Deep neural network Grad-Shafranov solver constrained with measured magnetic signals

Semin Joung,[1,*] Jaewook Kim,[1] Sehyun Kwak,[1,2] J.G. Bak,[3] S.G. Lee,[3]
H.S. Han,[3] H.S. Kim,[3] Geunho Lee,[4] Daeho Kwon,[1,4] and Y.-c. Ghim[1,†]

[1] *Department of Nuclear and Quantum Engineering, KAIST, Daejeon 34141, Republic of Korea*
[2] *Max-Planck-Institut für Plasmaphysik, Teilinstitut Greifswald, D-17491 Greifswald, Germany*
[3] *National Fusion Research Institute, Daejeon 34133, Republic of Korea*
[4] *Mobiis Co., Ltd., Seongnam-si, Gyeonggi-do 13486, Republic of Korea*

(Dated: November 6, 2019)

A neural network solving Grad-Shafranov equation constrained with measured magnetic signals to reconstruct magnetic equilibria in real time is developed. Database created to optimize the neural network's free parameters contain off-line EFIT results as the output of the network from 1,118 KSTAR experimental discharges of two different campaigns. Input data to the network constitute magnetic signals measured by a Rogowski coil (plasma current), magnetic pick-up coils (normal and tangential components of magnetic fields) and flux loops (poloidal magnetic fluxes). The developed neural networks fully reconstruct not only the poloidal flux function $\psi\left(R,Z\right)$ but also the toroidal current density function $j_\phi\left(R,Z\right)$ with the off-line EFIT quality. To preserve robustness of the networks against a few missing input data, an imputation scheme is utilized to eliminate the required additional training sets with large number of possible combinations of the missing inputs.

Keywords: Neural network, Grad-Shafranov equation, EFIT, poloidal flux, toroidal current, imputation, KSTAR

I. INTRODUCTION

Magnetic equilibrium is one of the most important information to understand the basic behavior of plasmas in magnetically confined plasmas, and the off-line EFIT [1] code has been extensively used to reconstruct such equilibria in tokamaks. Its fundamentals are basically finding a solution to an ideal magnetohydrodynamic equilibrium with toroidal axisymmetry, known as the Grad-Shafranov (GS) equation [2]:

$$\begin{aligned}\Delta^*\psi &\equiv \left(R\frac{\partial}{\partial R}\frac{1}{R}\frac{\partial}{\partial R}+\frac{\partial^2}{\partial Z^2}\right)\psi \\ &= -\mu_0 R j_\phi \qquad (1)\\ &= -\mu_0 R^2\frac{dp(\psi)}{d\psi}-F(\psi)\frac{dF(\psi)}{d\psi},\end{aligned}$$

where $\psi=\psi\left(R,Z\right)$ is the poloidal flux function, $j_\phi=j_\phi\left(R,Z\right)$ the toroidal current density function, $p(\psi)$ the plasma pressure. $F(\psi)$ is related to the net poloidal current. Here, R, ϕ and Z denote the usual cylindrical coordinate system. As the Δ^* is a two-dimensional nonlinear partial differential operator, the off-line EFIT [1] finds a solution with many numerical iterations and has been implemented in many tokamaks such as DIII-D [3], JET [4], NSTX [5], EAST [6] and KSTAR [7] to name some as examples.

With an aim of real-time control of tokamak plasmas, real-time EFIT (rt-EFIT) [8] code is developed to provide a magnetic equilibrium fast enough whose results are different from the off-line EFIT results. As pulse lengths of tokamak discharges become longer [9–15], demand on more elaborate plasma control is ever increased. Furthermore, some of the ITER relevant issues such as ELM (edge localized mode) suppression with RMP (resonant magnetic perturbation) coils [16] and the detached plasma scenarios [17, 18] require sophisticated plasma controls, meaning that the more accurate magnetic equilibria we have in real time, the better performance we can achieve.

There has been an attempt to satisfy such a requirement of acquiring a more accurate, i.e., closer to the off-line EFIT results compared to the rt-EFIT results, magnetic equilibrium in real-time using graphics processing units (GPUs) [19] by parallelizing equilibrium reconstruction algorithms. The GPU based EFIT (P-EFIT) [19] enabled one to calculate a well-converged equilibrium in much less time; however, the benchmark test showed similar results to the rt-EFIT rather than the off-line results [20].

Thus, we propose a reconstruction algorithm based on a neural network that satisfies the GS equation as well as the measured magnetic signals to obtain accurate magnetic equilibrium in real time. We note that usage of neural networks in fusion community is increasing rapidly, and examples are radiated power estimation [21], identifying instabilities [22], estimating neutral beam effects [23], classifying confinement regimes [24], determination of scaling laws [25, 26], disruption prediction [27–29], turbulent transport modelling [30–33], plasma tomography with the bolometer system [34, 35], coil current prediction with the heat load pattern in W7-X [36], filament detection on MAST-U [37], electron temperature profile estimation via SXR with Thomson scattering [38] and

[*] smjoung@kaist.ac.kr
[†] ycghim@kaist.ac.kr

equilibrium reconstruction [39–44] together with an equilibrium solver [45]. Most of previous works on the equilibrium reconstruction with neural networks have paid attention to finding the poloidal beta, the plasma elongation, positions of the X-points and plasma boundaries, i.e., last closed flux surface, and gaps between plasmas and plasma facing components, rather than reconstructing the whole internal magnetic structures we present in this work.

The inputs to our developed neural networks consist of plasma current measured by a Rogowski coil, normal and tangential components of magnetic fields by magnetic pick-up coils, poloidal magnetic fluxes by flux loops and a position in (R, Z) coordinate system, where R is the major radius, and Z is the height as shown in Figure 1. The output of the neural networks is a value of poloidal flux ψ at the specified (R, Z) position. To train and validate the neural networks, we have collected a total of $1,118$ KSTAR discharges from two consecutive campaigns, i.e., 2017 and 2018 campaigns. We, in fact, generate three separate neural networks which are NN_{2017}, NN_{2018} and $\mathrm{NN}_{2017,\ 2018}$ where subscripts indicate the year(s) of KSTAR campaign(s) that the training data sets are obtained from. Additional 163 KSTAR discharges (from the same two campaigns) are collected to test the performance of the developed neural networks.

We train the neural networks with the KSTAR off-line EFIT results, and let them be *accurate* magnetic equilibria. Note that disputing on whether the off-line EFIT results we use to train the networks are accurate or not is beyond the scope of this work. If we find more accurate EFIT results, e.g., MSE(Motional Stark Effect)-constrained EFIT or more sophisticated equilibrium reconstruction algorithms that can cope with current-hole configurations (current reversal in the core) [46–48], then we can always re-train the networks with new sets of data as long as the networks follow the trained EFIT data with larger similarity than the rt-EFIT results do. This is because supervised neural networks are limited to follow the training data. Hence, as a part of the training sets we use the KSTAR off-line EFIT results as possible examples of *accurate* magnetic equilibria to corroborate our developed neural networks.

To calculate the output data a typical neural network requires the same set of input data as it has been trained. Therefore, even a single missing input (out of input data set) can result in a flawed output [49]. Such a case can be circumvented by training the network with possible combinations of missing inputs. As a part of input data, we have 32 normal and 36 tangential magnetic fields measured by the magnetic pick-up coils. If we wish to cover a case with one missing input data, then we will need to repeat the whole training procedure with 68 $(32+36)$ different cases. If we wish to cover a case with two or three missing input data, then we will need additional $2,278$ and $50,116$ different cases to be trained on, respectively. This number becomes large rapidly, and it becomes formidable, if not impossible, to train the networks with reasonable computational resources. Since the magnetic pick-up coils are susceptible to damages, we have developed our networks to be capable of inferring a few missing signals of the magnetic pick-up coils in real-time by invoking an imputation scheme [50] based on Bayesian probability [51] and Gaussian processes [52].

In addition to reconstructing *accurate* magnetic equilibria in real-time, the expected improvements with our neural networks compared to the previous studies are at least fourfold: (1) the network is capable of providing whole internal magnetic topology, not limited to boundaries and locations of X-points and/or magnetic axis; (2) spatial resolution of reconstructed equilibria is arbitrarily adjustable within the first wall of KSTAR since (R, Z) position is a part of the input data; (3) the required training time and computational resources for the networks are reduced by generating a coarse grid points also owing to (R, Z) position being an input, and (4) the networks can handle a few missing signals of the magnetic pick-up coils using the imputation method.

