13th ITER International School
Nagoya Prime Central Tower, Nagoya, Japan
9-13 December 2024

Neural-network emulation of simulation codes with a high computational complexity

Emi Narita
Kyoto University
e-mail : narita.emi.5s@kyoto-u.ac.jp

Self-introduction

- Name: Emi NARITA
- Education and positions:

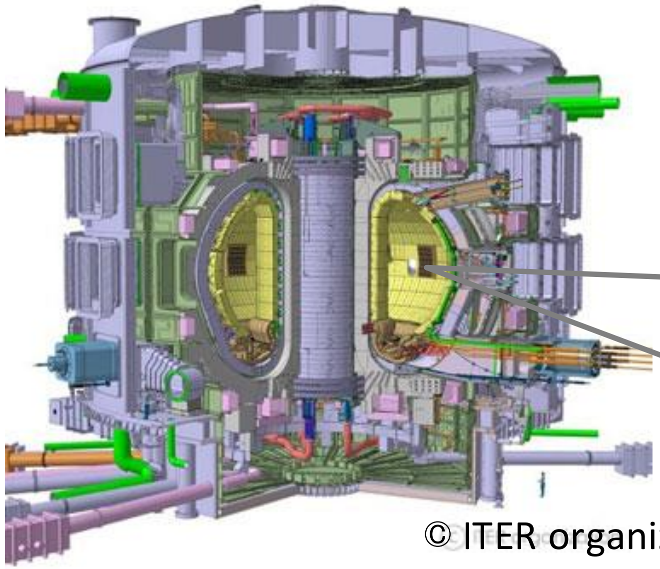
March 2011:	Bachelor of Engineering, Osaka University
March 2012:	Master of Engineering, Osaka University
March 2025:	Ph.D. of Engineering, Osaka University
April 2015 – March 2019:	Post-Doctoral Fellow, National Institutes for Quantum and Radiological Science and Technology (QST)
April 2019 – May 2023:	Researcher, National Institutes for Quantum and Radiological Science and Technology (QST)
June 2023 – Present:	Senior Lecturer, Kyoto University

- Keywords
  - Transport modeling
  - Machine learning
  - Integrated simulations

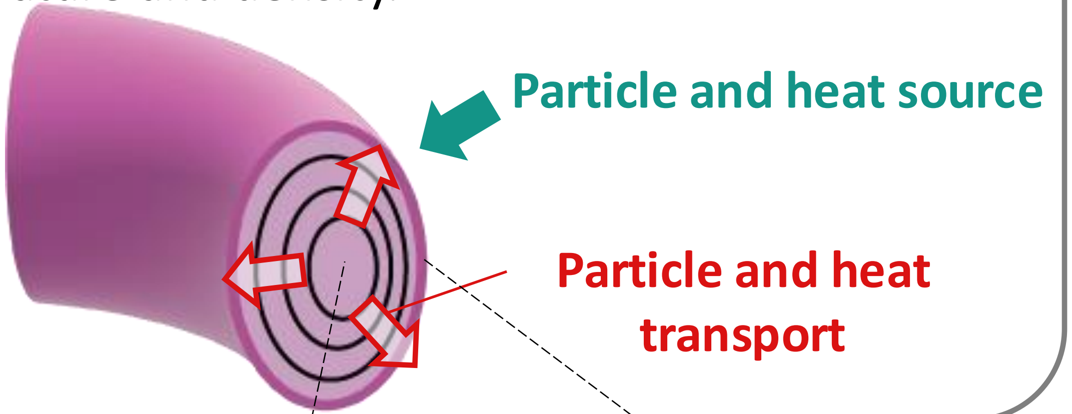
Transport in fusion plasmas

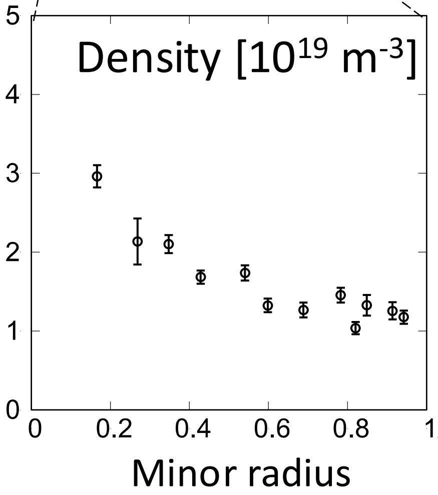
- Fusion power depends on the temperature and density of plasmas: $P_{\mathrm{NF}} \propto n^2\langle\sigma v\rangle$

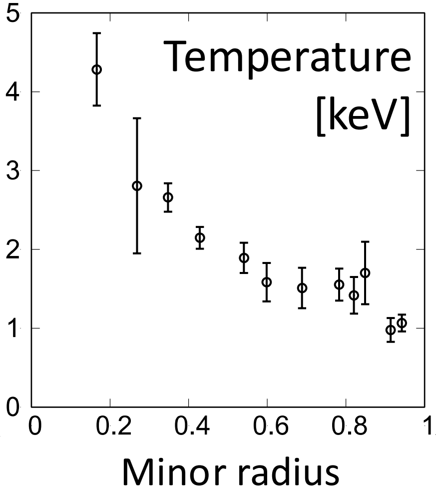
ITER



The balance between transport and sources determines the temperature and density.







✓ Prediction of heat and particle fluxes

➔ Prediction of performance in future devices

✓ Turbulent transport is dominant in tokamak plasmas.

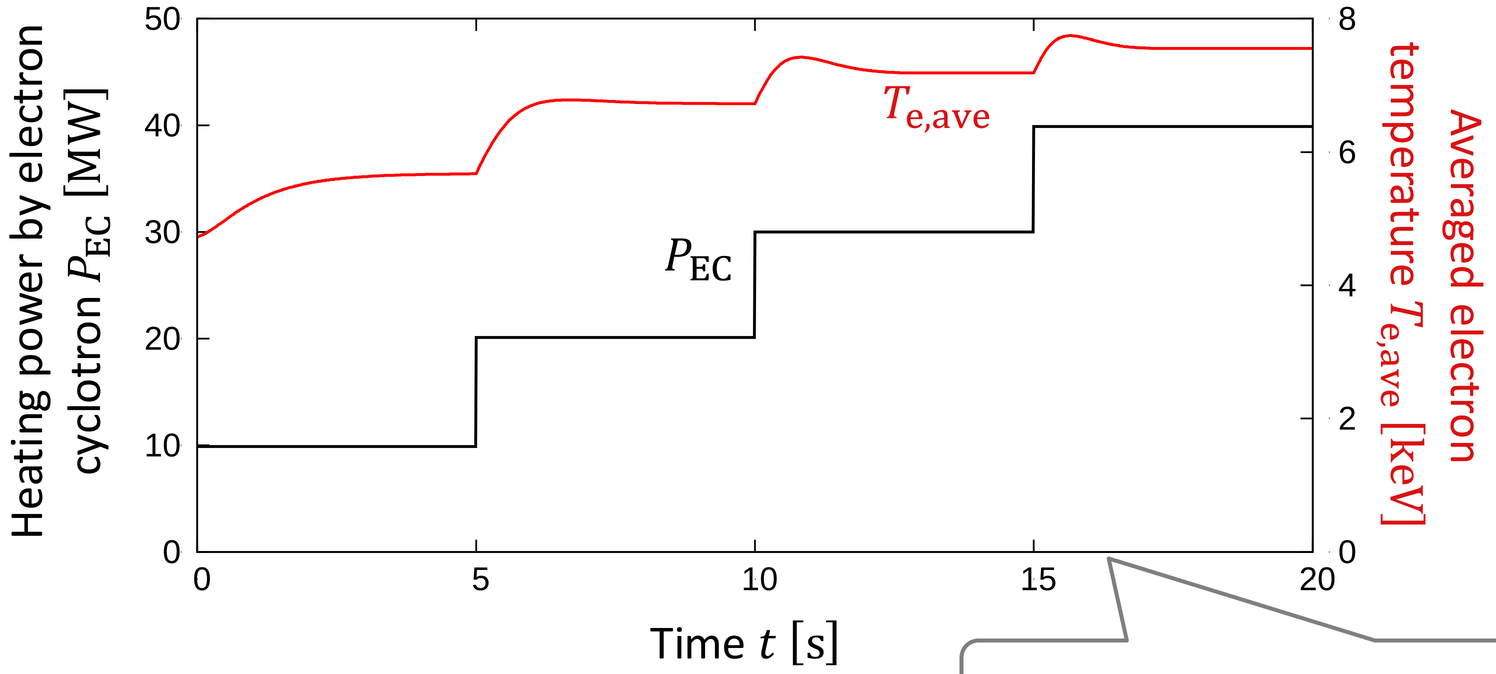
➔ Turbulent transport models are the key to the prediction performance.

