Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

SHRX: a neural surrogate of turbulent transport in tokamak plasma and its comparison with TORAX against experimental data

Beibarys Ermekuly

biba.ermek@gmail.com

Khoja Akhmet Yassawi International Kazakh-Turkish University

Research Article

Keywords: tokamak, turbulent transport, neural surrogate, QLKNN, TORAX, ITPA database

Posted Date: November 13th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11503430/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

SHRX: a neural surrogate of turbulent transport in tokamak plasma and its comparison with TORAX against experimental data

Beibarys Ermekuly[1]

[1] Department of Physics, Khoja Akhmet Yassawi International Kazakh-Turkish University, Turkistan, Kazakhstan

ORCID: https://orcid.org/0009-0005-0506-4120

Corresponding author: Beibarys Ermekuly, biba.ermek@gmail.com

Abstract

Integrated tokamak modelling needs fast surrogates of turbulent transport, but existing ones depend on JAX. We built SHRX, a neural network with 29,283 parameters that runs on NumPy alone, by distilling the QuaLiKiz neural network (QLKNN). A "bridge" converts the state of our one-and-a-half-dimensional simulator SINTEZ into the ten dimensionless QLKNN inputs. For stiff transport we introduce the error of the critical-gradient factor, δ_f, as a metric, and an ensemble of five networks gives an uncertainty estimate. We compare SINTEZ with SHRX against the TORAX code at three levels: identical local plasma states, an integrated ITER discharge with identical inputs, and 206 conventional and 41 spherical records of the H-mode database of the International Tokamak Physics Activity (ITPA). On machines held out of training, the median error of the heat diffusivity χ is 9% and the median δ_f is 2.3%; the ensemble lowers the χ error to 5.9–6.5%. The ITER fusion power is 492 MW with SINTEZ and SHRX, 478 MW with SINTEZ and QLKNN, and 532 MW with TORAX. On experimental records the two codes agree with each other ($r = 0.995$ in $\ln W$) better than with the measurements, and both are less accurate than the IPB98(y,2) regression ($\sigma_{\ln}$ of 0.55 and 0.56 against 0.21). On spherical tokamaks SHRX overestimates the stored energy by a median factor of 28. SHRX is a faster, JAX-free copy of QLKNN, not a better model than TORAX: most of the experimental error comes from inputs that both codes share.

Keywords: tokamak, turbulent transport, neural surrogate, QLKNN, TORAX, ITPA database

1 Introduction

Predicting the state of the plasma in a fusion device is the main task of integrated modelling. Heat and particle transport in the tokamak core is governed by turbulence driven by microinstabilities: the ion temperature gradient mode (ITG), the trapped electron mode (TEM) and the electron temperature gradient mode (ETG). The turbulent flux rises sharply once the gradient exceeds a critical value, so transport is "stiff": the real profile stays close to the critical gradient [1, 2]. What matters most in a transport model is therefore the position of the threshold, not the absolute value of the flux far above it.

The gyrokinetic quasilinear codes QuaLiKiz [3] and TGLF [4] reproduce the threshold well, but one call takes from milliseconds to seconds, and a single scenario needs hundreds of thousands of calls. The remedy is a neural surrogate: QLKNN [5] reproduces QuaLiKiz fluxes as a function of ten dimensionless variables, and TGLFNN [6, 7] does the same for TGLF. The TORAX code from DeepMind [8] couples QLKNN to a differentiable transport solver written in JAX; in this work it serves as the comparison benchmark.

The SINTEZ project is a one-and-a-half-dimensional tokamak simulator written in NumPy (SINTEZ; 12 devices, including the Kazakhstani KTM, the Kazakhstan Tokamak for Material testing). In its earlier version, transport was supplied by a "TORAX-style" surrogate (called SHRX v0 in this paper), a network trained on data generated from the project's own formulas. The drawback of that approach was measured (Section 4.2): for the same ITER regime SHRX v0 gave Q = 1.3, while TORAX gave Q = 10.6. The teacher of that network was a set of formulas, so it knows nothing about gyrokinetic physics.

The aim of this work is to build a transport model for SINTEZ that stands in for a true gyrokinetic surrogate and relies on NumPy alone, and to compare it with TORAX at three levels: (i) the local χ in identical plasma states; (ii) an integrated ITER discharge with identical inputs; (iii) real discharges of the ITPA H-mode confinement database, compared with experiment.

The contributions are as follows.

1. SHRX is a network of 29,283 parameters distilled from QLKNN (Sections 2.3 and 2.4). It works through a "bridge" that converts the SINTEZ state into the QLKNN variables, so one network serves any machine. The training data were collected over several rounds from states that the simulator itself visits (on-policy relabelling), with additional samples from the threshold region.
2. A quality measure suited to stiff transport, the critical-gradient factor error δ_f, Eq. (13). The flux error (and R^2) rated the first-round network as good while its ITER result disagreed; the threshold error captures that difference (Section 3.2).
3. An ensemble of five networks and the uncertainty estimate derived from it. QLKNN itself gives no error estimate.
4. A speed-up of the simulator that does not change the computed result by a single bit, together with the regression tests that protect it (Section 2.7).
5. A comparison with TORAX, including a publicly available experimental database, and an open report of the negative results: SHRX does not predict experimental data better than a regression scaling, and it is not suitable for spherical tokamaks or for KTM.

The paper is organised as follows. Section 2 describes the models and the comparison protocol, Section 3 the agreement of SHRX with QLKNN, and Section 4 the comparison with TORAX. Section 5 discusses limitations.

2 Models and methods

2.1 The SINTEZ simulator

SINTEZ is a time-dependent one-and-a-half-dimensional tokamak simulator written in NumPy. The electron and ion heat balance is solved with a fully implicit scheme in the normalised radius $\rho = \sqrt{\Phi/\Phi_{edge}}$ (Φ is the toroidal flux):

$$\frac{3}{2}\frac{\partial(n_s T_s)}{\partial t} = \frac{1}{V'}\frac{\partial}{\partial \rho}\left(\frac{V' n_s \chi_s}{a^2}\frac{\partial T_s}{\partial \rho}\right) + S_s - L_s, \qquad s \in \{e, i\}, \tag{1}$$

where $V' = dV/d\rho$, a is the minor radius, S_s are the heating sources (NBI, ICRH, ECRH, α-particles, ohmic power, electron–ion exchange) and L_s is the radiation loss. The model also includes particle balance, current diffusion, a periodically solved Grad–Shafranov equilibrium (it gives $q(\rho)$, l_i and β_p), the bootstrap current and neoclassical resistivity from the Sauter coefficients [9], and sawtooth and ELM events. In H-mode, χ_s is suppressed by a barrier in the pedestal region ($\rho > \rho_{ped}$). The pedestal height T_{ped} is set in one of two ways: by an EPED-like [10] "predictive" model, or by a prescribed value ("core-only" mode, Section 2.9).

The transport coefficients χ_e, χ_i come from interchangeable models. This work uses the default model anchored to the IPB98(y,2) scaling; SHRX v0 (the earlier network trained on the project's own formulas); SHRX, the network delivered here; and the comparison benchmark, the real QLKNN (`qlknn_ref`, run through JAX for evaluation only).

2.2 QLKNN and TORAX

QLKNN [5] is a family of networks trained on QuaLiKiz [3] calculations. The version shipped with TORAX is `qlknn_7_11_v1` (73,823 parameters); it returns the ion and electron heat fluxes of the ITG, TEM and ETG modes and the electron particle flux. Its inputs are ten dimensionless variables (Section 2.3) and its outputs are gyro-Bohm fluxes. TORAX combines them as follows:

$$q_i = q_i^{ITG} + q_i^{TEM}, \qquad q_e = q_e^{ITG} + q_e^{TEM} + f_{ETG}\, q_e^{ETG}, \qquad f_{ETG} = \frac{1}{3}. \tag{2}$$

TORAX 1.4.3 [8] was run for ITER with this set-up: the ITER CHEASE geometry, $I_p = 15$ MA, $B_0 = 5.3$ T, $P_{aux} = 50$ MW; the density profile, the pedestal ($T_{ped} = 4.29$ keV, $\rho = 0.93$) and Z_{eff} were taken from SINTEZ; a linear solver, a predictor–corrector step, Pereverzev–Corrigan extra diffusion, inner ($\rho < 0.2$) and outer ($\rho > 0.9$) patches, a sawtooth model and 25 cells. Ohmic power was not included in TORAX: in its example, `resistivity_multiplier=200` inflates the ohmic power to 113 MW, whereas in SINTEZ it is ≈ 1.7 MW. For the experimental database records the geometry is circular with a prescribed κ, and the q profile is fixed.

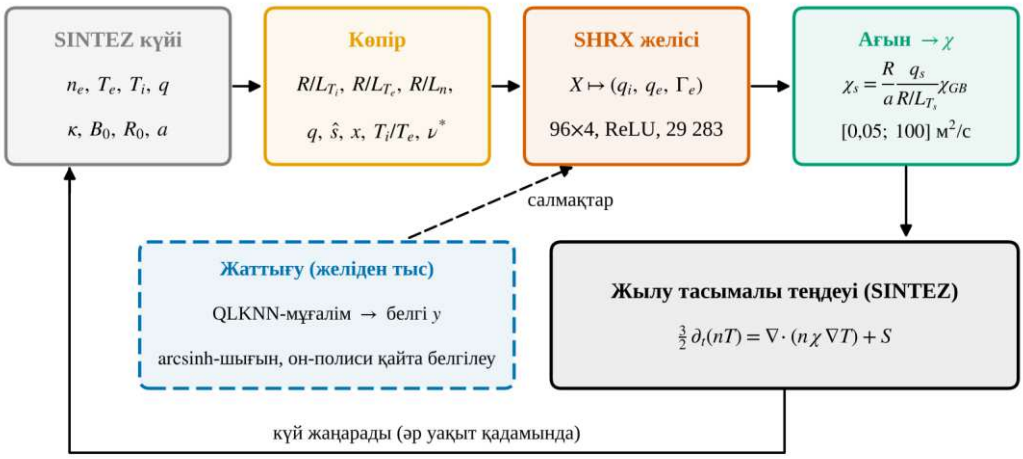


Fig. 1 SHRX inside SINTEZ. At each time step the simulator state is converted by the "bridge" into the ten dimensionless QLKNN variables (X); the network returns gyro-Bohm fluxes, which go back to the heat transport equation (1) as χ_e, χ_i. The dashed block is the teacher (QLKNN) and the training recipe, used during training only

2.3 The bridge: from the SINTEZ state to the QLKNN variables

SHRX uses the same ten variables, so the network is not tied to any particular machine. The bridge computes them from the SINTEZ state (n_e, T_e, T_i, q, geometry) with the formulas of TORAX itself. Logarithmic gradients are taken with respect to the half-width r in the mid-plane:

$$\frac{R}{L_y} = -\frac{R}{y}\frac{\partial y}{\partial r}, \qquad y \in \{T_i,\, T_e,\, n_e,\, n_i\}, \qquad \hat{s} = \frac{r}{q}\frac{dq}{dr}. \tag{3}$$

The effect of the non-circular cross-section is included through the relation between ρ and r; for elliptical surfaces

$$\rho = x_m \sqrt{K_0 + (1 - K_0)\, x_m^P}, \qquad x_m = \frac{r}{a}, \qquad K_0 = 0.74,\ P = 4.93, \tag{4}$$

where K_0 and P are fitted to the ITER CHEASE equilibrium supplied with TORAX (the largest error in ρ is 0.003). The radial variable of the network is rescaled to the aspect ratio at which QLKNN was trained ($\varepsilon_{NN} = 1/3$):

$$x = \frac{r/R}{\varepsilon_{NN}}. \tag{5}$$

The magnetic shear is taken with the Shafranov-shift correction (the van Mulders correction, as in TORAX):

$$\hat{s}_{eff} = \hat{s} - \frac{\alpha}{2}, \qquad \alpha = \frac{2\mu_0 q^2}{B_0^2} \sum_s n_s T_s \left(\frac{R}{L_{T_s}} + \frac{R}{L_{n_s}} \right). \tag{6}$$

The electron collisionality is defined, following Wesson, through the bounce time of a trapped particle (the period of one revolution along a banana orbit): $\nu^* = Z_{eff}\, \tau_b / \tau_e$, where

$$\tau_e = \frac{12\pi^{3/2} \varepsilon_0^2 \sqrt{m_e/2}\ T_e^{3/2}}{n_e e^4 \ln\Lambda}, \qquad \tau_b = \frac{qR}{\varepsilon^{3/2}} \sqrt{\frac{m_e}{T_e}}, \qquad \ln\Lambda = 31.3 - \frac{1}{2}\ln n_e + \ln T_e^{[eV]}, \tag{7}$$

and the network receives $\log_{10}\nu^*$. The remaining two inputs are T_i/T_e and n_i/n_e. In addition, inside $q < 1$ (the sawtooth region) q and the shear are corrected by hand, as in TORAX. The relation between the network output and χ is given by gyro-Bohm normalisation [11]:

$$\chi_{GB} = \frac{\sqrt{A\, m_u}}{(eB_0)^2}\frac{T_i^{3/2}}{a}, \qquad \chi_s = \frac{R}{a}\,\frac{q_s}{R/L_{T_s}}\,\chi_{GB}, \qquad \chi_s \in [0.05;\ 100]\ \mathrm{m}^2/\mathrm{s}, \tag{8}$$

where A is the mean ion mass (amu) and m_u is the atomic mass unit. The clipping bounds are the same as in TORAX. The bridge was checked against the inputs of TORAX itself (Section 3): T_i/T_e and q agree exactly, and the median absolute error of the other variables (each in its own units) does not exceed 0.027.