We, first, present how the data are collected to train the neural networks and briefly discuss real-time preprocessing of the measured magnetic signals in Section II. For the readers who are interested in thorough description of the real-time preprocessing, Appendix A provides the details. Then, we explain the structure of our neural networks and how we train them in Section III. In Section IV, we present the results of the developed neural network EFIT (nn-EFIT) in four aspects. First, we discuss how well the $\mathrm{NN}_{2017,\ 2018}$ network reproduces the off-line EFIT results. Then, we make comparisons among the three networks, NN_{2017}, NN_{2018} and $\mathrm{NN}_{2017,\ 2018}$, by examining in-campaign and cross-campaign performance. Once the absolute performance qualities of the networks are established, we compare relative performance qualities between nn-EFIT and rt-EFIT. Finally, we show how the imputation method support the networks when there exist missing inputs. Our conclusions are presented in Section V.

II. COLLECTION AND REAL-TIME PREPROCESSING OF DATA

Figure 1 shows locations where we obtain the input and the output data with the first wall (blue dotted line) on a poloidal cross-section of KSTAR. The green dotted line indicates a Rogowski coil measuring the plasma current (I_p). The green open circles and crosses show locations of the magnetic pick-up coils measuring 32 normal (B_n) and 36 tangential (B_t) components of magnetic fields, respectively, whereas the green triangles show 22 flux loops measuring the poloidal magnetic fluxes (Ψ_FL). These magnetic signals are selectively chosen out of all the magnetic sensors in KSTAR [53] whose performance has been demonstrated for many years, i.e., less susceptible to damages.

Although KSTAR calibrates the magnetic sensors

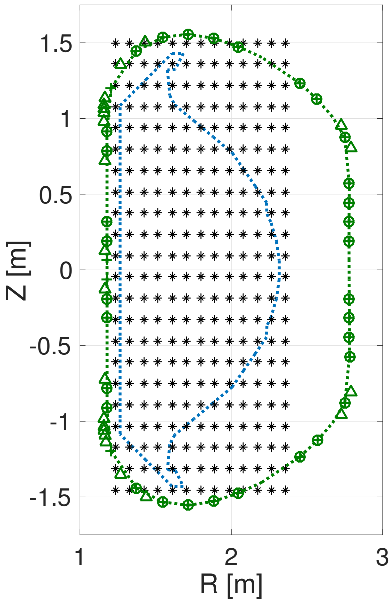

FIG. 1. A poloidal cross-section of KSTAR with the first wall (blue dotted line). Green dotted line indicates a Rogowski coil measuring the plasma current ($I_{\rm p}$). Green open circles and crosses depict locations of the magnetic pick-up coils measuring 32 normal ($B_{\rm n}$) and 36 tangential ($B_{\rm t}$) magnetic fields, respectively, whereas green triangles represent 22 flux loops measuring poloidal magnetic fluxes ($\Psi_{\rm FL}$). Black asterisks (22×13 spatial positions) show locations where we obtain the values of ψ from the off-line EFIT results.

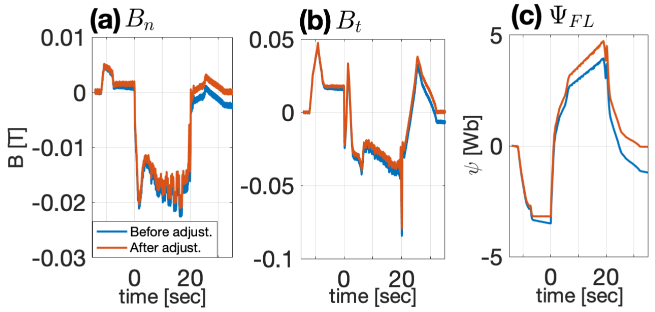

FIG. 2. Before (blue) and after (red) the magnetic signal adjustments for (a) normal and (b) tangential components of magnetic fields measured by the magnetic pick-up coils, and (c) poloidal magnetic flux measured by one of the flux loops. The signals return closer to zeros after the adjustment when all the external magnetic coils (except the toroidal field coils) are turned off at around 30 sec in this KSTAR discharge. See Appendix A for detailed description.

TABLE I. Summary of the data samples to train and validate the networks

Parameter	Definition	Data size	No. of samples
$I_{\rm p}$	Plasma current (Rogowski coil)	1	217,820 (time slices)
$B_{\rm n}$	Normal magnetic field (Magnetic pick-up coils)	32	
$B_{\rm t}$	Tangential magnetic field (Magnetic pick-up coils)	36	
$\Psi_{\rm FL}$	Poloidal magnetic flux (Flux loops)	22	
R	Position in major radius	1	286 (22×13 grids)
Z	Position in height	1	
Network Input size		93 (+1 for bias)	
Total no. of samples			62,296,520

(magnetic pick-up coils and flux loops) regularly during a campaign to remove drifts in the magnetic signals, it does not guarantee to fully eliminate such drifts. Thus, we preprocess the signals to adjust the drifts. Figure 2 shows examples of before (blue) and after (red) the drift adjustment for (a) normal and (b) tangential components of magnetic fields measured by the magnetic pick-up coils and (c) poloidal magnetic flux measured by one of the flux loops. Here, a KSTAR discharge is sustained until about 20 sec, and all the external magnetic coils (except the toroidal field coils) are turned off at about 30 sec. Therefore, we expect all the magnetic signals to return to zeros at around 30 sec. If not, we envisage that there has been residual drifts. This means that we need to be able to preprocess the magnetic signals in real-time so that the input signal characteristics for predictions are similar to the trained ones. Appendix A describes in detail how we preprocess the magnetic signals in real-time.

The black asterisks in Figure 1 show the 22×13 grid points where we obtain the values of ψ from the off-line EFIT results as outputs of the networks. We note that the original off-line EFIT provides the values of ψ with 65×65 grid points. The 22×13 grid points are selected such that the distances between the neighboring channels in R and Z directions are as similar as possible while covering whole region within the first wall. By generating such coarse grid points we can decrease the number of samples to train the network, thus consuming less amount of computational resources. Nevertheless, we do not lose the spatial resolution since (R, Z) position is an input, i.e., the network can obtain the value of ψ at any position within the first wall (see Section IV).

With an additional input for the spatial position R and Z, each data sample contains 93 inputs (and yet another input for bias) and one output which is a value of ψ at the specified (R, Z) location. We randomly collect a total of $1,118$ KSTAR discharges from 2017 and 2018 campaigns. Since each discharge can be further broken into many time slices, i.e., every 50 msec following the temporal resolution of the off-line EFIT, we obtain $217,820$ time slices. With a total of 286 value of ψ from 22×13 spatial points, we have a total of $62,296,520\,(= 217,820 \times 286)$ samples to train and validate the networks. 90% of the samples are used to train the networks, while the other 10% are used to validate the networks to avoid overfitting problems. Note that an overfitting problem can occur if a network is overly well trained to the *training data* following the very details of them. This inhibits generalization of the trained network to predict *unseen data*, and such a problem can be minimized with the validation data set. All the inputs except R and Z are normalized such that the maximum and minimum values within the whole samples become 1 and -1, respectively. We use the actual values of R and Z in the unit of meters.

Table I summarizes the training and validation samples discussed in this section. Additionally, we also have randomly collected another 163 KSTAR discharges in the same way discussed here which are different from the $1,118$ KSTAR discharges to test the performance of the networks.

III. NEURAL NETWORK MODEL AND TRAINING

A. Neural network model

We develop the neural networks that not only output a value of ψ but also satisfies Equation (1), the GS equation. With the total of 94 input nodes (91 for a plasma current and magnetic signals, two for R and Z position, one for the bias) and one output node for a value of ψ, each network has three fully connected hidden layers with an additional bias node at each hidden layer. Each layer contains 61 nodes including the bias node. The structure of our networks is selected by examining several different structures by error and trials.

Denoting the value of ψ calculated by the networks as ψ^{NN}, we have

$$\psi^{\mathrm{NN}} = s_0 + \sum_{l=1}^{60} s_l \times f\left(u_{l0} + \sum_{k=1}^{60} u_{lk} f\left(v_{k0} + \sum_{j=1}^{60} v_{kj} f\left(w_{j0} + \sum_{i=1}^{93} w_{ji} x_i\right)\right)\right), \tag{2}$$

where x_i is the i^{th} input value with $i = 1, \ldots, 93$, i.e., 91 measured values with the various magnetic diagnostics and two for R and Z positions. w_{ji} is an element in a 61×94 matrix, whereas v_{kj} and u_{lk} are elements in 61×61 matrices. s_l connects the l^{th} node of the third (last) hidden layer to the output node. w, v, u and s are the weighting factors that need to be trained to achieve our goal of obtaining accurate ψ. w_{j0}, v_{k0}, u_{l0} and s_0 are the weighting factors connecting the biases, where values of all the biases are fixed to be unity. We use a hyperbolic tangent function as the activation function f giving the network non-linearity [54]:

$$f(t) = \tanh(t) = \frac{2}{1+e^{-2t}} - 1. \tag{3}$$

The weighting factors are initialized as described in [55] so that a good training can be achieved. They are randomly selected from a normal distribution whose mean is zero with the variance set to be an inverse of total number of connecting nodes. For instance, our weighting factor w connects the input layer (94 nodes with bias) and the first hidden layer (61 nodes with bias), therefore the variance is set to be $1/(94+61)$. Likewise, the variances for v, u and s are $1/(61+61)$, $1/(61+61)$ and $1/(61+1)$, respectively.

