Development of a Surrogate Turbulent Transport Model and Its Usefulness in Transport Simulations*)

Mitsuru HONDA and Emi NARITA

National Institutes for Quantum and Radiological Science and Technology, Naka 311-0193, Japan

(Received 2 November 2020 / Accepted 15 November 2020)

For accelerating a transport simulation with an advanced physics turbulent transport model like TGLF, we have been developing a surrogate model that mimics the behavior of the model based on a neural network model. With a steady-state transport solver GOTRESS used, the surrogate model has shown its ability to successfully predict temperature profiles almost equivalent to those by TGLF. The performance of the surrogate model is improved by optimizing hyperparameters and eliminating outliers from training data. Extrapolability of the optimized model is examined by changing the normalized temperature gradient. The objective is to better investigate the nature of the model in addition to measuring its utility in transport simulations. The versatile model, which has been trained with data of multiple cases, is developed applicable to many situations. It shows the same reproducibility as the model specific to each individual case, a fact which unveils great potential of the surrogate model in transport simulations.

Keywords: global optimization, neural network model, surrogate model, hyperparameter optimization, turbulent transport model, transport simulation, tokamak

DOI: 10.1585/pfr.16.2403002

1. Introduction

Integrated transport simulations are indispensable for predicting the confinement performance of plasmas such as JT-60SA and ITER. For this purpose, we have been developing a steady state transport solver called GOTRESS [1, 2]. GOTRESS has a number of features not found in other conventional transport codes, the most salient of which is that it employs global optimization techniques, such as a genetic algorithm and the Nelder Mead method, to solve the steady-state transport equations. GOTRESS attempts to directly find out a set of the temperature T_s and its normalized gradient $1/L_{T_s} \equiv -(dT_s/d\rho)/T_s$ for species s that satisfies the transport equation at each grid point. Here, ρ means the normalized minor radius. In that sense, it is not necessary to differentiate a T_s profile by some numerical method to obtain $1/L_{T_s}$, unlike other transport codes. This feature is suitable for dealing with a stiff transport model where the output fluxes are severely dependent on the gradients of density, temperature and so on. The most advanced turbulent transport models from a physical point of view are the models classified as a stiff transport model and one of them is the TGLF model [3, 4]. Due to the model's complexity, TGLF is numerically costly and it is usually run in parallel. GOTRESS itself is parallelized by MPI as well. Therefore, GOTRESS with TGLF is typically run on a supercomputer with more than two thousands CPUs used to obtain steady state temperature profiles [2]. Since the use of supercomputers is ascribed to the numerical heaviness of TGLF, it is natural to build a surrogate model that mimics TGLF to achieve faster calculation. If a fast surrogate model with high reproducibility can be constructed, GOTRESS with the model can be run on handy computer clusters with a smaller number of CPUs than a supercomputer and many trials can be easily performed.

Due to the feature of GOTRESS that uses a genetic algorithm, GOTRESS has a disadvantage that it takes relatively a long time to compute, while it has an advantage to be able to generate a large amount of data. This advantage goes well with deep learning and thus the development of a neural-network based surrogate model. In the previous work, we have exhibited the methodology to build a surrogate model of TGLF using GOTRESS and then to improve the model by hyperparameter optimization [2]. In this paper, we will investigate the details of the properties of the surrogate model and proceed with the development of the model that has versatility applicable to various transport simulations.

The rest of this paper is organized as follows. After briefly revisiting the development of a surrogate model based on the neural network model in section 2, we will discuss improvement methods of the developed model in section 3. The extrapolability of the surrogate model is investigated in section 4. The development of the model that has versatility applicable to various cases is described in section 5, followed by summary.

author's e-mail: honda.mitsuru@qst.go.jp

*) This article is based on the presentation at the 29th International Toki Conference on Plasma and Fusion Research (ITC29).

2. Neural-Network Based Surrogate Model

The idea to speed up evaluation of a transport model by building a surrogate model of the transport model based on a neural network model is not new in itself. Attempts have been made to create a model that mimics the behavior of a reduced transport model [5–7], which is numerically heavy enough as compared to conventional transport models despite the name of a "reduced" model, and a model that uses the module simulating the results of gyrokinetic simulations that require huge computational resources [8, 9]. Prediction of diffusion coefficients or fluxes calculated by a transport model is a regression problem in the context of machine learning. A neural network model constructed for that purpose is basically a fully connected (FC) feed forward model, which consists of an input layer, an output layer and hidden layers inbetween. The number of units, or sometimes called neurons, in the input layer corresponds to that of inputs of a model and the number of units in the output layer also corresponds to that of outputs. Each hidden layer has the arbitrary number of units, strongly correlated with expressivity of the model. With regard to a FC model, each input unit is connected to each unit in the next hidden layer. Each unit in this hidden layer is also connected to each unit in the next hidden layer or the output layer, depending upon the structure of the neural network model. Therefore, increasing the number of hidden layers or increasing the number of units in the hidden layer will explode the number of connections and lead to the increase in expressivity, or sometimes called capacity, of the model. The higher the number of connections, however, the better the performance of the model will not necessarily be. If we set up a model that is too complex relative to the amount of training data, the model will not be able to learn sufficiently and be overfitted, whereas if we set up a model that is too simple, the model will have less expressivity and be underfitted. There exists the optimal number of connections associated with the amount of training data in question.