2.4 The SHRX network and its training

SHRX is a ReLU network with four hidden layers of 96 neurons each, $\mathbb{R}^{10} \to \mathbb{R}^3$: $X \mapsto (q_i, q_e, \Gamma_e)$, with 29,283 parameters in total (2.5 times fewer than the 73,823 of QLKNN). The inputs are clipped to the box on which QLKNN was trained ($X_{lo} \le X \le X_{hi}$) and standardised. The outputs go through an inverse-hyperbolic-sine compression, so that weak fluxes (near the threshold) and strong fluxes count equally in the loss:

$$\hat{y} = f_\theta\left(\frac{clip(X; X_{lo}, X_{hi}) - \mu}{\sigma}\right), \qquad y_k = s_k \sinh\hat{y}_k, \qquad \tilde{y}_k = arcsinh\frac{y_k}{s_k}, \qquad s_k = \frac{1}{4}\, med\,\{|y_k| : |y_k| > 10^{-3}\}. \tag{9}$$

The loss is a weighted mean squared error:

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N} w_i \parallel \hat{y}_i - \tilde{y}_i \parallel^2, \tag{10}$$

where the weight w_i depends on the type of data (Table 1). The optimiser is Adam [12] with an initial step of $2 \cdot 10^{-3}$ and cosine decay, a batch size of 4096 and 100 epochs; the weights are initialised by the He method [13].

The training data are (X, y) pairs labelled by the QLKNN "teacher" (passing knowledge from a large model to a small one is called distillation [14]). States taken from the simulator alone proved insufficient: the first-round network is good on the states it visited, but in an integrated run it changes its own states and leaves the training region. The data were therefore collected over several rounds with on-policy relabelling (similar to DAgger [15]), Table 1: in each round the previous network (or the real QLKNN) drives the simulator, QLKNN labels the visited states, and the network is retrained. After that, for stiff transport (the δ_f metric, Section 2.5), extra samples were drawn from the critical-gradient region.

Table 1 Composition of the SHRX training data. "Samples" is the cumulative total up to that round

Round	Data added	Samples	Result (ITER, P_{fus})
1	12 machines × 4 transport models (SINTEZ default, CGM, Bohm/gyro-Bohm, v0), 100 s each	1,648,000	267 MW ($Q = 5.3$): a clear deviation even on the states the network itself visited
2	+ states visited when the real QLKNN (and the round-1 SHRX) drove SINTEZ	2,272,000	466 MW (128×3 network)
3	+ states visited by the round-2 SHRX; 96×4 network, 100 epochs	2,584,000	492 MW (delivered)

The NSTX and DEMO machines do not take part in training (they are used for testing only), so the "held-out" results show how well the network generalises to machines with a new aspect ratio and size.

2.5 The threshold metric

In stiff transport the temperature gradient exceeds its critical value only slightly, so an error in the threshold position converts directly into an error in the gradient and, with it, in the stored energy. Consider the simple approximation

$$\frac{R}{L_T} = \left(\frac{R}{L_T}\right)_c + \left(\frac{q}{q_0}\right)^{1/\alpha}, \qquad T(\rho_0) = T_{ped}\exp\int_{\rho_0}^{\rho_{ped}} \frac{1}{R}\frac{R}{L_T}\frac{dr}{d\rho}\, d\rho. \tag{11}$$

When the flux term is small, a relative error δ in the critical gradient enters the increment of $\ln T$ along the profile as $\delta \cdot \Delta \ln T_c$ ($\Delta \ln T_c$ is the contribution of the critical term). We therefore measure the threshold directly: multiply both temperature gradients in a given state by a common factor f and find the smallest f at which the flux first exceeds $q_{on} = 1$ gyro-Bohm unit (bisection, $f \in [0.3;\ 3]$):

$$\begin{aligned} f_c(X) &= \min\{f:\ q_i(X_f) + q_e(X_f) > q_{on}\}, \\ X_f &= \left(f\,\frac{R}{L_{T_i}},\, f\,\frac{R}{L_{T_e}},\, \text{others unchanged}\right), \end{aligned} \tag{12}$$

$$\delta_f = \frac{f_c^{NN}}{f_c^{QLKNN}} - 1. \tag{13}$$

δ_f is computed only among the states for which the threshold is bracketed from both sides (that is, f_c can be found).

2.6 The ensemble and uncertainty

Following the deep-ensemble approach [16], $K = 5$ networks are trained separately, each with different initial weights and a random 85% sample of the training rows. The ensemble-mean flux and the relative spread of the members in χ are

$$\bar{q} = \frac{1}{K}\sum_{k=1}^{K} q^{(k)}, \qquad s = \frac{\sqrt{\frac{1}{K}\sum_k (\chi^{(k)} - \bar{\chi})^2}}{\bar{\chi}}, \qquad u = \mathbb{1}[s > \tau] \wedge \mathbb{1}[\text{turbulence on}]. \tag{14}$$

Here "on" means that χ is above the floor in at least one member. The threshold τ is chosen on one held-out machine (NSTX) and tested on the other (DEMO), and vice versa. How well the spread ranks the error is assessed with two measures: the area under the ROC curve [17] for the event "error above 25%", and the Spearman rank correlation:

$$AUROC = P(s_+ > s_-) + \frac{1}{2}P(s_+ = s_-), \qquad \rho_S = corr(rank\, s,\ rank\, e), \qquad e = \frac{|\chi^{NN} - \chi^{QLKNN}|}{\chi^{QLKNN}}, \tag{15}$$

where s_+ (s_-) is the spread at points whose error is larger (smaller) than 25%. Only the epistemic spread between the ensemble members is used in this work. It estimates how far SHRX deviates in regions that its training data do not cover; it does not estimate the training error of QLKNN.

2.7 Speed-up of the simulator and its accuracy

Profiling showed that most of the time goes to rebuilding the Grad–Shafranov operator at every solve, its LU factorisation, Python loops in the coefficient calculation, and tridiagonal systems. The speed-up consists of these changes: a compact (COO) operator structure with a cached factorisation, a cached grid, vectorised trapped-particle fractions in the bootstrap current and neoclassical resistivity, a vectorised slowing-down integral of fast ions, a vectorised residual of the two-point SOL model, and a tridiagonal solve with LAPACK `dgtsv`.

Any change in a simulator with stiff transport is delicate: machine rounding ($\varepsilon \approx 2.2 \cdot 10^{-16}$) is itself a small perturbation, and the system amplifies it:

$$\| \delta T(t) \| \approx \varepsilon\, e^{\lambda t} \quad \text{(until saturation)}, \qquad t_* = \frac{1}{\lambda} \ln \frac{\delta_*}{\varepsilon}, \tag{16}$$

where λ is a growth rate similar to a Lyapunov exponent [18] and δ_* is the saturation level (Section 4.3). The speed-up must therefore give a bit-for-bit identical result. The output of the code before the speed-up for 6 cases (ITER with three transport models, NSTX, DEMO and JET; 30 s of plasma time, 14 quantities at $t = 10$, 20 and 30 s) was fixed as a strict reference, and every modified primitive (vectorised convolutions, the integral, the cache) is also compared separately with the original version. The reference test is tied to the platform (the versions of NumPy, SciPy and Python, and the architecture): rounding differs on another platform, so the test is skipped there, while the platform-independent primitive tests remain.

This danger proved real. Vectorised integration with precomputed weights shifted quantities in the growth phase by up to 28% from the original result (swapping one file at a time showed that this change alone was the cause). The fix was to repeat exactly the same floating-point operations as the trapezoidal rule; after that, the deviation is exactly zero in all 6 cases.

2.8 Comparison protocol: TORAX and SINTEZ with SHRX

The two codes differ, so the comparison must rest on identical inputs. Three levels are used (Table 2).

- *A, local.* Compare QLKNN (through the bridge), SHRX and the own χ of TORAX on identical profiles taken from SINTEZ (one state in ρ) and from TORAX. This measures only the difference between networks.
- *B, integrated.* The same ITER regime: I_p, B_0, P_{aux}, the density profile, the pedestal and Z_{eff} are identical; the difference lies both in the transport model and in the code (solver, geometry, sawtooth, patches). There are three participants: TORAX (QLKNN), SINTEZ with QLKNN (the same network, a different code) and SINTEZ with SHRX (a different network, the same code). Their differences split into two causes: the network error $\epsilon_{net} = P^{SHRX}/P^{QLKNN} - 1$ and the code difference $\epsilon_{code} = P^{SINTEZ+QLKNN}/P^{TORAX} - 1$.
- *C, experimental.* Real discharges of the ITPA H-mode database (Section 2.9): SINTEZ with SHRX and TORAX are run with the same inputs and compared with the measured stored energy.

Table 2 Comparison levels: what is identical and what varies

Level	Identical	Varies	Measured
A (local)	profiles T_e, T_i, n_e, q, geometry	network (SHRX / QLKNN), TORAX's own code	χ_e, χ_i
B (ITER)	$I_p, B_0, P_{aux}, n_e(\rho), T_{ped}, Z_{eff}$	code (SINTEZ / TORAX), network	P_{fus}, Q, W_{th}, profiles
C (experiment)	machine, $I_p, B_0, \bar{n}_e, P_{loss}, Z_{eff}$, $n_e(\rho), T_{ped}$	code, network	W_{th}, T_{e0}, T_{i0}

In the figures, colour has a fixed meaning: QLKNN is blue, SHRX is orange, TORAX is green, the IPB98(y,2) regression is pink, the ensemble of five networks is purple, and the earlier SHRX version (v0) and a single network are grey; the lines $y = x$ and $\pm 30\%$ are black. Line style and marker also differ, so the figures stay readable in grey-scale print.

2.9 The experimental database and the measures

The experimental data are a publicly available sample of the ITPA H-mode database (DB5) [19] with 14,153 records. The selection: H-mode with or without ELMs, quasi-steady state ($|dW/dt| \leq 0.15\, P_{loss}$), all quantities needed for the simulation present, and at most 16 random records per device. By aspect ratio there are two sets: conventional ($R/a \geq 1.5$; 206 records, 14 devices) and spherical ($R/a < 1.5$; 41 records: NSTX, MAST, START). The input of a test run is the machine of the record, I_p, B_0, the isotope mass, the line-averaged density $\bar{n}_e$, Z_{eff} and the database's own P_{LTH} (the H_{98} in the database is computed with exactly this power). The H-mode of the simulator is imposed from outside: the question is what confinement the model predicts in the given H-mode, not whether the transition to H-mode takes place.

Power balance. P_{LTH} includes ohmic power, while SINTEZ adds its own ohmic power. The auxiliary power is therefore corrected in two passes:

$$P_{aux} = \max(P_{LTH} + P_{rad} - P_{ohm},\ 0.1\, P_{LTH}), \tag{17}$$

where P_{rad} and P_{ohm} are taken from the first pass. The density $\bar{n}_e$ is reached by gas puff feedback; the simulator can only raise the density, not lower it.

Validated run. A run is fit for comparison with experiment only if the line density deviates from the record by no more than 20%, the delivered power deviates from P_{LTH} by no more than 20%, and the plasma does not go into thermal runaway ($\beta_N < 3.5$ and bootstrap current fraction < 0.8; no H-mode in the database exceeded these values). The selection looks at the sanity of the run itself, not at the stored energy.

Measures. For the stored-energy prediction, $r_i = W_i^{pred}/W_i^{exp}$. The summary indicators are the median $\tilde{r}$, the standard deviation of $\ln r$, $\sigma_{\ln}$, and the share of records with $|r - 1| < 0.3$, f_{30}:

$$\sigma_{\ln} = \sqrt{\frac{1}{N}\sum_i \left(\ln r_i - \overline{\ln r}\right)^2}, \qquad f_{30} = \frac{1}{N}\sum_i \mathbb{1}\left[\,|r_i - 1| < 0.3\right]. \tag{18}$$

The IPB98(y,2) scaling [2] serves as the reference ceiling for these indicators. It is a regression fitted to exactly these data, so its result is that of a model that "knows" the data:

$$\tau_E^{IPB98} = 0.0562\, I_p^{0.93} B_0^{0.15} P^{-0.69} \bar{n}_{19}^{0.41} A^{0.19} R^{1.97} \varepsilon^{0.58} \kappa_a^{0.78}, \qquad W^{IPB98} = P\, \tau_E^{IPB98}, \tag{19}$$

where I_p is in MA, B_0 in T, P in MW, $\bar{n}_{19}$ in 10^{19} m^{-3}, R in m, A is the mean ion mass (amu) and κ_a is the elongation of the cross-section; hence $W^{IPB98}/W^{exp} = 1/H_{98}$.

Core-only mode. To separate the error of the pedestal prediction from the error of core transport, the temperature at the top of the pedestal is chosen so that the measured W_{th} is reproduced (a secant method in $\ln T_{ped}$; $|\ln W/W^{exp}| < 0.02$ is the stopping condition, with at most 7 passes):

$$T^*_{ped}: \quad \ln \frac{W^{pred}\left(T^*_{ped}\right)}{W^{exp}} = 0. \tag{20}$$

W now matches by construction, so only the shape of the core profile can be compared: the central temperatures T_{e0} and T_{i0} (given in the database in eV).