B. Training

With the aforementioned network structure, training (or optimizing) the weighting factors to predict the correct value of ψ highly depends on a choice of the cost function. A typical choice of such cost function would be:

$$\epsilon = \frac{1}{N}\sum_{i=1}^{N}\left(\psi_i^{\mathrm{NN}} - \psi_i^{\mathrm{Target}}\right)^2, \tag{4}$$

where ψ^{Target} is the target value, i.e., the value of ψ from the off-line EFIT results in our case, and N the number of data sets.

As will be shown shortly, minimizing the cost function ϵ does not guarantee to satisfy the GS equation (Equation (1)) even if ψ^{NN} and ψ^{Target} matches well, i.e., the network is well trained with the given optimization rule. Since $\Delta^*\psi$ provides information on the toroidal current density directly, it is important that $\Delta^*\psi^{\mathrm{NN}}$ matches $\Delta^*\psi^{\mathrm{Target}}$ as well. We have an analytic form representing ψ^{NN} as in Equation (2); therefore, we can analytically differentiate ψ^{NN} with respect to R and Z, meaning that we can calculate $\Delta^*\psi^{\mathrm{NN}}$ during the training stage. Thus, we introduce another cost function:

$$\begin{aligned}\epsilon^{\mathrm{new}} &= \frac{1}{N}\sum_{i=1}^{N}\left(\psi_i^{\mathrm{NN}} - \psi_i^{\mathrm{Target}}\right)^2 \\ &+ \frac{1}{N}\sum_{i=1}^{N}\left(\Delta^*\psi_i^{\mathrm{NN}} - \Delta^*\psi_i^{\mathrm{Target}}\right)^2,\end{aligned} \tag{5}$$

where we obtain the value of $\Delta^*\psi^{\mathrm{Target}}$ from the off-line EFIT results as well.

To acknowledge difference between the two cost functions ϵ and ϵ^{new}, we first discuss the results. Figure 3 shows the outputs of the two trained networks with the cost function (a) ϵ and (b) ϵ^{new}. It is evident that in both cases the network output ψ^{NN} (red dashed line) reproduces the off-line EFIT ψ^{Target} (black line). However, only the network trained with the cost function ϵ^{new} reproduces the off-line EFIT $\Delta^*\psi^{\mathrm{Target}}$. Both networks are trained well, but the network with the cost function ϵ does not achieve our goal, that is correctly predicting ψ^{Target} *and* $\Delta^*\psi^{\mathrm{Target}}$.

Since our goal is to develop a neural network that solves the GS equation, we choose the cost function to be ϵ^{new} to train the networks. We optimize the weighting factors by minimizing ϵ^{new} with the Adam [56] which is

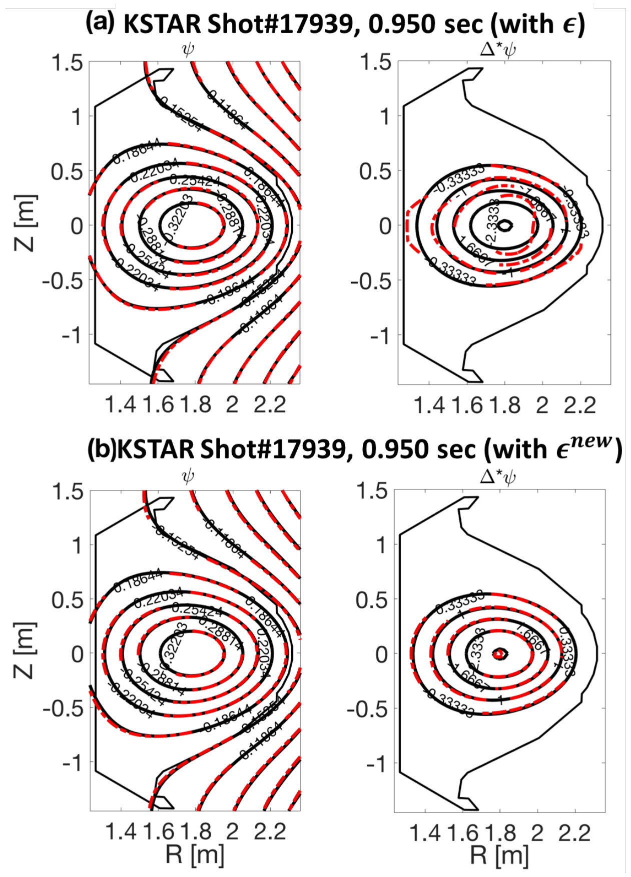

FIG. 3. An example of the two networks' results trained with the cost function (a) ϵ and (b) ϵ^{new} for KSTAR shot# 17939 at 0.950 sec. Both networks (red dashed line) reproduce the ψ^{Target} (black line) well (left panels), but only the network trained with ϵ^{new} reproduces $\Delta^*\psi^{\text{Target}}$ (right panels).

one of the gradient-based optimization algorithms. With 90% and 10% of the total data samples for training and validation of the networks, respectively, we stop training the networks with a fixed number of iterations that is large enough but not too large such that the validation errors do not increase, i.e., to avoid overfitting problems. The whole workflow is carried out with Python and Tensorflow [57].

With the selected cost function we create three different networks that differ only by the training data sets. NN_{2017}, NN_{2018} and $\text{NN}_{2017,\,2018}$ refer to the three networks trained with the data sets from only 2017 (744 discharges), from only 2018 (374 discharges) and from both 2017 and 2018 (744 + 374 discharges) campaigns, respectively.

The descending feature of the cost function ϵ^{new} as a function of the training iteration for $\text{NN}_{2017,2018}$ network is shown in Figure 4. Both the training errors (blue line) and validation errors (red dashed line) decrease together with similar values which means that the network is well generalized. Furthermore, since the validation errors do not increase, the network does not have an overfitting problem. Note that fluctuations in the errors, i.e., standard deviation of the errors, are represented as shaded areas.

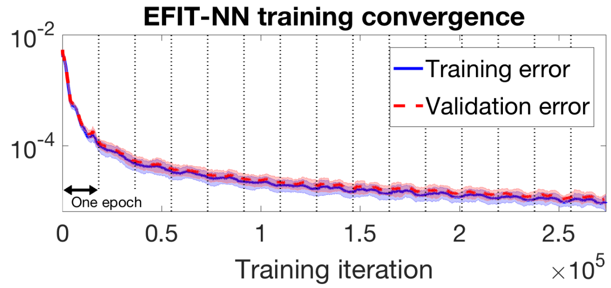

FIG. 4. The descending feature of training (blue line) and validation (red dashed line) errors as a function of iterations. Shaded areas represent standard deviation of the errors.

Small undulations repeated over the iterations in Figure 4 are due to the mini-batch learning. Contrary to the batch learning, i.e., optimizing the network with the entire training set in one iteration, the mini-batch learning divides the training set into some number of small subsets (1, 000 subsets for our case) to optimize the networks sequentially. One cycle that goes through all the subsets once is called an epoch. The mini-batch learning helps to escape from local minima in the weighting factor space [58] via the stochastic gradient descent scheme [59].

IV. PERFORMANCE OF THE DEVELOPED NEURAL NETWORKS: BENCHMARK TESTS

In this section, we present how well the developed networks perform. Main figures of merit we use are peak signal-to-noise ratio (PSNR) and mean structural similarity (MSSIM) as have been used pervиously [34] in addition to the usual statistical quantity R^2, coefficient of determination. We note that obtaining full flux surface information $\psi\,(R, Z)$ on 22×13 or 65×65 spatial grids with our networks takes less than 1 msec on a typical personal computer.

First, we discuss the benchmark results of the $\text{NN}_{2017,2018}$ network. Then, we compare the performance of NN_{2017}, NN_{2018} and $\text{NN}_{2017,2018}$ networks. Here, we also investigate cross-year performance, for instance, applying the NN_{2017} network to predict the discharges obtained from 2018 campaign and vice versa. Then, we evaluate the performance of the networks against the rt-EFIT results to examine possibility of supplementing or even replacing the rt-EFIT with the networks. Finally, we show how the imputation scheme supports the networks' performance. Here, all the tests are performed with the unseen (to all three networks, i.e., NN_{2017}, NN_{2018} and $\text{NN}_{2017,2018}$) KSTAR discharges which are 88 and 75 KSTAR discharges from 2017 and 2018 campaigns, respectively.