One of the methods to reduce overfitting is to introduce the dropout rate, an element that makes up hyperparameters. The dropout refers to randomly dropping out units during the training process of a neural network and the dropout rate means the ratio of units that are dropped out to all units. Introducing the dropout adequately also makes the model more robust to unknown data that have not been used for training. The appropriate dropout rate of the model is likely to depend upon the amount of training data, and thus there also exists an optimal value.

Our previous work [2] distinguished itself from the previous research that not only developed was the neural network model that mimics the behavior of a transport model highly accurately but also optimized were hyperparameters of the developed model for further improvement. In the next section, after revisiting the methodology of hyperparameter optimization, we will consider how to improve the model other than optimization.

3. Improvement of Reproducibility of the Model

3.1 Hyperparameter optimization

The first thing that comes to mind as a way to improve the neural-network based surrogate model is hyperparameter optimization. With regard to the model trained with the data obtained by the simulation of GOTRESS with TGLF for JT-60U #39117 discharge, which is an H-mode plasma, we have sought an optimized set of hyperparameters of the model, which had been given manually relying on our experience and intuition [2]. There are lots of techniques to find out a set of hyperparameters giving a better performance, such as a grid search technique, a random search technique, a genetic algorithm and so on and so forth. For example, the traditional grid search technique is a method in which a set of pre-determined parameter candidates are tested on after another to find the best one. However, the fact that one trial of the grid search technique is equivalent to one training session for an entire neural-network model means that such exhaustive search techniques require huge amounts of hours and makes fine tuning of the model infeasible. The technique we have chosen is therefore a Sequential Model Based Optimization (SMBO) with Tree-structured Parzen Estimator (TPE) [10], which is a formalism of Bayesian optimization. In this algorithm, we first develop a probabilistic surrogate model approximating the loss of the original model, based on a small number of trials, i.e., entire processes of neural-network model training. By constructing a probabilistic surrogate model, the computational cost that would have been required to evaluate the model can be significantly reduced. This procedure is called SMBO. Expected Improvement (EI) as a measure of the amount of improvement of the model originally maximizes the conditional probability, but it is converted to maximize likelihood by Bayes' theorem. It is TPE that gives the expression of this likelihood. Details are consulted with the original paper [10]. In practice, hyperparameter optimization was performed using Hyperopt [10, 11], an implementation of SMBO with TPE, and its Keras wrapper, Hyperas [12] on the TensorFlow framework [13].

It takes about 8 hours for 100 optimization trials. The best model obtained in 100 trials gives the Mean Squared Logarithmic Error (MSLE) loss 0.00740, which is less than half of the loss of 0.0166 in the original model. Even if it were stopped after 20 trials, the loss of the best model would be 0.00777, which is comparable to that of 100 trials, while the calculation time would be only less than 2 hr. Considering the trade-off between calculation time of the optimization process and performance of the optimized model, it may be possible to stop after about 20 trials. Since the detailed list of hyperparameters is sum-

marized in Table II of [2], we here give an interpretation of how these values were obtained by optimization. The number of units at each hidden layer increases about from 200 to 400 on average, while the optimization process did not choose to add a hidden layer. In other words, the number of hidden layers remained three. The increase in units implies that the expressivity, i.e., complexity, of the model corresponding to a huge amount of training data should be augmented. As a result, the number of trainable parameters becomes fourfold. The dropout rate decreases from 0.25 to about 0.1 on average, indicating that there is no need to increase generalization performance for unknown data because training data is already large enough. The batch size increases eightfold to thin out potentially bad influence of outliers that may be included in each batch on neural-network model training. The performance improvement of the optimized model over the original model can be quantified by looking at the coefficient of determination, R^2, of the electron and ion heat fluxes. R^2's are improved from 0.962 to 0.987 for electrons and from 0.897 to 0.942 for ions, respectively [2], demonstrating the effectiveness of hyperparameter optimization.

3.2 Eliminating outliers from the training data to improve the model

One of the findings in the process of hyperparameter optimization is that the outliers contained in the training data deteriorate the performance of the surrogate model. We note that in this paper the word "outlier" is not given a quantitative definition and refers to values that are far out of the average in a general sense. In the genetic algorithm, the range of input data is set in advance. Outliers of the output data, i.e., heat fluxes, tend to occur near the boundaries of that range. In our case, $-2 \leq 1/L_{T_s} \leq 40$ and $0.01 \leq T_s \leq 20$ for $s = \mathrm{e}, \mathrm{i}$. Due to the nature of the genetic algorithm, it would be desirable to allow a wide range of input values. It is likely, however, that values near the upper limit of $1/L_{T_s}$ are potentially outside the scope of the physics model in TGLF. Moreover, $T_\mathrm{i}/T_\mathrm{e}$ is one of the inputs to TGLF and GOTRESS chooses T_e and T_i independently within the defined range. The range of $T_\mathrm{i}/T_\mathrm{e}$ could therefore be $5 \times 10^{-4} \leq T_\mathrm{i}/T_\mathrm{e} \leq 2 \times 10^3$. Considering $T_\mathrm{i}/T_\mathrm{e} \sim O(1)$ in actual sense, this range of the input is too wide, as clearly seen in the following figure, and its extrema are again outside the scope of TGLF.