2.10 The second teacher: TGLFNN-STEP and its bridge

QLKNN was trained on data with a fixed aspect ratio $R/a = 3$, so a second teacher was considered for spherical geometry: the UKAEA TGLF-EM surrogate, trained on JETTO calculations of the STEP ramp-up phase (SPR45; $R/a \approx 1.8$, $\kappa \approx 2.3$) [4, 7, 20]. The weights are distributed as a dictionary of NumPy arrays: for each flux (electron heat, ion heat, ion particle) there is an ensemble of five networks, each with six linear layers (width 512) and ReLU, and the output is a normalised mean and a softplus variance. The input has 15 variables:

$$X_{TGLF} = \left(\frac{a}{L_{n_e}}, \frac{a}{L_{T_e}}, \frac{a}{L_{T_i}}, \frac{T_i}{T_e}, \frac{r}{a}, \frac{dR_0}{dr}, q, \hat{s},\ XNUE,\ \kappa,\ s_\kappa,\ \delta,\ s_\delta,\ \beta_e,\ Z_{eff}\right). \tag{21}$$

The transformation is in the TGLF "GYRO" units: lengths are normalised by a and fluxes by $n_e T_e c_s (\rho_s/a)^2$, where c_s and ρ_s are computed for deuterium and with B_{unit}:

$$\begin{aligned} B_{unit} &= B_0 \frac{\rho_{eff}}{r}\frac{d\rho_{eff}}{dr}, \qquad \rho_{eff} = \rho\, a\sqrt{\kappa_a}, \\ XNUE &= \frac{\nu_e}{c_s/a}, \qquad \nu_e = 2.91\cdot 10^{-6}\, n_e \ln\Lambda\, T_e^{-3/2}, \qquad \beta_e = \frac{2\mu_0 n_e T_e}{B_{unit}^2} \end{aligned} \tag{22}$$

(ν_e in s^{-1}, n_e in cm^{-3}, T_e in eV, $\ln\Lambda = 24 - \ln\left(\sqrt{n_e}/T_e\right)$). The gradient of the Shafranov shift, $dR_0/dr = -(r/R_0)\left(\beta_p^{loc} + l_i/2\right)$, is obtained from the local pressure and the poloidal field of a circular cross-section, and the shape variables are given by the $\kappa(x)$ profile of the QLKNN bridge and by $\delta(x) = \delta_{edge} x^2$. The transformation of the geometry inputs is therefore a model assumption, not a validated bridge. From the network fluxes, χ is

$$\chi_e = \frac{q_e^{GB}}{a/L_{T_e}}\frac{c_s\rho_s^2}{a}, \qquad \chi_i = \frac{q_i^{GB}}{(n_i/n_e)(T_i/T_e)\, a/L_{T_i}}\frac{c_s\rho_s^2}{a}, \qquad \frac{c_s\rho_s^2}{a} = \frac{\sqrt{m_D}\, T_e^{3/2}}{e^2 B_{unit}^2\, a}, \tag{23}$$

where T_e is in joules. The box of training inputs of the teacher is not published, so the input is clipped to ± 3.5 standard deviations of the training mean, and a point is counted as out of range when $|z_j| > 3.5$ for $z_j = \left(X_j - \mu_j\right)/\sigma_j$. The forward pass in NumPy was compared with the ONNX export (a multi-machine model of the same family) and agreed to a ratio of 10^{-5} on average. The weights are loaded with a restricted unpickler that only reconstructs NumPy arrays: the standard `pickle.load` would execute any code embedded in the file.

2.11 Software implementation

The whole system is written in Python: the SINTEZ simulator and SHRX use NumPy and SciPy only (JAX is needed just for the comparison with QLKNN and for training), and the backend is FastAPI. The 1.5-dimensional simulator and SHRX are available through an HTTP API: `/simulate` (the `transport_model` field takes `scaling_anchored`, `cgm`, `shrx`, `shrx_ens`, `shrx_v0` and others), `/ml/shrx` (χ_e, χ_i from one set of local inputs) and `/machines` (12 devices). Fig. 4 shows an example

response of `/ml/shrx` as used in this work: besides fluxes and χ, it returns an `out_of_range` flag that tells whether the input left the QLKNN box.

The interactive web interface “SINTEZ” (Figs. 2 and 3) runs a 0-dimensional physics model in the browser and does not use neural-network transport. It is only the visual layer of the software platform: a three-dimensional view of the reactor, heating sources (NBI, ICRH, ECRH), gas puffing, I_p, B_t, and disruption and pellet tests. All comparisons were carried out with the 1.5-dimensional simulator in the Python backend.

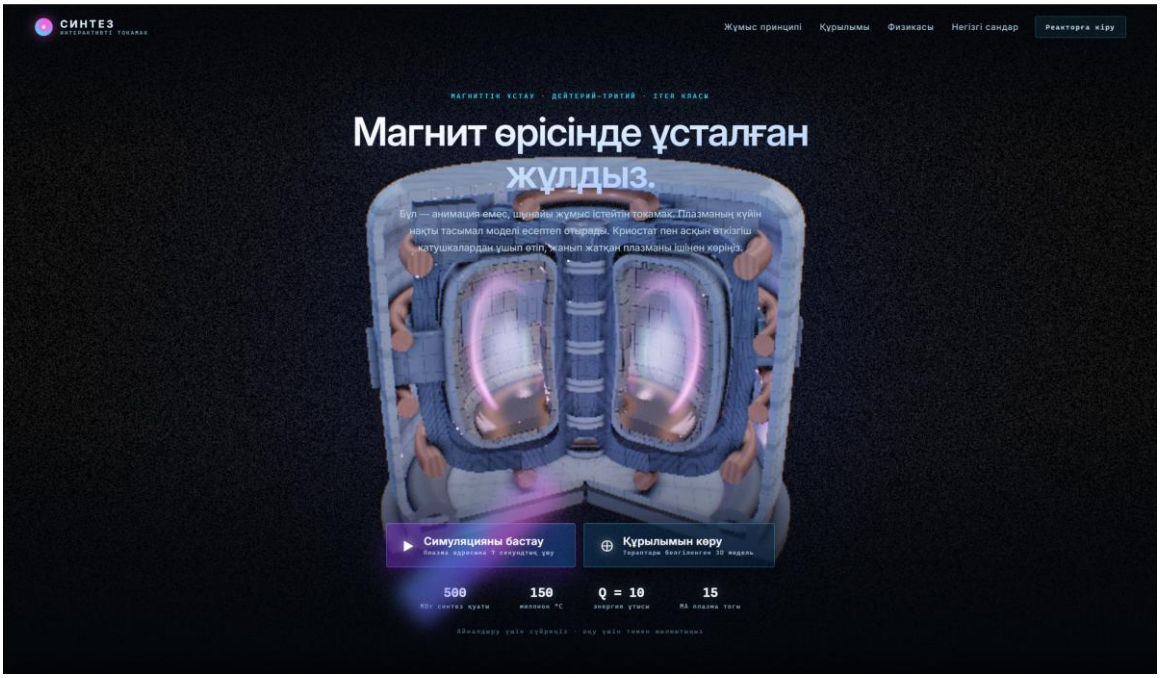


Fig. 2 The “SINTEZ” interface: home page (screenshot of the Kazakh-language interface)

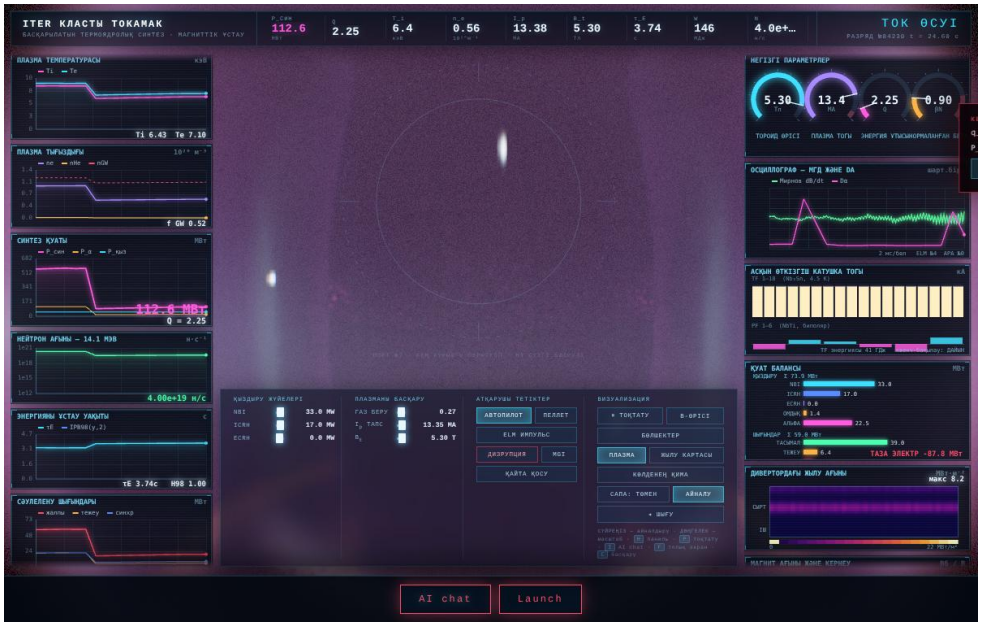


Fig. 3 The “SINTEZ” interface: simulation mode. On the left, plots of plasma temperature, density, fusion power, neutron flux, energy confinement time and radiation losses; on the right, the main parameters, the power balance and the divertor heat flux; at the bottom, the controls for heating, gas, I_p and B_t (screenshot of the Kazakh-language interface)

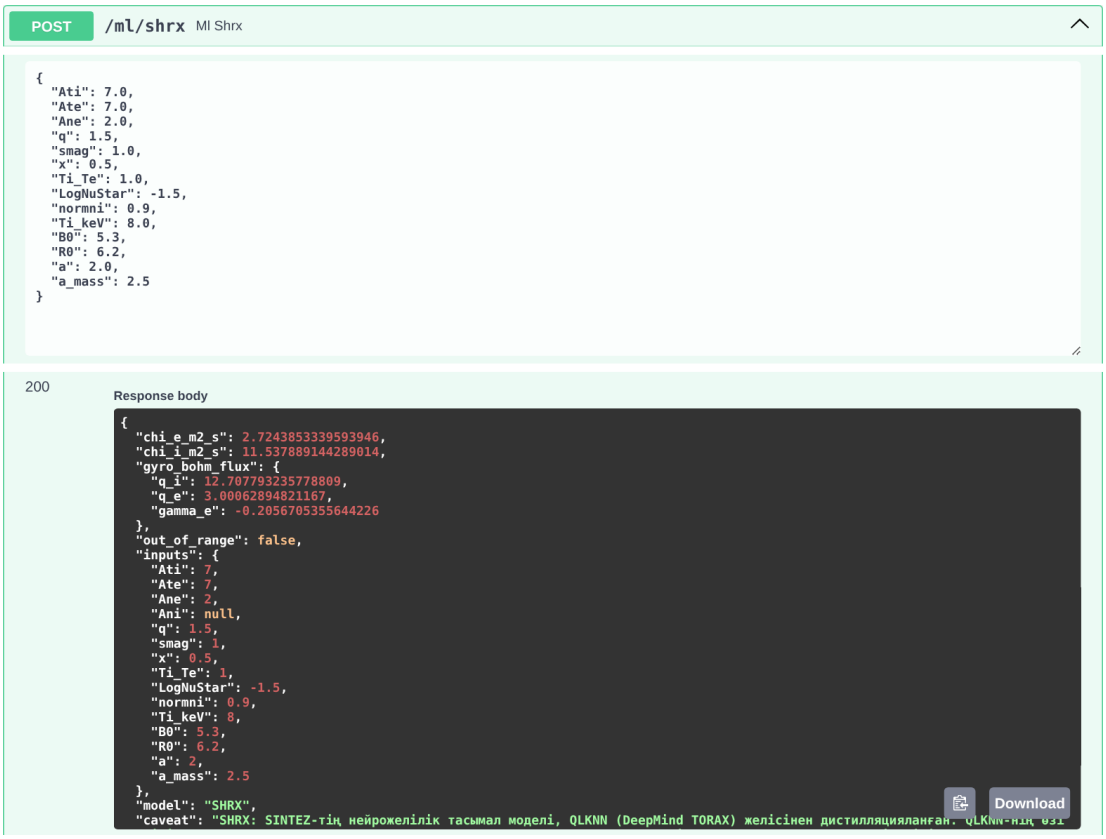


Fig. 4 A local call of SHRX through the HTTP API (`POST /ml/shrx`, Swagger UI screenshot, ITER-like inputs): the response contains χ_e, χ_i (m^2/s), the gyro-Bohm fluxes and the `out_of_range` flag

3 Agreement of SHRX with QLKNN

This section measures the SHRX network against its teacher, QLKNN. Here the "right answer" is QLKNN, not experiment, so however good SHRX becomes, it cannot exceed QLKNN in gyrokinetic accuracy; it only reproduces QLKNN in a form convenient for SINTEZ. The comparison with experiment is in Section 4.4.

3.1 Local accuracy

Table 3 gives the agreement between SHRX and QLKNN fluxes. On the machines that were not part of the training (NSTX, DEMO), $R^2 \approx 0.988$ and 0.987 for the electron and ion fluxes. At points where turbulence is on, the median relative error of χ is 8.7% (electrons) and 9.4% (ions), and 80% (electrons) and 75% (ions) of the points deviate from QLKNN by less than 25%. On the training machines the error is 1.6 times smaller (5.4%): the network generalises to a new machine, but not as accurately as it reproduces the training machines. The points outside the 25% corridor (Fig. 5) lie mostly in the region of weak flux ($q_e \lesssim 1$ in gyro-Bohm units), that is, near the threshold.

Table 3 Agreement of SHRX and QLKNN fluxes. R^2 is computed on gyro-Bohm fluxes; the median error and the share within $\pm 25\%$ refer to χ at points where QLKNN turbulence is on (Section 2.4)

Set	Points	R^2, q_e	R^2, q_i	Error, χ_e	Error, χ_i	$\pm 25\%, \chi_e$
Held out (NSTX, DEMO), SINTEZ states	91,000	0.988	0.987	8.7%	9.4%	80%
Held out, perturbed states	273,000	0.981	0.978	10.3%	10.9%	80%
Training machines, SINTEZ states	455,000	0.995	0.995	5.4%	4.7%	92%
Uniform points over the whole box	400,000	0.976	0.969	11.7%	12.3%	79%

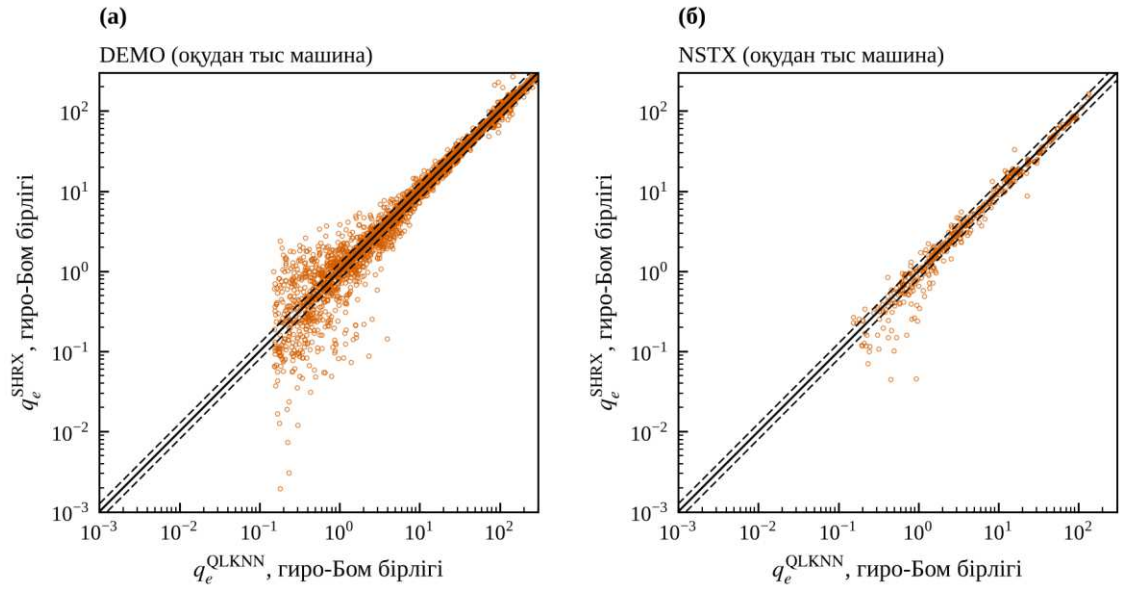


Fig. 5 SHRX and QLKNN electron heat flux (q_e) on machines held out of training: (a) DEMO, (b) NSTX. The solid line is $y = x$ and the dashed lines are $\pm 25\%$. Each point is one state visited by the simulator

3.2 Error in the threshold

For stiff transport, the mean error in flux does not predict the integrated result: the round-1 network gave 267 MW in the ITER calculation, against 478 MW for QLKNN (Table 1). This motivated the δ_f metric defined by Eq. (13). After samples from the threshold region were added, the median δ_f was 2.3% on the held-out machines and 1.1% on the training machines (Fig. 6). On the held-out machines, $|\delta_f| < 10\%$ in 91% of the states (93% for the ensemble). The share of states with $|\delta_f| < 3\%$ on the held-out machines rises from 59% to 67% when moving to the ensemble.

