Collaborative approaches between numerical plasma simulations and machine learning techniques

M. Honda[1], E. Narita[2], S. Maeyama[3] and T.-H. Watanabe[3]

[1] *Kyoto University, Kyoto, Kyoto 615-8530, Japan*

[2] *Kyoto University, Kyoto, Kyoto 615-8540, Japan*

[3] *Nagoya University, Nagoya, Aichi 464-8602, Japan*

The progress in understanding fusion plasmas and achieving quantitative predictions has been accomplished through numerical simulations. Despite improvements in numerical algorithms and computer performance over the years, the advancement of physics models tends to increase computation time, and there are still some numerically challenging problems to address. Deep learning techniques, fundamentally distinct from the conventional numerical schemes and algorithms used in physics problems, could assist numerical simulations.

We have been focusing on the large amount of data generated by a local flux-tube gyrokinetic simulation and developing a deep learning program that effectively utilizes this data. For instance, the program can be employed to optimize a set of input parameters for a gyrokinetic simulation, aiming to minimize calculation time. This optimization can be achieved if a model is developed to predict the time at which turbulent fluxes reach saturation using data from the early stages of the simulation. The time-series numerical data in 5D phase space produced by the gyrokinetic simulation is too extensive to handle in its original form. However, by visualizing the amplitude of perturbed distribution functions as $|\tilde{f}|^2$ and selecting specific combinations, such as $|\tilde{f}|^2(k_x, k_y)$ where k_x and k_y represent the radial and poloidal wavenumbers, respectively, we can significantly reduce the data volume processed in the program.

To handle these images, we employed a convolutional neural network (CNN) model. Using the images as input, we trained the model based on EfficientNet-B4 [1], which is a state-of-the-art CNN model. We utilized transfer learning and fine-tuning techniques during the training process. The model successfully predicted the time corresponding to the input image and also predicted the time to saturation [2].

The previous model was unable to predict the magnitude of turbulent heat fluxes accurately because the input images were normalized at each time step and did not include information about the absolute values of quantities necessary to predict fluxes. To predict the electron and ion turbulent heat fluxes, denoted as Q_e and Q_i, respectively, the model needs to incorporate numerical values related to the fluxes as input. In this regard, we selected the amplitude of the electrostatic potential, $|\tilde{\phi}|^2$, as input and developed a multimodal model capable of simultane-

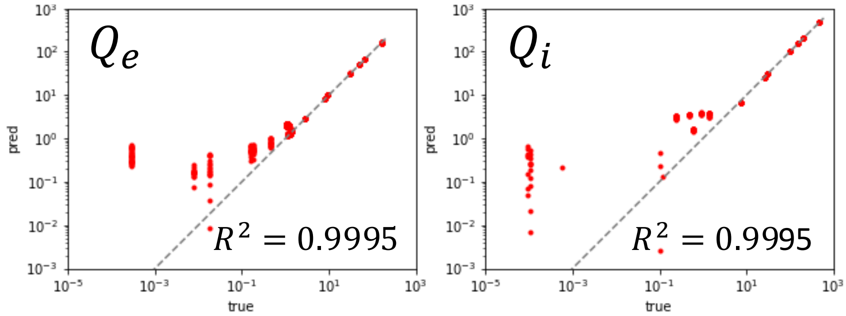

Figure 1: Regression plots of the predicted Q_e and Q_i versus the true values for the test data, along with the corresponding R^2 scores.

ously processing images and $|\tilde{\phi}|^2$. The image should contain sufficient information to independently predict Q_e and Q_i. We combined monochrome images of $|\tilde{f}_e|^2$, $|\tilde{f}_i|^2$, and $|\tilde{\phi}|^2$ to create a "pseudo" three-color-channel image, which served as input to the multimodal model. The developed model demonstrated very good performance in predicting Q_e, and Q_i for the test dataset [3], as shown in fig. 1. It also displayed some predictive ability for unknown datasets, although the extent of its accuracy varied depending on the cases. To enhance predictability further, we are currently exploring the use of different types of images and developing a new model that explicitly incorporates time-series information.

Modern state-of-the-art turbulent transport models, such as TGLF, can predict turbulent fluxes much faster than gyrokinetic codes, but they are still computationally intensive despite being reduced models. For instance, TGLF typically requires approximately $\sim 0.2\,$s for evaluation using 21 parallel CPU cores. On the other hand, a neural network (NN) model has the capability to mimic the behavior of the original model when provided with abundant and diverse training data. Such a NN-based surrogate model is significantly faster than the original model, depending on the complexity of the NN model. In our previous work, we successfully developed a surrogate model of TGLF tailored to a specific problem with high reproducibility. Now, we are aiming to create a more versatile model that can be applied across a wide range of scenarios. To achieve this, we generated an exhaustive dataset by varying 15 out of 23 input parameters, resulting in over 90 million data points. To enhance the quality of the training data, we employed the interquartile range (IQR) method to prune outliers. The resulting surrogate model, incorporated into the steady-state transport code GOTRESS [4], exhibited a computational speed approximately 100 times faster than the original TGLF model. Additionally, the GOTRESS simulations converged regardless of the number of radial grid points employed. This result shows the usefulness of the surrogate model in practical applications.