- Integrated simulations of fusion plasmas
- Turbulent transport simulations
- Development of transport models with machine learning
- Application of machine-learning-based transport models

Integrated simulations

- Fusion plasmas are governed by a wide variety of physical phenomena.
- Integrated codes include several models that express each physical phenomenon.
- Integrated codes are used to predict plasma performance.

Prediction of temperature with a change in the heating power



Models for heating and transport are used.

Roles of transport models

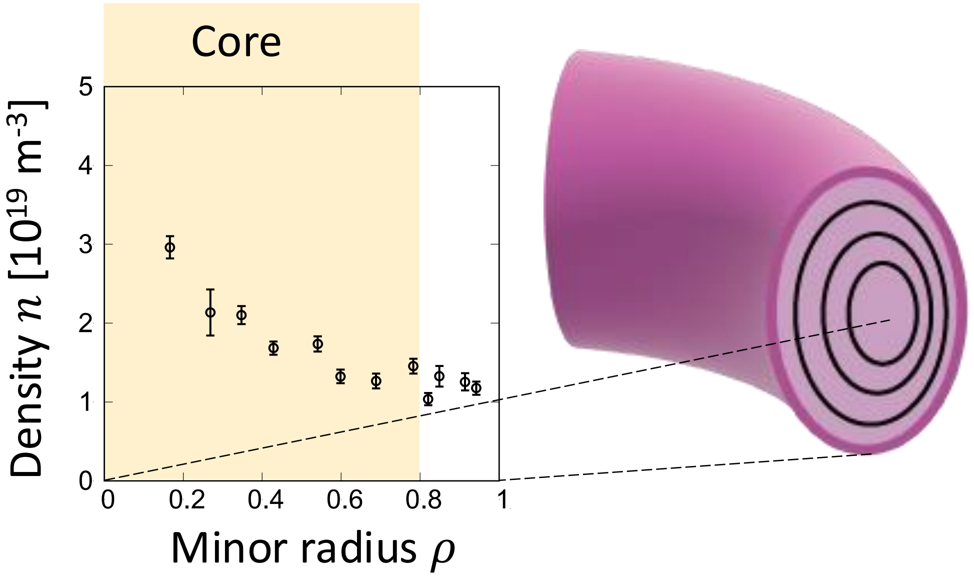
- Core plasma: $\rho \lesssim 0.8$
- The temperature and density in the core plasma are predicted with the transport equations.

Transport eq. for the energy

$$\underbrace{\frac{\partial}{\partial t}\left(\frac{3}{2}n_a T_a\right)}_{\text{Time evolution}} = \underbrace{-\frac{\partial \rho}{\partial V}\frac{\partial}{\partial \rho}\left(\frac{\partial V}{\partial \rho}Q_a\right)}_{\text{Transport}} + \underbrace{S_{\mathrm{e},a}}_{\text{Source}}$$

Transport eq. for the density

$$\frac{\partial n_a}{\partial t} = -\frac{\partial \rho}{\partial V}\frac{\partial}{\partial \rho}\left(\frac{\partial V}{\partial \rho}\Gamma_a\right) + S_{\mathrm{p},a}$$



✓ Evaluation of the heat flux Q and particle flux Γ is the role of transport models.

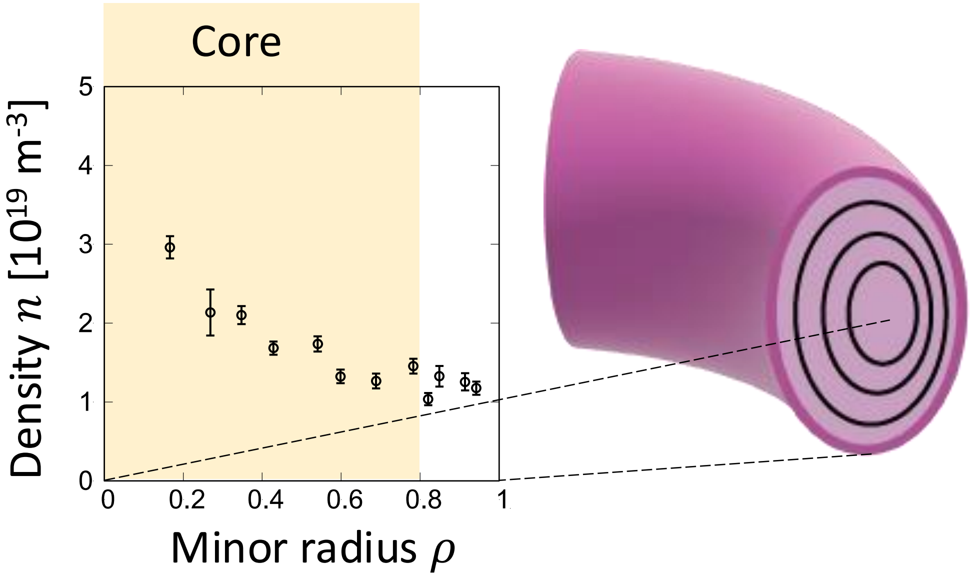
Other models required to solve transport equations

Transport eq. for the energy

$$\frac{\partial}{\partial t}\left(\frac{3}{2}n_a T_a\right) = -\frac{\partial \rho}{\partial V}\frac{\partial}{\partial \rho}\left(\frac{\partial V}{\partial \rho}Q_a\right) + S_{\mathrm{e},a}$$

Transport eq. for the density

$$\frac{\partial n_a}{\partial t} = -\frac{\partial \rho}{\partial V}\frac{\partial}{\partial \rho}\left(\frac{\partial V}{\partial \rho}\Gamma_a\right) + S_{\mathrm{p},a}$$



- Each parameter like density n is averaged over the flux surface.
  - → Models for the magnetic equilibrium
- Energy and particle sources S
  - → Models for RF heating, NBI, pellets and alpha particles
- Boundary condition
  - → Models for the edge region

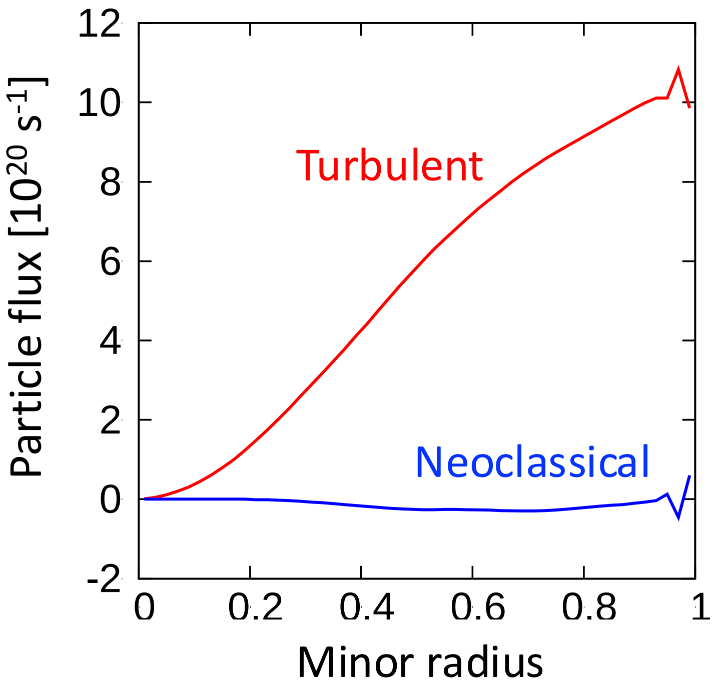
✓ The transport equations need to be solved consistently with several models.