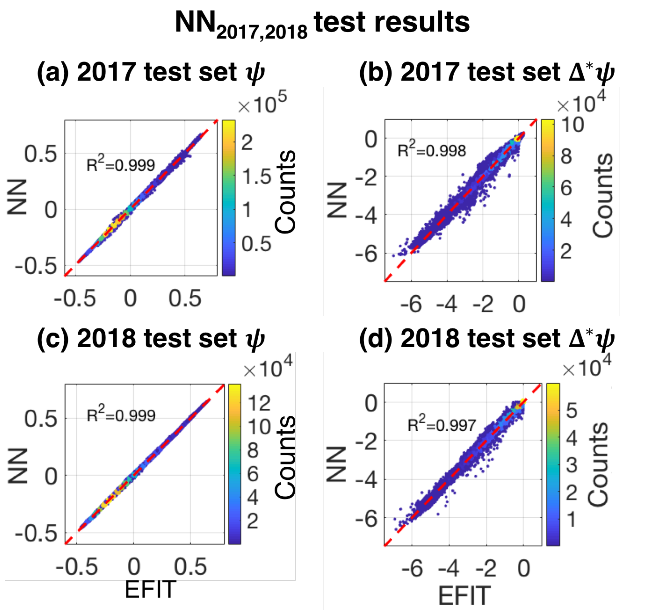

FIG. 5. Performance tests of the $\mathrm{NN}_{2017,2018}$ network on the unseen KSTAR discharges from (a)(b) 2017 campaign and (c)(d) 2018 campaign. The values of R^2 and histograms of (a)(c) ψ^{NN} vs. ψ^{Target} and (b)(d) $\Delta^*\psi^{\mathrm{NN}}$ vs. $\Delta^*\psi^{\mathrm{Target}}$ with colors representing number of counts manifest goodness of the $\mathrm{NN}_{2017,2018}$ network. Red dashed line is the $y = x$ line.

A. Benchmark results of the $\mathrm{NN}_{2017,2018}$ network

Figure 5 show the benchmark results of the $\mathrm{NN}_{2017,2018}$ network, i.e., network trained with the data sets from both 2017 and 2018 campaigns. (a) and (b) show the results with the test discharges from 2017 campaign; while (c) and (d) present the results with the test discharges from 2018 campaign. Histograms of (a)(c) ψ^{NN} vs. ψ^{Target} and (b)(d) $\Delta^*\psi^{\mathrm{NN}}$ vs. $\Delta^*\psi^{\mathrm{Target}}$ are shown with colors representing the number of counts. For instance, there is a yellow colored point in Figure 5(a) around $(-0.1, -0.1) \pm \varepsilon$, where ε is a bin size for the histogram. Since yellow represents about 2×10^5 counts, there are approximately 2×10^5 data whose neural network values and EFIT values are $-0.1 \pm \varepsilon$ simultaneously within our test data set. Note that each KSTAR discharge contains numerous time slices whose number depends on the actual pulse length of a discharge, and each time slice generates the total of $22 \times 13 = 286$ data points. The values of ψ^{Target} and $\Delta^*\psi^{\mathrm{Target}}$ are obtained from the off-line EFIT results. It is clear that the network predicts the target values well.

As a figure of merit, we introduce the R^2 metric (coefficient of determination) defined as

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{L} \left(y_i^{\mathrm{Target}} - y_i^{\mathrm{NN}} \right)^2}{\sum_{i=1}^{L} \left(y_i^{\mathrm{Target}} - \frac{1}{L} \sum_{j=1}^{L} y_j^{\mathrm{Target}} \right)^2}, \qquad (6)$$

where y takes either ψ or $\Delta^*\psi$, and L is the number of test data sets. The calculated values are written in Figure 5, and they are indeed close to unity, implying the existence of very strong linear correlations between the predicted (from the network) and target (from the off-line EFIT) values. Note that $\mathrm{R}^2 = 1$ means the perfect prediction. The red dashed lines on the figures are the $y = x$ lines.

Figure 6 is an example of reconstructed magnetic equilibria using KSTAR shot #18057 from 2017 campaign. (a) shows the evolution of the plasma current. The vertical dashed lines indicate the time points where we show and compare the equilibria obtained from the network (red) and the off-line EFIT (black) which is our target. (b) and (c) are taken during the ramp-up phase, (d) and (e) during the flat-top phase, and (f) and (g) during the ramp-down phase. In each sub-figure from (b) to (g), left panels compare ψ, and right panels are for $\Delta^*\psi$. We mention that the equilibria in Figure 6 are reconstructed with 65×65 grid points even though the network is trained with 22×13 grid points demonstrating how spatial resolution is flexible in our networks.

For a quantitative assessment of the network, we use an image relevant figure of merit that is peak signal-to-noise ratio (PSNR) [60] (see Appendix B) originally developed to estimate a degree of artifacts due to an image compression compared to an original image. Typical PSNR range for the JPEG image, which preserves the original quality with a reasonable degree, is generally in 30–50 dB [34, 61]. For our case, the networks errors relative to the off-line EFIT results can be treated as artifacts. As listed on Figure 6(b)-(g), PSNR for ψ is very good, while we achieve acceptable values for $\Delta^*\psi$.

B. The NN_{2017}, NN_{2018} and $\mathrm{NN}_{2017,2018}$ networks

Similar to shown in Figure 5, we show the benchmark results of the NN_{2017} (trained with the data sets from 2017 campaign) and the NN_{2018} (trained with the data sets from 2018 campaign) in Figures 7 and 8, respectively. R^2 metric is also provided on the figures. Again, overall performance of the networks are good.

The NN_{2017} and NN_{2018} networks are trained with only in-campaign data sets, e.g., NN_{2018} with the data sets from only 2018 campaign, and we find slightly worse results, but still good, on predicting cross-campaign magnetic equilibria, e.g. NN_{2018} predicting equilibria for 2017 campaign. Notice that the NN_{2017} seems to predict cross-campaign equilibria better than in-campaign ones by comparing Figure 7(a) and (c) which contradicts our intuition. Although the histogram in Figure 7(c) seems tightly aligned with the $y = x$ line (red dashed line), close inspection reveals that the NN_{2017} network, in general, underestimates the off-line EFIT results from 2018 campaign marginally. This will be evident when we compare image qualities.