Kernel density plots of the inputs to TGLF are shown in Figs. 1 (a) and (b), which correspond to $1/L_{T_s}$ and $T_\mathrm{i}/T_\mathrm{e}$, respectively. Each distribution consists of 976,087 data points. Even though $1/L_{T_\mathrm{e}}$ and $1/L_{T_\mathrm{i}}$ are certainly widely distributed in Fig. 1 (a), $\mu_\mathrm{e} + (-)2\sigma_\mathrm{e} \simeq 11.4\,(1.68)$ and $\mu_\mathrm{i} + (-)2\sigma_\mathrm{i} \simeq 10.7\,(0.680)$, indicating that most of the data is localized below about 10. Note that these distributions are non-Gaussian and these values do not come from the simple sum or difference of the mean μ and the standard deviation σ. With regard to $T_\mathrm{i}/T_\mathrm{e}$ shown in Fig. 1 (b),

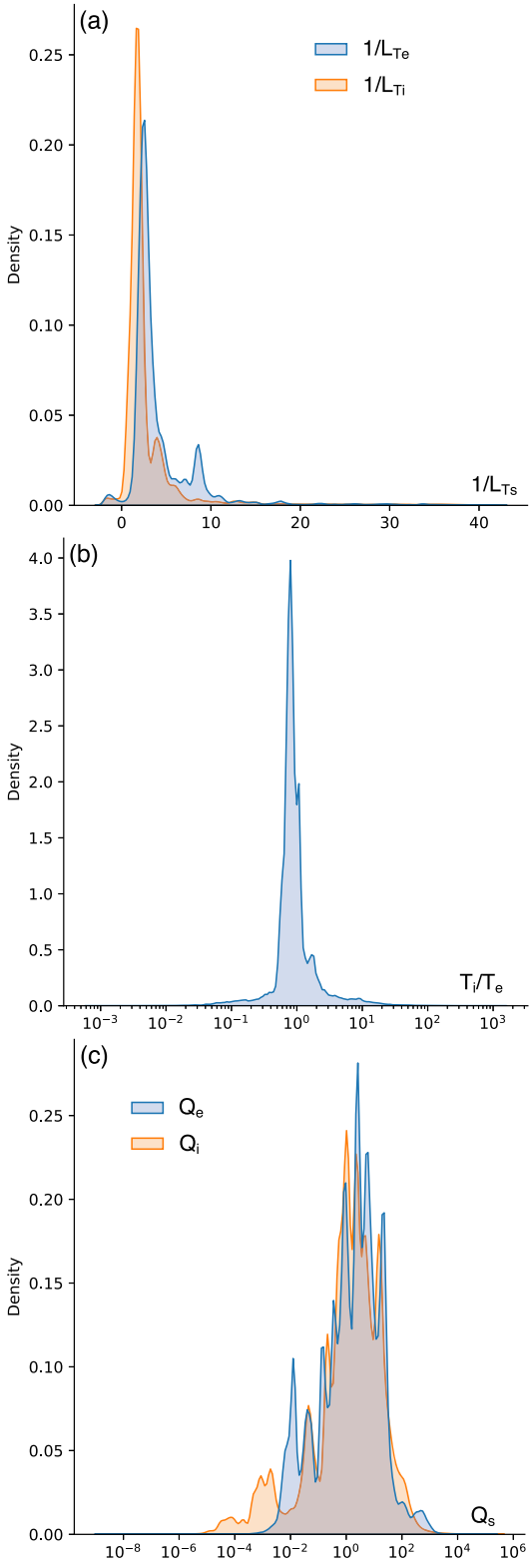

Fig. 1 Kernel density plots of the inputs to TGLF: (a) $1/L_{T_s}$ and (b) $T_\mathrm{i}/T_\mathrm{e}$, and the plot of the output from TGLF: (c) Q_s.

$\mu + (-)2\sigma \simeq 3.47\,(0.368)$, indicating that $T_\mathrm{i}/T_\mathrm{e}$ values are concentrated in a narrow range. Note that the input data has been garnered at all radial positions and thus $T_\mathrm{i}/T_\mathrm{e}$ values should have a spread to some extent and may have multiple peaks. Nonetheless, the small dispersion of $T_\mathrm{i}/T_\mathrm{e}$ means