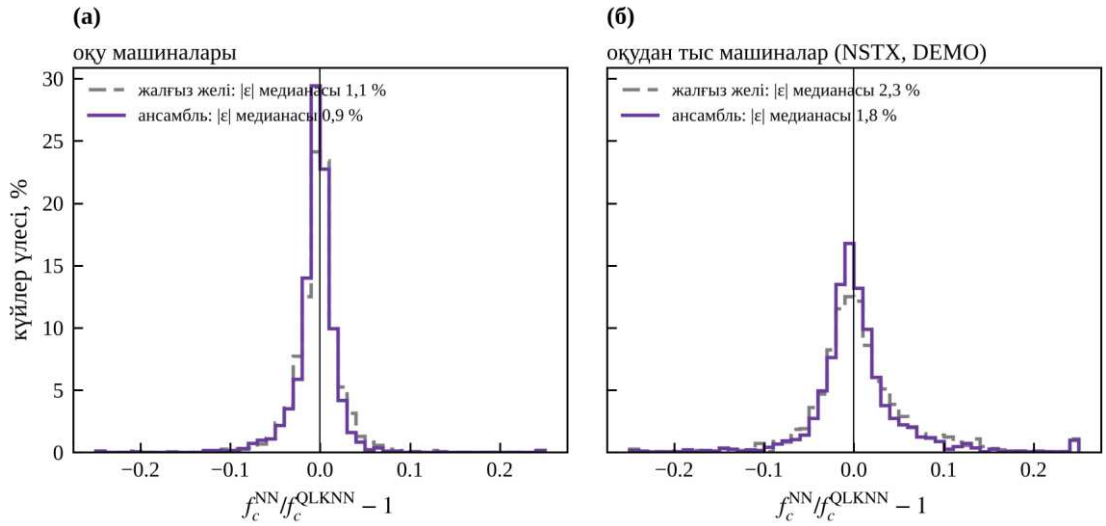


Fig. 6 Error of the critical-gradient factor $\delta_f = f_c^{NN}/f_c^{QLKNN} - 1$ (13): (a) training machines, (b) machines held out of training. The dashed line is a single network (member zero of the ensemble) and the solid line is the ensemble of five members. The numbers of states are 2,801 (a) and 2,019 (b)

3.3 The ensemble and uncertainty

The mean of five members is more accurate than a single network (Fig. 7a): the median error of χ_e is 6.5% on NSTX (8.7% for a single network) and 5.9% on DEMO (8.9%); the share of points with an

error below 25% grows from 85% to 91% (NSTX) and from 84% to 88% (DEMO). The spread between members s goes together with the error: the median error grows with the decile of s (Fig. 7b), and for the event "error > 25%" the AUROC is 0.73 on NSTX and 0.89 on DEMO; the Spearman correlations are 0.33 and 0.49.

The spread threshold τ was chosen on one machine and tested on the other (Table 4). The chosen values (0.21 and 0.19) are close to each other, so $\tau = 0.20$ was taken for the delivered ensemble. The median error of the flagged points is 2.7–4.5 times larger than that of the unflagged ones, but the precision is low: only 27–45% of the flagged points really belong to the large errors. The uncertainty is therefore useful as a "take a look" mark and not as a "do not trust" verdict.

Table 4 Cross-calibration of the spread threshold (chosen on one machine and tested on the other). The error (χ_e, ensemble mean) counts as "large" when it exceeds 25%

Select → test	τ	Flagged share	Precision	Recall	Median error: flagged / unflagged
NSTX → DEMO	0.21	18%	45%	71%	22.1% / 4.9%
DEMO → NSTX	0.19	12%	27%	36%	15.8% / 5.8%

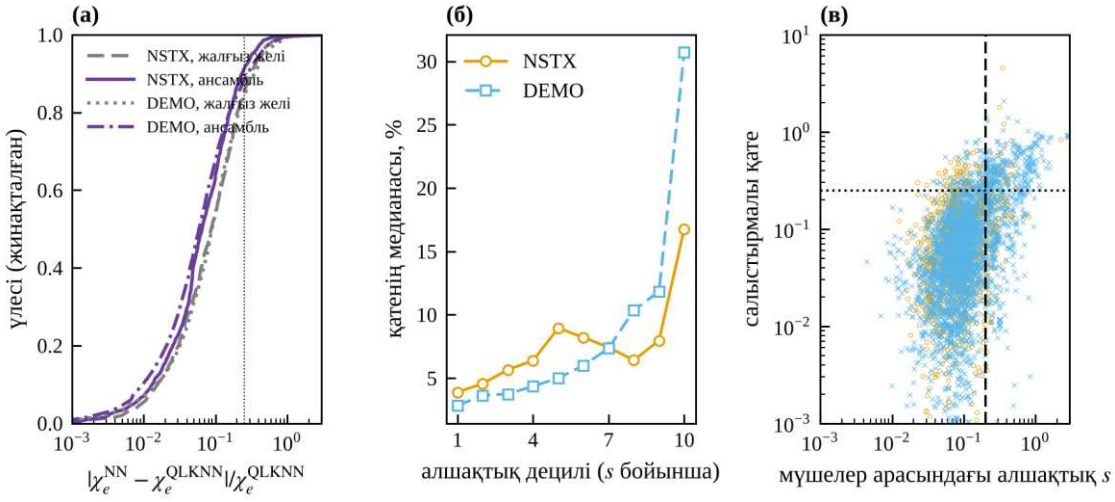


Fig. 7 Ensemble and uncertainty: (a) cumulative distribution of the relative error of χ_e (single network dashed or dotted, ensemble solid or dash-dotted); (b) median error by decile of the member spread; (c) spread against error for each point (two machines), with the vertical dashed line at $\tau = 0.20$ and the horizontal dotted line at 25% error

The effect of the ensemble on the integrated result is hard to measure (Section 4.3). In ITER the five members give 475–494 MW for P_{fus} (±2.0% about the mean), the ensemble gives 497 MW, the delivered network 494 MW and QLKNN 478 MW. Here P_{fus} is the mean over the flat top, whereas in Section 4.2 it is the final time value of the discharge, which is why the delivered network is quoted there at 492 MW.

3.4 Speed

One call of the SHRX network on 65 radial nodes takes 96 µs (one thread, float32). QLKNN compiled with jit in JAX takes 220 µs in float32 and 475 µs in float64, so SHRX is 2.3 and 5.0 times faster (on 26 nodes, 2.7 and 4.6 times). For the full call (bridge, network and conversion to χ) the factor is 4.9 (275 against 1356 µs). The network has 2.5 times fewer parameters than QLKNN and works without JAX. The measurement refers to the single-thread CPU execution of JAX; on a GPU or TPU, QLKNN may behave differently. The main result is not the speed itself but the fact that SHRX can be embedded in the simulator without a JAX dependency.

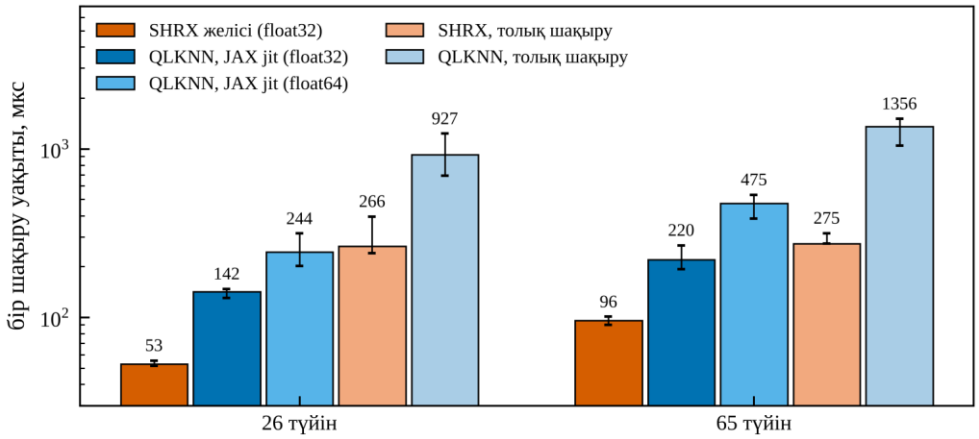


Fig. 8 Time per call (one thread, CPU; median, bars show the range). The rows are the SHRX network; QLKNN (JAX, jit, float32 and float64); and the full call of SHRX and QLKNN

3.5 The networks in the same simulator

We compared SHRX with the real QLKNN inside SINTEZ on the scripted discharges of 12 machines (100 s), Fig. 9. Among the conventional tokamaks (ITER, DEMO, SPARC, JET, DIII-D, EAST, KSTAR, T-15MD), the difference in W_{th} is below 30% on 5 of 8 machines (0.1% for ITER, 6.9% for JET). On the remaining three, DIII-D (64%), EAST (−48%) and KSTAR (−62%), the scripted discharge is itself a stiff system: a change of 10^{-6} in the initial state shifts the result by tens of percent (Section 4.3), so the difference cannot be put down to the network. The difference on spherical tokamaks is of the same kind (Section 4.5).

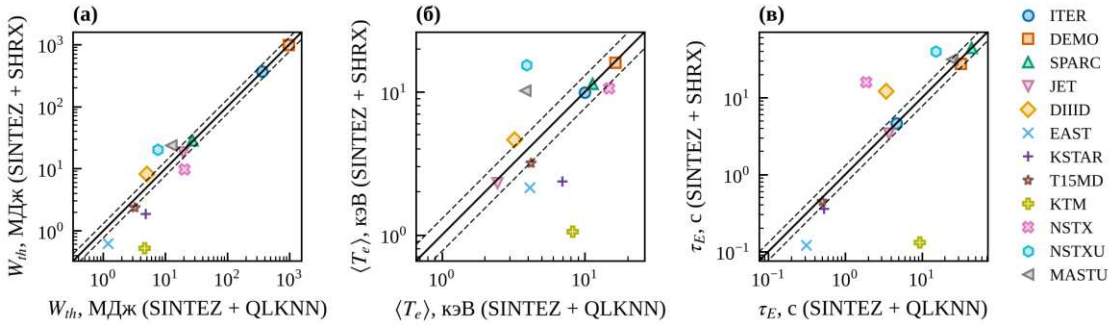


Fig. 9 SHRX and the real QLKNN inside SINTEZ, from scripted discharges of 12 machines: (a) stored energy, (b) mean electron temperature, (c) energy confinement time. The solid line is $y = x$ and the dashed lines are $\pm 30\%$

3.6 Speed-up of the simulator

The changes described in Section 2.7 speed up the same SINTEZ discharge (Table 5, Fig. 10): for ITER with the SHRX model, 26 s instead of 69 s (2.6 times), and 2.6–2.8 times in all four cases. The measurement was made on a machine with no other load, in one thread (`OMP_NUM_THREADS=1`); the "before" and "after" versions were run alternately (3 times per case). The main requirement is an identical result: in all four cases the final P_{fus}, Q, T_e, W_{th} and β_N of the two versions agree to the printed decimal, and in the six reference cases the deviation is exactly zero.

Table 5 Run time of one discharge (median, s) before and after the speed-up. “Identical” means that the printed results of the two versions match exactly

Case	Before, s	After, s	Speed-up	Identical result
iter, shrx	68.6	26.2	2.62×	yes
jet, shrx	82.3	30.5	2.70×	yes
demo, shrx	81.9	29.8	2.75×	yes
iter, scaling_anchored	64.6	22.7	2.85×	yes

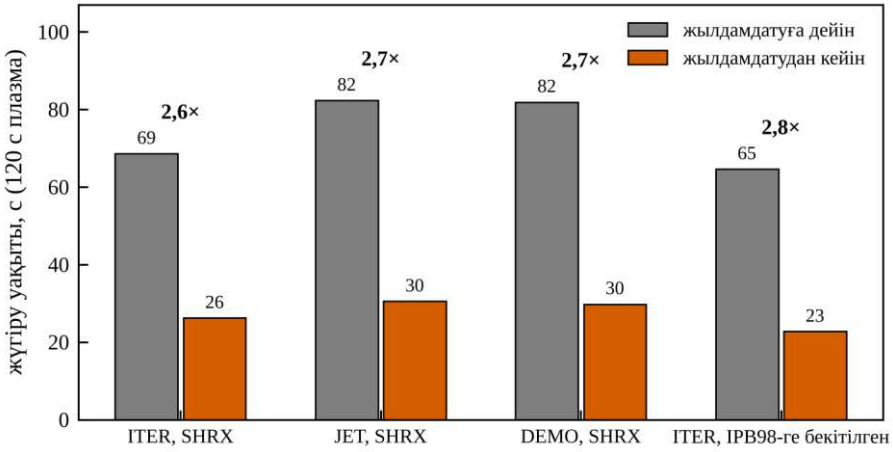


Fig. 10 Run time of a simulator discharge before and after the speed-up (120 s of plasma time, one thread); the speed-up is given above the bars

4 Comparison with TORAX

The comparison is made at the three levels described in Section 2.8: the local χ in identical states (A), an integrated ITER discharge with identical inputs (B), and real discharges of the experimental database (C). At each level we say plainly what is being compared: the network, the code, or both at once.

