

RESEARCH ARTICLE | JULY 07 2023

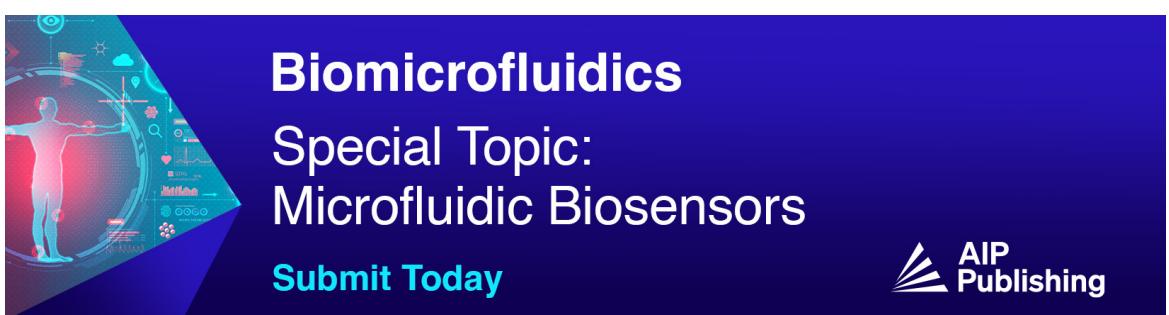
Fast equilibrium reconstruction by deep learning on EAST tokamak

Jingjing Lu  ; Youjun Hu   ; Nong Xiang  ; Youwen Sun 

 Check for updates

AIP Advances 13, 075007 (2023)

<https://doi.org/10.1063/5.0152318>



Biomicrofluidics
Special Topic:
Microfluidic Biosensors

Submit Today



Fast equilibrium reconstruction by deep learning on EAST tokamak

Cite as: AIP Advances 13, 075007 (2023); doi: 10.1063/5.0152318

Submitted: 29 March 2023 • Accepted: 17 June 2023 •

Published Online: 7 July 2023



View Online



Export Citation



CrossMark

Jingjing Lu,^{1,2} Youjun Hu,^{2,a} Nong Xiang,^{2,b} and Youwen Sun²

AFFILIATIONS

¹ University of Science and Technology of China, Hefei 230026, China

² Institute of Plasma Physics, Chinese Academy of Sciences, Hefei 230031, China

^aAuthor to whom correspondence should be addressed: yjhu@ipp.ac.cn

^bElectronic mail: nxiang@ipp.ac.cn

ABSTRACT

A deep neural network is developed and trained on magnetic measurements (input) and EFIT poloidal magnetic flux (output) on the EAST tokamak. In optimizing the network architecture, we use automatic optimization to search for the best hyperparameters, which helps in better model generalization. We compare the inner magnetic surfaces and last-closed-flux surfaces with those from EFIT. We also calculated the normalized internal inductance, which is completely determined by the poloidal magnetic flux and can further reflect the accuracy of the prediction. The time evolution of the internal inductance in full discharge is compared with that provided by EFIT. All of the comparisons show good agreement, demonstrating the accuracy of the machine learning model, which has high spatial resolution compared with the off-line EFIT while still meeting the time constraint of real-time control.

© 2023 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1063/5.0152318>

I. INTRODUCTION

Reconstructing the magnetic configuration using magnetic measurements is a routine task of tokamak operation. There are many equilibrium solvers, e.g., EFIT,^{1–6} that can perform this kind of reconstruction by solving the Grad-Shafranov equation under the constraint of magnetic measurements. In recent years, accumulation of data resulting from these reconstruction practices, along with the development of machine learning algorithms, software frameworks, and computing power, has made it possible to train deep neural networks to provide reconstruction as accurate as that by EFIT. This has been demonstrated on KSTAR⁷ and DIII-D.⁸

On the EAST tokamak,⁹ EFIT has been routinely used in tokamak operations for more than ten years, and substantial equilibrium data have been accumulated.^{10–13} In this paper, we report the results of magnetic reconstruction by a deep neural network trained on the magnetic measurements and EFIT reconstructed 2D magnetic poloidal flux.

There are two versions of EFIT used on EAST: one for real-time control and one for off-line analysis. The former is often restricted to lower accuracy due to the time constraint of real time control, while the latter is of higher accuracy. In this work, we use the off-line EFIT

data in training the neural network. The model trained this way has higher accuracy than the off-line EFIT while still meeting the time constraint of real-time control.

There are many hyperparameters in a neural network that usually need to be set manually, such as the number of hidden layers, units per layer, mini-batch size, learning rate, and number of epochs of training. In recent years, there appeared optimization libraries that can automatically set the values of these hyperparameters. In this work, we use the Optuna optimization framework¹⁴ in setting the hyperparameters. The hyperparameters found this way turn out to be much better than our previously manually set ones in terms of model accuracy. The size of the network architecture found by automatic hyperparameter tuning turns out to be relatively small (with less than 2×10^6 parameters). This small size allows for very fast equilibrium construction that can be easily deployed.

The input to the network is limited to only magnetic measurements. The output of the network is the 2D poloidal magnetic flux function, $\Psi(R, Z) \equiv A_\phi R$, which is related to the poloidal magnetic field, B_R and B_Z , by

$$B_R = -\frac{1}{R} \frac{\partial \Psi}{\partial Z}, \quad (1)$$

$$B_Z = \frac{1}{R} \frac{\partial \Psi}{\partial R}, \quad (2)$$

where (R, ϕ, Z) are the cylindrical coordinates. The 2D contours of Ψ in the (R, Z) plane correspond to the magnetic surfaces. We compare the inner magnetic surfaces and the last-closed-flux surfaces (LCFSs) with those given by EFIT in order to evaluate the accuracy of Ψ predicted by the network. We also calculate the normalized internal inductance, l_i , which is a quantity that is solely determined by Ψ and thus can reflect how accurate the predicted Ψ is. The time evolution of the internal inductance in full discharge is compared with that provided by EFIT. All of the comparisons show good agreement, demonstrating the accuracy of the machine learning model, which has high spatial resolution compared with

TABLE I. The inputs and outputs of the model.

Signal	Measure method	Signal meaning	Num. Of values
Input			
Ψ_{FL}	Flux loop	Poloidal magnetic flux	35
MF	Magnetic probe	Poloidal magnetic field	34
PF	Rogowski loop	Poloidal field coil current	14
I_p	Rogowski loop	Plasma current	1
Output			
$\Psi(R, Z)$	EFIT	Poloidal magnetic flux	16 641

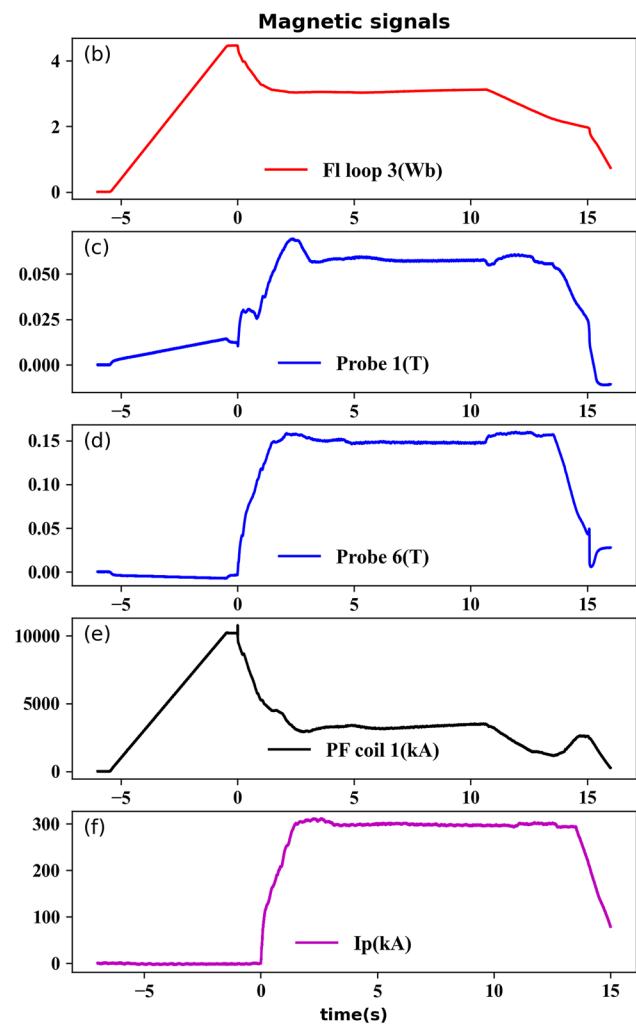
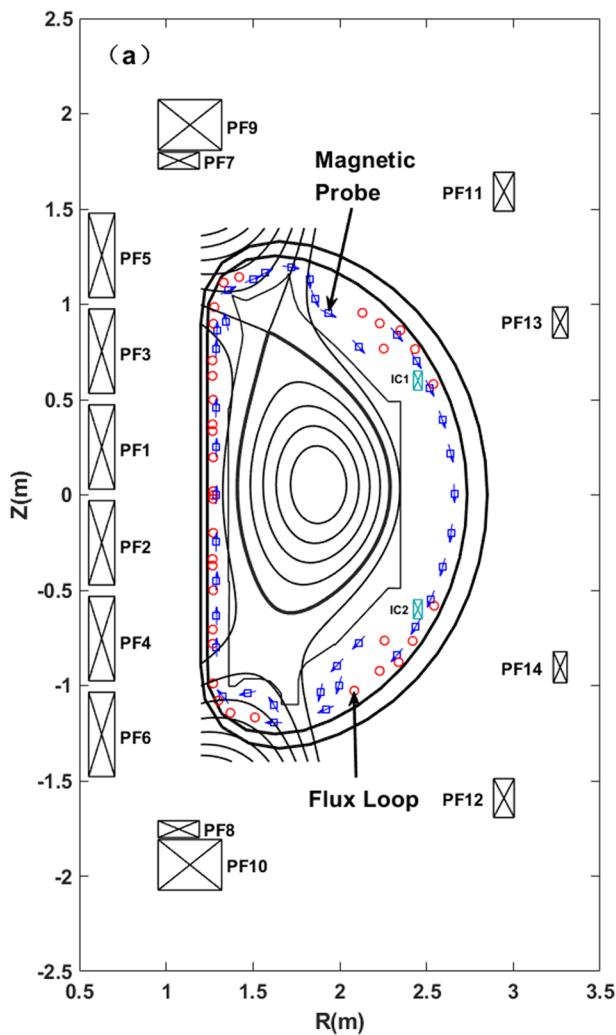


FIG. 1. (a) Locations of the magnetic probes, flux loops, and poloidal coils on EAST. A typical time evolution of some of the magnetic measurements from EAST discharge 113 019 are plotted in (b)–(f). There are a total of 38 magnetic probes measuring the equilibrium poloidal magnetic field, and only 34 of them are working and are used in this work.