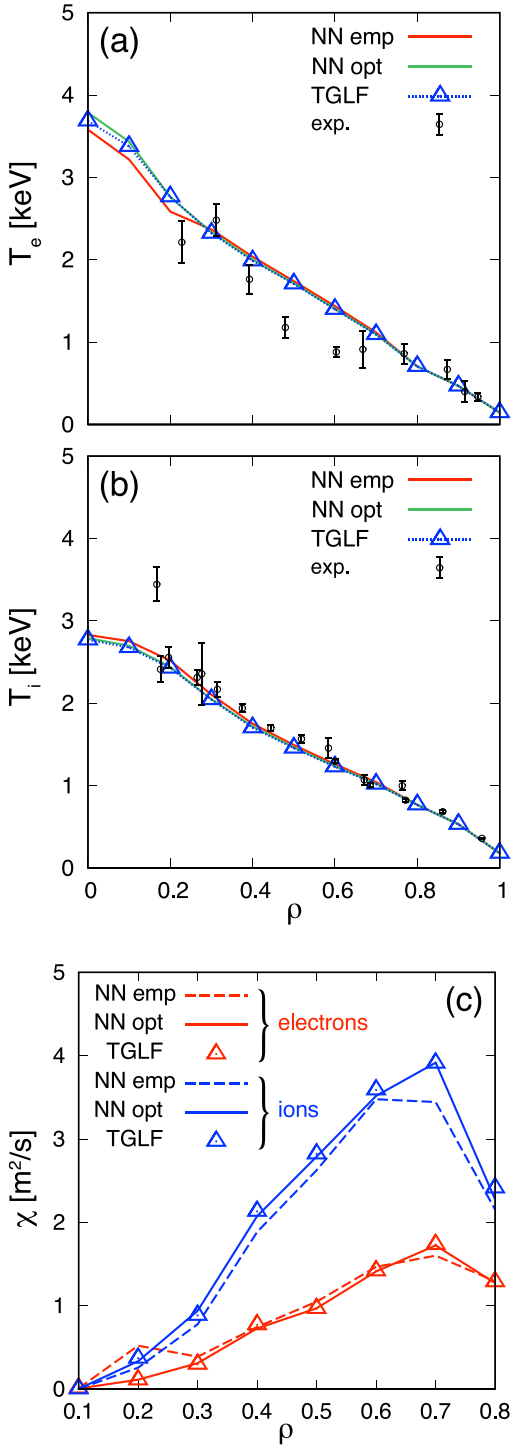

Fig. 2 Predicted temperature profiles of the JT-60U H-mode discharge #39117 for (a) electrons and (b) ions and (c) the heat diffusivity profiles using the narrow model with hyperparameters empirically chosen (NN emp), that with hyperparameters optimized (NN opt) and TGLF. Experimentally-measured points are also depicted in (a) and (b).

that GOTRESS has selectively generated realistic temperatures. Figure 1 (c) shows that a fairly wide range of flux outputs can be obtained from the relatively narrow input ranges, even though the realistic flux range is somewhat limited in actual. The deviation of the fluxes, to the larger ones in particular, brings about outliers and has an adverse effect on the training of the neural network model.

To exclude outliers of the fluxes, we generate the data by limiting the range of the input values as $0 \le 1/L_{T_s} \le 10$ and $0.5 \le T_s \le 5$, which are fed into another neural network model for training. Recalling that outliers come from inputs that are far from the solution, we can naturally reduce outliers of the output fluxes just by narrowing the range of inputs adequately. Hereafter, we call the original model "the wide model" and a newly-developed surrogate model with a narrower range of inputs "the narrow model". Although the number of training data is reduced from 976,087 used for the original model to 658,519, R^2's of the narrow model with hyperparameters empirically chosen improve significantly to 0.990 and 0.982 for electrons and ions, respectively, and those with hyperparameters optimized further improve to 0.999 and 0.997.

Shown are the results of GOTRESS simulations using the developed narrow models and TGLF in Fig. 2. The surrogate model with hyperparameters empirically chosen, denoted by "NN emp" in the figure, shows high reproducibility and, on top of that, that with hyperparameters optimized, denoted by "NN opt", shows almost exactly the same results as when using TGLF. Comparing the reproducibility of the wide model, as seen in Figs. 11 and 14 of [2], and the narrow model, it can be found that the latter one is obviously high. Hereafter, the narrow model will be used unless otherwise specified.

4. Extrapolability of the Surrogate Model

A neural-network based model is usually applied to problems within the range of data used for training of the model and is not used for extrapolating purpose. On the other hand, on purpose or inadvertently, the model is sometimes applied to the range outside of the training range when performing transport simulations. Extrapolation by the model may be possible if the trained model captures the essential (physical) trends of the original one hidden behind the data. However, it cannot be taken for granted without confirmation. It is therefore worth examining the extrapolability of the model developed. Before proceeding, we note that such a parameter survey conducted below is not necessary at all for the purpose to perform GOTRESS simulations with the surrogate model shown in the previous section.