4.1 Level A: local χ in identical states

Fig. 11 shows $\chi_e(\rho)$ and $\chi_i(\rho)$ for three plasma states: (A) the stationary ITER state found by TORAX itself; (B1) the final state of a cold SINTEZ discharge run with the neural network; and (B2) a $Q \approx 10$ state run with the default SINTEZ model. For each state we took QLKNN through the bridge, SHRX and, in the (B) states, the own χ of TORAX. In B1 turbulence is off at every node (floor 0.05 m^2/s), so the three curves coincide, which proves nothing; the information comes from the other two states.

In state (B2), QLKNN through the bridge and the own χ of TORAX differ by a median of 11% in χ_e (80% of the points in $\rho \in [0.25; 0.85]$ are within 25%) and by 6% in χ_i (87%). This is a combined measure of the accuracy of the bridge and of the intricate differences in how TORAX computes its input variables (geometry, numerical limits on gradients, corrections in the sawtooth region). The median deviation of SHRX from QLKNN in χ is 7% (χ_e) and 4% (χ_i). The error of the bridge is thus of the same order as the error of the network.

In the TORAX state (A), the medians of SHRX and QLKNN for χ_e coincide (ratio 1.00), but only 73% of the points are within 25%, and for χ_i the median ratio is 1.29 with 40% of the points within 25%. This state lies near the critical gradient, and the training data saw that region only through SINTEZ states; for exactly this reason extra samples were added along the threshold (Section 2.5).

The correlation is high ($r = 0.85$ and 0.86): the network gets the shape right, and the difference lies in the exact position of the threshold.

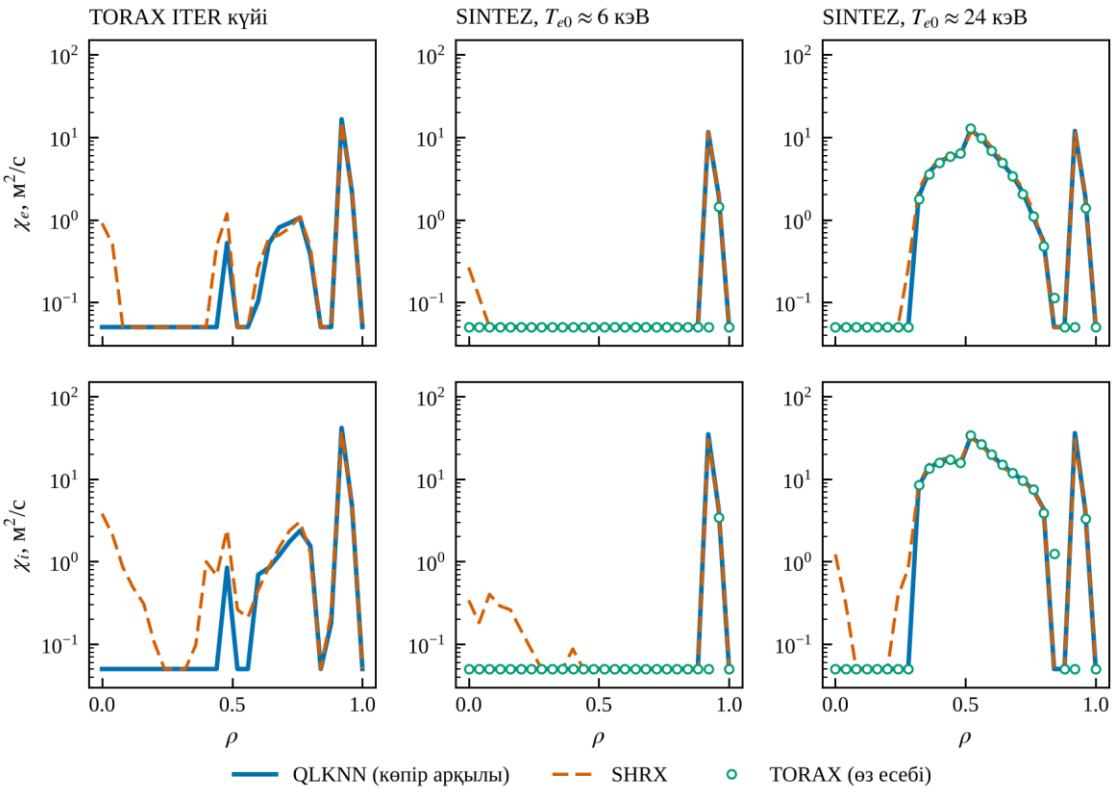


Fig. 11 χ_e (top row) and χ_i (bottom row) in identical plasma states: QLKNN (through the bridge, thick solid line), SHRX (dashed line) and the own calculation of TORAX (circles; absent in state (A), because that state is TORAX's own). Columns: the stationary ITER state of TORAX; the cold state of SINTEZ with SHRX ($T_{e0} \approx 6$ keV); the $Q \approx 10$ state of SINTEZ ($T_{e0} \approx 24$ keV). The lower bound is $\chi_{min} = 0.05$ m^2/s

4.2 Level B: integrated ITER with identical inputs

For the same ITER regime ($I_p = 15$ MA, $B_0 = 5.3$ T, $P_{aux} = 50$ MW, identical $n_e(\rho)$, pedestal and Z_{eff}), four results were obtained (Table 6, Fig. 12). TORAX with the real QLKNN gave $P_{fus} = 532$ MW ($Q = 10.6$); SINTEZ with QLKNN (the same network, a different code) gave 478 MW ($Q = 9.6$); SINTEZ with SHRX gave 492 MW ($Q = 9.8$); and the earlier SHRX v0 gave 67 MW ($Q = 1.3$), 8 times less than TORAX. The difference splits into two causes (Section 2.8):

$$\epsilon_{net} = \frac{P^{SHRX}}{P^{QLKNN}} - 1 = 3.1\,\%, \qquad \epsilon_{code} = \frac{P^{SINTEZ+QLKNN}}{P^{TORAX}} - 1 = -10.2\,\%. \tag{24}$$

The contribution of the network is smaller than that of the code. SHRX is 7.5% below TORAX, which is not far from the ITER design value of 500 MW (−1.5%), nor from the TORAX value itself (6.4%). This is a result for one particular regime: no general conclusion about the quality of the network follows from it (Section 4.3).

Table 6 Results for the same ITER regime. "Design" is the ITER $Q = 10$ design point. H_{98} is computed in every row with the SINTEZ definition ($P_{loss} = P_{aux} + P_{\alpha} - P_{rad}$)

Code and network	P_{fus}, MW	Q	W_{th}, MJ	$\langle T_e \rangle$, keV	T_{e0}, keV	β_N	H_{98}
ITER design	500	10.0	325	8.9	25	1.77	1.00
TORAX + QLKNN	532	10.6	356	8.5	15.5	1.78	0.97
SINTEZ + QLKNN	478	9.6	364	9.9	25.4	1.99	1.13
SINTEZ + SHRX	492	9.8	367	10.0	26.9	2.01	1.13
SINTEZ + SHRX v0	67	1.3	186	4.7	5.6	0.96	0.79
SINTEZ, anchored to IPB98	526	10.5	339	8.9	24.5	1.86	0.99

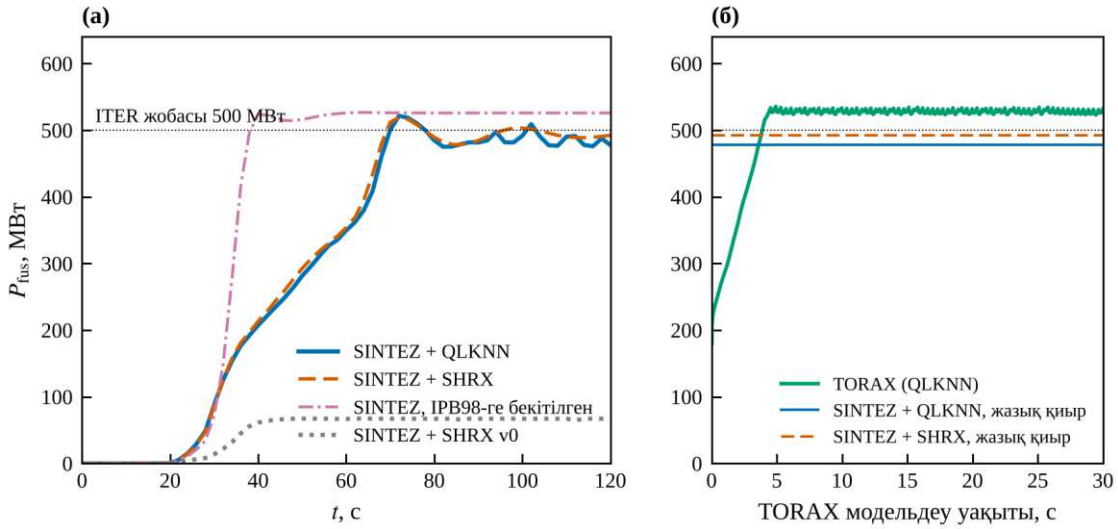


Fig. 12 Fusion power in ITER: (a) four SINTEZ models (120 s, until saturation, including the growth phase); (b) the accumulation of TORAX and the stationary values of SINTEZ with QLKNN and SINTEZ with SHRX. In TORAX the discharge starts from a warm initial state, so the time axes are not comparable and only stationary values are compared

The profiles (Fig. 13) differ much more. In SINTEZ both networks give a central temperature $T_{e0} \approx$ 25–27 keV, while TORAX gives 15.5 keV ($T_{i0} = 13.0$ keV); for $\rho > 0.4$ the profiles agree. In TORAX $q_0 = 0.78$ and the sawtooth model produces 76 crashes; with the sawtooth switched off, $q_0 \approx 0.4$ and $P_{fus} = 816$ MW, which is not physical. With added radiation (P_{rad} raised to 36 MW), P_{fus} is 526 MW. The difference in the central region could be attributed to the inner patch of TORAX ($\chi = 1$ m^2/s for $\rho < 0.2$) and its sawtooth model, and in SINTEZ to differences in the sawtooth and current drive, but this is a hypothesis that we did not test separately.

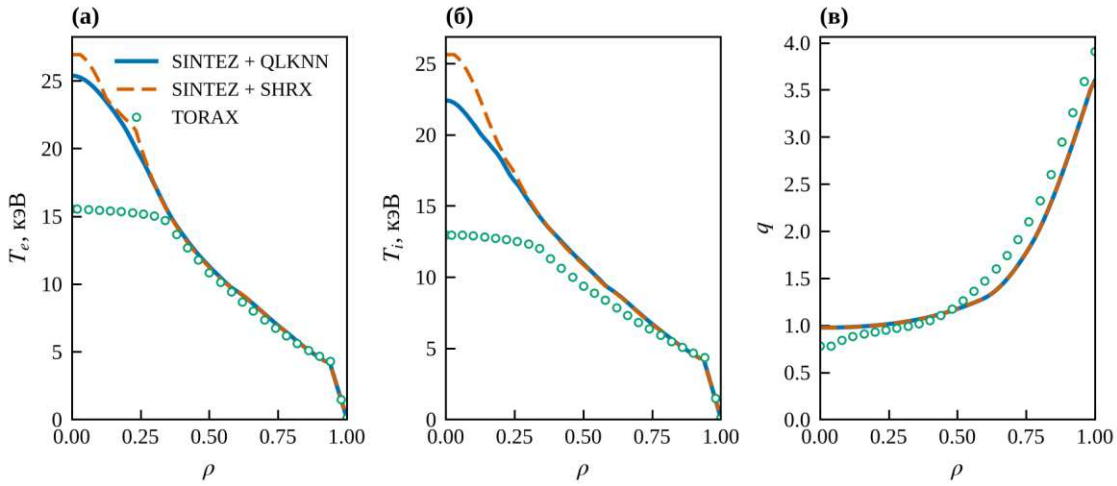


Fig. 13 Profiles for the same ITER regime: (a) electron temperature, (b) ion temperature, (c) safety factor. SINTEZ with QLKNN (solid), SINTEZ with SHRX (dashed), TORAX (circles)

4.3 Sensitivity and the limits of the comparison

To read an integrated comparison correctly, one needs to know how sensitive the simulator itself is to a small perturbation. We perturbed the initial electron temperature by relative amounts $\delta = 10^{-12}$, 10^{-9} and 10^{-6} (Fig. 14) and measured $\max_\rho |T_e - T_e^{ref}| / \max_\rho T_e^{ref}$ for each machine. A rounding-level perturbation ($\delta = 10^{-12}$) is amplified within 1–2 s: at $t \approx 1.8$ s it reaches 2.2×10^{-6} in ITER, 6.7×10^{-6} in JET, 2.2×10^{-8} in DIII-D and 2.5×10^{-9} in KSTAR (3.4–6.8 orders above the initial 10^{-12}). Another jump occurs at $t \approx 14$ s, after which the deviation settles: its median for $t \geq 25$ s is 0.09% in ITER, 0.0% in JET, 0.03% in DIII-D and 88% in KSTAR, which means that the whole profile moves to a different state. When the initial perturbation is increased to 10^{-6}, the settled level in ITER, JET and KSTAR is practically unchanged (Fig. 14): it depends on the bifurcation events of the system and not on the size of the perturbation (we did not investigate the cause of this phenomenon further). This is why the speed-up has to be bit-for-bit identical (Section 2.7), and why a single run is not enough to compare networks on small and spherical machines.