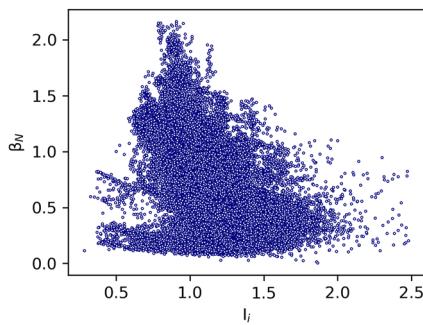


FIG. 2. Distribution of the EFIT equilibrium data (training set + validation set + testing set) in the (l_i, β_N) plane.

the off-line EFIT while still meeting the time constraint of real-time control.

The rest of this paper is organized as follows: Sec. II presents how the data are collected and normalized. Section III explains the structure of our neural network and how the hyperparameters are chosen by automatic optimization. In Sec. IV, we test the predicting capability of the trained network. Section V discusses a small network used to predict some volume-averaged quantities, namely, the plasma stored energy W_{mhd} , normalized toroidal beta β_N , and edge safety factor q_{95} . A brief summary is given in Sec. VI.

II. DATA COLLECTION AND NORMALIZATION

Figure 1(a) illustrates the poloidal locations of the magnetic measurements used as inputs in our model. A typical time evolution of some of the magnetic measurements from EAST discharge 113 019 is plotted in Figs. 1(b)–1(f).

The inputs (features) to the neural network (NN) are 84 magnetic measurements: 35 poloidal magnetic flux (Ψ_{FL}) values measured by flux loops, 34 equilibrium poloidal magnetic field (MF) values measured by magnetic probes, 14 poloidal field (PF) coil currents, and 1 plasma current (I_p) measured by a Rogowski loop.

The outputs (targets) of the neutral network are the values of the poloidal magnetic flux Ψ at $R \times Z = 129 \times 129 = 16\,641$ spatial locations. The Ψ used in the training process is computed by off-line EFIT and is downloaded from the EAST MDSplus server (mds.ipp.ac.cn). The input and the output signals are interpolated to the same time slices before they are fed to the NN.

The inputs and outputs are summarized in Table I.

The data used in training, validation, and testing processes were downloaded from the EAST MDSplus server by using Python, which scans a series of discharges and automatically skips discharges where necessary signals are missing. Specifically, we scan every five discharges among all the discharges spanning from #114 000 to #117 000, resulting in a total of 45 544 equilibria (time slices). These discharges are from experiments performed in one EAST campaign from June to July in 2022. This range is casually chosen with no particular criterion, except that we prefer recent discharges and avoid old discharges because locations of some magnetic probes were changed in previous campaigns. The auxiliary heating methods on EAST used in this campaign include neutral beam injection

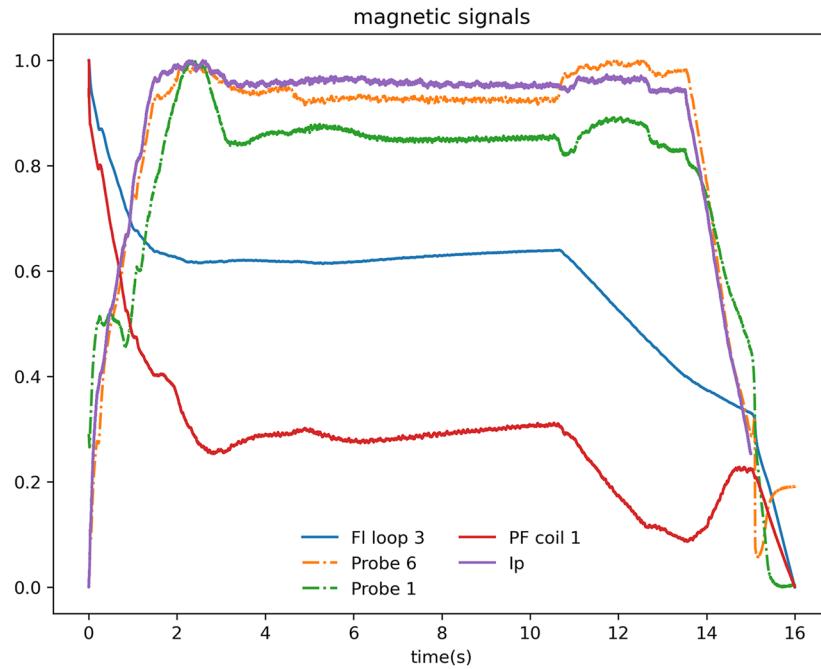


FIG. 3. Normalized signals corresponding to those in Figs. 1(b)–1(f). The sudden change in the poloidal magnetic flux (FI Loop 3) after 10 s is brought about by the PF1 coil current, which is actively adjusted by the plasma control system to provide the Ohm field to maintain a constant plasma current.

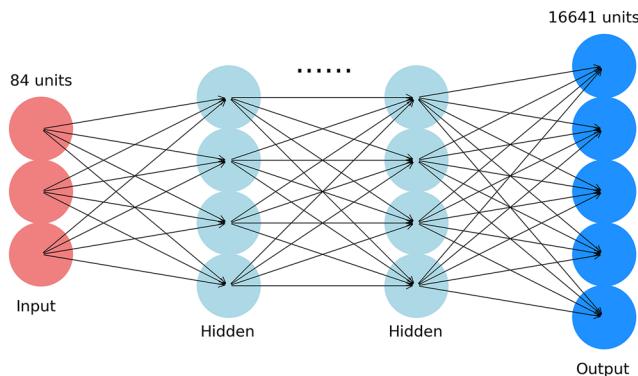


FIG. 4. Fully connected feed-forward neural network used in this work.

(50–70 keV deuterium beam), lower-hybrid waves (2.45 and 4.6 GHz), electron cyclotron waves (140 GHz), and ion cyclotron waves (25–70 MHz). The typical values of the total heating source power are between 4 and 10 MW.

TABLE II. Final values of hyperparameters of the model. The hyperparameters^a are fixed during the fine tuning. The above-mentioned activation function refers to that used in the hidden layers. For the output layer, the linear activation function is used.

Hyperparameter	Meaning	Final values
n_layers ^a	Number of hidden layers	4
n_units	Number of nodes per hidden layer	86
Activation ^a	Activation function	tanh
Optimizer ^a	Optimizer type	Adam
η	Learning rate	2.26×10^{-5}
Loss ^a	Loss function	MSE
batch_size	Number of samples used in a step	16
Epochs	Number of epochs	97

^aFixed hyperparameters during fine tuning.

Figure 2 shows the distribution of the EFIT equilibrium data in the (l_i, β_N) plane, where l_i is the normalized internal induction and β_N is the normalized plasma beta.

The collected data are split into three sets—training set (81%), validation set (9%), and testing set (10%)—where the training set

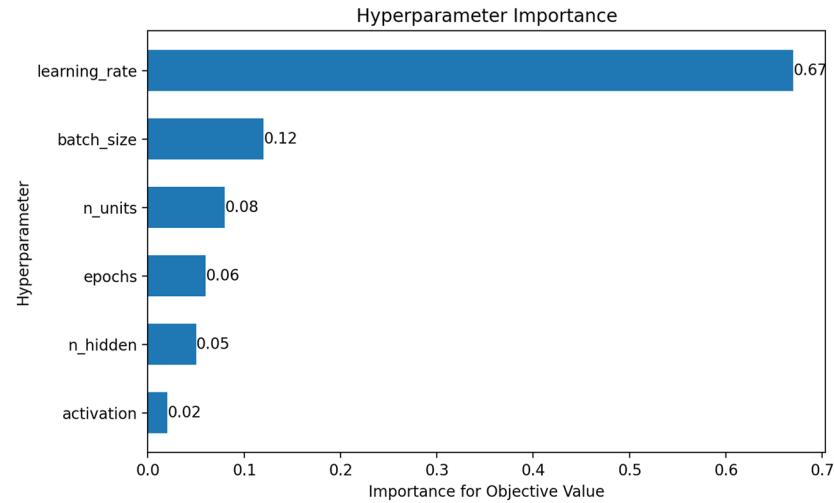


FIG. 5. Hyperparameter importance distribution. Model accuracy is not sensitive to the activation functions for this case.