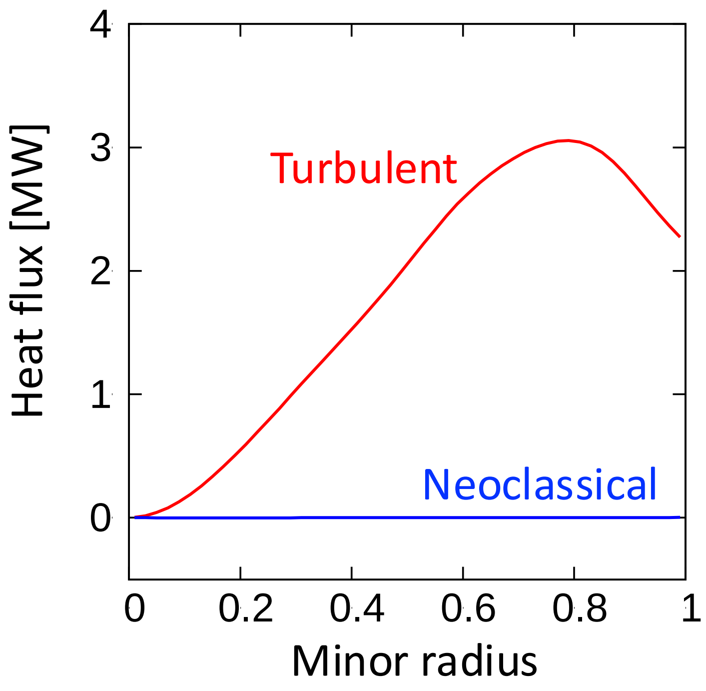
- Integrated simulations of fusion plasmas
- Turbulent transport simulations
- Development of transport models with machine learning
- Application of machine-learning-based transport models

Turbulent transport is dominant in tokamak plasmas.

<u>Transport in fusion plasmas</u>

- Neoclassical:
  - ✓ Collisions between particles and distortion of the particle orbits
- Turbulent:
  - ✓ Electrostatic and electromagnetic fluctuations





The first principle of plasma turbulence: Gyrokinetic theory (1/2)

- Drift-wave turbulence
  - ✓ is driven by temperature and density gradient.
  - ✓ has a lower frequency than the gyro-frequency.
  - ✓ has a perpendicular wavelength that is shorter than parallel one.

- Boltzmann equation : $\left(\frac{\partial}{\partial t} + \dot{\boldsymbol{x}} \cdot \nabla + \dot{\boldsymbol{v}} \cdot \frac{\partial}{\partial \boldsymbol{v}}\right) \underline{\mathcal{F}_a(\boldsymbol{x}, \boldsymbol{v}, t)} = C_a(\mathcal{F}_a)$

  Distribution function in the 6D phase space at time t

- The distribution function is divided into the background and perturbed parts : $\mathcal{F}_a(\boldsymbol{x}, \boldsymbol{v}, t) = F_a(\boldsymbol{x}, \boldsymbol{v}) + f_a(\boldsymbol{x}, \boldsymbol{v}, t)$

- The electrostatic potential and the magnetic field are also divided into the background and perturbed parts

- The perturbed distribution function satisfies: $\left[\frac{\partial}{\partial t} + \boldsymbol{v} \cdot \nabla + \frac{e_a}{m_a}\left(\boldsymbol{E} + \tilde{\boldsymbol{E}} + \frac{1}{c}\boldsymbol{v} \times \boldsymbol{B}\right) \cdot \frac{\partial}{\partial \boldsymbol{v}}\right] f_a = C_a^L(f_a) - \frac{e_a}{m_a}\tilde{\boldsymbol{E}} \cdot \frac{\partial}{\partial \boldsymbol{v}} F_a$ – (1)

✓ Gyrokinetic ordering : $\frac{\omega}{\Omega} \sim \frac{L_\perp}{L_\parallel} \sim \frac{f}{F} \sim \frac{e\tilde{\phi}}{T} \sim \frac{\tilde{B}}{B} \ll 1$

The first principle of plasma turbulence: Gyrokinetic theory (2/2)

- Introduction of the coordinate (ε, μ, ξ)
  - ✓ ε : kinetic energy + electric potential energy, μ : magnetic moment, ξ: gyrophase
  - ✓ The gyrokinetic motion is faster than the drift wave.
- The following gyrokinetic equation is obtained by taking a gyrophase average of Eq. (1):

$$\left(\frac{\partial}{\partial t} + v_\parallel \boldsymbol{b}\cdot\nabla + \mathrm{i}\boldsymbol{k}_\perp\cdot\boldsymbol{v}_{\mathrm{D}a}\right)\hat{g}_{a\boldsymbol{k}_\perp} - C_a$$

$$= \left(\frac{\partial}{\partial t} + \mathrm{i}\omega_{*T_a}\right)\frac{e_a\left(\hat{\phi}_{\boldsymbol{k}_\perp} - v_\parallel \hat{A}_{\parallel\boldsymbol{k}_\perp}\right)}{T_a} J_{0a}F_{\mathrm{M}a} + \underbrace{\frac{c}{B}\sum_{\boldsymbol{k}'_\perp+\boldsymbol{k}''_\perp=\boldsymbol{k}_\perp}\left[\boldsymbol{b}\cdot\left(\boldsymbol{k}'_\perp\times\boldsymbol{k}''_\perp\right)\right] J_{0a}\left(\hat{\phi}_{\boldsymbol{k}_\perp} - v_\parallel \hat{A}_{\parallel\boldsymbol{k}_\perp}\right)\hat{g}_{a\boldsymbol{k}_\perp}}_{\text{Nonlinear term}}$$

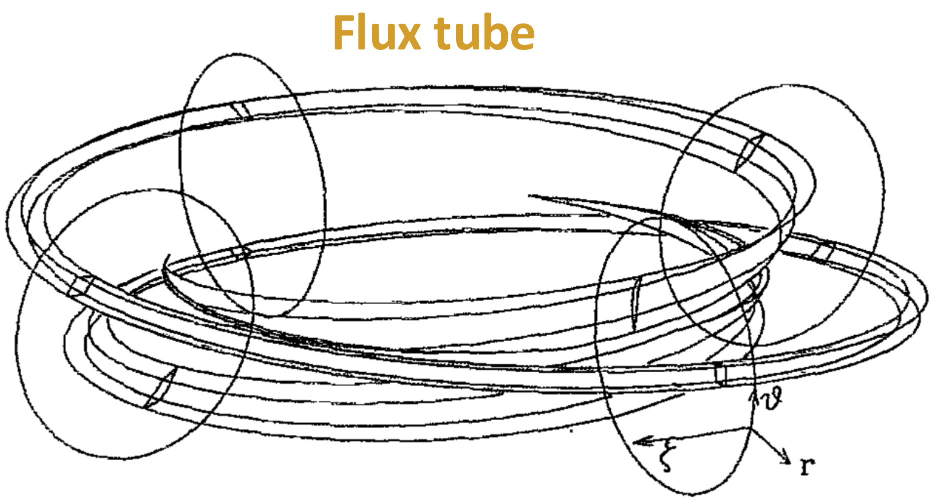
- The heat and particle fluxes are calculated by solving the gyrokinetic, Poisson and Ampere equations:

$$Q_a = \sum_{\boldsymbol{k}_\perp}\left\langle \int \mathrm{d}^3\boldsymbol{v}\mathrm{Re}\left[\frac{\mathrm{i}k_\theta(\hat{\phi}^*_{\boldsymbol{k}_\perp} - v_\parallel \hat{A}^*_{\parallel\boldsymbol{k}_\perp})}{B}\frac{m_a v^2}{2}J_{0a}\hat{g}_{a\boldsymbol{k}_\perp}\right]\right\rangle_{\mathrm{s}}$$