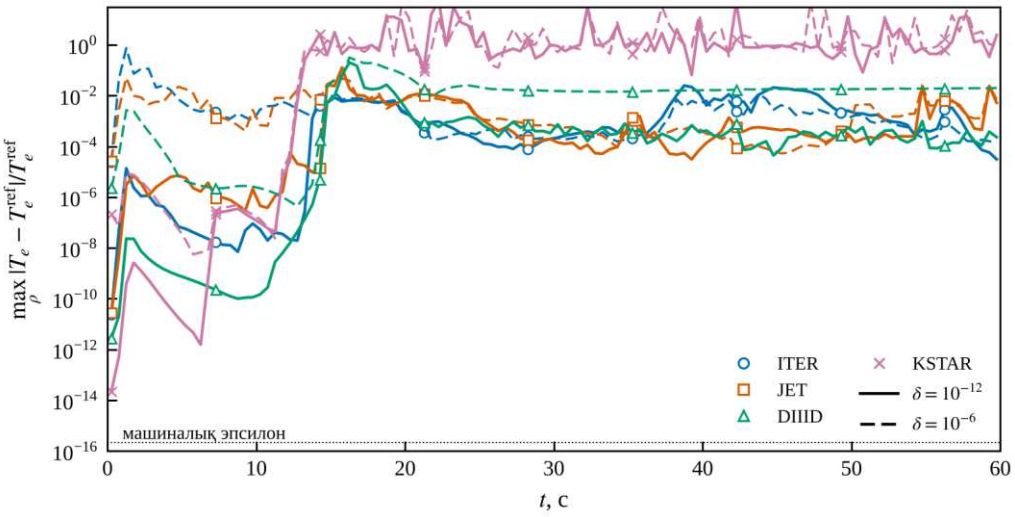


Fig. 14 Sensitivity of the simulator to the initial T_e perturbation (with the SHRX model): $\max_\rho |T_e - T_e^{ref}| / \max_\rho T_e^{ref}$ against time. The solid line is $\delta = 10^{-12}$ and the dashed line is 10^{-6}; each machine has its own marker. The dotted line is machine epsilon

This bears directly on the comparison of networks (Fig. 15). ITER and JET are stable: the five members lie within ±2.0% in P_{fus}, and in JET within 2.32–2.50 keV in $\langle T_e \rangle$ (the ensemble gives 2.43 keV). On DIII-D, KSTAR and KTM, networks with the same local accuracy give results whose $\langle T_e \rangle$ differs by factors of 2.1, 4.2 and 8.5; perturbing the initial T_e by 10^{-6}–10^{-3} alone gives factors of 1.45, 1.29 and 1.25. The spread between members is larger than that caused by the initial perturbation, so small differences between networks are amplified on these machines; and the ensemble result may even fall outside the range of the members (DIII-D). The conclusion is that the advantage of the ensemble shows in the local and threshold metrics (Section 3.3), but in the integrated result on stiff machines it cannot be isolated, and such machines are not suitable for judging the quality of a network.

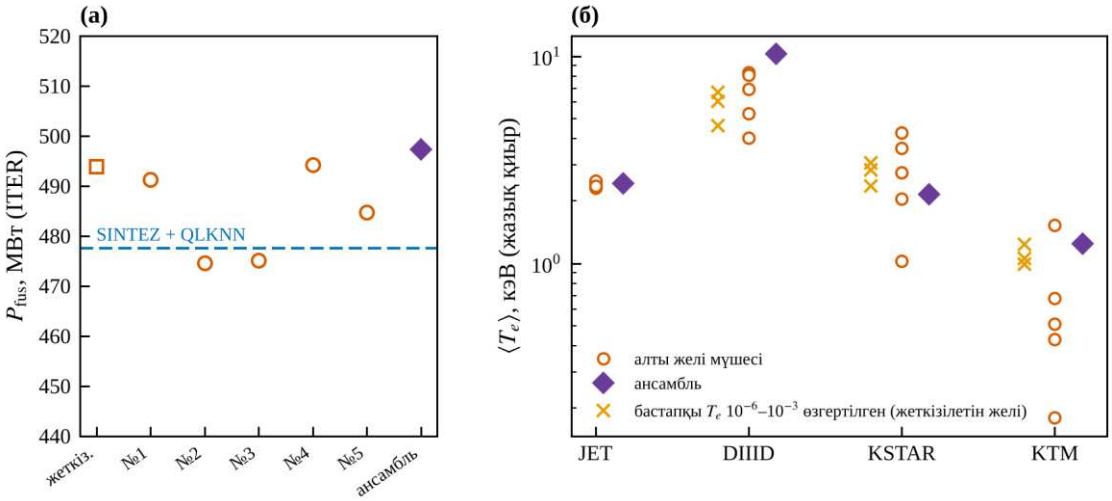


Fig. 15 Sensitivity of the integrated result to the network weights: (a) P_{fus} in ITER (delivered network, five members, ensemble; the dashed line is SINTEZ with QLKNN); (b) stationary $\langle T_e \rangle$ on four machines (circles are the five network members, diamonds the ensemble, crosses the runs of the delivered network with the initial T_e perturbed by 10^{-6}–10^{-3})

4.4 Level C: the ITPA H-mode database

4.4.1 Predictive pedestal

SINTEZ with SHRX ran on all 206 records, but only 67 of them (33%) met the validated-run criterion (Section 2.9). Most of the rest went into thermal runaway (Table 7): 129 runs have $\beta_N \geq 3.5$ or a bootstrap current fraction ≥ 0.8. The median pedestal-top temperature in such runs is 5.5 keV, against 0.63 keV in the valid ones. This is the error of an EPED-like pedestal model in regimes beyond its usual capability (low I_p, high β_p, old limiter devices): because of the outer boundary condition, the stored energy is determined mainly by T_{ped}, and core transport changes it only slightly. The devices with no valid run at all are ASDEX, JFT2M, PDX, T10, TDEV and TFTR.

Table 7 Results by device (conventional aspect ratio). “Valid” is the number of runs that meet all three criteria; “Density”, “Power” and “No runaway” are the numbers of runs that meet each criterion separately. The ratios $\tilde{r} = W^{pred}/W^{exp}$ are medians by device; $1/H_{98}$ is the IPB98(y,2) ratio in the database (H_{98} is absent from the AUG records)

Device	a/R	N	Valid	Density	Power	No runaway	$\tilde{r}$, SHRX (all)	$\tilde{r}$, SHRX (valid)	$\tilde{r}$, TORAX	$1/H_{98}$
ASDEX	0.24	16	0	16	16	0	8.52	–	–	1.02
AUG	0.31	16	12	16	15	12	1.41	1.25	1.16	–
CMOD	0.33	16	14	16	14	16	1.26	1.25	1.65	0.92
COMPASS	0.30	16	6	16	6	13	3.03	2.35	2.58	1.28
D3D	0.37	16	13	16	16	13	0.95	0.94	0.93	0.95
JET	0.32	16	14	16	16	14	1.69	1.76	1.63	1.12
JFT2M	0.21	16	0	16	16	0	4.07	–	–	0.95
JT60U	0.26	16	1	12	15	1	4.19	5.29	4.39	1.32
PBXM	0.16	16	4	16	16	4	2.65	0.36	0.34	0.85
PDX	0.29	16	0	16	16	0	7.30	–	–	1.04
T10	0.20	4	0	0	4	0	39.43	–	–	1.15
TCV	0.25	16	3	16	10	4	3.40	2.08	3.85	1.27
TDEV	0.26	10	0	10	9	0	8.21	–	–	0.95
TFTR	0.33	16	0	13	14	0	9.29	–	–	0.75

Table 8 gives the summary indicators. On the 67 validated runs, SINTEZ with SHRX overestimates the stored energy by a median of 24%, with a spread $\sigma_{\ln} = 0.55$, and only 45% of the records lie within ±30%. The IPB98(y,2) regression, fitted to exactly these data, gives a spread of 0.21 and 84% (±30%) on the same records. So *with a predictive pedestal, SINTEZ with SHRX is weaker than the regression*: it does not “know” the data and cannot predict the pedestal accurately (Figs. 16 and 17).

Table 8 Stored-energy prediction: $\tilde{r} = W^{pred}/W^{exp}$ (18). The last three rows are the set of records common to the comparison with TORAX (validated SINTEZ and SHRX runs for which TORAX was also computed)

Prediction	N	$\tilde{r}$	$\sigma_{\ln}$	f_{30}
SINTEZ + SHRX, all runs	206	3.65	0.98	15%
SINTEZ + SHRX, validated runs	67	1.24	0.55	45%
IPB98(y,2) regression, same records	55	0.99	0.21	84%
Common set: SINTEZ + SHRX	64	1.25	0.51	47%
Common set: TORAX	64	1.55	0.56	36%
Common set: IPB98(y,2)	52	1.00	0.22	83%

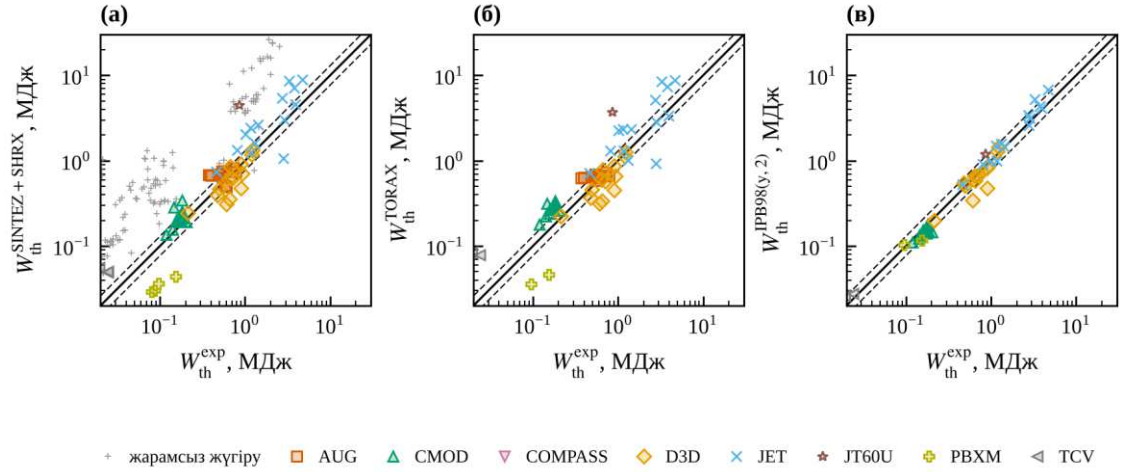


Fig. 16 Stored energy: prediction against experiment. (a) SINTEZ with SHRX: crosses are runs that failed validation, markers are validated runs (by device); (b) TORAX and (c) IPB98(y,2) for the same validated records. The solid line is $y = x$ and the dashed lines are $\pm 30\%$

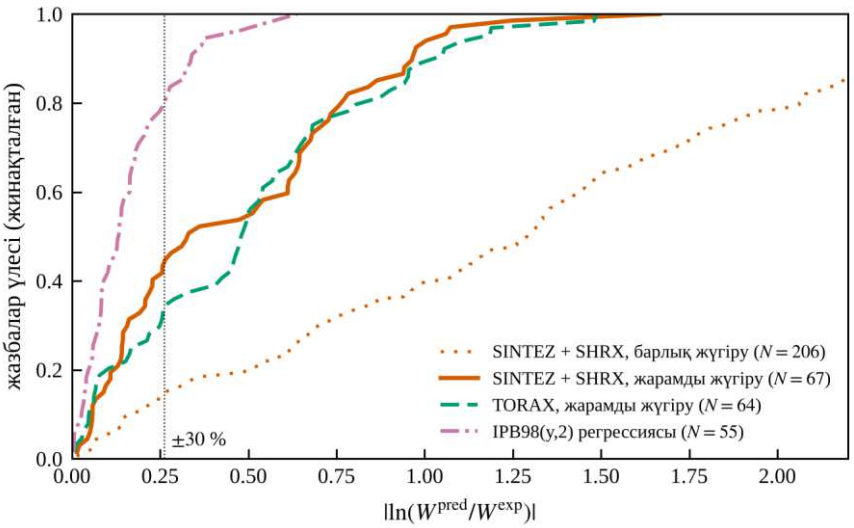


Fig. 17 Cumulative distribution of the prediction error $|\ln(W^{pred}/W^{exp})|$. The dotted line is all runs and the others are validated records. The vertical dotted line is ±30% ($\ln 1.3 = 0.26$)

Comparison with TORAX. TORAX was run on 64 records with the density, pedestal and power set by the same SINTEZ, but with its own code (circular cross-section, fixed q, no current diffusion). Against experiment, TORAX has $\tilde{r} = 1.55$, $\sigma_{\ln} = 0.56$ and $f_{30} = 36\%$, whereas on the same set SINTEZ with SHRX has 1.25, 0.51 and 47%. The two codes are much closer to each other than to experiment (Fig. 18): the correlation in $\ln W$ is $r = 0.995$, the median ratio is 1.02 and $\sigma_{\ln} = 0.18$ ($N = 64$), so the difference between the two models ($\sigma_{\ln} = 0.18$) is 2.8–3.1 times smaller than the difference of each from experiment (0.51 and 0.56). With identical inputs, most of the deviation from experiment comes from something that the two codes share (the pedestal, the density profile, the distribution of heating sources) and not from the transport network.

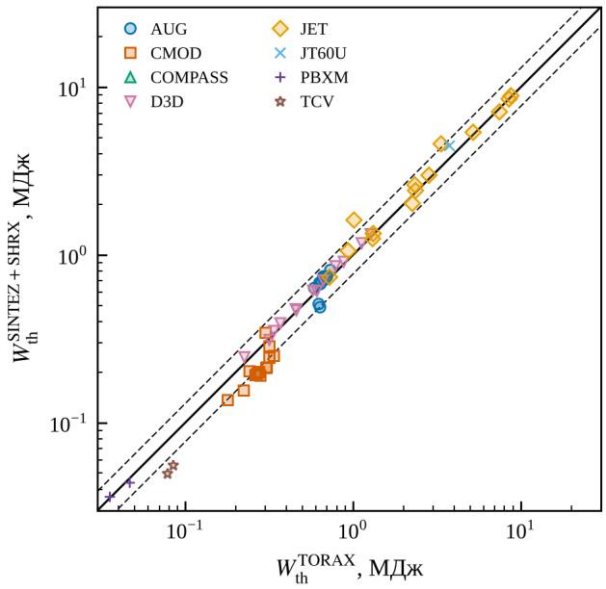


Fig. 18 Stored energy predicted by SINTEZ with SHRX and by TORAX with identical inputs (validated records). The solid line is $y = x$ and the dashed lines are $\pm 30\%$

4.4.2 Core-only mode

To separate the pedestal error from the transport error, the temperature at the top of the pedestal was chosen so that the measured W_{th} is reproduced (20). Of the 94 records with a measured central

temperature, 39 passed the cleanliness criteria (density 72, power 51, no runaway 59, and the pedestal stayed within the search range) and were compared with TORAX on 37 records. With the same SINTEZ pedestal, TORAX also reproduces the stored energy well (the median W^{TORAX}/W^{exp} is 1.04, the spread 0.15, and 86% of records are within ±30%): the pedestal determines most of the energy.