Mean structural similarity (MSSIM) [62] (see Ap-

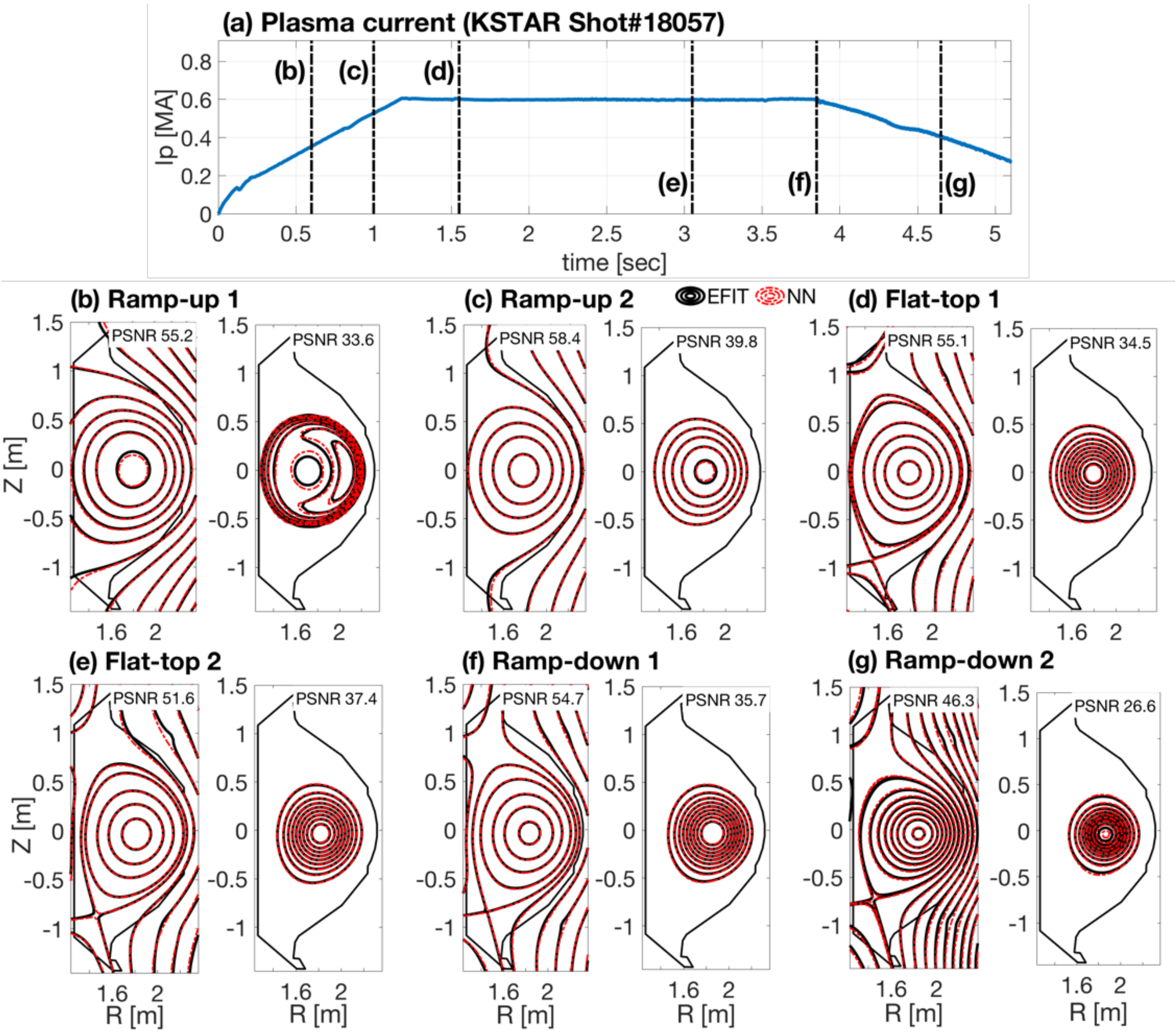

FIG. 6. The actual reconstruction results for the KSTAR shot#18057, comparing the network results and off-line EFIT reconstructions for ramp-up ((b) and (c)), flat-top ((d) and (e)), ramp-down ((f) and (g)) phases following (a) the plasma current evolution. Black lines indicate the flux surfaces from the off-line EFIT, overlaid with the red dotted lines which stand for the NN reconstructions. As a figure of merit, magnitudes of PSNR metric are written on each figure.

pendix B) is another image relevant figure of merit used to estimate perceptual similarity (or perceived differences) between the true and reproduced images based on inter-dependence of adjacent spatial pixels in the images. MSSIM ranges from zero to one, where the closer to unity the better the reproduced image is.

Together with PSNR, Figure 9 shows MSSIM for (a) NN_{2017}, (b) NN_{2018} and (c) $\mathrm{NN}_{2017,2018}$ where the off-line EFIT results are used as reference. Notice that counts in all the histograms of MSSIM and PSNR in this work correspond to the number of reconstructed magnetic equilibria (or a number of time slices) since we obtain a single value of MSSIM and PSNR from one equilibrium; whereas counts in Figures 5, 7 and 8 are much bigger since $286(= 22 \times 13)$ data points are generated from each time slice. Red (green) line indicates the test results on the data sets from 2017 (2018) campaign. In general, whether the test data sets are in-campaign or cross-campaign, image reproducibility of all three networks, i.e., predicting the off-line EFIT results, is good as attested by the fact that MSSIM is quite close to unity and PSNR for ψ ($\Delta^*\psi$) ranges approximately 40 to 60 (20 to 40). It is easily discernible that in-campaign results are better for both NN_{2017} and NN_{2018} unlike what we noted in Figure 7(a) and (c). Not necessarily guaranteed, we find that the $\mathrm{NN}_{2017,2018}$ network works equally well for both campaigns as shown in Figure 9(c).

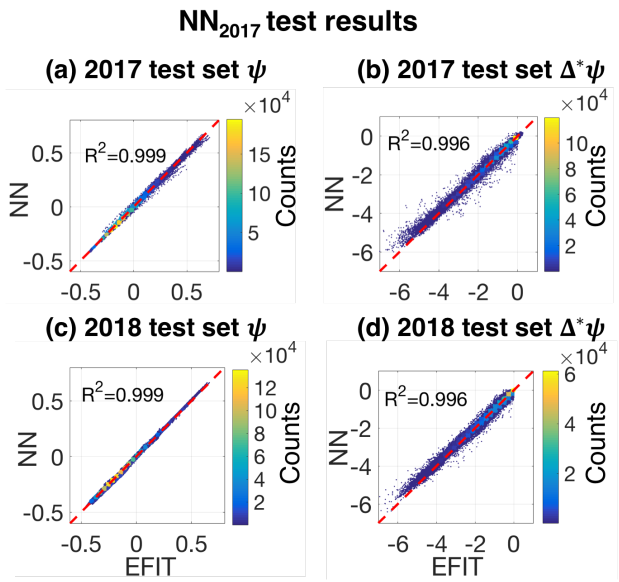

FIG. 7. Same as Figure 5 for the the NN_{2017} network, i.e., trained with the data sets from 2017 campaign.

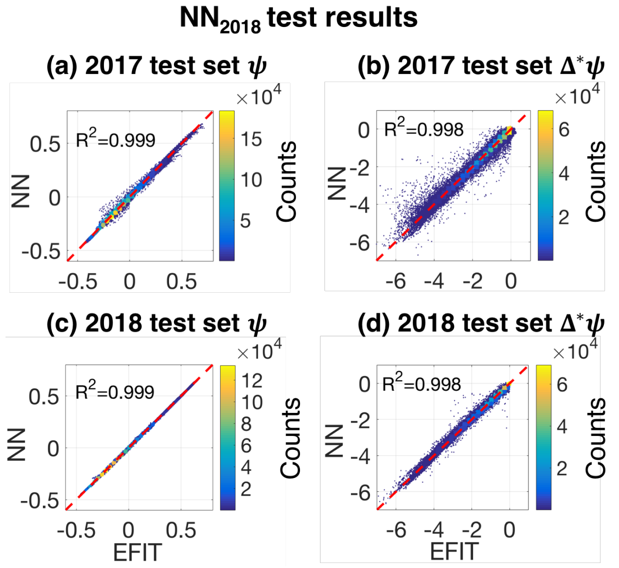

FIG. 8. Same as Figure 5 for the the NN_{2018} network, i.e., trained with the data sets from 2018 campaign.

C. Comparisons among nn-EFIT, rt-EFIT and off-line EFIT

It is widely recognized that rt-EFIT results and off-line results are different from each other. If we allow the off-line EFIT results used to train the networks to be accurate ones, then the reconstruction of equilibria with the neural networks (nn-EFIT) must satisfy the following criterion: nn-EFIT results must be more similar to the off-line EFIT results than rt-EFIT results are to the off-line EFIT as mentioned in Section I. Once this criterion

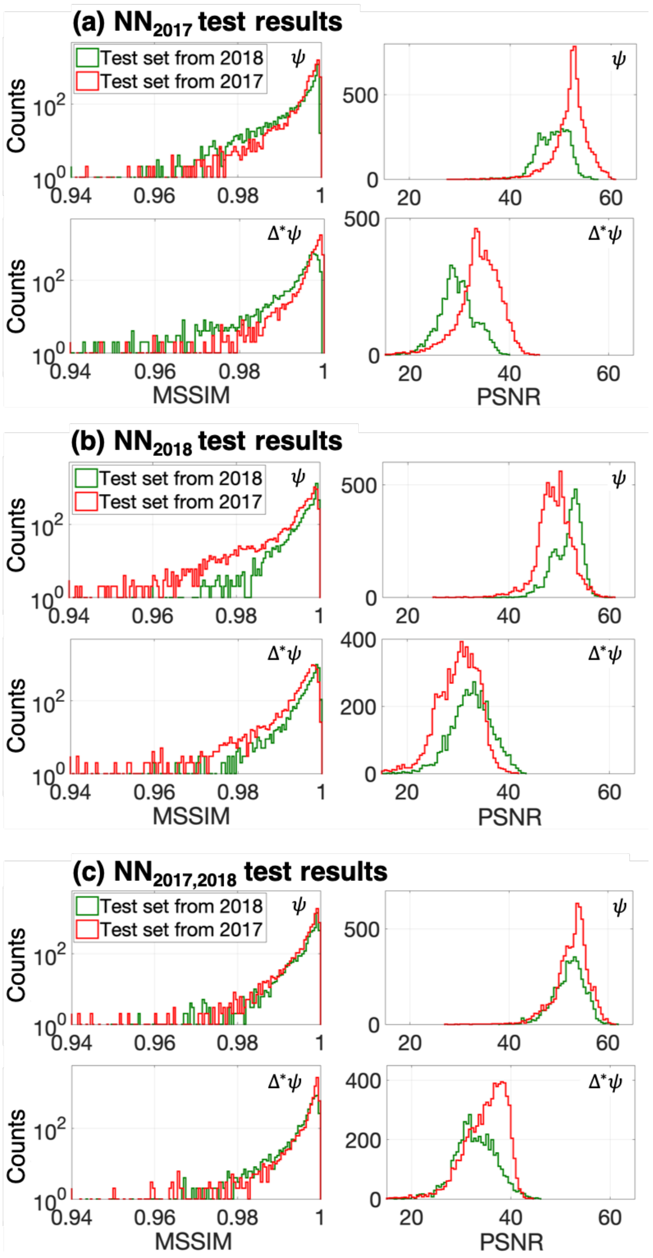

FIG. 9. Histograms of MSSIM (left panel) and PSNR (right panel) for (a) NN_{2017}, (b) NN_{2018} and (c) $\mathrm{NN}_{2017,2018}$. Red (green) line indicates the test results on the data sets from 2017 (2018) campaign. In each sub-figure, top (bottom) panel show the results for ψ ($\Delta^*\psi$). The off-line EFIT results are used as reference.

is satisfied, then we can always improve the nn-EFIT as genuinely more accurate EFIT results are collected. For this reason, we make comparisons among the nn-EFIT, rt-EFIT and off-line EFIT results.