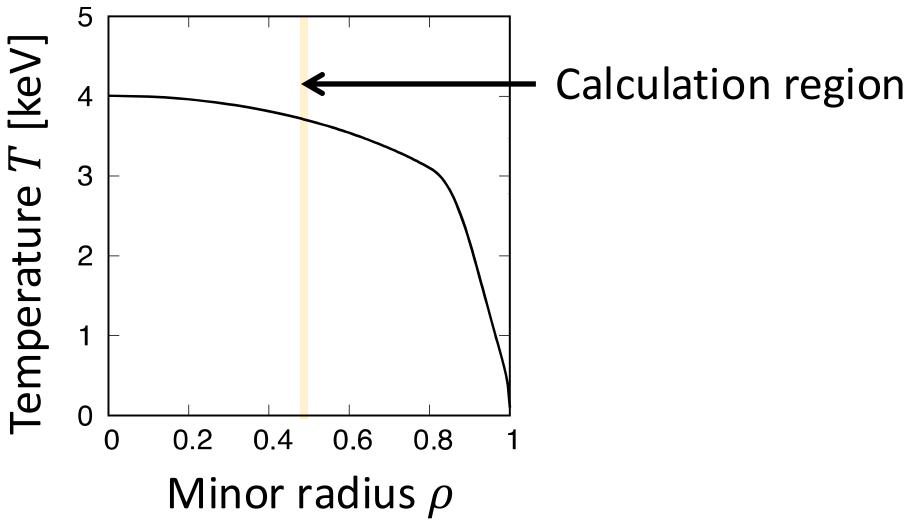
$$\Gamma_a = \sum_{\boldsymbol{k}_\perp}\left\langle \int \mathrm{d}^3\boldsymbol{v}\mathrm{Re}\left[\frac{\mathrm{i}k_\theta(\hat{\phi}^*_{\boldsymbol{k}_\perp} - v_\parallel \hat{A}^*_{\parallel\boldsymbol{k}_\perp})}{B}J_{0a}\hat{g}_{a\boldsymbol{k}_\perp}\right]\right\rangle_{\mathrm{s}}$$

Local gyrokinetic codes

- **Local limit**
  - ✓ The wavelength perpendicular to the magnetic field is comparable to the gyroradius: $L_\perp \sim \rho_\mathrm{s}$
  - ✓ The gyroradius is smaller than the machine size.
- **Local (flux tube) gyrokinetic codes**
  - ✓ The gyrokinetic equation is solved in a flux tube along a magnetic field. → Low computational cost
  - ✓ The calculations are not validated beyond the local limit.
  - ✓ Physical phenomena outside the flux tube are not considered.



[Beer PoP1995]



Can we use gyrokinetic codes as a transport model?

- Evaluation of the heat flux Q and particle flux Γ is the role of transport models.
- The transport equations are solved repeatedly (10^3~10^6 times) to predict the temperature and density.
- Q and Γ also need to be evaluated repeatedly.

Transport eq. for the energy

$$\frac{\partial}{\partial t}\left(\frac{3}{2}n_a T_a\right) = -\frac{\partial \rho}{\partial V}\frac{\partial}{\partial \rho}\left(\frac{\partial V}{\partial \rho}Q_a\right) + S_{\mathrm{e},a}$$

Transport eq. for the density

$$\frac{\partial n_a}{\partial t} = -\frac{\partial \rho}{\partial V}\frac{\partial}{\partial \rho}\left(\frac{\partial V}{\partial \rho}\Gamma_a\right) + S_{\mathrm{p},a}$$

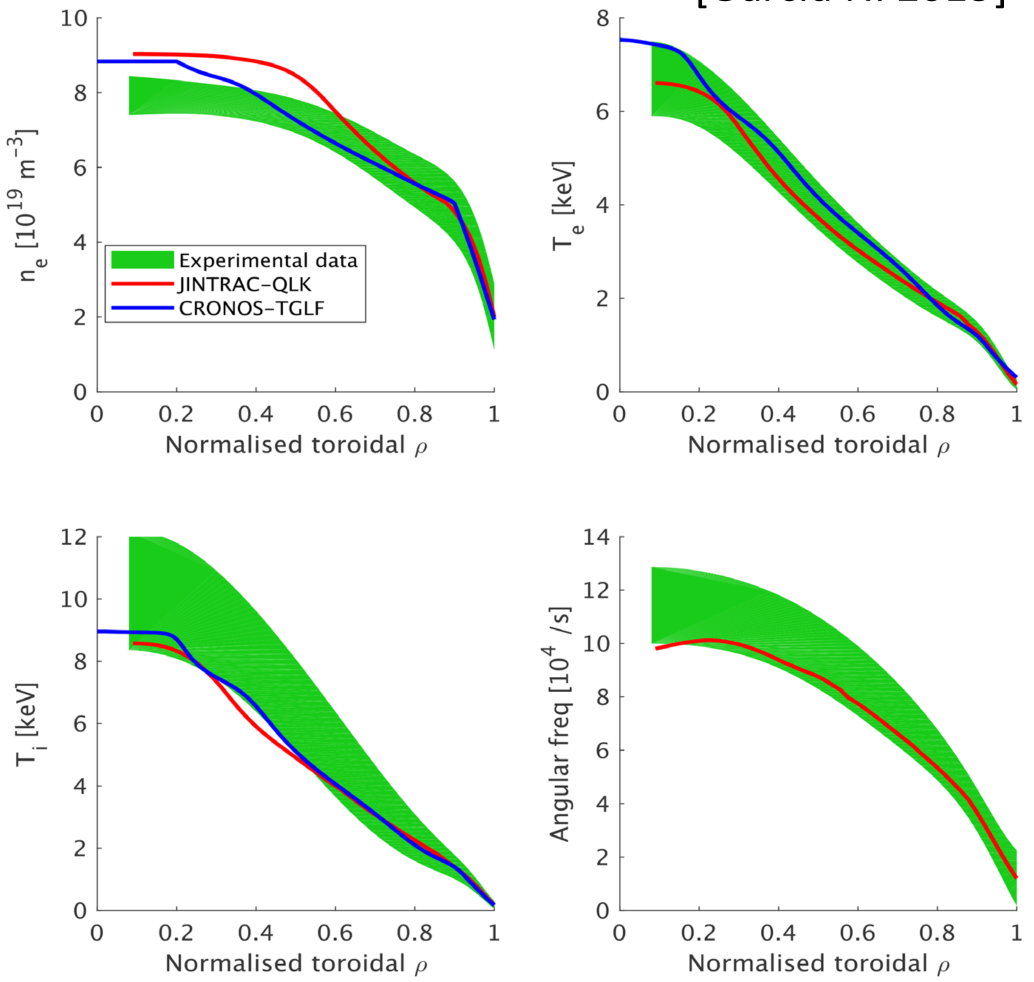
Evaluation of the heat flux Q and particle flux Γ with local gyrokinetic codes:

- ✓ It takes several days on a supercomputer.
- ✓ It is unrealistic to use them as a transport model.

Reduced transport models

- Reduced transport models based on gyrokinetic codes are used as a transport model.
  - ✓ Reduction using the linear gyrokinetic equation
  - ✓ Adjustment to nonlinear gyrokinetic simulations
  - ✓ E.g.: TGLF, QuaLiKiz

[Garcia NF2019]



✓ Reasonable agreement with experiments

✓ It takes several hours or days to predict the temperature and density even with the “reduced” models.

✓ The transport models can be a computational bottleneck.

→ Acceleration with machine learning

Acceleration with machine learning

Time to estimate Q and Γ

Gyrokinetic codes

10^5 core hours: several days

Reduced models

$10^{-3\sim1}$ core hours: a few seconds or minutes

Machine learning models

10^{-6} core hours: 10^{-3} seconds

Simulation time
(Calculate fluxes 10^3~10^6 times):

Unrealistic

Several hours or days
(with parallel computation)

< A few hours

✓ 10^{11} times speedup with machine learning

- Integrated simulations of fusion plasmas
- Turbulent transport simulations
- Development of transport models with machine learning
- Application of machine-learning-based transport models

Progress in machine-learning-based transport models

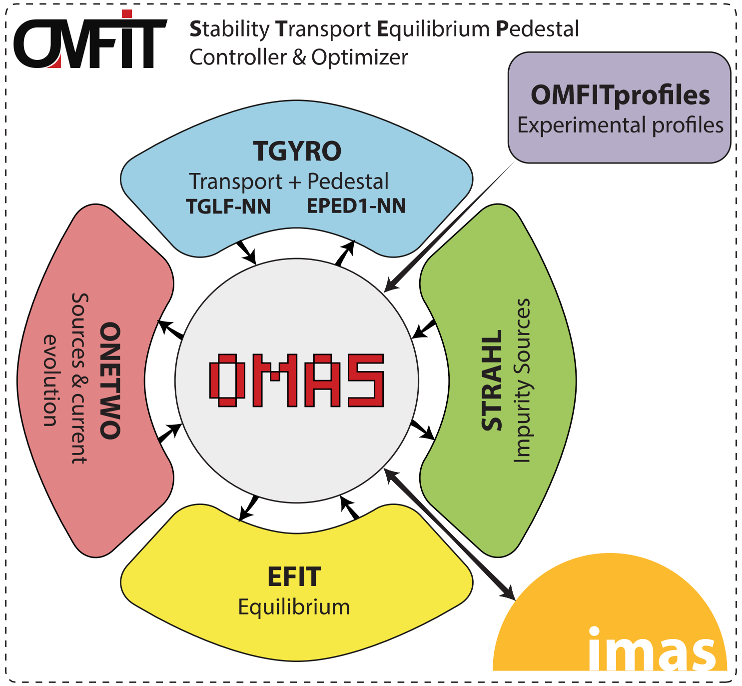
- Neural networks (NNs) are used as a machine-learning model.
- The NN-based transport models are implemented in the integrated simulations.