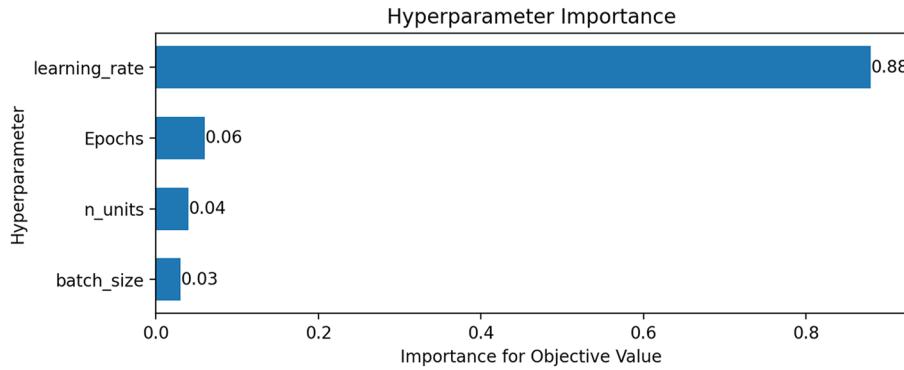


FIG. 6. Relative importance of the hyperparameters in determining the model accuracy in the fine tuning.

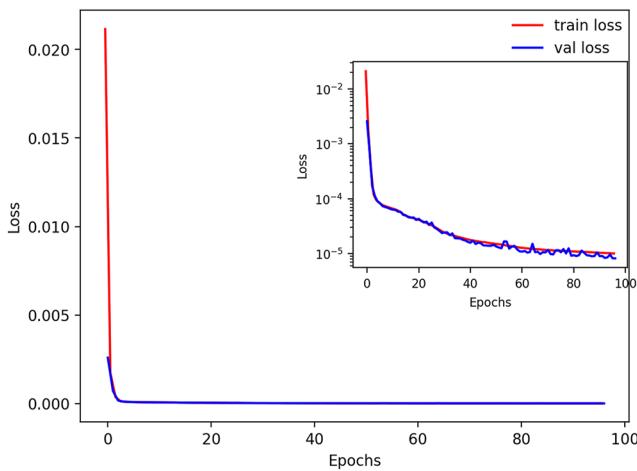


FIG. 7. Training history showing the training and validation loss function values vs the training epochs. One epoch corresponds to go through all the samples in the training set. In the mini-batch stochastic gradient descent method used here, each gradient descent step uses randomly selected 16 samples (a mini batch), and the loss function values shown here are obtained by summation over a mini-batch of samples in the training set or validation set.

is used in training the NN, the validation set is used in monitoring the potential overfitting and tuning hyperparameters, and the testing set is used in testing the predicting capability of the trained model.

Figures 1(b)–1(f) show that there is a difference of up to six orders of magnitude in the values of the input signals. In order to eliminate scale differences among features, we use the min-max normalization method to normalize the input data. The general formula is given by

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \quad (3)$$

where x is the original value of the feature, x' is the normalized value, and x_{\min} and x_{\max} are, respectively, the minimal and maximal value of a feature in the datasets excluding the testing set. x_{\min} and x_{\max} obtained here are then used to normalize the input data in the testing set when performing prediction using the trained NN.

Figure 3 plots the time evolution of the normalized input signals corresponding to those in Figs. 1(b)–1(f).

The magnitude of the output (Ψ in SI units) is near 1, so no normalization is applied to it.

III. MODEL ARCHITECTURE AND AUTOMATIC HYPERPARAMETER TUNING

An artificial neural network is a kind of computational network, which usually consists of multiple layers: input layer, one or more inner layers (known as hidden layers), and an output layer. Each layer is made of units. Each unit in the computing layers (hidden and output layers) receives information and processes the information using some linear transform (matrix multiplication) and some nonlinear transform (activation function).

A fully connected feed-forward network shown in Fig. 4 is used here to predict the poloidal magnetic flux Ψ based on the magnetic measurements. Here, “fully connected” means that each unit of a computing layer receives information from all the units of the previous layer. “Feed-forward” means that information moves in only one direction (from the input layer to the hidden layers and to the output layers), with no cycles or loops and no intra-layer connections.

Each unit (neuron or node) in the computing layers has trainable parameters, often called weights and biases. The output of the

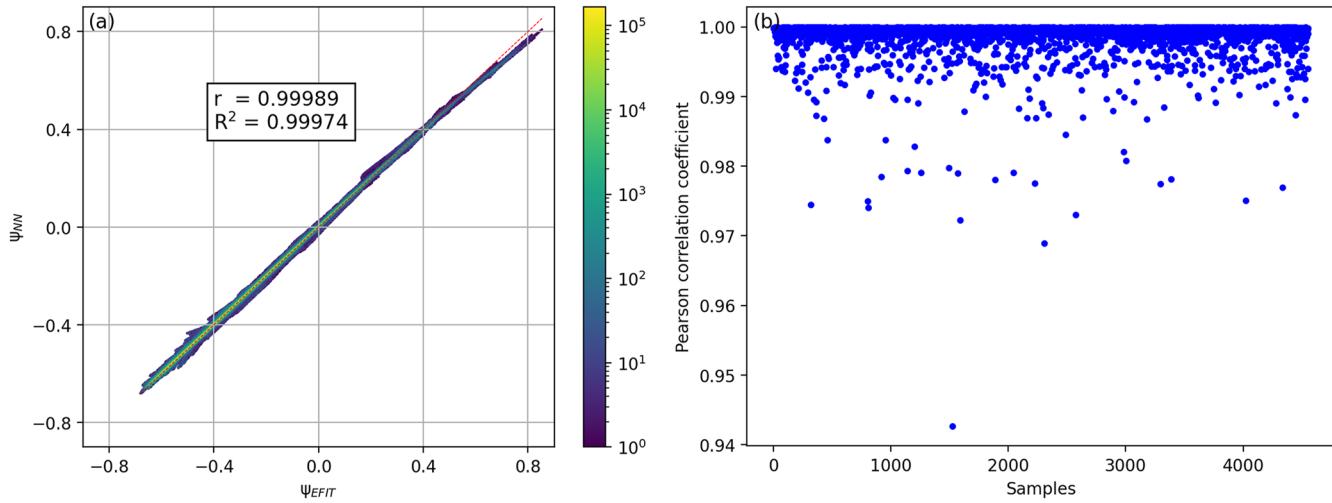


FIG. 8. (a) Neural network prediction of the poloidal flux Ψ_{NN} vs EFIT results Ψ_{EFIT} for the testing set (total 4555 equilibria). The red dashed line is the $y = x$ line. The total number of points shown here is $4555 \times 129 \times 129$. Color represents the density of data points. (b) The distribution of the correlation coefficient for each equilibrium of the 4555 equilibria in the testing set.

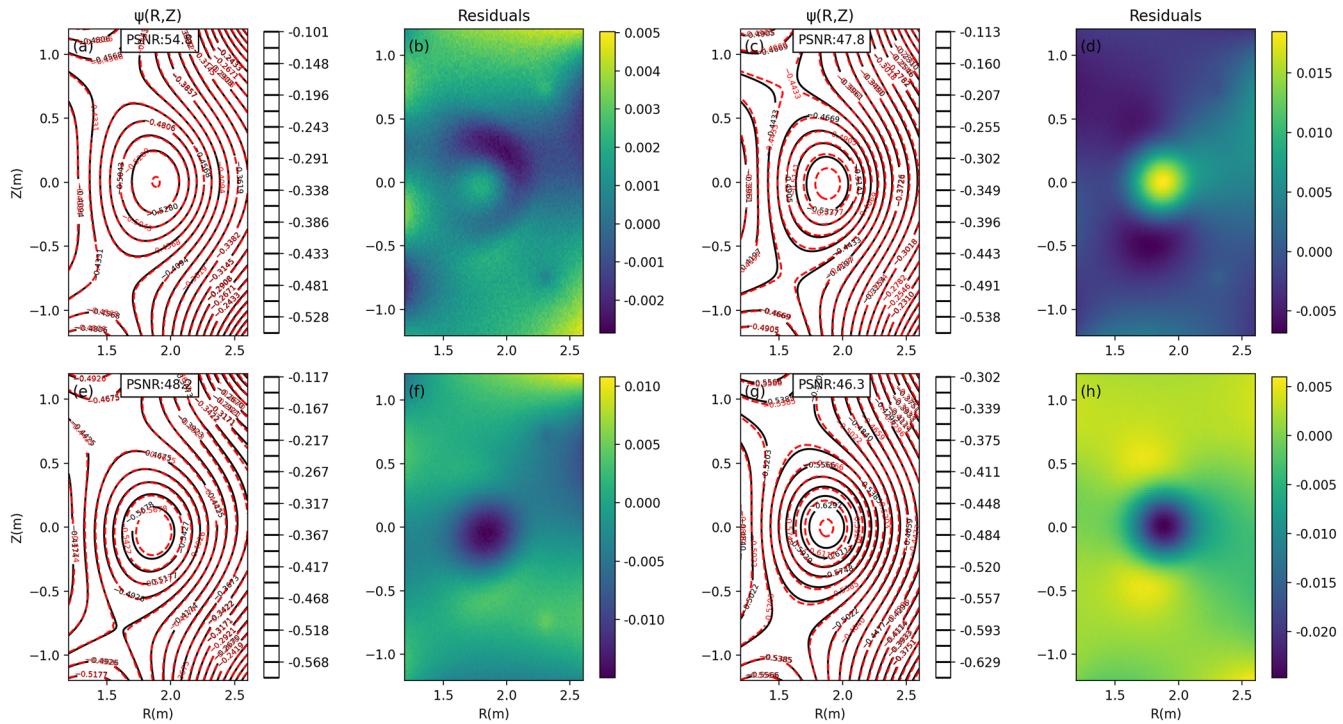


FIG. 9. Panels (a), (c), (e), and (g) compare the magnetic surfaces from EFIT (solid black lines) with those from the NN (dashed red lines) for four randomly selected time slices from the 4555 testing samples. Panels (b), (d), (f), and (h) show the corresponding normalized residua [$\Psi(R, Z)\Psi^{\text{NN}}(R, Z)/\max(|\Psi(R, Z)|)$].