Figure 10 shows an example of reconstructed magnetic equilibria for (a) rt-EFIT vs. off-line EFIT and (b) nn-EFIT (the $\mathrm{NN}_{2017,2018}$ network) vs. off-line EFIT for KSTAR shot #17975 at 0.7 sec with ψ (left panel) and $\Delta^*\psi$ (right panel). Green, red and black lines indicate

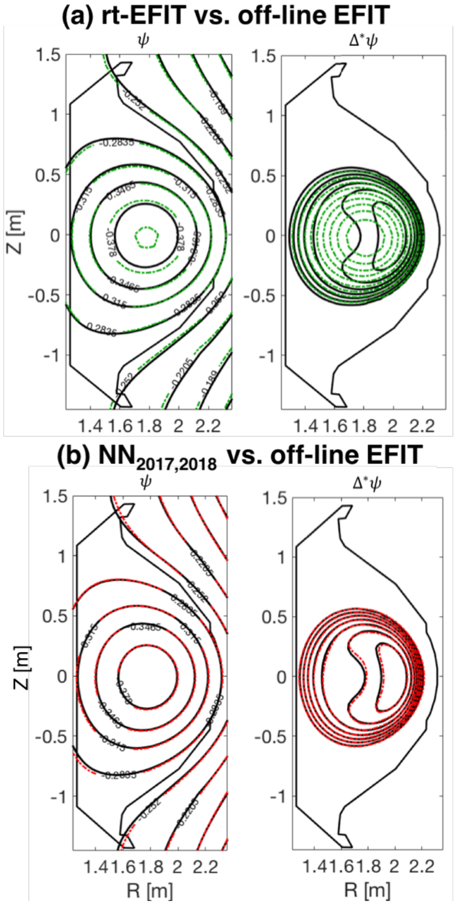

FIG. 10. An example of reconstructed $\psi(R,Z)$ (left panel) and $\Delta^*\psi(R,Z)$ (right panel) for KSTAR shot #17975 at 0.7 sec comparing (a) rt-EFIT (green) and off-line EFIT (black) and (b) nn-EFIT ($\mathrm{NN}_{2017,2018}$) (red) and off-line EFIT (black).

rt-EFIT, nn-EFIT and off-line EFIT results, respectively. This simple example shows that the nn-EFIT is more similar to the off-line EFIT than the rt-EFIT is to the off-line EFIT, satisfying the aforementioned criterion.

To validate the criterion statistically, we generate histograms of MSSIM and PSNR for the nn-EFIT and the rt-EFIT with reference to the off-line EFIT. This is shown in Figure 11 as histograms, where MSSIM (left panel) and PSNR (right panel) of ψ (top) and $\Delta^*\psi$ (bottom) are compared between the nn-EFIT (black) and the rt-EFIT (green). Here, the nn-EFIT results are obtained with the $\mathrm{NN}_{2017,2018}$ network on the test data sets. We confirm that the criterion is satisfied with the $\mathrm{NN}_{2017,2018}$ network as the histograms in Figure 11 are in favour of the nn-EFIT, i.e., larger MSSIM and PSNR are obtained by the nn-EFIT. This is more conspicuous for $\Delta^*\psi$ than ψ.

We perform the similar statistical analyses for the other two networks, NN_{2017} and NN_{2018}, which are shown in Figures 12 and 13. Since these two networks are trained with the data sets from only one campaign, we

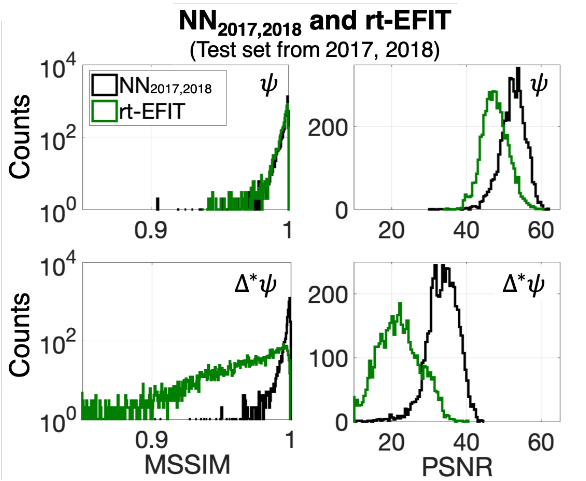

FIG. 11. Histograms of MSSIM (left panel) and PSNR (right panel) of ψ (top) and $\Delta^*\psi$ (bottom) calculated by the nn-EFIT (black) and the rt-EFIT (green), where the nn-EFIT is the $\mathrm{NN}_{2017,2018}$. For both the nn-EFIT and the rt-EFIT, the off-line EFIT is treated as reference.

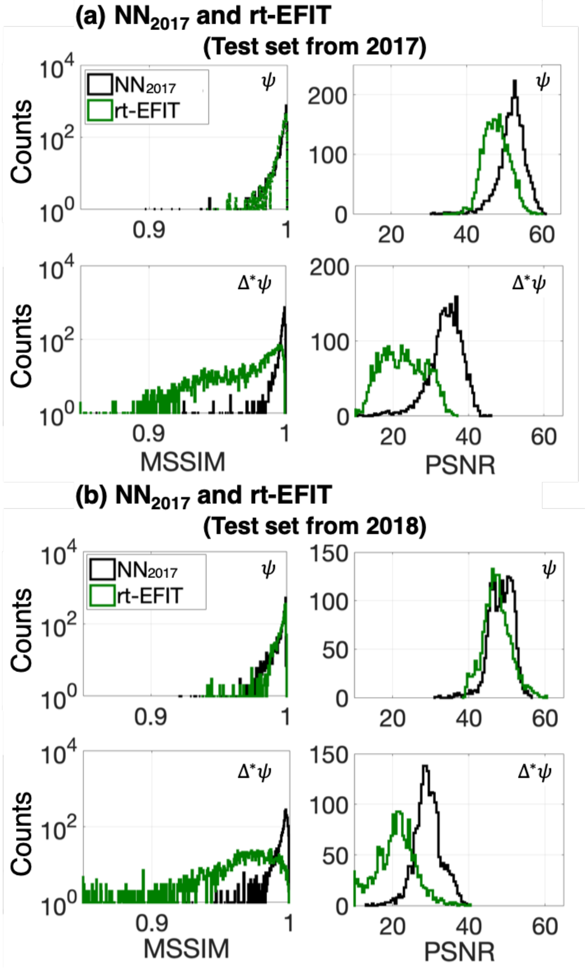

FIG. 12. Same as Figure 11 with the NN_{2017} as the nn-EFIT where the test data sets are obtained from (a) 2017 campaign and (b) 2018 campaign.