Development timeline

- The first model predicts heat fluxes estimated for the DIII-D plasmas [Meneghini NF2014].
- QLKNN learns QuaLiKiz calculations [Citrin NF2015].
- TGLF-NN learns TGLF calculations [Meneghini NF2017].
- DeKANIS learns gyrokinetic calculations and particle fluxes estimated for JT-60U plasmas [Narita NF2019].
- ...

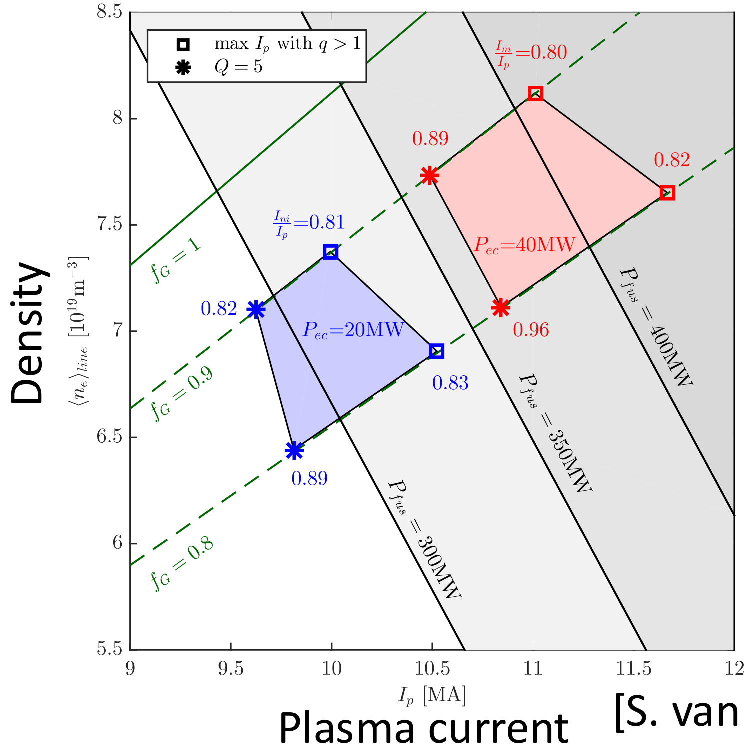
NN-based transport models are practically in use.

- The original model: TGLF
- TGLF-NN is implemented in the integrated-modeling framework OMFIT.



[O. Meneghini NF2021, B.C. Lyons PoP2023]

- The original model: QuaLiKiz
- QLK-NN is applied to optimizing of the ITER operation scenario.



[S. van Mulders NF2021]

✓ NN-based transport model is an option of the transport models used in integrated simulations.

Neural network

Neural network (NN) is

- one of the machine learning models.
- the backbone of the deep-learning algorithms.

A simple model

- y is calculated with weighted sum of the input x:

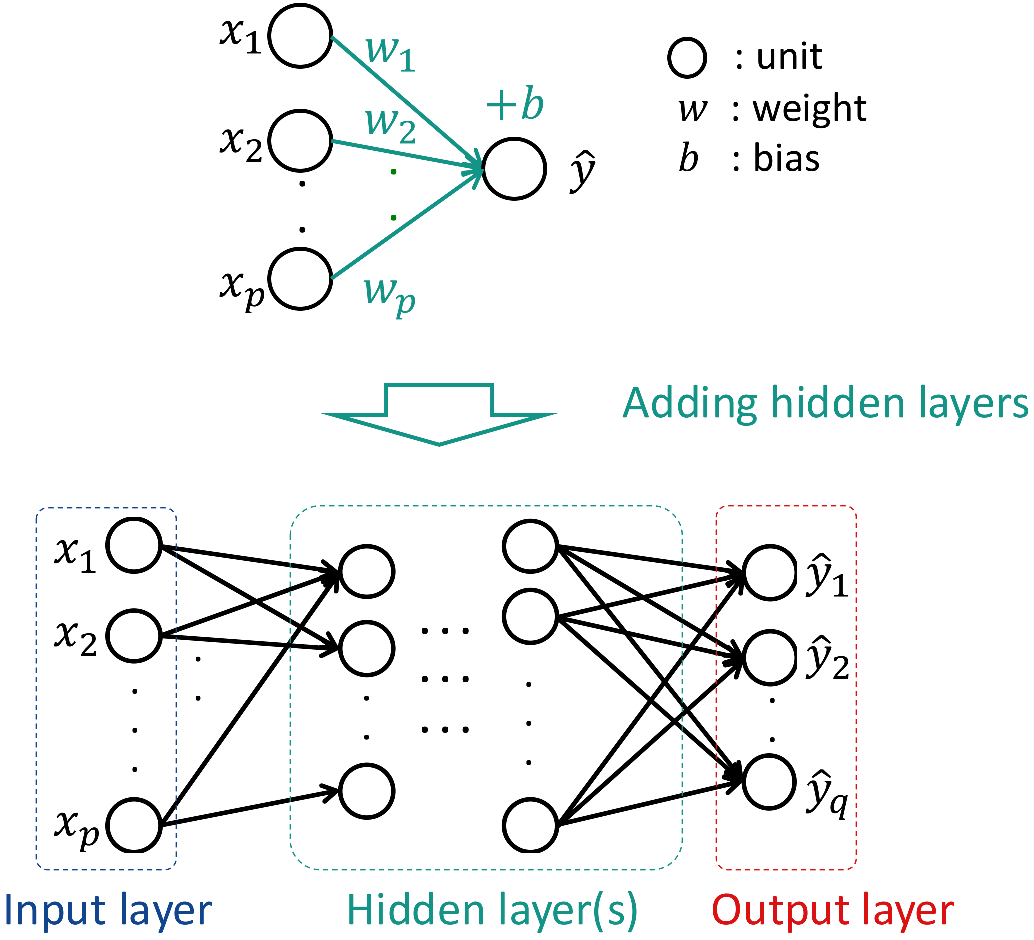
$$\hat{y} = \underbrace{f}_{\text{Activation function}}\big(\underbrace{w_1 \times x_1 + w_2 \times x_2 + \cdots + w_p \times x_p + b}_{\text{Liner model}}\big)$$

A model with hidden layers

- y is calculated by computing the weighted sum repeatedly.
- If one hidden layer is added,

$$\hat{y}_k = f^{[2]} \sum_{j=1}^{h} f^{[1]}\left(\sum_{i=1}^{p} x_i w_{i,j}^{[1]} + b_j^{[1]}\right) w_{j,k}^{[2]} + b_k^{[2]}.$$

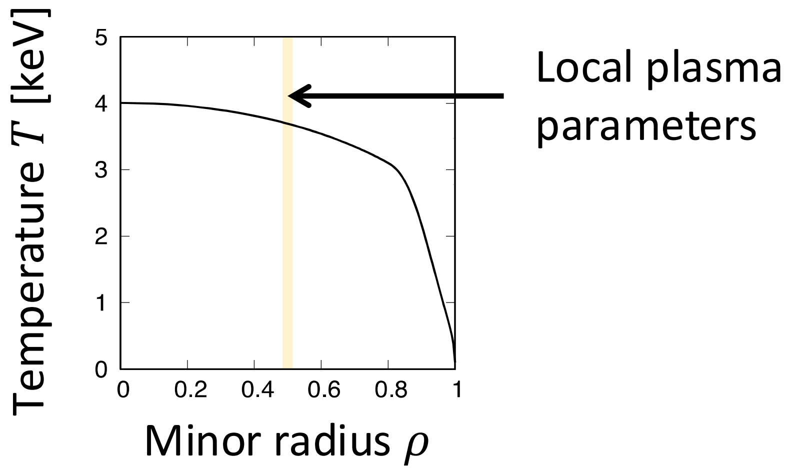
- Weights w and biases b are optimized by learning.



