High-accuracy transport model surrogates via deep kernel learning in Gaussian process regression

M. Honda[1], S. Maeyama[2], and E. Narita[1,2]

[1]*Graduate School of Engineering, Kyoto University, Kyoto 615-8530, Japan*

[2]*National Institute for Fusion Science, Toki, Gifu 509-5292, Japan*

Introduction

Integrated transport codes solve a set of transport equations to predict the kinetic profiles and assess operation scenarios in tokamaks such as JT-60SA. The fidelity of such predictions hinges on the turbulent transport model used to supply the heat and particle fluxes. Advanced quasi-linear models such as TGLF [1] provide these fluxes, but they are computationally expensive and can dominate the cost of a full scenario calculation when called inside an iterative solver. Our steady-state solver GOTRESS [2] directly determines the kinetic profiles consistent with the transport fluxes and heating sources. It has recently been extended in two respects: electron particle transport now allows temperature and density to be solved together (*density-inclusive* simulations), and a local Levenberg–Marquardt (LM) optimizer has been added alongside the global genetic algorithm/Nelder–Mead (GA/NM) search.

For a JT-60U discharge (shot 39117), using electromagnetic TGLF with the SAT2 saturation rule, the LM method converges to essentially the same solution as the global GA/NM search while requiring far fewer model evaluations: for the heat-transport-only problem it converges about 64$\times$ faster (1,097 s versus 69,901 s). Solving temperature and density together is harder to converge, raising the cost to 4,489 s. This bottleneck motivates the use of a fast surrogate.

The LM acceleration also changes the *data regime*. Conventional neural-network (NN) surrogates for TGLF [2] are trained on very large datasets, of order 10^6 samples, which a broad global search such as GA/NM naturally generates. Because LM stays close to the solution trajectory, it samples only a small, localized region of input space, so those broadly distributed points are no longer available. This motivates the central problem addressed in this work: constructing a surrogate that performs well from *limited, localized* data.

Deep kernel learning with Gaussian processes

Gaussian process regression (GPR) is a nonparametric Bayesian method that performs well in small-data regimes and provides calibrated uncertainty, making it a natural starting point for a low-data surrogate; its performance depends strongly on the kernel, which acts as a prior over functions. A connection between deep NNs and GPs follows from the central limit theorem: in

the infinite-width limit a fully connected NN becomes a GP, and closed-form "deep kernels" for error-function and ReLU activations propagate weight and bias variances layer by layer [3], giving GPR deep-network expressivity while retaining probabilistic interpretability.

Even with such kernels, a GP defined directly on the 23-dimensional reduced input space used for the TGLF surrogate can be insufficient, because Euclidean distance in this space does not necessarily reflect similarity in the TGLF flux response. We therefore adopt Deep Kernel Learning (DKL) [4, 5], in which a neural-network feature extractor $g(\mathbf{x};\mathbf{w})$ (a small multilayer perceptron, here $23 \to 32 \to 16$) is placed in front of the base kernel,

$$k_{\mathrm{DKL}}(\mathbf{x},\mathbf{x}') = k_{\mathrm{base}}\big(g(\mathbf{x};\mathbf{w}), g(\mathbf{x}';\mathbf{w});\theta\big)\,, \tag{1}$$

so that the kernel acts on a learned feature space rather than on the raw inputs. Crucially, the network weights $\mathbf{w}$ and the kernel hyperparameters θ are optimized jointly by maximizing the marginal likelihood or its variational lower bound, so that the network learns a task-relevant representation while the GP supplies uncertainty on top of it.

We implement all of the above in dgpr [6], a custom GP framework in Python/JAX with automatic differentiation and GPU acceleration. It supports exact GPR and the Stochastic Variational Gaussian Process (SVGP) [7], composite kernels, the Intrinsic Coregionalization Model (ICM) for multi-output prediction, and the deep kernels above, and agrees well with GPy and GPflow in benchmarks. The DKL feature map is a JAX/Flax MLP regularized by weight decay and a spectral normalization. For scalability, SVGP represents the GP posterior using M inducing points for N training samples, reducing the training cost from the prohibitive $\mathcal{O}(N^3)$ of exact GPR to approximately $\mathcal{O}(NM^2 + M^3)$, with $M \ll N$.

Low-data DKL surrogate for density-inclusive transport

Training data are generated directly from the density-inclusive LM optimization runs of GOTRESS, which solve temperature and density together. Each sample has 23 input dimensions and 4 output fluxes: electron, ion, and impurity heat fluxes (Q_e, Q_i, Q_I) and the electron particle flux (Γ_e). After five LM iterations and de-duplication to five significant figures, only about 4,800 samples remain, which is small for a 23-dimensional, multi-output regression problem. On these data we train a DKL surrogate whose feature extractor is a $23 \to 32 \to 16$ MLP (Swish activations), with an SVGP on the 16-dimensional features.