A parameter survey was conducted with $-2 \le 1/L_{T_s} \le 40$ divided into 1,000 meshes. The $1/L_{T_i}$ is left fixed at the "nominal" value when the $1/L_{T_e}$ is changed, and vice versa. Here, the nominal value of the gradient corresponds to its steady-state solution of GOTRESS at $\rho = 0.5$. Single CPU calculations using the wide and narrow NN models took 0.674 second and 1.30 seconds, respectively, whereas TGLF with 21 CPUs used took 732 seconds. We

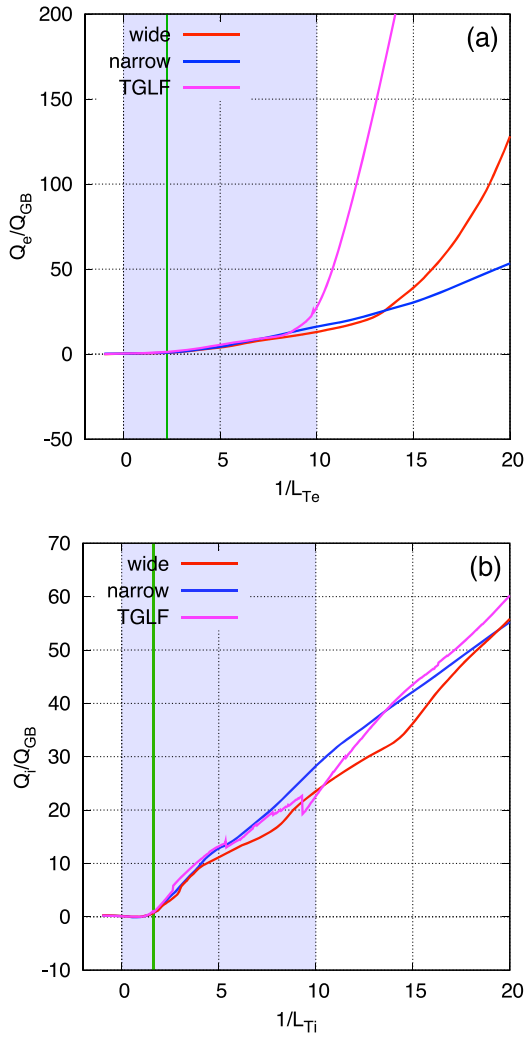

Fig. 3 (a) Dependence of the electron heat flux on $1/L_{T_e}$ and (b) that of the ion heat flux on $1/L_{T_i}$ at $\rho = 0.5$, predicted by the wide model, the narrow model and TGLF. The heat fluxes are normalized by the gyro-Bohm flux, Q_{GB}. The horizontal range of the blue hatched area corresponds to the range of the data used to train the narrow model. The green vertical line denotes the nominal value at $\rho = 0.5$.

now compare the dependence of the normalized electron and ion heat fluxes, Q_e/Q_{GB} and Q_i/Q_{GB}, on $1/L_{T_s}$, predicted by the three models, viz., the wide model, the narrow model and TGLF. Here, Q_{GB} is the gyro-Bohm flux. Figure 3 shows the dependence of Q_s/Q_{GB} on $1/L_{T_s}$ for each species s at $\rho = 0.5$. In the figure, the range of the horizontal axis is narrowed down to 20 for the sake of visibility. The horizontal range of the blue hatched area corresponds to the range of the data used to train the narrow model and the green vertical line denotes the nominal value. The reason that the position of the nominal value is plotted is that this surrogate model can well reproduce the TGLF behavior around the position because there are plenty of data generated around it. On the other hand, the more out of nominal value, the smaller the amount of data is, the less the surrogate model reproducibility is expected to be. Looking at the blue hatched area, we can see that both surrogate models reproduce the TGLF dependence almost perfectly. In contrast, in the region beyond the upper limit of the data used to train the narrow model, i.e., $1/L_{T_s} = 10$, both surrogate models significantly underpredict Q_e compared to the TGLF result, whereas they can capture the trend of the TGLF result for Q_i. It is natural that they are highly extrapolable to Q_i because Q_i essentially increases almost linearly and monotonically with respect to $1/L_{T_i}$ in this case. When we take a look at the dependence of Q_e on $1/L_{T_e}$ around $1/L_{T_e} \approx 10$, we find Q_e abruptly and nonlinearly increasing. This may be due to the change in the dominant instability that causes the heat flux predominantly. In order to confirm this speculation, the real frequency and the linear growth rate of the most unstable modes are investigated in the cases of the nominal $1/L_{T_e} \approx 2.24$ and $1/L_{T_e} = 12$. It reveals that the ion temperature gradient (ITG) mode is the most unstable mode in the former case, where the mixing length estimate of the electron heat diffusivity culminates at $k_\perp \rho_i \approx 0.2$, while the ITG/TEM hybrid mode predominates in the latter, where its peak is located around $k_\perp \rho_i \approx 1.4$. Here, TEM is the acronym of the trapped electron mode, $k_\perp$ is the poloidal wave number and ρ_i, the ion Larmor radius. The switch of the dominant mode by the increase in $1/L_{T_e}$ is confirmed. Of course, there is no way for the narrow model to know this change, and it is not surprising that it fails to reproduce the TGLF result. On the other hand, the wide model captures the tendency of nonlinear increase in Q_e, while it fails to reproduce the position where the increase is triggered and the amount of increase, even though it should know the data up to $1/L_{T_e} = 40$. Let us look at Fig. 1 (a) to understand the misprediction. Indeed the wide model has been trained with the data up to 40, most of the data resides within 10: There exists a small fraction of the data over 10. It is therefore concluded that the reason for this misprediction is that the wide model did not have enough data to capture the flux surge in the region above 10.