Emulation of reduced models

Input: plasma parameters on the flux surface in question

- Temperature gradient
- Density gradient
- Minor radius
- Temperature ratio
- …



Reduced model

Replacement

Neural-network model

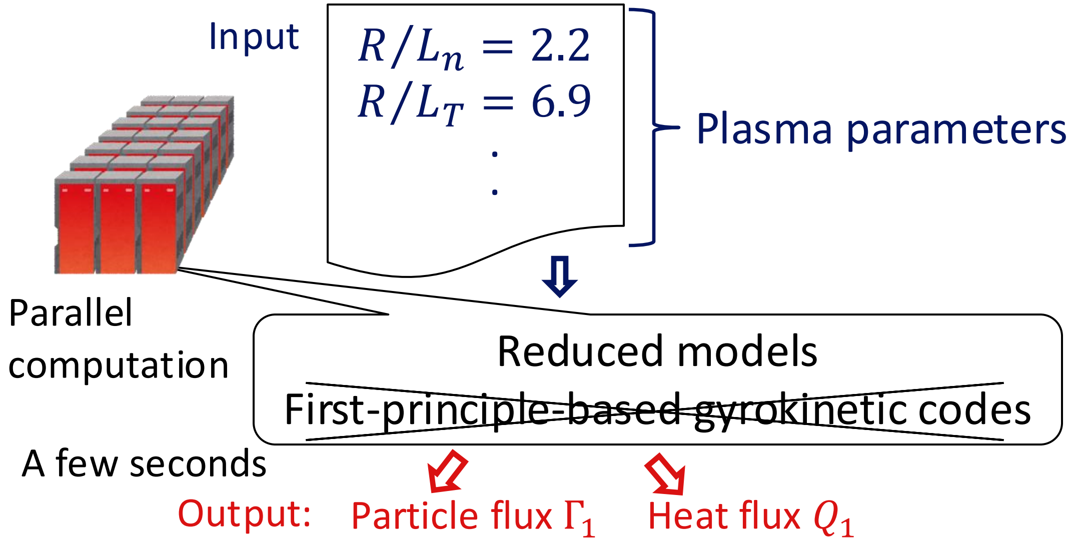
Output

- Heat flux Q
- Particle flux Γ

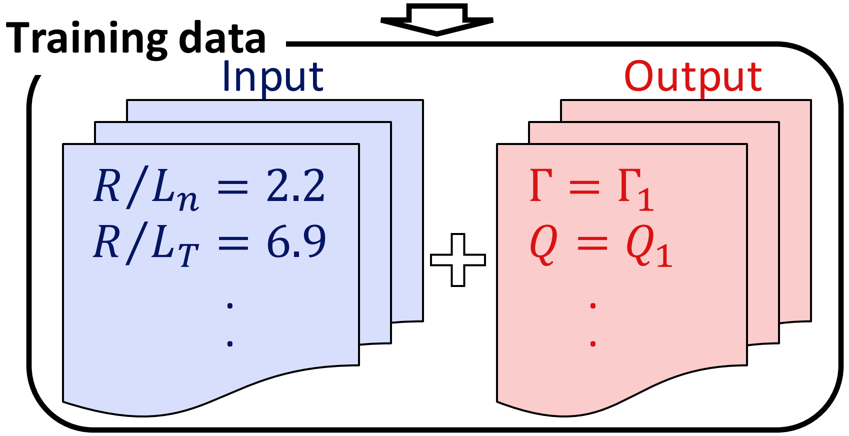
✓ Learning on the calculation results from reduced models: Supervised learning

Construction of NN-based transport models

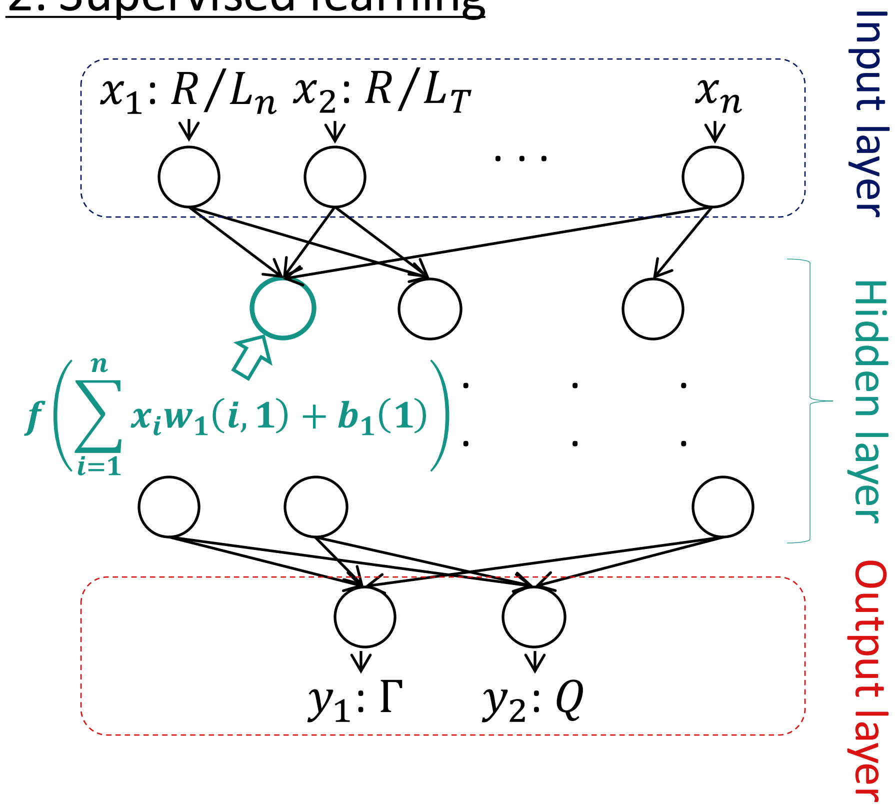
1. Training datasets created by existing models



✓ Multiple calculations in a wide range of plasma parameters



2. Supervised learning



- ✓ Weights w and biases b are optimized to reproduce the training datasets.
- ✓ Γ and Q are calculated in about 10^{-3} seconds, mimicking the reduced model.
  → Surrogate models

Programming language

- Construction of the NN models: Python
  - ✓ A wide variety of machine learning related packages
  - ✓ Slow calculation speed
- Integrated codes: Fortran
  - ✓ High calculation speed and high maintainability
    - → Numerical simulations in science and technology are often performed with Fortran programs.

✓ The NN models constructed with Python are converted into a Fortran program, and introduced to the integrated codes.

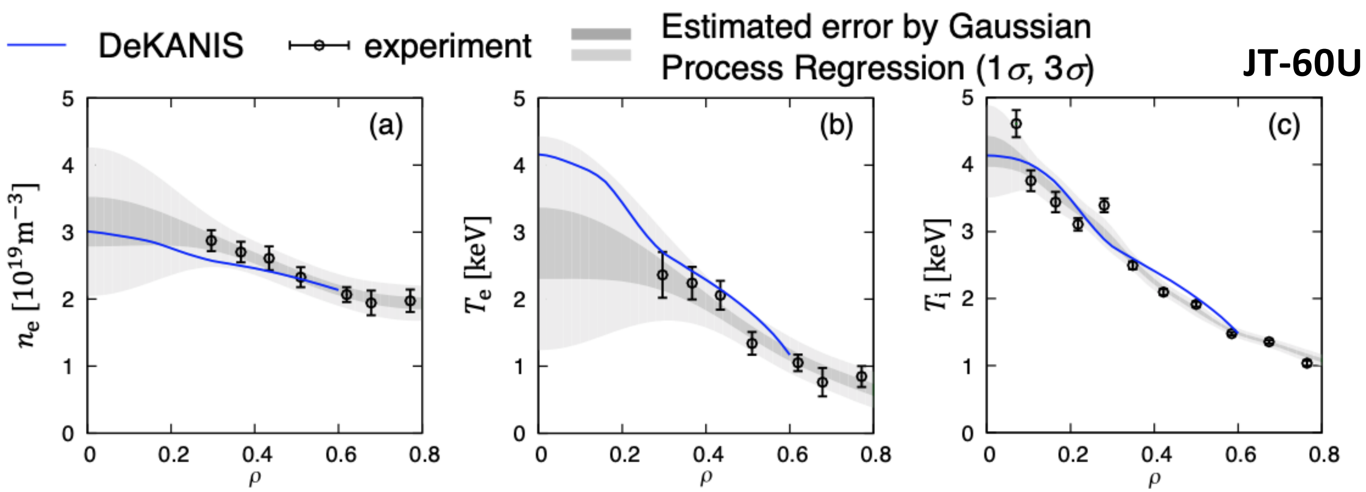
- Integrated simulations of fusion plasmas
- Turbulent transport simulations
- Development of transport models with machine learning
- Application of machine-learning-based transport models

NN-based transport model DeKANIS

Most of the NN-based transport models are built to mimic the existing models (surrogate models), but the DeKANIS project started with the quasilinear flux modeling.