jth neuron in the lth layer is denoted by a_j^l ; then a neural network model assumes that a_j^l is related to a^{l-1} (output of the previous layer) via

$$a_j^l = \sigma \left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l \right), \quad (4)$$

where w_{jk}^l and b_j^l are the weight and bias, respectively, the summation is over all neurons in the $(l-1)$ th layer, and σ is a function called the activation function. The weights and biases will be adjusted in the training process by gradient descent methods to reduce the loss (cost or error) function, which is defined in this work as

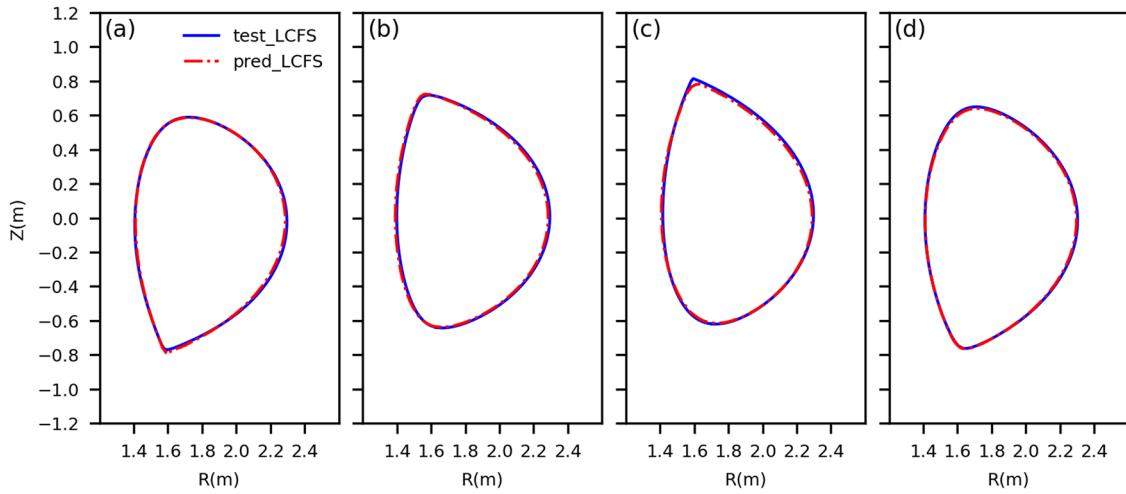


FIG. 10. Comparison of LCFSs between the NN prediction and EFIT result. The four panels (a)–(d) correspond to the four equilibria shown in Fig. 9. In locating the LCFSs from the NN prediction, we analyze a series of Ψ contours and determine the outermost contour that is near the magnetic separatrix.

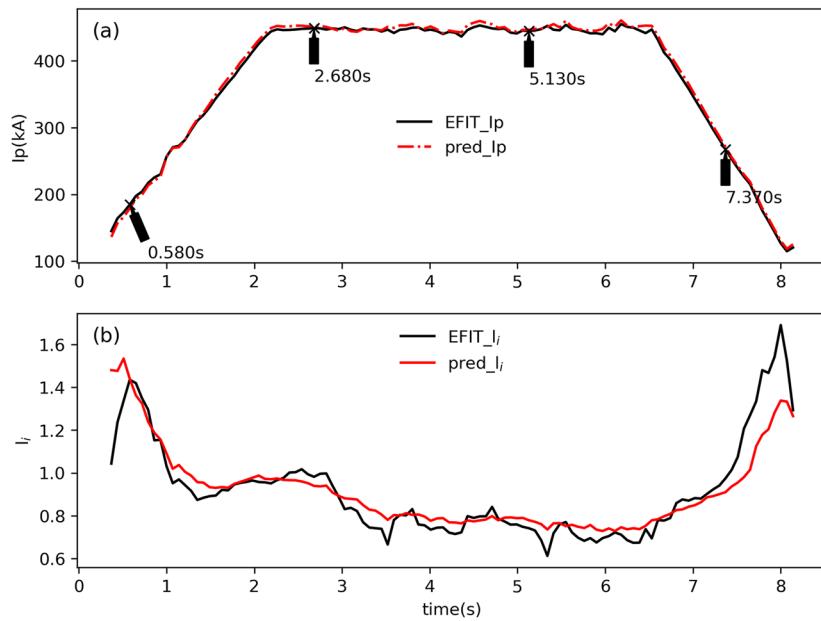


FIG. 11. (a) Comparison of time evolution of plasma current given by the NN and EFIT. (b) Comparison of time evolution of I_i given by the NN and EFIT. Four time slices are indicated on the graph, which are time slices selected for the magnetic configuration comparison shown in Figs. 12 and 13.

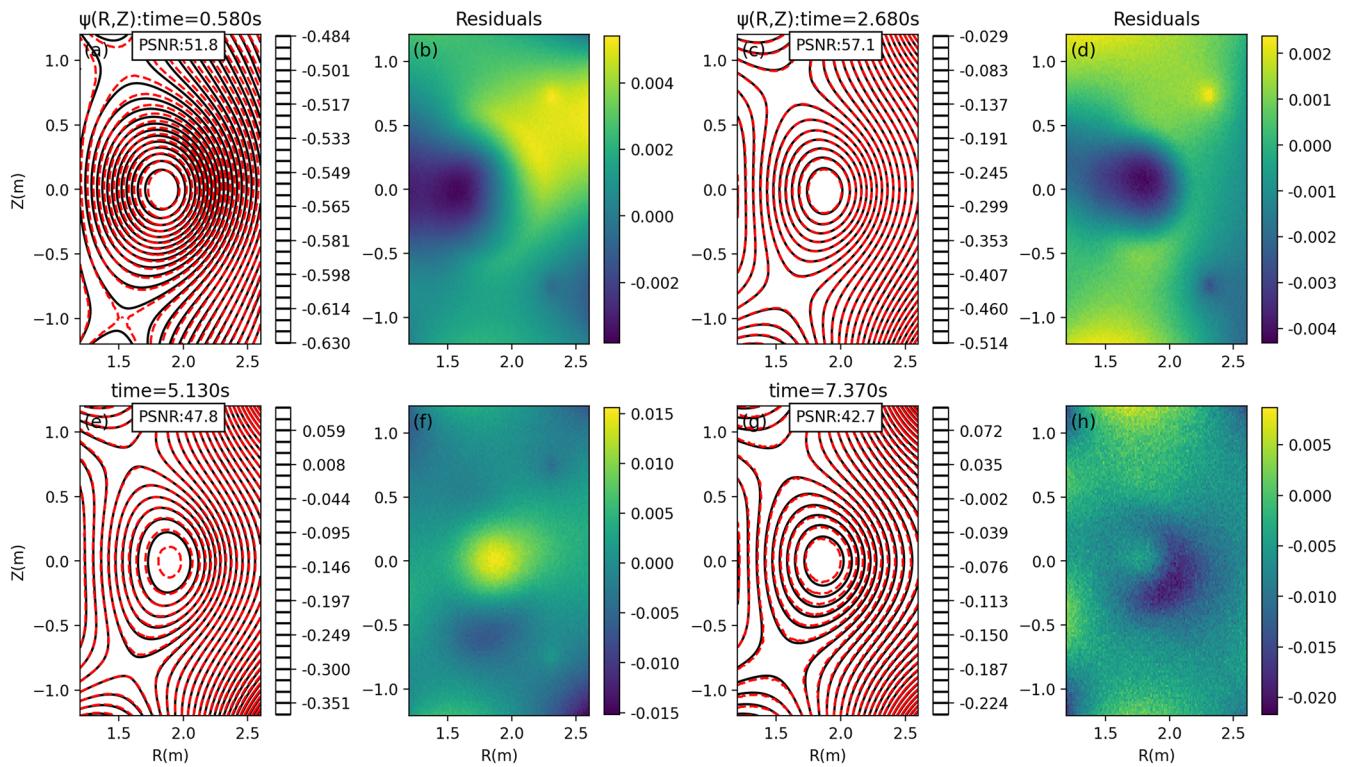


FIG. 12. Panels (a), (c), (e), and (g) compare the magnetic surfaces from EFIT (solid black lines) with those from the NN (dashed red lines) for discharge #113 388. Ψ at four different time slices during the discharge—0.580 s (early ramp-up), 2.680 s, 5.130 s (flat top), and 7.370 s (ramp-down)—is shown. Panels (b), (d), (f), and (h) show the corresponding relative error $(\Psi_{\text{EFIT}} - \Psi_{\text{NN}})/\max(|\Psi_{\text{EFIT}}|)$.