Figure 4 shows the dependence of Q_s/Q_{GB} on $1/L_{T_{s'}}$ for each species $s \neq s'$ at $\rho = 0.5$. Considering this figure with Fig. 3, it is seen that the flux is correctly predicted when $1/L_{T_i}$ is changed, but not when $1/L_{T_e}$ is changed, regardless of whether it is the electron or ion heat flux. The predictive performance of the heat flux is clearly independent of the particle species thereof. On the other hand, it is largely dependent on the particle species of the temperature to be changed, since the dominant instability may change with the change in the temperature gradient.

Also, we find in the figures that the flux predictions of TGLF are not continuous with the change in the gradients, and they sometimes change in a stepwise fashion. This reason is expected to be due to the difficulty in creating a model in which the results are continuous on a hyperplane consisting of all input parameters of TGLF. In contrast, the neural-network based surrogate models do not show such a tendency. The role of the neural network model as

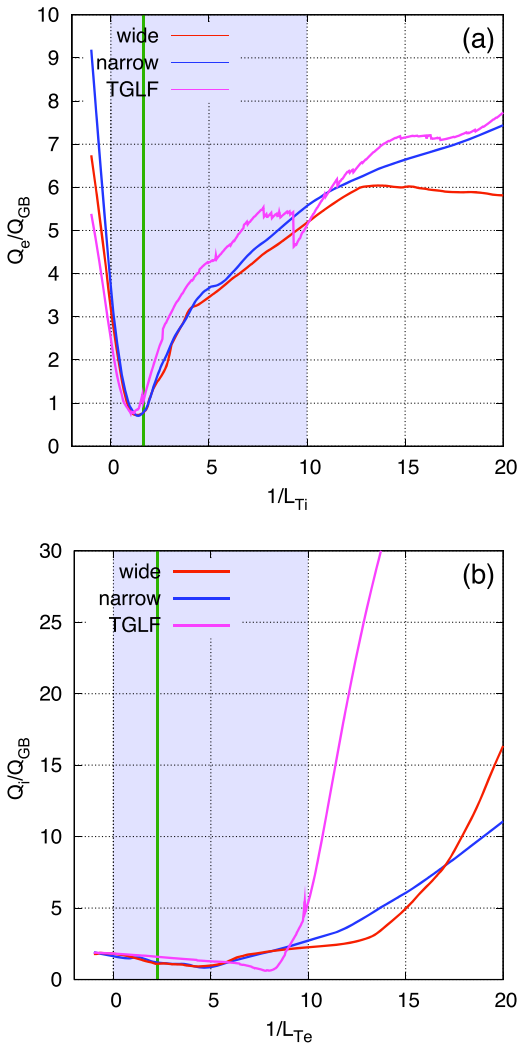

Fig. 4 (a) Dependence of Q_e/Q_{GB} on $1/L_{T_i}$ and (b) that of Q_i/Q_{GB} on $1/L_{T_e}$ at $\rho = 0.5$, predicted by the wide model (red line), the narrow model (blue) and TGLF (magenta).

a smoother in the hyperplane presumably contributes to a stable transport simulation when incorporated into a transport code [14].

5. Versatile Surrogate Model Trained with Data in the Multiple Cases

So far, we have built a surrogate model with very high reproducibility at the cost of specializing only in a specific case, viz., #39117 in this case, as schematically depicted in Fig. 5. Instead, this model is in general less versatile. When applied to other cases, the calculation results are often completely off the mark, rather than poorly reproducible. At worst, a convergent solution may not be obtained.

In order to make a surrogate model widely applicable to transport simulations, it is necessary for the model to have versatility while maintaining high reproducibility. Therefore, as depicted in Fig. 6, we create training data that

Fig. 5 Diagram showing the flow chart of building a surrogate model with the data for a specific case (JT-60U #39117) and performing prediction.

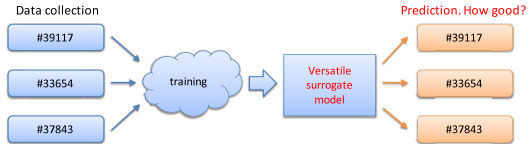

Fig. 6 Diagram showing the flow chart of building a versatile surrogate model with the data for multiple cases (JT-60U #39117, #33654 and #37843) and performing predictions for all cases.

mixes the data generated in multiple cases, viz., #39117, #33654 and #37843, and build a single surrogate model based on it. Note that these discharges are H-mode plasmas. This model, which is hereafter called a versatile model, is examined whether it is reproducible for each of the original cases. Note that the amount of data for training a versatile model is about three times as much as previous cases, of course. Therefore, the time required for training the surrogate model also increases. The optimization gives us a best set of hyperparameters, showing that the numbers of units of the hidden layers are 550, 700 and 600 and the dropout rates thereof are 0.00147, 0.0138, 0.154, respectively. It can be seen that as the amount of training data increases, so does the capacity of the optimized model. As mentioned earlier, the ranges of $1/L_{T_s}$ and T_s included in the training data are still narrow, but their upper limits are extended to 15 and 20, respectively. This is because the temperatures of #33654 and #37843 are higher than those of #39117 and the adequate ranges of inputs have to be wider accordingly. The number of outliers included in the data is likely to be small and therefore the optimized batch size results in at most 4,096, which is smaller than that found in section 3.1 albeit the increase in the data size.