DeKANIS (Detailing Kinetic fluxes with Artificial Neural networks for Insights into Simulations) [Narita NF2019, 2021, CPP2023, IAEA FEC2023]

- is founded on a combination of gyrokinetic calculations and experimental data.
- can detail transport processed related to density and temperature profile predictions.
- is recently expanded to include hydrogen isotope effects.



Other gyrokinetic calculation based NN model: GENE-NN [Citrin PoP2023]

Fluxes predicted by DeKANIS

- Particle and heat fluxes given by the linear gyrokinetic equation

$$\Gamma_a = \left\langle \int \mathrm{d}^3\boldsymbol{v}\mathrm{Re}\left[\underline{\left(\tilde{\phi}_{k_\theta} - v_\parallel \tilde{A}_{\parallel k_\theta}\right)^2} \frac{i k_\theta^2 J_0^2 F_{a,\mathrm{M}}}{B^2 R \mathcal{L}_a} \left(\frac{R}{L_{n_a}} + \left(\frac{m_a v^2}{2T_a} - \frac{3}{2} \right) \frac{R}{L_{T_a}} + \frac{e_a B R}{T_a k_\theta}\omega \right) \right] \right\rangle_\mathrm{s}$$

$$Q_a = \left\langle \int \mathrm{d}^3\boldsymbol{v}\mathrm{Re}\left[\underline{\left(\tilde{\phi}_{k_\theta} - v_\parallel \tilde{A}_{\parallel k_\theta}\right)^2} \frac{i k_\theta^2 J_0^2 F_{a,\mathrm{M}}}{B^2 R \mathcal{L}_a} \frac{m_a v^2}{2} \left(\frac{R}{L_{n_a}} + \left(\frac{m_a v^2}{2T_a} - \frac{3}{2} \right) \frac{R}{L_{T_a}} + \frac{e_a B R}{T_a k_\theta}\omega \right) \right] \right\rangle_\mathrm{s}$$

Simplified for electrons

Particle flux: $\bar{\Gamma}_\mathrm{e} = \bar{D}\left(\frac{R}{L_{n_\mathrm{e}}} + C_\mathrm{T}\frac{R}{L_{T_\mathrm{e}}} + C_\mathrm{P} \right)$

Heat flux: $\bar{Q}_\mathrm{e} = \bar{\chi}_\mathrm{e}\left(C_\mathrm{N}\frac{R}{L_{n_\mathrm{e}}} + \frac{R}{L_{T_\mathrm{e}}} + C_\mathrm{HP} \right)$

$\propto (\tilde{\phi} - v_\parallel \tilde{A}_\parallel)^2$
: The turbulent fluctuation amplitude

: diagonal : off-diagonal

- Ion heat flux:

$$\bar{Q}_\mathrm{i} = \frac{\bar{\chi}_\mathrm{i,eff}}{\bar{\chi}_\mathrm{e,eff}} \bar{\chi}_\mathrm{e,eff} \frac{R}{L_{T_\mathrm{i}}} \frac{n_\mathrm{i}}{n_\mathrm{e}} \frac{T_\mathrm{i}}{T_\mathrm{e}}$$

$\bar{\cdots}$: normalized parameter

✓ C_T, C_P, C_N and C_HP are not constant.
✓ $\bar{\Gamma}_\mathrm{e}$ and $\bar{Q}_\mathrm{e}$ are given by the linear combination of the nonlinear terms.

Structure of DeKANIS

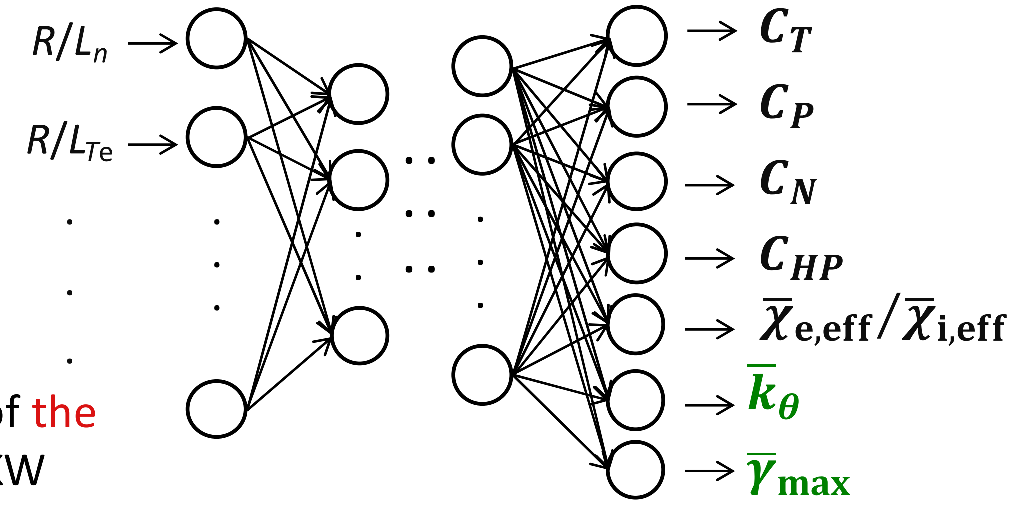
$$\bar{\Gamma}_{\mathrm{e}} = \bar{D}\left(\frac{R}{L_{n_{\mathrm{e}}}} + C_{\mathrm{T}}\frac{R}{L_{T_{\mathrm{e}}}} + C_{\mathrm{P}}\right)$$

$$\bar{Q}_{\mathrm{e}} = \bar{\chi}_{\mathrm{e}}\left(C_{\mathrm{N}}\frac{R}{L_{n_{\mathrm{e}}}} + \frac{R}{L_{T_{\mathrm{e}}}} + C_{\mathrm{HP}}\right)$$

$$\bar{Q}_{\mathrm{i}} = \frac{\bar{\chi}_{\mathrm{i,eff}}}{\bar{\chi}_{\mathrm{e,eff}}}\bar{\chi}_{\mathrm{e,eff}}\frac{R}{L_{T_{\mathrm{i}}}}\frac{n_{\mathrm{i}}}{n_{\mathrm{e}}}\frac{T_{\mathrm{i}}}{T_{\mathrm{e}}}$$

- ✓ 6 coefficients (C_{T}, C_{P}, C_{N}, C_{HP}, $\bar{\chi}_{\mathrm{e,eff}}/\bar{\chi}_{\mathrm{i,eff}}$, $\bar{D}$) are estimated with a NN and $\bar{D}_{\mathrm{model}}$.
- ✓ $\bar{\chi}_{\mathrm{e}}$ is calculated not to break a restriction.