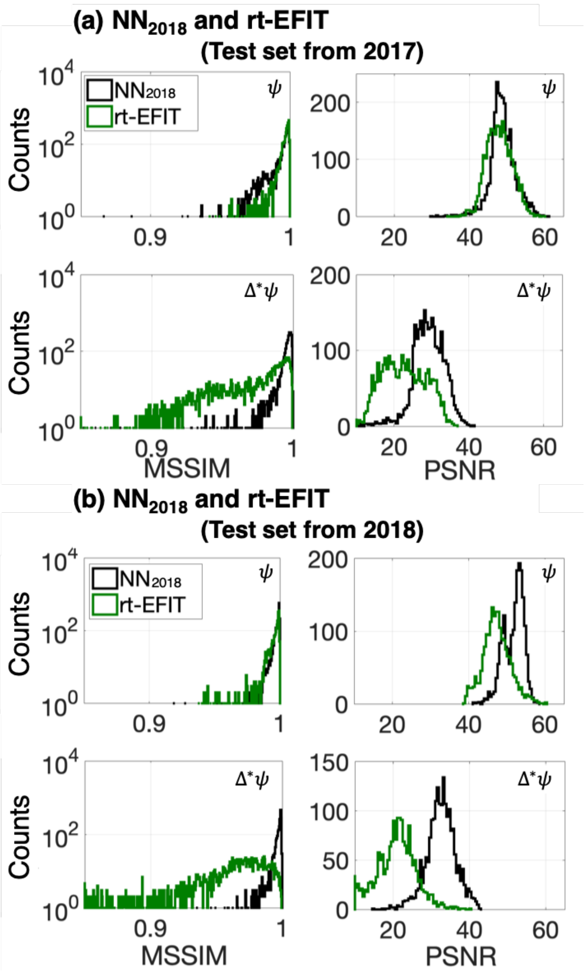

FIG. 13. Same as Figure 11 with the NN$_{2018}$ as the nn-EFIT where the test data sets are obtained from (a) 2017 campaign and (b) 2018 campaign.

show the results where the test data sets are prepared from (a) 2017 campaign or (b) 2018 campaign so that in-campaign and cross-campaign effects can be assessed separately. We find that whether in- or cross-campaign, the criterion is fulfilled for both ψ and $\Delta^*\psi$.

D. The NN$_{2017,2018}$ network with the imputation scheme

If one or a few magnetic pick-up coils which are a part of the inputs to the nn-EFIT are impaired, then we will have to re-train the network without the damaged ones or hope that the network will reconstruct equilibria correctly by padding a fixed value, e.g., zero-padding, to the broken ones. Of course, one can anticipate training the network by considering possible combinations of impaired magnetic pick-up coils. With the total number of 68 signals from the magnetic pick-up coils being inputs to the network in our case, we immediately find that the number of possible combinations increases too quickly to consider it as a solution.

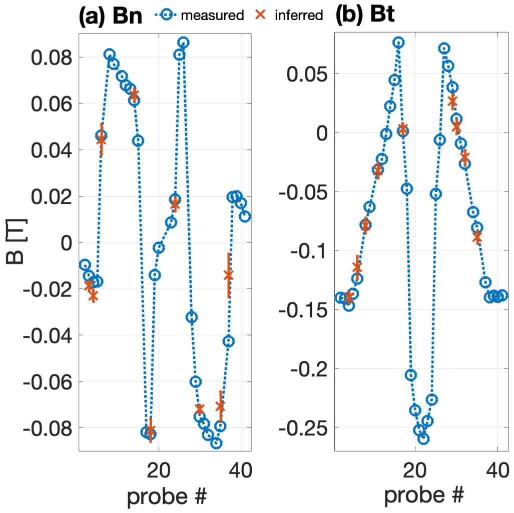

FIG. 14. Measured (blue open circles) and inferred with the imputation method [50] (red crosses with their uncertainties) values for (a) B_n and (b) B_t. Probe # on the horizontal axis is used as an identification index of magnetic pick-up coils. Inferred probes are Probe #3, 4, 6, 14, 18, 24, 30, 35, 37 for B_n and Probe #4, 6, 8, 11, 17, 29, 30, 32, 35 for B_t.

TABLE II. The imputation results shown in Figure 14 with KSTAR shot #20341 at 2.1 sec.

B_n [T] $\times 10^{-2}$			B_t [T] $\times 10^{-2}$		
No.	Measured	Inferred	No.	Measured	Inferred
3	-1.45	-1.88±0.22	4	-14.69	-13.97±0.47
4	-1.72	-2.31±0.24	6	-12.38	-11.42±0.97
6	4.62	4.45±0.65	8	-7.82	-7.88±0.67
14	6.13	6.36±0.27	11	-3.15	-3.22±0.65
18	-8.27	-8.11±0.48	17	0.10	0.30±0.52
24	1.86	1.65±0.30	29	3.84	2.65±0.64
30	-7.52	-7.19±0.18	30	1.15	0.49±0.61
35	-7.93	-7.08±0.65	32	-2.65	-2.11±0.62
37	-4.27	-1.41±0.93	35	-8.07	-8.87±0.55

Since inferring the missing values is better than the null replacement [49], we resolve the issue by using the recently proposed imputation method [50] based on Gaussian processes (GP) [52] and Bayesian inference [51], where the likelihood is constructed based on Maxwell's equations. The imputation method infers the missing values fast enough, i.e., less than 1 msec to infer at least up to nine missing values on a typical personal computer; thus, we can apply the method during a plasma discharge by replacing the missing values with the real-time inferred values.

We have applied the imputation method to KSTAR shot #20341 at 2.1 sec for the normal (B_n) and tangential (B_t) components of the magnetic pick-up coils as an example. We have randomly chosen nine signals from the 32 B_n measurements and another nine from the 36 B_t

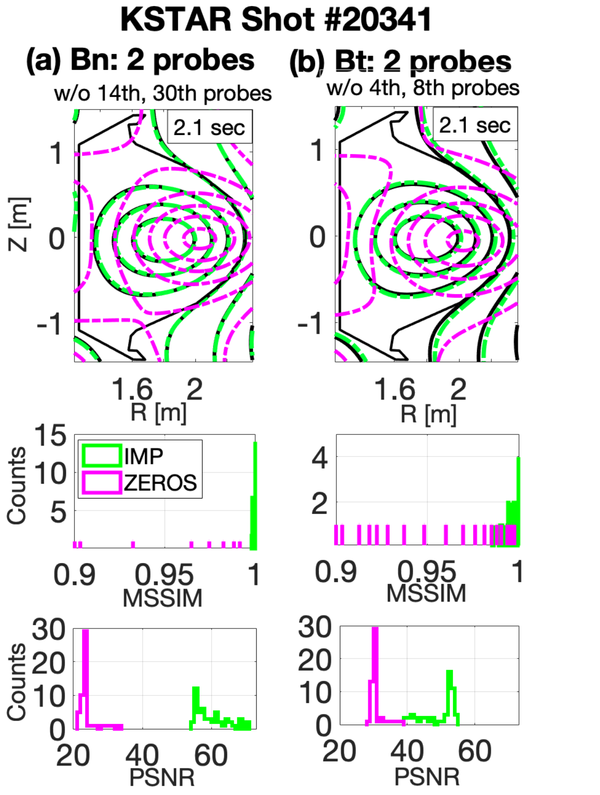

FIG. 15. Top panel: nn-EFIT ($\mathrm{NN}_{2017,2018}$ network) reconstructed equilibria without any missing values (black line), and with two missing values replaced with the inferred values using the imputation method (green line) or with the zeros using the zero-padding method (pink dashed line), where the missing values are (a) B_{n} Probe #14 and 30 (left panel) and (b) B_{t} Probe #4 and 8 (right panel). Bottom panels: histograms of MSSIM and PSNR using the imputation method (green) and the zero-padding method (pink) for all the equilibria obtained from KSTAR shot #20341, where the reference values are those obtained using nn-EFIT without any missing values. Note that there are many more counts less than 0.9 for MSSIM with the zero-padding method.

measurements and pretended that all of them $(9+9)$ are missing simultaneously. Figure 14 shows the measured (blue open circles) and the inferred (red crosses with their uncertainties) values for (a) B_{n} and (b) B_{t}. Probe # on the horizontal axis is used as an identification index of the magnetic pick-up coils. Table II provides the actual values of the measured and inferred ones for better comparisons. We find that the imputation method infers the correct (measured) values very well except Probe #37 of B_{n}. Inferred (missing) probes are Probe #3, 4, 6, 14, 18, 24, 30, 35, 37 for B_{n} and Probe #4, 6, 8, 11, 17, 29, 30, 32, 35 for B_{t}. Here, we provide all the Probe #'s used for the neural network: B_{n} Probe #[2, . . . , 6, 8, 9, 11, . . . , 15, 17, . . . , 20, 23, . . . , 26, 28, . . . , 32, 34, 35, 37, . . . , 41] (a total of 32) and B_{t} Probe #[2, . . . , 6, 8, 9, 11, . . . , 32, 34, 35, 37, . . . , 41] (a total of 36).

Comparisons between the nn-EFIT without any missing values, which we treat as reference values, and the nn-EFIT with the imputation method or with the zero-padding method are made. Here, nn-EFIT results are obtained using the $\mathrm{NN}_{2017,2018}$ network. Top panel of Figure 15 shows $\psi(R,Z)$ obtained from the nn-EFIT without any missing values (black line) and from the nn-EFIT with the two missing values replaced with the inferred values (green line), i.e., imputation method, or with zeros (pink dashed line), i.e., zero-padding method for (a) B_{n} (left panel) and (b) B_{t} (right panel) at 2.1 sec of KSTAR shot #20341. Probe #14 and 30 for B_{n} and Probe #4 and 8 for B_{t} are treated as the missing ones. Bottom panels compare histograms of MSSIM and PSNR using the imputation method (green) and the zero-padding method (pink) for all the equilibria obtained from KSTAR shot #20341.