Shown in Fig. 7 are the temperature profiles predicted by GOTRESS with the versatile surrogate model, the original one and TGLF, for three cases. The original surrogate model has been customized specific to each case. As is clear from the figure, in all cases, the sole versatile model shows the same predictive performance as the individually trained models. Furthermore, in #33654 and #37843 cases, it can be seen from the comparison with TGLF results that the predictive performance of the versatile model is even better than the original model. This is probably because the versatile model was trained with more data to better learn the behavior of TGLF.

This tendency of reproducibility improvement can be seen more clearly by carrying out a parameter survey of the

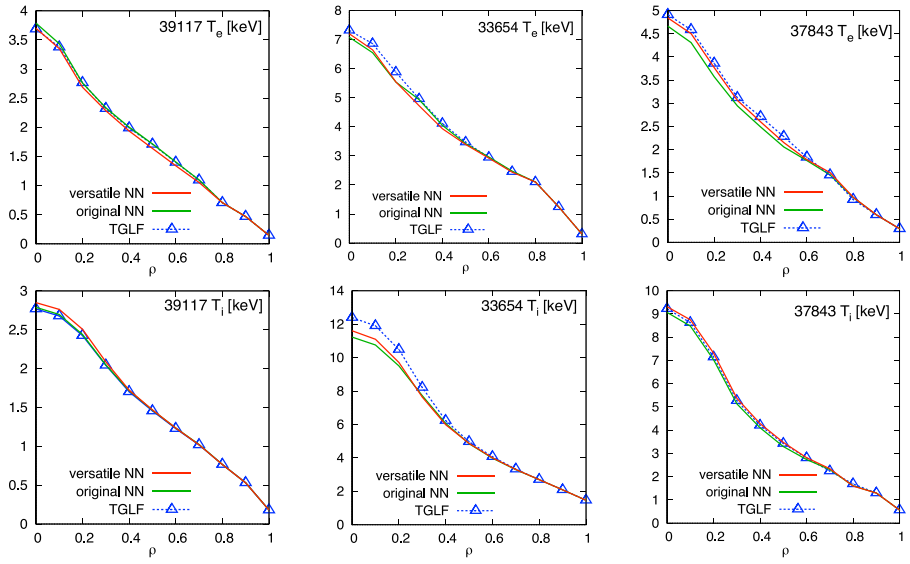

Fig. 7 Temperature profiles predicted with the versatile surrogate model (red line), the original surrogate model (green) and TGLF (blue dotted) for three cases.

models conducted in section 4 for the versatile model. The results of focusing on the $1/L_{T_e}$ dependence of the models are shown in Fig. 8. As is clear from the comparison with Fig. 3 (a) and Fig. 4 (b), the $1/L_{T_e}$ dependence of the versatile model is much closer to that of TGLF. It is true that the variable range of $1/L_{T_e}$ has been expanded to the upper limit of 20 by adding data other than those of #39117, but the training data specific to this #39117 has not increased at all to train the versatile model. This fact indicates that the increased diversity and total amount of data have made it possible for the surrogate model to better reproduce the behavior of TGLF for any case.

6. Summary and Future Work

We have used a deep learning technique to build a surrogate model for accelerating transport simulations which have been performed using a transport model based on advanced physics. It has been confirmed that optimizing hyperparameters of the model and removing outliers from training data are effective means of improving the performance of the model. The R^2's of the heat fluxes predicted by the model to which both improvement methods has been applied exceed 0.997 for the JT-60U #39117 H-mode plasma, showing very high reproducibility. The parameter survey of the surrogate model to examine its extrapolability revealed the following. Even outside the range of the training data, unless the dominant instability changes, the surrogate model shows reasonable extrapolability. On the other hand, the model cannot capture the tendency of TGLF, when the dominant instability changes outside the range of the training data, where the heat fluxes sharply increase in this case. However, we also found that training with the data of multiple cases to some extent mitigates a deficiency in extrapolability. This fact indicates

