKNN-hyper nets compounds errors
  - Solves challenge: QLKNN-hyper training strategy allows for non-physical freedom
- We want to approximate the full QuaLiKiz model
  - Challenge: Hypercube dataset scales poorly with input dimensionality, multiple TCPUh!
  - Solution: Base on experimental data $\Rightarrow$ *QLKNN-jetexp*, on A. Ho *et al.* PoP 2021
  - New challenge: Need large database of profiles of different machines

QLKNN family: Include relevant physics in different ways

- We want to approximate the full QuaLiKiz model *including isotopes, impurities*
  - Challenge: Needs larger dataset, make sure QuaLiKiz is okay
  - Solution: Extend dataset $\Rightarrow$ QLKNN-hyper-*11D*
- We want to extend incorporation of known physics constraints
  - Solution: Include physics in network architecture itself: *QLKNN-HornNet* (P. Horn *et al.*, in preparation)
  - Solves challenge: Combing QLKNN-hyper nets compounds errors
  - Solves challenge: QLKNN-hyper training strategy allows for non-physical freedom
- We want to approximate the full QuaLiKiz model
  - Challenge: Hypercube dataset scales poorly with input dimensionality, multiple TCPUh!
  - Solution: Base on experimental data $\Rightarrow$ *QLKNN-jetexp*, on A. Ho *et al.* PoP 2021
  - New challenge: Need large database of profiles of different machines

QLKNN-jetexp-15D highlight



Has been applied for T-campaign JET hybrid scenario ramp-up optimization [A. Ho *et al.* APS invited 2021]

Section D

Wrap-up

1. Surrogates in integrated modelling (introducing QLKNN)
2. Application of PoP QLKNN to ITER cases
3. Extensions to QLKNN model
4. Wrap-up

Conclusion

QLKNN family of models:

- are *ready for exploitation* in RAPTOR and JINTRAC
- are being integrated in *multiple frameworks*
- enables QuaLiKiz approximation *3-5 orders of magnitude faster*

Improvements to QuaLiKiz and its family of surrogate models is ongoing

qualikiz.com　　gitlab/QLKNN-fortran　　gitlab/QLKNN-develop