NN model



Training data:
linear calculations of the gyrokinetic code GKW

+

Saturation model:

$$\bar{D}_{\mathrm{model}} = f(\bar{k}_\theta, \bar{\gamma}_{\mathrm{max}}, \ldots)$$

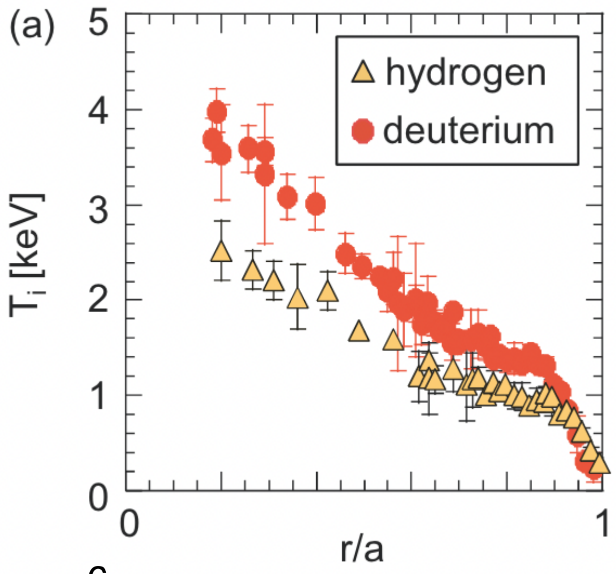
- ✓ The linear calculations are ~10^3 times faster than the nonlinear ones.
- ✓ Based on the experimental data

Hydrogen isotope effects

- Most fusion plasma experiments have been performed with hydrogen or deuterium.
- The effective fusion reaction is expected with deuterium and tritium (DT) plasmas.

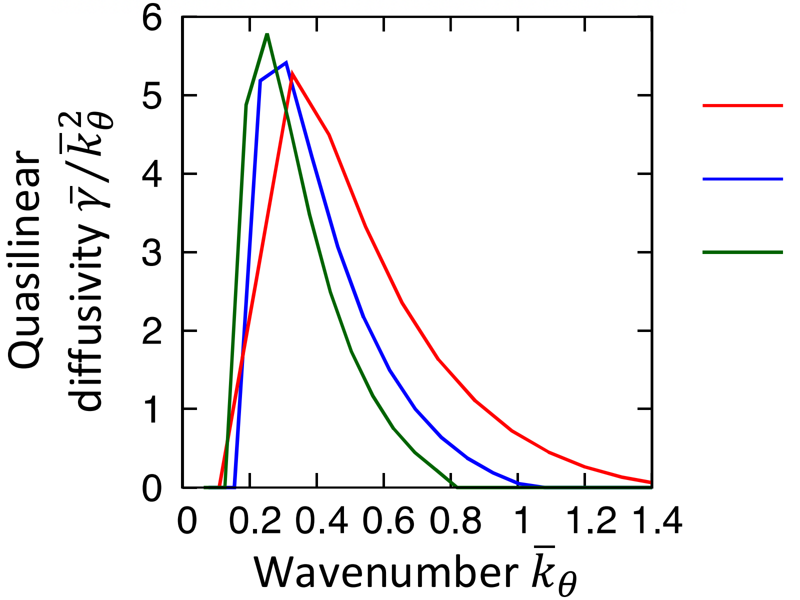
➔ Isotope effects are crucial in predicting DT plasma performance with our knowledge.

- Positive isotope effects are observed in several devices.
  - ✓ The higher temperature and density with the heavier isotope
- The gyrokinetic codes can capture the experimental trend [Nakata PRL2017, Garcia PPCF2022].
  - ✓ The lower flux (~ diffusivity) with the heavier isotope
  - ✓ DeKANIS is able to emulate the trend using the NN model trained on the gyrokinetic calculations.



JT-60U

[Urano NF2013]



Training datasets of DeKANIS

Input parameters:
R/L_n, R/L_{T_e}, R/L_{T_i}, $n_\mathrm{i}/n_\mathrm{e}$, $T_\mathrm{e}/T_\mathrm{i}$, β, $\bar{\nu}_\mathrm{ee}$, q, s, ε, κ, δ, m_i

Plasma parameters used for inputs (13 columns)

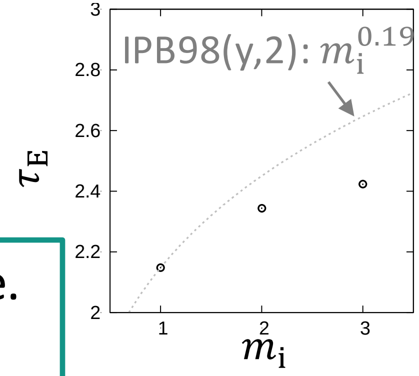
Transport coefficients used for outputs (7 columns)

R/L_{n_e}	R/L_{T_e}	$\cdots$	m_i	C_T	C_P	C_N	C_HP	$\frac{\chi_\mathrm{e,eff}}{\chi_\mathrm{i,eff}}$	$\bar{k}_\theta$	$\bar{\gamma}_\mathrm{max}$
Experimental values (Group I: 306 rows)				**Gyrokinetic code GKW**						
Parameter scan based on group I (Group II: 24,051 rows)										

- ✓ Experimental values in group I are taken from JT-60U and JET D plasmas.
- ✓ Density gradient R/L_n, electron and ion temperature gradient $R/L_{T_{e,i}}$ and the ion mass number m_i change in the parameter scan in group II.

Integrated simulations including isotope effects

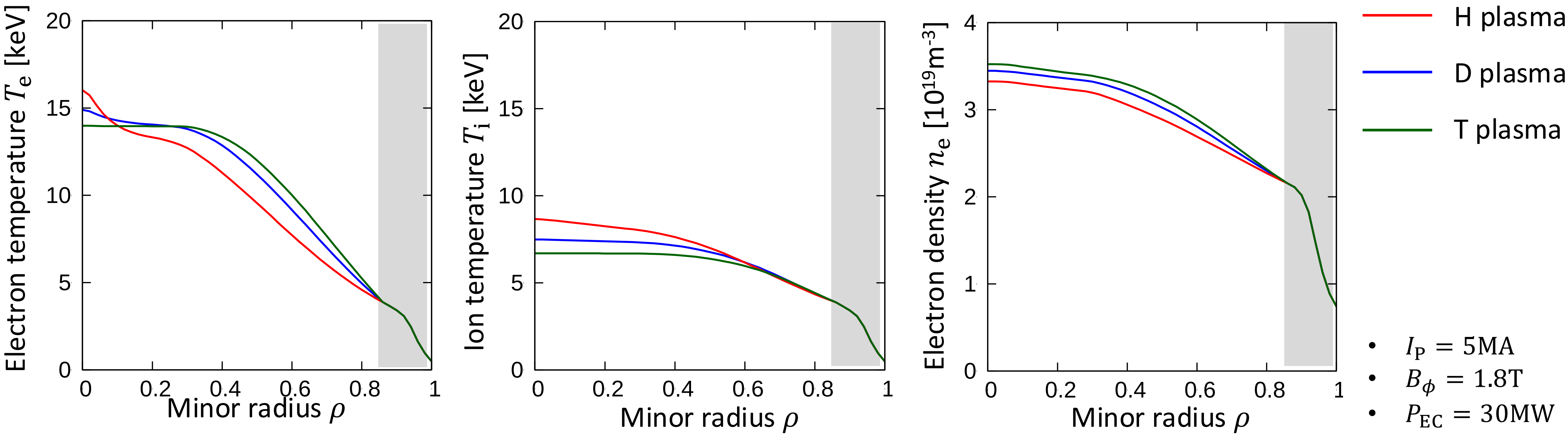
- DeKANIS is implemented in the integrated code TOPICS.
- Integrated simulations of ITER plasmas heated by electron cyclotron resonance



✓ The electron temperature T_e and density n_e are higher for the heavier ion mass case.

✓ T_i is higher for the H case due to the the collisional equipartition. [cf. Schneider NF2017]

τ_E increases with m_i.



- $I_P = 5$MA
- $B_\phi = 1.8$T
- $P_{EC} = 30$MW

Summary

- Integrated simulations of fusion plasmas
  - ✓ Integrated codes are used to predict plasma performance considering a wide variety of physical phenomena consistently.
- Turbulent transport simulations
  - ✓ Gyrokinetic theory: The first principle of plasma turbulence
  - ✓ Transport models be a computational bottleneck.
- Development of transport models with machine learning
  - ✓ The neural-network models have dramatically accelerated integrated simulations.
- Application of machine-learning-based transport models: Improving both accuracy and speed
  - ✓ The hydrogen isotope effect captured by the gyrokinetic codes has been incorporated into integrated simulations.