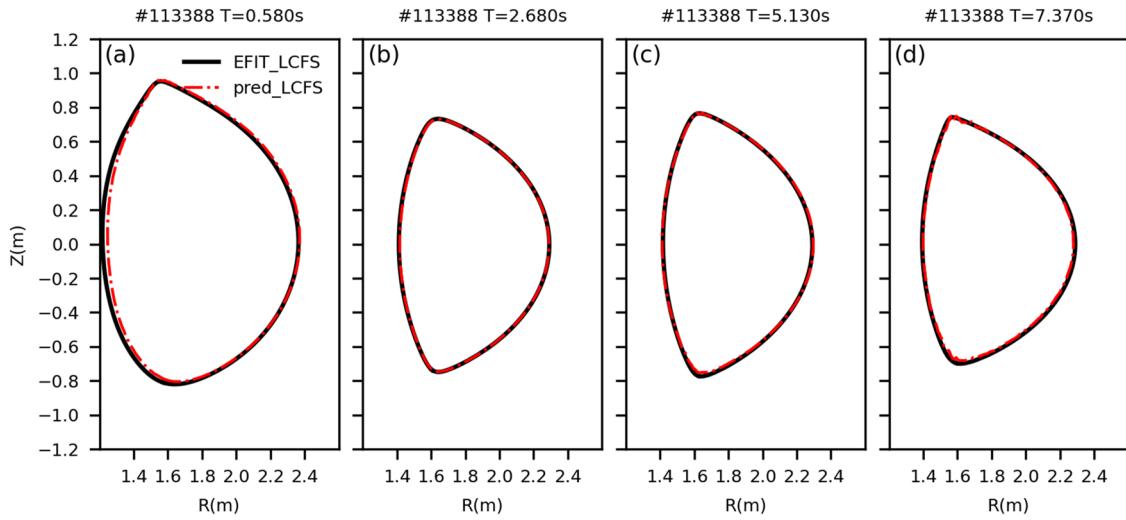


FIG. 13. NN reconstruction of the LCFS (dotted-dashed red) for EAST shot #113 388 overlaid against the EFIT LCFS (solid black). LCFSs from four different times in discharge #113 388 are shown.

$$L(\mathbf{w}, \mathbf{b}) \equiv \frac{1}{2n} \sum_{i=1}^n \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|^2, \quad (5)$$

where \mathbf{y}_i is the EFIT poloidal magnetic flux, $\hat{\mathbf{y}}_i$ is the NN output, and the summation is over all the samples in the training set. The loss function in Eq. (5) is the mean squared error (MSE). The loss function measures the derivation of the approximate solution away

from the desired exact solution. Hence, the goal of the learning algorithm is to find weights and biases that minimize the loss function. To minimize the loss function over (\mathbf{w}, \mathbf{b}) using the gradient descent method, we need to compute the partial derivatives $\partial L / \partial w_{jk}^l$ and $\partial L / \partial b_j^l$, which can be efficiently computed by the well-known back-propagating method.^{15,16} The back-propagating algorithm and the

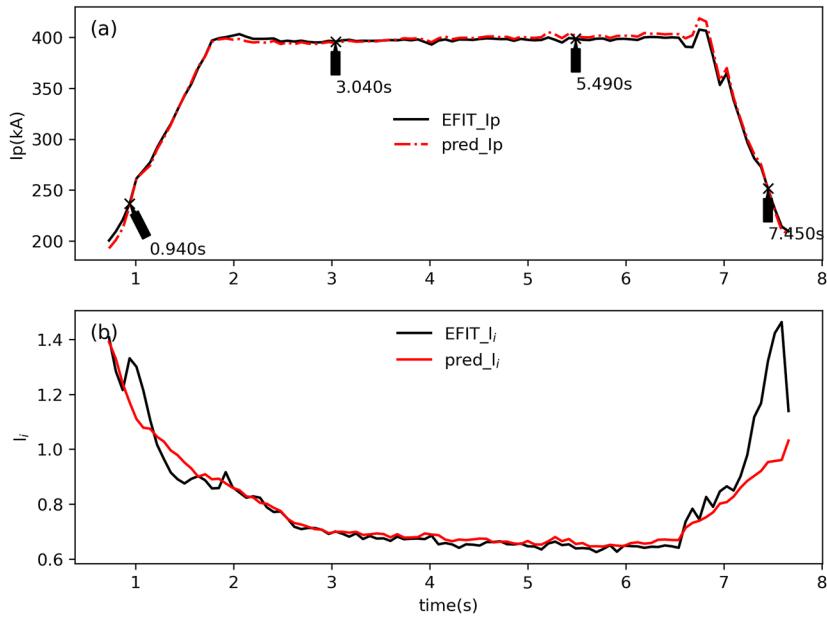


FIG. 14. Same as Fig. 11, except that the discharge is #117 016.

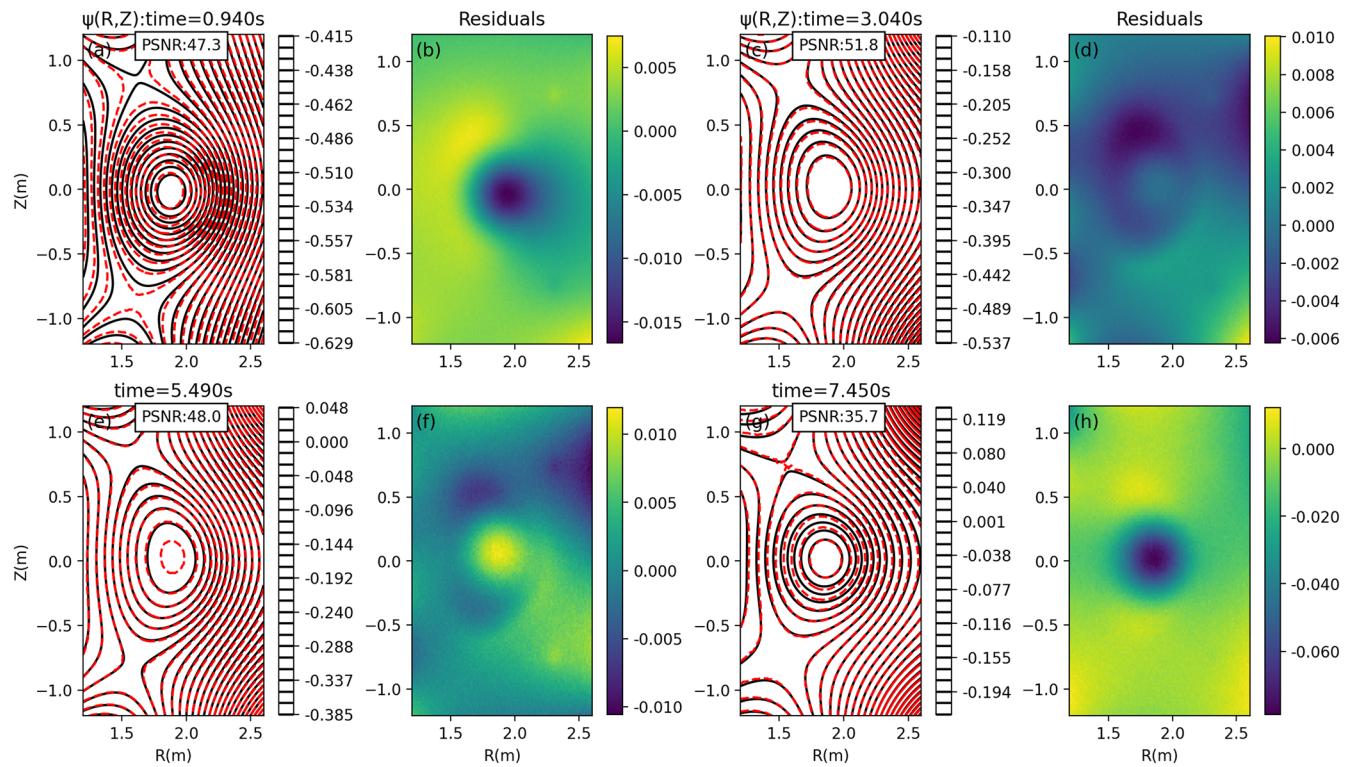


FIG. 15. Same as Fig. 12, except that the discharge is #117 016.

corresponding gradient descent method are the core algorithms in all deep learning software frameworks.

Besides the trainable parameters, there are various hyperparameters in an NN that usually need to be set manually, such as the number of hidden layers, units per layer, activation function,

NN optimizers, learning rate, batch size, and number of epochs of training. In recent years, there appeared automatic optimization libraries that can search for the best combination of hyperparameters. In this work, we use the Optuna optimization framework¹⁴ in setting the hyperparameters. Optuna automates the hyperparameter

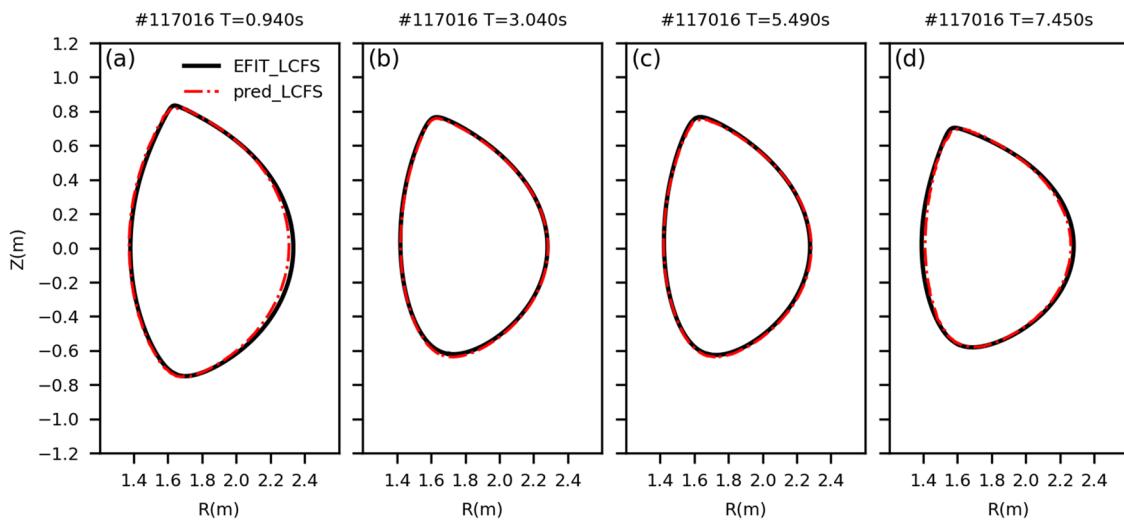


FIG. 16. Same as Fig. 13, except that the discharge is #117 016.