In the field of plasma transport simulation, it has long been recognized that one of the challenging issues is achieving stable transport simulations when employing a stiff turbulent transport model [4, 5, 6]. The term "stiff" refers to the sensitivity of turbulent fluxes to temperature and density gradients. Stiff transport models often exhibits spikes in diffusion coefficient profiles, rendering the transport simulations meaningless once these spikes happen. Unless special treatments proposed in [5, 6] are employed, the only solution to avoid numerical oscillations is to decrease the time step of the transport simulation, resulting in significantly increased computation time. To address this challenge and provide a new approach to handle stiff transport models, we devised an idea utilizing Physics Informed Neural Networks (PINNs) [7]. PINNs usually belong to the category of categorized as unsupervised learning, where patterns are learned from unlabeled data. By leveraging the powerful technique of automatic differentiation inherent in neural network models, a neural network can be trained to represent the behavior of dependent variables governed by specific equations. The training process aims to minimize losses associated with partial differential equations, as well as initial and boundary conditions, by bringing them as close to zero as possible. Remarkably, this approach does not require providing any data explicitly. To validate the effectiveness of the solution obtained using PINNs, a simple transport model exhibiting stiffness against temperature gradients similar to that used in Ref. [5] is prepared, expressed as

$$\chi(T') = \begin{cases} k(|T'| - |T'_c|)^{\alpha} + \chi_0 & \text{for } |T'| > |T'_c| \\ \chi_0 & \text{for } |T'| \leq |T'_c| \end{cases}, \tag{1}$$

where $\chi_0 = 0.1$, $\alpha = 1.5$, $k = 1$, and $T'_c = 0.5$. Subsequently, the diffusive transport equations for electrons and ions were solved, taking into account collisional equipartition processes between species. The initial electron temperature T_e was set to twice the ion temperature T_i to simulate the temperature equilibration through the equipartiton process without any auxiliary heatings. The solution derived from PINNs is the compared to the results obtained from the transport code TRESS in order to validate its accuracy and reliability. TRESS utilizes the finite element method with the cubic Hermite function as the basis function to discretize and solve the diffusive transport equations. In order to prevent the occurrence of numerical spikes in the diffusion coefficient profiles, TRESS typically requires a very small time step, such as $\Delta t = 10^{-6}$ s. Figure 2 demonstrates the comparison between TRESS and PINN, indicating that PINNs can handle the stiff transport model effectively, even with coarse grids in both space and time dimensions. The computation time with PINN was approximately half that of TRESS. The apparent ease with which PINNs handle stiff transport models is noteworthy. The training process of PINNs

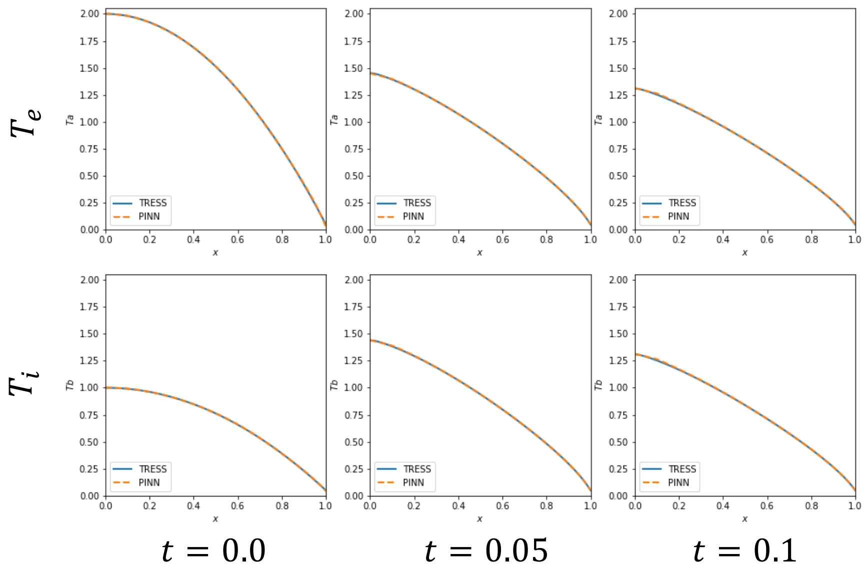

Figure 2: Comparison of the electron and ion temperature profiles at $t = 0.0, 0.05$, and $0.1\,$s between TRESS and PINN.

results in a neural network that represents the spatiotemporal solution governed by the diffusion equation. It is intriguing because it seems as if the solution is determined instantaneously without explicitly considering the causality inherent in the diffusion equation. While certain treatments are necessary to achieve stable solutions in PINNs, we believe it holds great promise as a novel method for dealing with stiff transport models in transport simulations.

References

[1] M. Tan and Q.V. Le, 2019. arXiv:1905.11946.

[2] E. Narita, M. Honda, S. Maeyama, and T.-H. Watanabe. *Nucl. Fusion*, 62:086037, 2022.

[3] M. Honda, E. Narita, S. Maeyama, and T.-H. Watanabe. *Contrib. Plasma Phys.*, page 23e2022001373, 2023.

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

[5] S.C. Jardin, G. Bateman, G.W. Hammett, and L.P. Ku. *J. Comput. Phys.*, 227:8769, 2008.

[6] G.V. Pereverzev and G. Corrigan. *Comput. Phys. Commun.*, 179:579, 2008.

[7] M. Raissi, P. Perdikaris, and G.E. Karniadakis. *J. Comput. Phys.*, 378:686, 2019.