The shape of the core profile is shown by the central temperatures (Table 9, Fig. 19). Both codes overestimate the central electron temperature: the median $T_{e0}^{pred}/T_{e0}^{exp}$ is 1.45 ($\sigma_{\ln} = 0.38$) for SINTEZ with SHRX and 1.32 (0.30) for TORAX. For the central ion temperature the median is 0.85 with SHRX and 1.05 with TORAX, but the scatter is large ($\sigma_{\ln} = 0.73$ and 0.56). A clear systematic error is in the ratio T_i/T_e: the measured mean $T_{i0}/T_{e0} \approx 0.97$, whereas SINTEZ with SHRX gives 0.47 ($N = 23$). The two codes are close to each other: the SHRX/TORAX ratio is 1.09 ($\sigma_{\ln} = 0.17$) for T_{e0} and 1.11 ($\sigma_{\ln} = 0.44$) for T_{i0}. It would therefore be wrong to call the imbalance between T_e and T_i an error of the network alone: it is shared by QLKNN and TORAX, so it may come from the simulator (heating sources, electron–ion exchange) and from the electron channel of QLKNN. We did not test this hypothesis separately.

Table 9 Prediction of the central temperatures in core-only mode (pedestal fitted to the measured W_{th}): prediction over measurement

Quantity	**Code**	N	**Median**	$\sigma_{\ln}$	f_{30}
T_{e0}	SINTEZ + SHRX	37	1.45	0.38	30%
T_{e0}	TORAX	35	1.32	0.30	37%
T_{i0}	SINTEZ + SHRX	25	0.85	0.73	28%
T_{i0}	TORAX	24	1.05	0.56	58%

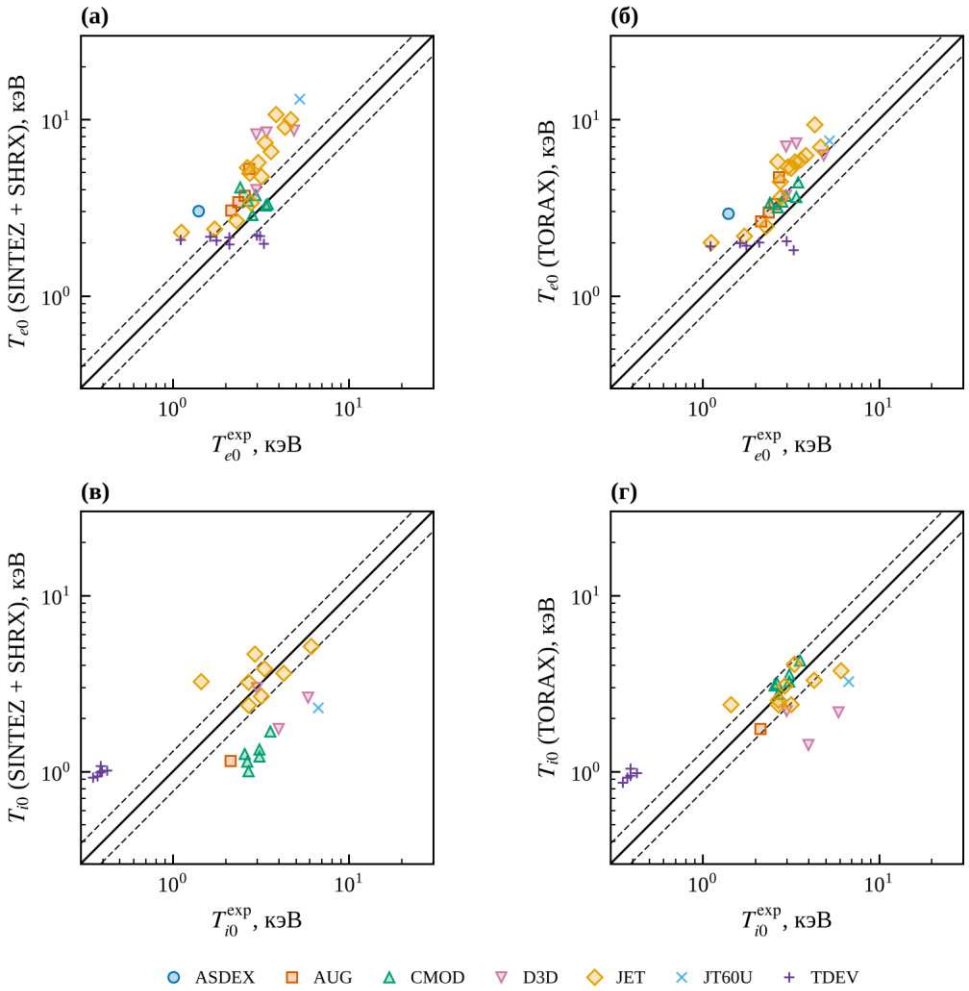


Fig. 19 Core-only mode: predicted and measured central temperatures: (a, b) T_{e0}, (c, d) T_{i0}; SINTEZ with SHRX on the left and TORAX on the right. The solid line is $y = x$ and the dashed lines are $\pm 30\%$. Values below 0.2 keV in the database are not measurements (filler values) and were excluded

4.5 Spherical tokamaks and KTM

This section reports a negative result: the present SHRX, which rests on QLKNN and on the UKAEA TGLFNN for STEP, cannot predict spherical tokamaks or KTM reliably.

SHRX (QLKNN) on spherical records. For the 41 records of NSTX, MAST and START, SINTEZ with SHRX overestimates the stored energy by a median factor of 28 (22 for MAST, 31 for NSTX, 62 for START; not one record is within ±30%). The plasma goes into thermal runaway: the central electron temperature is 20–40 keV and the ion temperature 5–6 keV (T_i/T_e is below the lower bound of the training box, 0.25; at all points in NSTX and at 50% of the points in MAST). We fed these states to both SHRX and the real QLKNN (Table 10): QLKNN switches turbulence off at every point inside the box ($\chi = 0.05$ m^2/s), whereas SHRX in NSTX gives the upper limit (100 m^2/s) for χ_e. The comparison is one-sided, because these states are the product of a SHRX run, but it shows how far the network departs from QLKNN when it leaves its training range. A possible cause (not tested separately) is that the small but non-zero flux of the network at the edge of the box is multiplied by the large gyro-Bohm scale ($\propto T_e^{3/2}/B_0^2$; $B_0 = 0.45$ T in NSTX).

Table 10 SHRX and QLKNN in the final states of SHRX runs for spherical tokamaks: the median χ (m^2/s) at radial points that lie inside the training box, and the share of those points at the floor (for χ_e)

Group	a/R	N	Points outside the box	χ_e, SHRX	χ_e, QLKNN	Floor, SHRX	Floor, QLKNN
MAST	0.69	16	50%	100.00	0.05	9%	100%
NSTX	0.71	16	100%	100.00	0.05	0%	100%
START	0.76	9	100%	0.39	0.05	45%	100%

SHRX now reports this situation: `shrx.machine_validity` gives a "valid / borderline / outside" label by a/R_0 and κ (KTM is borderline, and NSTX, NSTX-U and MAST-U are outside), and the simulator writes one warning and prints the numbers `shrx_validity` and `shrx_ood_core` (the share of radial points inside the box that are out of range) at every step.

The second teacher: TGLFNN-STEP. We first believed that there was no public surrogate covering spherical geometry, which was wrong: UKAEA has published the weights of a TGLFNN version for the STEP ramp-up phase (SPR45, $R/a \approx 1.8$, $\kappa \approx 2.3$, finite β) [7, 20]. It is a deep ensemble of TGLF-EM with 15 inputs and five networks per flux (Section 2.10). We added it to SINTEZ as the `tglfnn_ref` model (for evaluation only; the weights are not part of the repository). On the scripted discharges of KTM, NSTX and MAST-U it gives a physically plausible state with no thermal runaway (Table 11).

Table 11 Scripted discharge of spherical machines when the TGLFNN-STEP "teacher" is connected to SINTEZ (60 s; no thermal runaway, in contrast with the box run of SHRX)

Machine	a/R	$\langle T_e \rangle$, keV	W_{th}, MJ	β_N	H_{98}	Points out of range
KTM	0.50	0.80	0.11	2.44	0.43	100%
NSTX	0.79	0.40	0.08	1.75	0.23	100%
MAST-U	0.76	0.58	0.29	1.38	0.18	100%

However, the training range of the teacher does not cover the states of these machines (Fig. 20). Among its inputs, for the collisionality ($XNUE = \nu_{ee}/(c_s/a)$) the STEP training mean is 0.0087 ± 0.0083 (STEP is hot and collisionless), whereas in the KTM states produced by SINTEZ the median is 2.3, in NSTX 0.5 and in MAST-U 0.9; for the spherical records, NSTX has 3.2 and MAST 1.1. For a simple box range on the input (training mean $\pm 3.5\sigma$), one input is out of range at 92% (KTM) and 100% (NSTX records) of the points that lie inside the box. These numbers refer to the states produced by SINTEZ, which are cold ($T_e \approx 0.5$–1 keV), so the collisionality is high. For a real KTM the problem depends on T_e: $XNUE \propto n_e\, a \ln\Lambda\, T_e^{-2}$, and for $n_e = 5 \cdot 10^{19}$ m^{-3} and $a = 0.43$ m it equals 0.52, 0.14, 0.036 and 0.016 at $T_e = 0.5$, 1, 2 and 3 keV. The upper bound of the teacher's box range (training mean $+3.5\sigma$) is 0.038, so KTM enters the teacher's range only at $T_e \gtrsim 1.9$ keV; for comparison, STEP ($T_e = 10$ keV, $n_e = 8 \cdot 10^{19}$ m^{-3}, $a = 2$ m) gives 0.012.

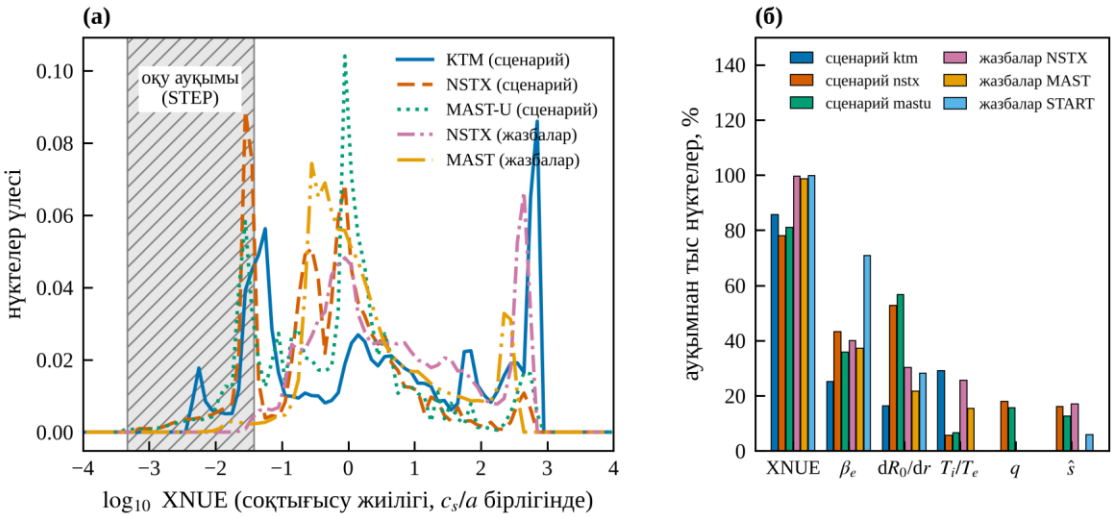


Fig. 20 Training range of TGLFNN-STEP and the spherical states of SINTEZ: (a) distribution of the collisionality input ($\log_{10}XNUE$; the hatched region lies between -1σ and $+3.5\sigma$ of the STEP training mean); (b) share of points outside the training range, over several inputs

Distilling TGLFNN-STEP for KTM would therefore make sense only in the hot regime ($T_e \gtrsim 2$ keV), while the present KTM states of SINTEZ are much colder: for the latter the network would only repeat the extrapolation of the teacher. We therefore developed only the structure of the SHRX-ST trainer (state collection, dataset construction, training; the bridge is described in Section 2.10) and do not deliver a network for KTM: its correctness is not verified. A teacher suitable for KTM must cover both spherical geometry and the collisionality of present-day devices, and that requires new TGLF or QuaLiKiz calculations; they were not carried out because this environment has no Fortran compiler. The geometric range of TGLFNN-STEP is not verified either for NSTX and MAST-U ($R/a \approx 1.3$), since R/a is fixed in its training data and its exact value is not published.

The teacher in the loop on spherical records. SINTEZ with TGLFNN-STEP underestimates the stored energy on the same 41 records by a median factor of 2.2 (SHRX overestimates it by 28), with $\sigma_{\ln} = 0.36$ and only 2% of the records within ±30% (Fig. 21). The sign of the error is opposite to that of SHRX, but its size is also large. Because the inputs of the teacher lie outside its training range (see above), this result does not show whether TGLFNN is right or wrong: it is an extrapolation. Neither teacher therefore gives the stored energy of spherical tokamaks at a usable accuracy relative to experiment.