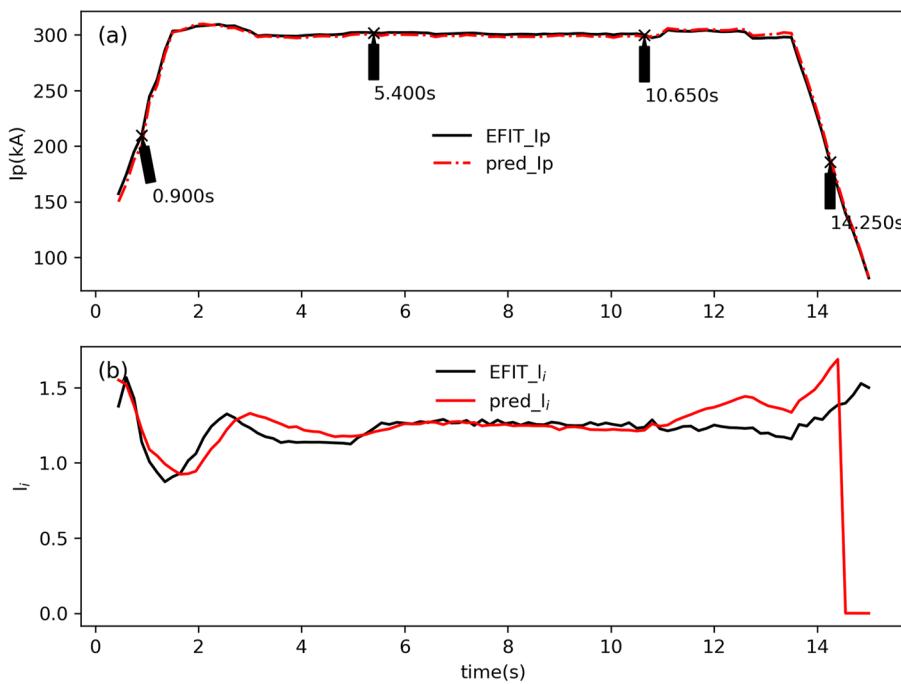


FIG. 17. Same as Fig. 11, except that the discharge is #113 019.

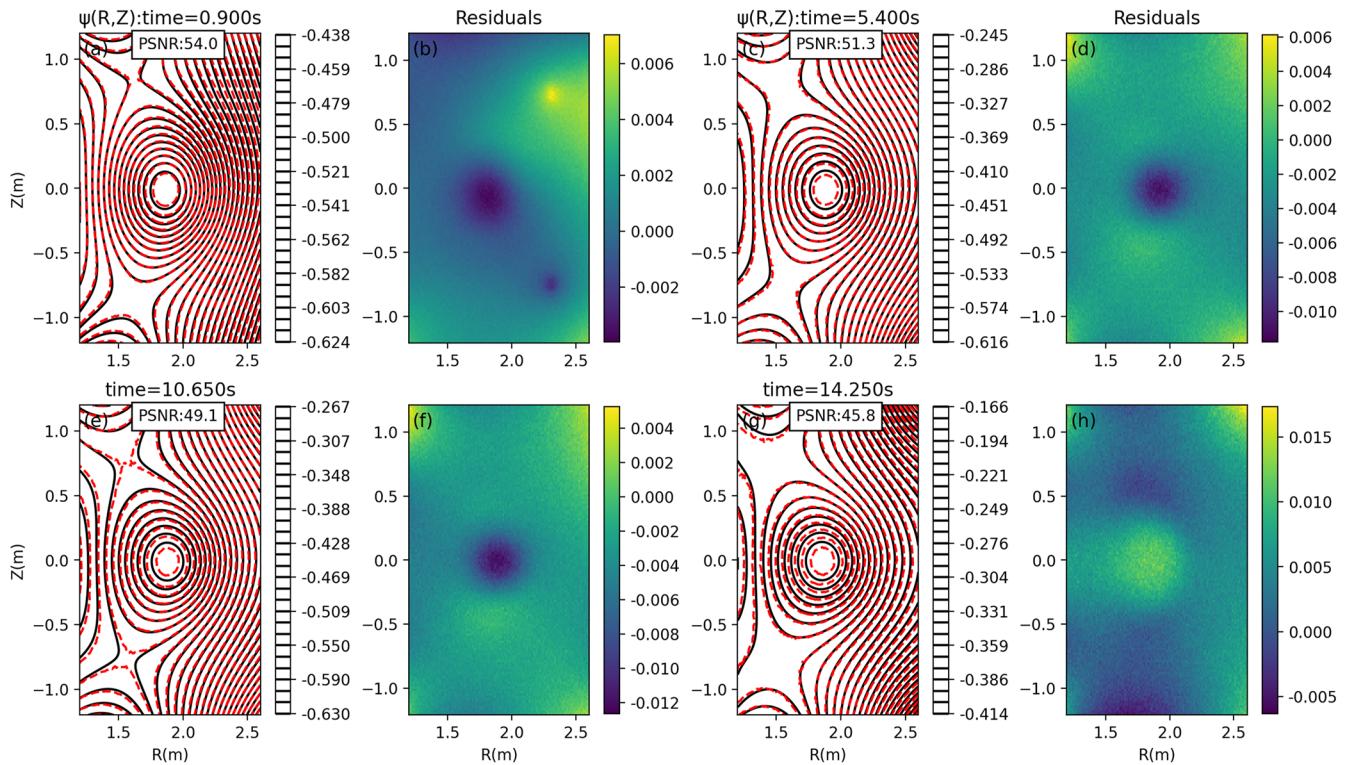


FIG. 18. Same as Fig. 12, except that the discharge is #113 019.

optimization process by defining a search space of hyperparameters and exploring the space using efficient searching algorithms. The tree-structured Parzen estimator (TPE) algorithm is used in this work. This algorithm models the relationship between hyperparameters and their corresponding performance metrics and makes efficient decisions on which hyperparameters to try next.

Optuna automates the selection of the best hyperparameter combination. After multiple experiments, we have found that the model accuracy is not sensitive to the number of hidden layers, activation functions, and optimizers (an example showing the relative importance of these hyperparameters is given in Fig. 5). Therefore, these hyperparameters are fixed in the fine-tuning step in order to improve the speed of the model selection process and explore more hyperparameter regimes to which the model may be sensitive. For other hyperparameters, we use the Optuna framework to find the optimal combination of hyperparameters. The relative importance of these hyperparameters is shown in Fig. 6. The above-mentioned results indicate that learning rate is the dominant factor that determines the model performance.

The final values of the hyperparameters used in the model are shown in Table II.

The network is constructed and trained using Keras and TensorFlow2,^{17,18} which are broadly adopted open source deep learning frameworks in the industry and research community. Figure 7 plots the loss function values as a function of the training epochs. The loss function is also evaluated on the validation set, which serves as a monitor for possible overfitting. The validation loss follows the same trend as the training loss, indicating no overfitting.

IV. PERFORMANCE OF THE NEURAL NETWORK

A. Performance of the model on the testing set

After the model is trained on the training set, we assess its prediction capability on the data that are not seen by the training process. To evaluate the reconstruction quality, we employ three widely adopted metrics: the Pearson correlation coefficient r

(definition is given in Appendix A), the coefficient of determination R^2 (definition is given in Appendix B), and the peak signal-to-noise ratio (PSNR) (definition is given in Appendix C).

Figure 8(a) plots the NN prediction of the poloidal flux Ψ_{NN} vs EFIT results Ψ_{EFIT} for the testing set (total 4555 equilibria, each with 16 641 values). The Pearson correlation coefficient r and the coefficient of determination R^2 are also shown in the figure, which are very close to 1, indicating strong predictive capability. Figure 8(b) plots the distribution of the correlation coefficient r between NN predictions and EFIT results for each equilibrium of the 4555 equilibria in the testing set. The results indicate that the majority of the values are greater than 0.998, indicating good correlation between the NN prediction and EFIT result for each equilibrium.

To test the accuracy of the model in predicting the plasma magnetic surface, we compare the 2D contours of the poloidal magnetic flux predicted by the NN with those given by EFIT. The results are shown in Figs. 9(a), 9(c), 9(e), and 9(g), where the NN predictions of Ψ contours are overlaid on the Ψ contours of EFIT. It displays four randomly selected samples from the 4555 equilibria in the testing set (the four displayed samples may not necessarily come from the same discharge). Since our reconstruction results take the form of images with the resolution determined by the spatial grid points, it is also useful to use the PSNR in evaluating the reconstruction quality of the magnetic surface. The values of the PSNR for the four equilibrium are shown in the figure.

To further assess the accuracy of the mode, we locate the LCFSs predicted by the NN model and compare them with those given by EFIT. The LCFSs corresponding to the four equilibria shown in Fig. 9 are shown in Fig. 10, which indicates that the NN and EFIT results are in good agreement. Minor discrepancies appear near the X points.

B. Performance of the model on four complete discharges

In this section, we arbitrarily select three full discharges that are not in the dataset used above to examine the time evolution of the

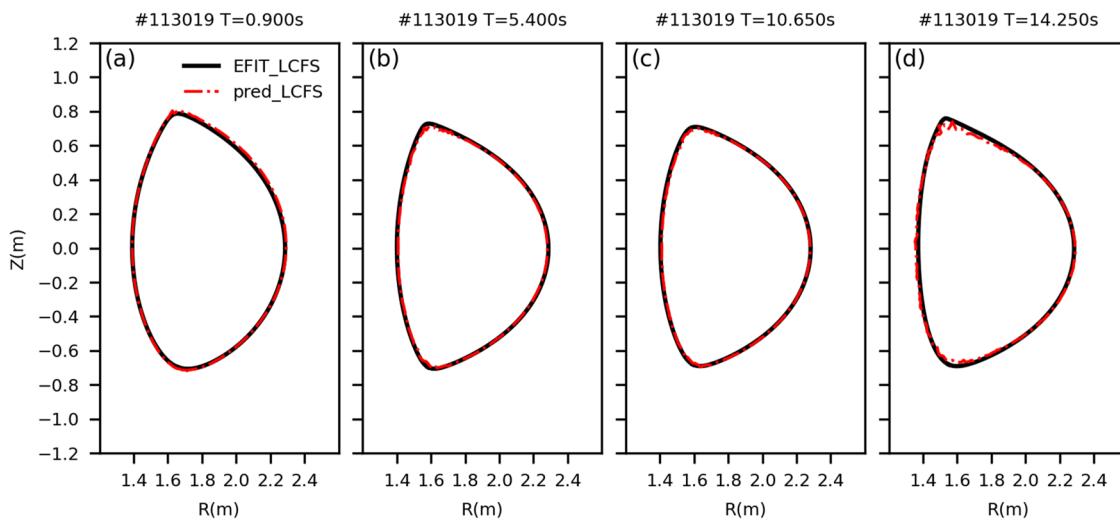


FIG. 19. Same as Fig. 13, except that the discharge is #113 019.

magnetic configuration during an entire discharge (from ramp-up to flat-top and then to ramp-down).