Judged by the regression metric alone the surrogate looks excellent: the DKL model gives per-channel $R^2 = (0.9625, 0.9957, 0.9987, 0.9982)$ for $(\Gamma_e, Q_e, Q_i, Q_I)$. A high global R^2, however, does not guarantee that GOTRESS converges when the surrogate replaces TGLF. Near the magnetic axis the transport fluxes are tiny and span a very wide dynamic range, and there

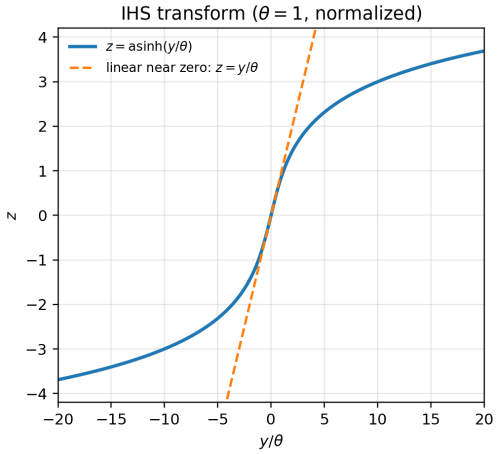

Figure 1: Inverse-hyperbolic-sine (IHS) transform $z = \mathrm{asinh}(y/\theta)$ applied to the small core fluxes. The transform is linear near zero and logarithmic for large $|y|$, preserving resolution for very small fluxes while avoiding the artificial near-zero resolution floor introduced by linear output scaling.

the surrogate is not accurate enough; the average fit hides a local failure in the core, and the density-inclusive solve fails to converge.

Inner–global hybrid surrogate

To resolve the tiny core fluxes without distorting the high-flux outer region, we introduce an *inner–global hybrid* surrogate: the global DKL model spans the full radial range, while a dedicated *inner* model is trained for the core, and at run time GOTRESS uses the inner model for $\rho \leq 0.3$ and the global model further out. Three ingredients make the inner model effective: the core is oversampled; an inverse-hyperbolic-sine (IHS) transform $z = \mathrm{asinh}(y/\theta)$ is applied to the small flux outputs (Fig. 1), being linear near zero and logarithmic for large arguments so that it spans their many-orders-of-magnitude dynamic range; and the core data are enriched near the LM trajectory with gradient perturbations (about 180 added samples). Crucially, the IHS transform also avoids the artificial near-zero resolution floor produced by linear output scaling, preserving the local flux–gradient sensitivity needed for LM convergence. Together these restore the per-channel core-flux reproducibility and, decisively, the convergence of the density-inclusive solve.

Results and summary

Putting these elements together, the hybrid DKL surrogate reproduces the full TGLF reference solution of the density-inclusive GOTRESS simulation, namely the T_e, T_i, and n_e profiles in Fig. 2, and does so more accurately than a single, monolithic NN surrogate built without the

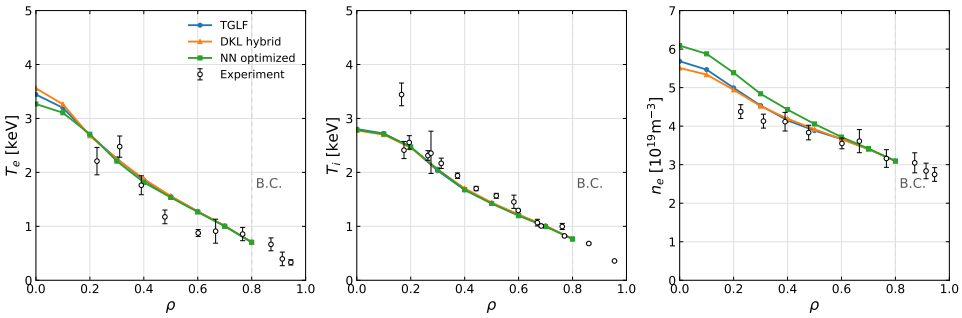

Figure 2: T_e, T_i, and n_e profiles for JT-60U #39117 from density-inclusive GOTRESS. The hybrid DKL surrogate (orange) reproduces the TGLF reference (blue) more closely than a monolithic NN surrogate (green); open circles are experimental data, and the dashed line marks the boundary condition (B.C.).

inner/global split. The cost breaks down as data generation (1,644 s), GPU training of the global and inner models ($458 + 485$ s), and the surrogate GOTRESS run (4.2 s), giving an end-to-end total of 2,591 s compared with 4,489 s for the direct TGLF calculation. This corresponds to a speedup of almost twofold, even when the data-generation and training costs are included.

In summary, the LM acceleration changes the data regime to a low-data one in which the conventional large-data NN approach is no longer suitable. Deep kernel learning, which couples a neural network to a Gaussian process and trains them jointly, provides an accurate surrogate from these limited data. In addition, an inner–global hybrid with an IHS-transformed core model recovers the tiny core fluxes that a high global R^2 alone misses, offering a practical route to reference-quality, density-inclusive transport simulation more than three times faster than direct GOTRESS with TGLF.

References

[1] G. M. Staebler, J. E. Kinsey, and R. E. Waltz. *Phys. Plasmas*, 14:055909, 2007.

[2] M. Honda and E. Narita. *Phys. Plasmas*, 26:102307, 2019.

[3] G. Pang, L. Yang, and G. E. Karniadakis. *J. Comput. Phys.*, 384:270–288, 2019.

[4] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Deep Kernel Learning, 2015. arXiv:1511.02222.

[5] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Stochastic Variational Deep Kernel Learning, 2016. arXiv:1611.00336.

[6] M. Honda, S. Maeyama, and E. Narita. *Phys. Plasmas*, 32:103906, 2025.

[7] J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian Processes for Big Data, 2013. arXiv:1309.6835.