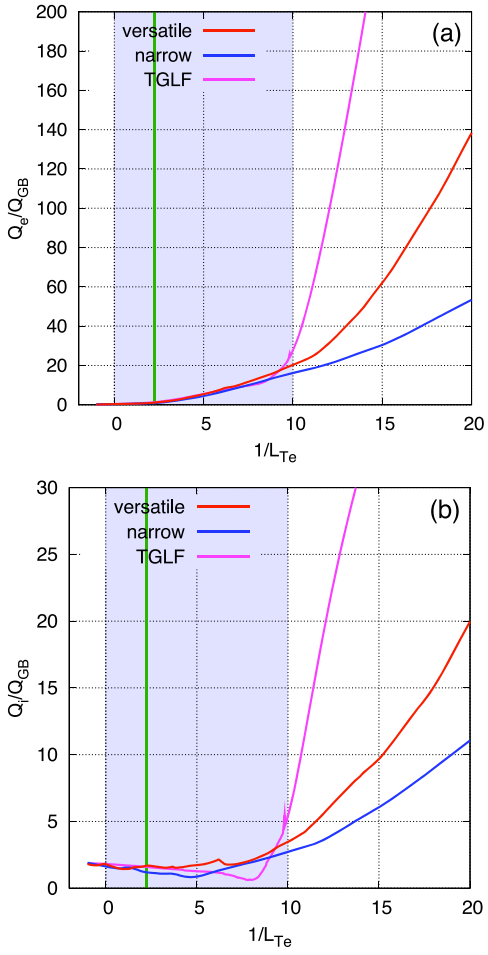

Fig. 8 (a) Dependence of (a) Q_e/Q_{GB} and (b) Q_i/Q_{GB} on $1/L_{T_e}$ at $\rho = 0.5$, predicted by the versatile model (red line), the narrow model (blue) and TGLF (magenta). The results of the narrow model and TGLF in Figs. (a) and (b) are identical to those shown in Fig. 3 (a) and Fig. 4 (b), respectively.

that even without further additional data on the case we are currently dealing with, the increased diversity and total amount of data has allowed the model to learn more about the behavior of TGLF, leading to the improvement of the reproducibility even on this case. The versatile surrogate model has the same high reproducibility as the model specific to each individual case. It can be said that it unveils great potential of a neural-network based surrogate model employed in transport simulations.

There are still challenges in further advancing the versatile model. This time training the model was conducted using the data of three cases. Then, let us imagine the case that the data of 100 cases is garnered. The total amount of data becomes enormous, and the time required for training increases tremendously. Of course, the capacity of the model must also increase as the amount of data increases. To make matters worse, when the data of one more case is additionally available and is fed into the model trained with 100 cases data, the model must forget all the information from the previous 100 cases and then results in being optimized for the newly added data only. This is called catastrophic forgetting or catastrophic inference. Several methods have been proposed to solve this problem (see e.g. [15–17]), but none have been conclusive yet. At the same time, techniques such as knowledge distillation [18] will also have to be considered to miniaturize the model that would have been huge for practical use.

Acknowledgements

They are grateful to Mr. T. Kuwata for numerical assistance. This work was mainly carried out using the JFRS-1 supercomputer system at Computational Simulation Centre of International Fusion Energy Research Centre (IFERC-CSC) in Rokkasho Fusion Institute of National Institutes for Quantum and Radiological Science and Technology (QST), Aomori, Japan. This work was partly supported by MEXT as "Program for Promoting Researches on the Supercomputer Fugaku" (Exploration of burning plasma confinement physics, hp200127), by JSPS KAKENHI Grant Number 17K07001 and by Institute for Quantum Life Science (iQLS), QST.

[1] M. Honda, Comput. Phys. Commun. **231**, 94 (2018).
[2] M. Honda and E. Narita, Phys. Plasmas **26**, 102307 (2019).
[3] G.M. Staebler, J.E. Kinsey and R.E. Waltz, Phys. Plasmas **12**, 102508 (2005).
[4] G.M. Staebler, J.E. Kinsey and R.E. Waltz, Phys. Plasmas **14**, 055909 (2007).
[5] O. Meneghini, C.J. Luna, S.P. Smith and L.L. Lao, Phys. Plasmas **21**, 060702 (2014).
[6] J. Citrin *et al.*, Nucl. Fusion **55**, 092001 (2015).
[7] O. Meneghini *et al.*, Nucl. Fusion **57**, 086034 (2017).
[8] E. Narita, M. Honda, M. Nakata, M. Yoshida, H. Takenaga and N. Hayashi, Plasma Phys. Control. Fusion **60**, 025027 (2018).
[9] E. Narita, M. Honda, M. Nataka, M. Yoshida, N. Hayashi and H. Takenaga, Nucl. Fusion **59**, 106018 (2019).
[10] J. Bergstra, R. Bardenet, Y. Bengio and B. Kégl, in Proceedings of the Advances in Neural Information Processing Systems 24 (NIPS 2011), Granada, Spain (2011), Vol.24.
[11] http://jaberg.github.io/hyperopt/
[12] http://maxpumperla.com/hyperas/
[13] M. Abadi *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems", preprint arXiv:1603.04467 (2016).
[14] K.L. van de Plassche *et al.*, Phys. Plasmas **27**, 022310 (2020).
[15] J. Kirkpatrick *et al.*, preprint arXiv:1612.00796 (2017).
[16] C. Atkinson *et al.*, "Pseudo-Recursal: Solving the Catastrophic Forgetting Problem in Deep Neural Networks", preprint arXiv:1802.03875 (2018).
[17] C. Atkinson *et al.*, "Pseudo-Rehearsal: Achieving Deep Reinforcement Learning without Catastrophic Forgetting", preprint arXiv:1812.02464 (2019).
[18] G. Hinton, O. Vinyals and J. Dean, "Distilling the Knowledge in a Neural Network", preprint arXiv:1503.02531 (2015).