Besides the plasma currents I_p , we also calculate the normalized internal inductance l_i (definition is given in Appendix D), which is a quantity that is solely determined by Ψ and thus can further reflect how accurate the predicted Ψ is. We plot the time evolution of l_i and compare it with the EFIT results. By doing this, we can assess the accuracy of the NN in predicting the time evolution of some key volume-integrated quantities characterizing magnetic configuration.

[Figure 11](#) compares the time evolution of I_p and l_i predicted by the NN and that by EFIT for discharge #113 388.

[Figure 12](#) compares the contours of Ψ given by the NN model and that given by EFIT at 4 time slices (indicated in [Fig. 11](#)) in discharge #113 388. The results indicate that the relative error between the NN and EFIT results is less than 2%.

[Figure 13](#) compares the LCFSs given by the NN model and that given by EFIT at four time slices in discharge #113 388. The results show good agreement between the two models. Minor differences usually appear in the ramp up/down phase and near the X-points.

Similar results for discharge #117 016 are shown in [Figs. 14–16](#).

Similar results for discharge #113 019 are shown in [Figs. 17–19](#).

V. NEURAL NETWORK PREDICTION OF W_{mhd} , β_N , AND q_{95}

Besides the l_i discussed above, there are some other global parameters that can be constructed from the magnetic measurements, namely, the plasma stored energy W_{mhd} , normalized plasma beta β_N , and edge safety factor q_{95} . These parameters depend on information beyond the poloidal magnetic flux, namely, the toroidal magnetic field and plasma pressure. Therefore, they cannot be fully determined by using only the poloidal magnetic flux predicted from the above-mentioned network. Following Ref. 8, we construct a new NN for predicting these parameters (called NN2 in the following; the previous one will be called NN1), where the network has only three output values, namely, W_{mhd} , β_N , and q_{95} . The input to NN2 includes a new signal, the current in the toroidal field (TF) coils, which determines the toroidal field (In the NN1, this signal is not included because it has negligible effect on the prediction of the poloidal magnetic flux.). NN2 has only one hidden layer consisting of 16 units, and uses the sigmoid as the activation function for both the hidden and output layers. The input and output signals of NN2 are normalized by using the same min-max scalar used for NN1.

The training data consist of about 1/4 randomly selected part of the data used for NN1. We found that using larger dataset makes this small network prone to overfitting. The testing set consists of 1000 time slices. [Figure 20](#) plots the NN2 predictions against the EFIT values for the testing set. The results indicate that the NN2 predictions are in reasonable agreement with the EFIT values for all the three parameters. The NN2 predictions of q_{95} are a little worse than those of the other two parameters, judging from the values of r and R^2 .

To evaluate the accuracy of NN2 prediction for a full discharge, we arbitrarily chose a discharge and compare the time evolution of W_{mhd} , β_N , and q_{95} between the NN2 predictions and EFIT values. The results are shown in [Fig. 21](#), which shows good agreement between the network predictions and EFIT values.

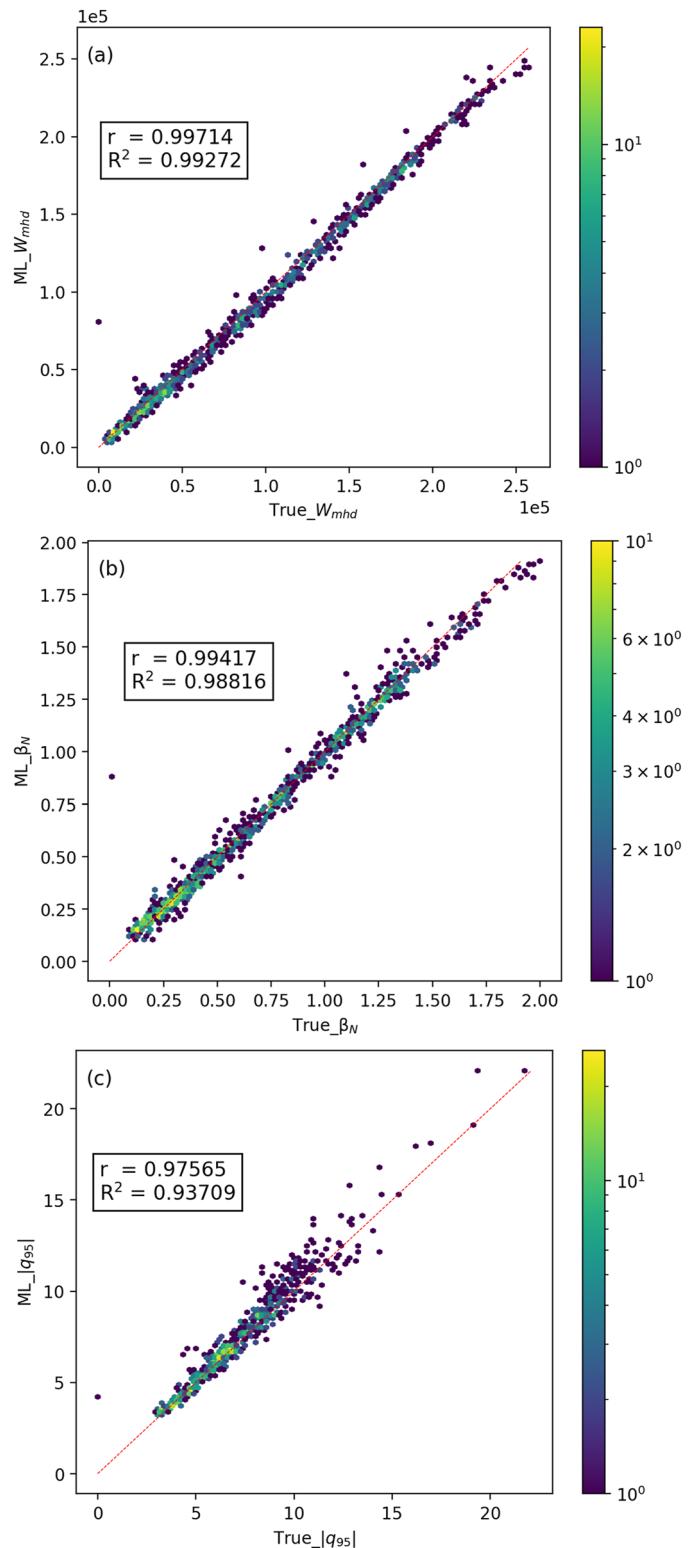


FIG. 20. NN2 predictions of $W_{\text{mhd}}(J)$ (a), β_N (b), and q_{95} (c) against the EFIT values for the testing set.

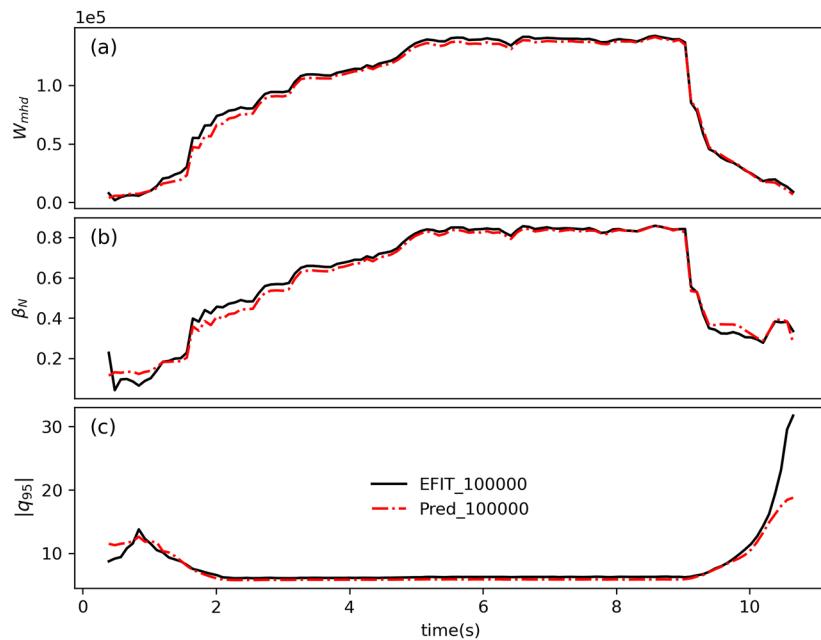


FIG. 21. Time evolution of plasma stored energy W_{mhd} (J) (a), normalized beta β_N (b), and edge safety factor q_{95} (c) in EAST discharge #100 000. Solid lines are EFIT values, and dashed-dotted lines are network predictions.

VI. SUMMARY AND DISCUSSION

In this work, we train a multiple-layer neural network on the magnetic measurements (input) and EFIT poloidal magnetic flux (output) on the EAST tokamak. The prediction capability of the network is examined by comparing the reconstructed magnetic surfaces, last closed flux surfaces, plasma current, and normalized internal inductance with those of EFIT. The neural network shows good agreement with EFIT for the data unseen in the training process.

In constructing the neural network, we use automatic optimization in searching for the best hyperparameters of the model. The hyperparameters found this way turn out to be better than our previously manually set hyperparameters in terms of the model accuracy.