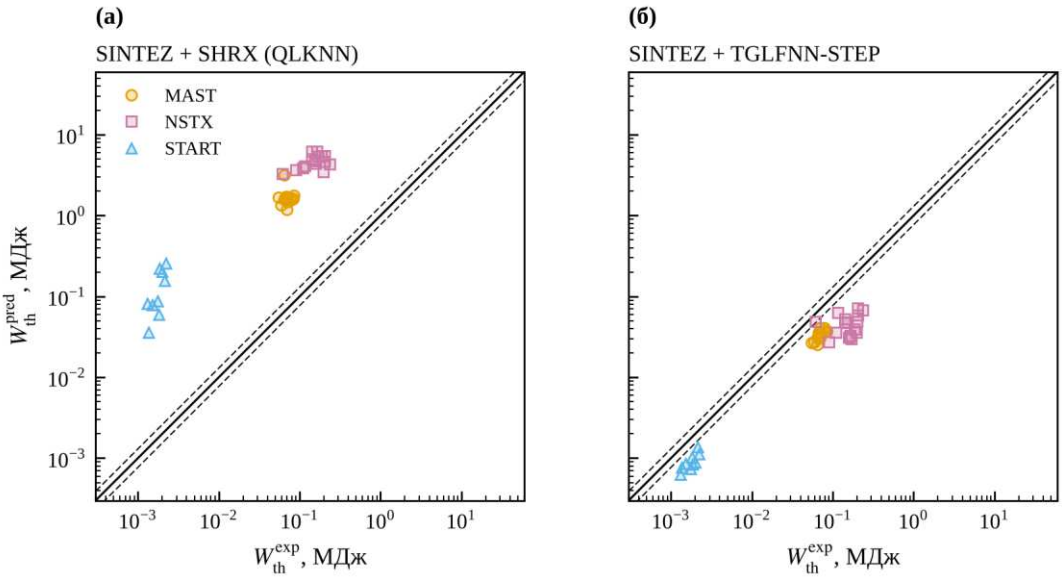


Fig. 21 Stored energy on spherical tokamaks: (a) SINTEZ with SHRX (QLKNN); (b) SINTEZ with TGLFNN-STEP (with the predictive SINTEZ pedestal). The solid line is $y = x$ and the dashed lines are $\pm 30\%$

5 Discussion

What SHRX does and does not do. SHRX is a compressed copy of QLKNN: on machines held out of training the median error in χ is 9%, and in the critical gradient 2.3% (Sections 3.1 and 3.2). It enters SINTEZ without JAX, and it is 2.3–5.0 times faster than QLKNN run by JAX on a single CPU thread. But its teacher is QLKNN, so it inherits the gyrokinetic errors of QLKNN and cannot exceed its physical accuracy. To be "stronger than TORAX" it would need other ingredients: new gyrokinetic calculations, a correction against experiment, or an error estimate. This paper produced only one kind of the last (the spread between members).

Network, code and data: where the error comes from. In ITER (Section 4.2) the contribution of the network is $\epsilon_{net} = 3.1\,\%$ and that of the code $\epsilon_{code} = -10.2\,\%$; in the profiles, the difference in the central region is much larger between the two codes than between the networks. On experimental records (Section 4.4), SINTEZ with SHRX and TORAX are much closer to each other ($\sigma_{\ln} = 0.18$) than to experiment ($\sigma_{\ln} = 0.51$ and 0.56). Both results lead to one conclusion: the error of an integrated prediction comes more from the input data (the pedestal, the density profile, the heating sources) and from the code itself than from the network. Improving the transport network, for example with more training data, would therefore not improve the comparison with experiment noticeably, whereas a better model of the pedestal and of the sources would. This conclusion applies only to the comparison domain studied here.

Performance against experiment. The IPB98(y,2) regression gives $\sigma_{\ln} = 0.21$ on similar records, and SINTEZ with SHRX gives 0.55 (Table 8). The regression is fitted to the data of the ITPA database, whereas SHRX and TORAX have "not seen" these records, so the comparison favours the regression. Still, the task of a transport model is to reach an accuracy no worse than a scaling law, because a physical model predicts profiles as well as the confinement time. The present result is still far from that level, and its main cause is the pedestal and not transport (Section 4.4.1).

Sensitivity and the limits of the comparison. On stiff machines (DIII-D, KSTAR, KTM, spherical tokamaks) the simulator result is sensitive to a 10^{-12} change in the initial state (Section 4.3): a single run is not enough to compare networks, and different seeds of one network give T_e that differ by factors of 2.1–8.5. ITER and JET are stable, so an integrated comparison is meaningful only on them.

This is a warning that a comparison with TORAX "in one test" is insufficient; it needs many runs and statistics, whereas we compared ITER only once.

Uncertainty. The ensemble spread shows the deviation of the network in regions not covered by its training data (AUROC 0.73–0.89), but it does not show the error of QLKNN as a teacher, and it gives many false alarms (precision 27–45%). TORAX and QLKNN themselves give no uncertainty estimate; the UKAEA TGLFNN trained with NLL provides a mean and a variance, but we used only the ensemble mean.

Spherical tokamaks. The present models, which rely on QLKNN or on TGLFNN-STEP, are unfit to predict KTM (Section 4.5): for QLKNN the ranges of aspect ratio and T_i/T_e are exceeded, and for TGLFNN-STEP the present KTM states of SINTEZ are out of range in collisionality. The `machine_validity` and `shrx_ood_core` diagnostics were added to keep SHRX in check here, but a "warned" result is not thereby a correct one. The realistic ways forward for KTM are (i) new TGLF or QuaLiKiz calculations covering the geometry and collisionality of KTM, and (ii) a correction against KTM's own measurements, if they are available.

Limitations. (i) Only one ITER regime was compared directly with TORAX. (ii) The TORAX set-up stayed at its initial form (simple geometry, fixed q), and many of its parameters were not tuned. (iii) Of 206 records, only 67 passed validation, and the comparison with TORAX was made on 64 records (TORAX did not finish on the remaining 3: in one record it exceeded the memory and time limits, and the next ones in the queue did not finish in time; in core-only mode, on 37 of 39 records; the records not computed belong to the devices PBXM, TCV and TDEV, so the share of these devices is somewhat lower). (iv) The DB5 records are compared only for stored energy and central temperatures; profiles were not compared. (v) The results refer only to simulation and to this sample of the open database.

6 Conclusions

1. SHRX is a network of 29,283 parameters distilled from QLKNN; it works inside SINTEZ without JAX. On machines held out of training the median error in χ is 9% and in the critical gradient 2.3%; one call is 2.3–5.0 times faster than QLKNN.
2. For the same ITER regime, SINTEZ with SHRX gives $P_{fus} = 492$ MW; SINTEZ with QLKNN gives 478 MW, TORAX 532 MW, and the earlier SHRX v0 67 MW. The network error (3.1%) is smaller than the code difference (−10.2%).
3. On the experimental database, SINTEZ with SHRX and TORAX are close to each other ($r = 0.995$ in $\ln W$, $\sigma_{\ln} = 0.18$) and far from experiment; both are weaker than the IPB98(y,2) regression ($\sigma_{\ln}$ of 0.55 and 0.56 against 0.21). The main cause of the error is the predictive pedestal and the inputs shared by the codes: 67% of the records were not valid in SINTEZ.
4. The ensemble reduces the error of the network outside its training range (from 8.7–8.9% to 5.9–6.5%) and the threshold error, and the spread between members separates the errors (AUROC 0.73–0.89). In the integrated result on stiff machines this effect is lost within the sensitivity of the simulator.
5. The speed-up of the simulator gives a bit-for-bit identical result: the deviation is exactly zero in 6 reference cases, which protects against the risk that a rounding-level change alters a run.
6. The present SHRX (QLKNN) overestimates the stored energy of spherical tokamaks (NSTX, MAST, START) by a median factor of 28. The second teacher from UKAEA, TGLFNN-STEP, was connected: its training range does not cover the present KTM states of SINTEZ,

and in the loop it underestimates the stored energy on the same records by a factor of 2.2. No network is delivered for KTM.

Declarations

Funding. This research received no external funding. Author to confirm or replace.

Competing interests. The author has no relevant financial or non-financial interests to disclose. Author to confirm.

Ethics approval and consent to participate. Not applicable. The study uses simulations and a public database and involves no human participants or animals.

Author contributions. The sole author conceived the study, developed the software, performed the simulations and analysis, and wrote the manuscript. Author to confirm.

Data availability. The SHRX weights (`backend/tokamak/shrx/weights.npz`, `ensemble.npz`) are in the project repository repository URL or DOI to be added. The selected records of the experimental database are in `docs/shrx/db5_selection/`; the source data are the open sample of the ITPA H-mode database distributed with FUSE.jl [19]. The TGLFNN-STEP weights are not included in the repository: they were published by UKAEA (LGPL) and are specified through the `TGLFNN_STEP_PKL` variable.

Code availability. Every number in the paper is taken by a macro from the files `paper_shrx/data/*.json`; the figures are built by `make_figures.py` and the text by `build.py`. The experiments are run by the scripts in `backend/ml/shrx/` and `backend/ml/shrx/db5/` (training: `train_shrx.py`; threshold: `threshold_eval.py`; ensemble: `make_ensemble.py`, `uncertainty_eval.py`; comparison with experiment: `db5/sintez_runner.py`, `db5/torax_runner.py`, `db5/analyse.py`, `db5/core_analyse.py`; spherical tokamaks: `db5/st_frozen.py`, `st/collect_tglf_states.py`). The tests are `backend/tests/test_shrx.py`, `test_solver_regression.py` and `test_speed_equivalence.py`.

Appendix A SHRX hyperparameters and reproduction

Table 12 SHRX network and training parameters

Parameter	Value
Input	$X \in \mathbb{R}^{10}$ $(R/L_{T_i}, R/L_{T_e}, R/L_{n_e}, R/L_{n_i}, q, \hat{s}, x, T_i/T_e, \log_{10}\nu^*, n_i/n_e)$
Output	(q_i, q_e, Γ_e), gyro-Bohm units
Architecture	4 hidden layers × 96 neurons, ReLU
Parameters	29,283 (QLKNN: 73,823)
Training rows	2,131,259
Machines held out	NSTX, DEMO
Optimiser	Adam, initial step $2 \cdot 10^{-3}$, cosine decay
Batch / epochs	4096 / 100
Ensemble	5 networks, different seeds, 85% sample of rows; $\tau = 0.20$
Teacher	QLKNN `qlknn_7_11_v1` (Apache-2.0), through TORAX 1.4.3

Software environment. NumPy [21] and JAX [22] were used: Python 3.11.17, NumPy 2.4.6; TORAX 1.4.3 (JAX 0.10.2, CPU). All calculations ran on one 4-core x86-64 virtual machine (16 GB of memory); timings were taken in a single thread.

References

1. J. Wesson, Tokamaks, 4th edn. (Oxford University Press, Oxford, 2011)
2. ITER Physics Expert Group on Confinement and Transport, ITER Physics Expert Group on Confinement Modelling and Database, ITER Physics Basis Editors, ITER Physics Basis. Chapter 2: Plasma confinement and transport. Nucl. Fusion 39, 2175 (1999). https://doi.org/10.1088/0029-5515/39/12/302
3. C. Bourdelle, J. Citrin, B. Baiocchi et al., Plasma Phys. Control. Fusion 58, 014036 (2016). https://doi.org/10.1088/0741-3335/58/1/014036
4. G.M. Staebler, J.E. Kinsey, R.E. Waltz, Phys. Plasmas 14, 055909 (2007). https://doi.org/10.1063/1.2436852
5. K.L. van de Plassche, J. Citrin, C. Bourdelle et al., Phys. Plasmas 27, 022310 (2020). https://doi.org/10.1063/1.5134126; arXiv:1911.05617
6. O. Meneghini et al., Nucl. Fusion 57, 086034 (2017). https://doi.org/10.1088/1741-4326/aa7776
7. L. Zanisi, F. Casson, A. Snodin et al., Data-efficient digital twinning strategies and surrogate models of quasilinear turbulence in JET and STEP, in IAEA Fusion Energy Conference (FEC 2025), Chengdu, China, 13–18 October 2025. https://conferences.iaea.org/event/392/contributions/36059/
8. J. Citrin, I. Goodfellow, A. Raju et al., TORAX: a fast and differentiable tokamak transport simulator in JAX. arXiv:2406.06718 (2024)
9. O. Sauter, C. Angioni, Y.R. Lin-Liu, Phys. Plasmas 6, 2834 (1999). https://doi.org/10.1063/1.873240; erratum: Phys. Plasmas 9, 5140 (2002)
10. P.B. Snyder et al., Nucl. Fusion 51, 103016 (2011). https://doi.org/10.1088/0029-5515/51/10/103016
11. J.W. Connor, J.B. Taylor, Nucl. Fusion 17, 1047 (1977). https://doi.org/10.1088/0029-5515/17/5/015
12. D.P. Kingma, J.L. Ba, Adam: a method for stochastic optimization, in International Conference on Learning Representations (ICLR 2015). arXiv:1412.6980
13. K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on ImageNet classification, in Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2015), pp. 1026–1034. https://doi.org/10.1109/ICCV.2015.123
14. G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network. arXiv:1503.02531 (2015)
15. S. Ross, G. Gordon, D. Bagnell, A reduction of imitation learning and structured prediction to no-regret online learning, in Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS), PMLR 15 (2011), pp. 627–635
16. B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, in Advances in Neural Information Processing Systems 30 (NeurIPS 2017) (2017), pp. 6402–6413
17. T. Fawcett, Pattern Recognit. Lett. 27(8), 861 (2006). https://doi.org/10.1016/j.patrec.2005.10.010
18. E.N. Lorenz, J. Atmos. Sci. 20, 130 (1963). https://doi.org/10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2

19. G. Verdoolaege, S.M. Kaye, C. Angioni et al., Nucl. Fusion 61, 076006 (2021). https://doi.org/10.1088/1741-4326/abdb91. The open sample used in this work is distributed with FUSE.jl. https://github.com/ProjectTorreyPines/FUSE.jl (file sample/HDB5_compressed.csv, accessed 19 October 2026). The database is also archived at https://doi.org/10.11578/1814950
20. UKAEA, tglfnn-ukaea: neural network surrogates of TGLF. https://github.com/ukaea/tglfnn-ukaea (accessed 19 October 2026)
21. C.R. Harris, K.J. Millman, S.J. van der Walt et al., Nature 585, 357 (2020). https://doi.org/10.1038/s41586-020-2649-2
22. J. Bradbury, R. Frostig, P. Hawkins et al., JAX: composable transformations of Python+NumPy programs (2018). https://github.com/jax-ml/jax (accessed 19 October 2026)