Based on the model's good prediction capability and efficiency in terms of computational time [about 0.5 ms per equilibrium on a desktop computer using an 11th Gen Intel(R) Core(TM) i5-11500 @ 2.70 GHz central processing unit (CPU) with a single thread], it looks promising to apply the neural network to real-time magnetic configuration control. The above-mentioned computational time does not include the time used for tracing boundary/internal magnetic surfaces and other related calculations to obtain l_i . These computations (not optimized in this work) seem too inefficient to be used in real-time control. The purpose of computing l_i for the NN1 model is to evaluate the accuracy of the predicted Ψ . To predict these volume-integrated parameters, one usually uses an additional small network, as we did in Sec. V, which is efficient enough for real-time control because the network size is usually very small.

This work is limited to magnetic measurements. We plan to add more diagnostics related to the inner safety factor profiles and pressure profiles into the model in order to construct more realistic equilibria. This will rely on the kinetic EFIT output. We are accumulating these kind of training data.

ACKNOWLEDGMENTS

The authors thank Ting Lan, Tonghui Shi, Chengguang Wan, Zhengping Luo, Yao Huang, Guoqiang Li, and Jingping Qian for useful discussions. This work was supported by the Comprehensive Research Facility for Fusion Technology Program of China, under Contract No. 2018-000052-73-01-001228, by the Users with Excellence Program of the Hefei Science Center CAS, under Grant No. 2021HSC-UE017, and by the National Natural Science Foundation of China, under Grant No. 11575251.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Jingjing Lu: Data curation (equal); Formal analysis (equal); Software (equal); Validation (equal); Visualization (equal). **Youjun Hu:**

Conceptualization (equal); Data curation (equal); Formal analysis (equal); Methodology (equal); Software (equal); Supervision (equal); Writing – original draft (equal). **Nong Xiang:** Conceptualization (equal); Formal analysis (equal); Funding acquisition (equal); Supervision (equal). **Youwen Sun:** Formal analysis (equal); Methodology (equal); Project administration (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

APPENDIX A: PEARSON CORRELATION COEFFICIENT r

The Pearson correlation coefficient r is a statistical measure used to assess the strength and direction of a linear relationship between the predicted and true values of the data. It ranges from -1 to 1 , where 1 indicates a perfect positive correlation, 0 indicates no correlation, and -1 indicates a perfect negative correlation. The formula for r is

$$r = \frac{\sum_{i=1}^n (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2} \sqrt{\sum_{i=1}^n (\hat{y}_i - \bar{\hat{y}})^2}}, \quad (A1)$$

where n is the number of data in the testing set, y_i is the value given by EFIT, \hat{y}_i is the prediction by the NN, $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$ is the mean value of the values given by EFIT, and $\bar{\hat{y}} = \frac{1}{n} \sum_{i=1}^n \hat{y}_i$ is the mean value predicted by the NN.

APPENDIX B: COEFFICIENT OF DETERMINATION R^2

Another relevant metric used to assess how well a model fits the data is the coefficient of determination R^2 , which is defined by

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \quad (B1)$$

where n , y_i , \hat{y}_i , and \bar{y} mean the same as those in Sec. A1. The value of R^2 ranges from arbitrary negative values to 1 , where 1 represents a perfect fit between the model predictions and the actual data points. A higher value of R^2 suggests that the model is a better fit for the data. The coefficient of determination R^2 is usually not equal to the squared Pearson correlation coefficient except in some specific cases.

APPENDIX C: PEAK SIGNAL-TO-NOISE RATIO (PSNR)

The PSNR is a metric that measures the quality of an image by comparing the original image to a reconstructed version. A higher PSNR value indicates a higher quality reconstruction. It is defined by

$$\begin{aligned} \text{PSNR} &= 10 \times \log_{10} \left(\frac{\max(y_i)^2}{\text{MSE}} \right) \\ &= 10 \times \log_{10} \left(\frac{\max(y_i)^2}{\frac{1}{M} \sum_{i=1}^M (y_i - \hat{y}_i)^2} \right), \end{aligned} \quad (C1)$$

where $\max(y_i)$ is the maximum value of Ψ given by EFIT in the (R, Z) plane and MSE is the mean squared error between the EFIT and NN.

APPENDIX D: NORMALIZED INTERNAL INDUCTANCE

The normalized internal inductance l_i is defined by

$$l_i = \frac{\langle B_\theta^2 \rangle_P}{\langle B_\theta^2 \rangle_S}, \quad (D1)$$

where P is the integration over the plasma volume, $\langle B_\theta^2 \rangle_S$ is the surface average of poloidal field over the plasma boundary, and l_i reflects the peakness of the plasma current density profile; a small value of l_i corresponds to a broad current profile.

For circular cross section with minor radius a and assuming B_θ is independent of the poloidal angle, then Ampere's law gives $B_\theta(a) = \mu_0 I / (2\pi a)$. Then $\langle B_\theta^2 \rangle_S$ is approximated as

$$\langle B_\theta^2 \rangle_S \approx B_0^2(a) = \frac{\mu_0^2 I^2}{4\pi^2 a^2}. \quad (D2)$$

Using this and noting $V \approx \pi a^2 2\pi R_0$, where R_0 is the major radius of the device, Eq. (D1) is written as

$$l_i = \frac{4\pi^2 a^2}{\mu_0^2 I^2} \langle B_\theta^2 \rangle_P = \frac{4\pi^2 a^2 R_0}{\mu_0^2 I^2 R_0} \langle B_\theta^2 \rangle_P = \frac{2V}{\mu_0^2 I^2 R_0} \langle B_\theta^2 \rangle_P. \quad (D3)$$

Equation (D3) is used in this work to calculate l_i .

REFERENCES

- ¹L. L. Lao, H. S. John, R. D. Stambaugh, A. G. Kellman, and W. Pfeiffer, *Nucl. Fusion* **25**, 1611 (1985).
- ²L. L. Lao, J. R. Ferron, R. J. Groebner, W. Howl, H. S. John, E. J. Strait, and T. S. Taylor, *Nucl. Fusion* **30**, 1035 (1990).
- ³L. L. Lao, H. E. S. John, Q. Peng, J. R. Ferron, E. J. Strait, T. S. Taylor, W. H. Meyer, C. Zhang, and K. I. You, *Fusion Sci. Technol.* **48**, 968 (2005).
- ⁴J. Qian, B. Wan, L. L. Lao, B. Shen, S. A. Sabbagh, Y. Sun, D. Liu, B. Xiao, Q. Ren, X. Gong, and J. Li, *Plasma Sci. Technol.* **11**, 142 (2009).
- ⁵D. P. O'Brien, L. L. Lao, E. R. Solano, M. Garribba, T. S. Taylor, J. G. Cordey, and J. J. Ellis, *Nucl. Fusion* **32**, 1351 (1992).
- ⁶Y. S. Park, S. A. Sabbagh, J. W. Berkery, J. M. Bialek, Y. M. Jeon, S. H. Hahn, N. Eidietis, T. E. Evans, S. W. Yoon, J.-W. Ahn *et al.*, *Nucl. Fusion* **51**, 053001 (2011).
- ⁷S. Joung, J. Kim, S. Kwak, J. G. Bak, S. G. Lee, H. S. Han, H. S. Kim, G. Lee, D. Kwon, and Y.-C. Ghim, *Nucl. Fusion* **60**, 016034 (2019).
- ⁸L. L. Lao, S. Kruger, C. Akcay, P. Balaprakash, T. A. Bechtel, E. Howell, J. Koo, J. Leddy, M. Leinshauser, Y. Q. Liu, S. Madireddy, J. McClenaghan, D. Orozco, A. Pankin, D. Schissel, S. Smith, X. Sun, and S. Williams, *Plasma Phys. Controlled Fusion* **64**, 074001 (2022).
- ⁹B. N. Wan, Y. F. Liang, X. Z. Gong, J. G. Li, N. Xiang, G. S. Xu, Y. W. Sun, L. Wang, J. P. Qian, H. Q. Liu, X. D. Zhang, L. Q. Hu, J. S. Hu, F. K. Liu, C. D. Hu, Y. P. Zhao, L. Zeng, M. Wang, H. D. Xu, G. N. Luo, A. M. Garofalo, A. Ekedahl, L. Zhang, X. J. Zhang, J. Huang, B. J. Ding, Q. Zang, M. H. Li, F. Ding, S. Y. Ding, B. Lyu, Y. W. Yu, T. Zhang, Y. Zhang, G. Q. Li, T. Y. Xia, and EAST Team and Collaborators, *Nucl. Fusion* **57**, 102019 (2017).

- ¹⁰Z. Zheng, N. Xiang, J. Chen, S. Ding, H. Du, G. Li, Y. Wang, H. Liu, Y. Li, B. Lyu, and Q. Zang, "Kinetic equilibrium reconstruction for the NBI- and ICRH-heated H-mode plasma on EAST tokamak," *Plasma Sci. Technol.* **20**, 065103 (2018).
- ¹¹G. Q. Li, Q. L. Ren, J. P. Qian, L. L. Lao, S. Y. Ding, Y. J. Chen, Z. X. Liu, B. Lu, and Q. Zang, *Plasma Phys. Controlled Fusion* **55**, 125008 (2013).
- ¹²Z. Luo, B. Xiao, Y. Zhu, and F. Yang, *Plasma Sci. Technol.* **12**, 412 (2010).
- ¹³C. Wan, Z. Yu, A. Pau, O. Sauter, X. Liu, Q. Yuan, and J. Li, *Nucl. Fusion* **63**, 056019 (2023).
- ¹⁴T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, *arXiv:1907.10902* (2019).
- ¹⁵D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Nature* **323**, 533 (1986).
- ¹⁶M. A. Nielsen, *Neural Networks and Deep Learning* (Determination Press, San Francisco, CA, 2015), Vol. 25.
- ¹⁷F. Chollet *et al.*, Keras, <https://keras.io>, 2015.
- ¹⁸M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015, Software available from tensorflow.org.