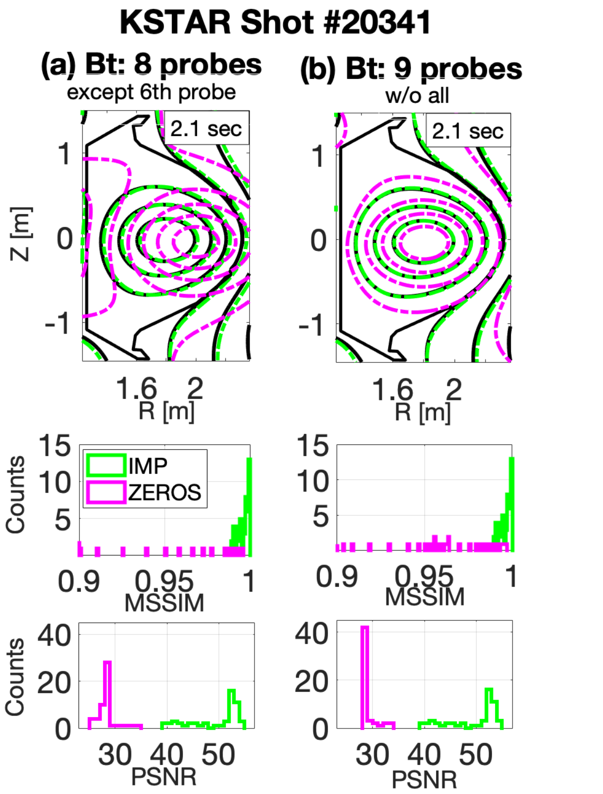

FIG. 16. Same color code as in Figure 15. Missing values are (a) eight B_{t} (except only Probe #6) and (b) all nine B_{t}.

It is clear that nn-EFIT with the imputation method (green line) is not only much better than that with the zero-padding method (pink dashed line) but it also reconstructs the equilibrium close to the reference (black). In fact, the zero-padding method is too far off from the reference (black line) to be relied on for plasma controls.

Motivated by such a successful result of the nn-EFIT with the imputation method on the two missing values, we have increased number of missing values as shown in Figures 16 and 17 for the same KSTAR discharge, i.e., KSTAR shot #20341. Let us first discuss Figure 16 which

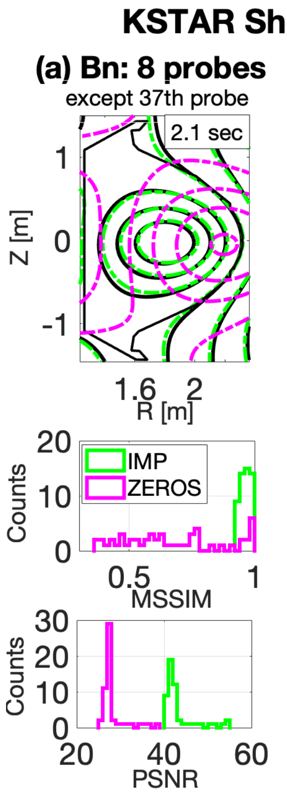

FIG. 17. Same color code as in Figure 15. Missing values are (a) eight B_n (except only Probe #37), (b) all nine B_n.

are with (a) the eight (except only Probe #6) and (b) nine (all) missing values of B_t. Color codes are same as in Figure 15, i.e., the reference is black, and nn-EFIT with the imputation method green or with the zero-padding method pink. It is evident that the nn-EFIT with the imputation method performs well at least up to nine missing values. Such a result is, in fact, expected since the imputation method has inferred the missing values well as shown in Figure 14(b) in addition to the fact that a well-trained neural network typically has a reasonable degree of resistance on noises. Again, the nn-EFIT with the zero-padding method is not reliable.

Figure 17 (a) and (b) are results with the eight (except only Probe #37) and nine (all) missing values of B_n, respectively. Color codes are same as in Figure 15. We find that the nn-EFIT with the eight missing values reconstructs the equilibrium similar to the reference one, while the reconstruction quality becomes notably worse for nine missing values. This is caused mostly due to poor inference of Probe #37 by the imputation method (see Figure 14(a)). Nevertheless, the result is still better than the zero-padding method. Figure 18 shows the reconstruction results with the same color codes as in Figure 15 when we have (a) 4+4 and (b) 9+9 combinations of B_n and B_t missing values simultaneously.

All these results suggest that the nn-EFIT with the imputation method reconstructs equilibria reasonably well

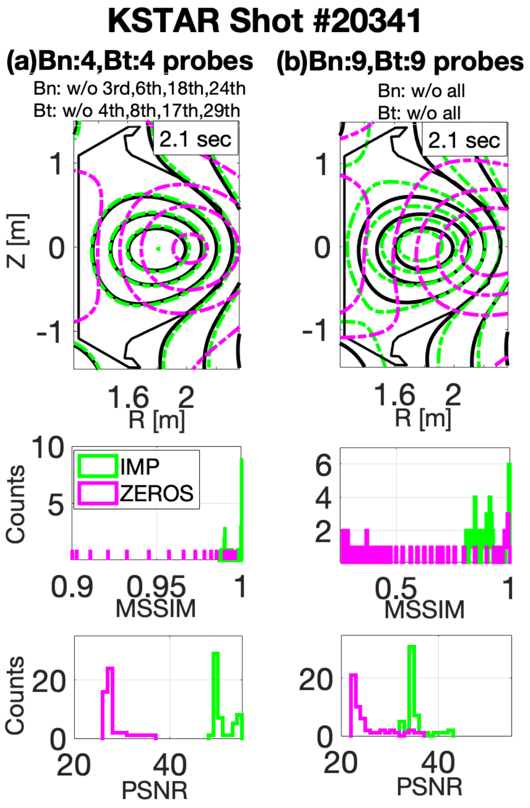

FIG. 18. Same color code as in Figure 15. Combinations of missing B_n and B_t are examined: (a) four missing B_n and four mssing B_t case and (b) nine missing B_n and nine missing B_t case.

except when the imputation infers the true value poorly, e.g., B_n Probe #37 in Figure 14(a) and Table II. In fact, the suggested imputation method [50] infers the missing values based on the neighboring intact values (using Gaussian processes) while satisfying the Maxwell's equations (using Bayesian probability theory). Consequently, such a method becomes less accurate if (1) the neighboring channels are also missing *AND* (2) the true values change fast from the neighboring values. In fact, B_n Probe #37 happens to satisfy these two conditions, i.e., Probe #35 is also missing, and the true values of Probe #35, #37 and #38 are changing fast as one can discern from Figure 14(a).

V. CONCLUSIONS

We have developed and presented the neural network based Grad-Shafranov solver constrained with the measured magnetic signals. The networks take the plasma current from a Rogowski coil, 32 normal and 36 tangential components of the magnetic fields from the magnetic pick-up coils, 22 poloidal fluxes from the flux loops, and (R, Z) position of the interest as inputs. With three fully connected hidden layers consisting of 61 nodes each layer, the network outputs a value of poloidal flux ψ. We set the cost function used to train the networks to be a function of not only the poloidal flux ψ but also the Grad-Shafranov equation $\Delta^*\psi$ itself. The networks are trained and validated with $1,118$ KSTAR discharges from 2017 and 2018 campaigns.

Treating the off-line EFIT results as accurate magnetic equilibria to train the networks, our networks fully reconstruct magnetic equilibria, not limited to obtaining selected information such as positions of magnetic axis, X-points or plasma boundaries, more similar to the off-line EFIT results than the rt-EFIT is to the off-line EFIT. Owing to the fact that (R, Z) position is a part of the input, our networks have adjustable spatial resolution within the first wall. The imputation method supports the networks to obtain the nn-EFIT results even if there exist a few missing inputs.

As all the necessary computation time is approximately 1 msec, the networks have potential to be used for real-time plasma control. In addition, the networks can be used to provide large number of automated EFIT results fast for many other data analyses requiring magnetic equilibria.

ACKNOWLEDGEMENT

This research is supported by National R&D Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (grant numbers NRF-2017M1A7A1A01015892 and NRF-2017R1C1B2006248) and the KUSTAR-KAIST Institute, KAIST, Korea.

Appendix A: Real-time preprocess on magnetic signals

As shown in Figure 2 and discussed in Section II, normal $(B_{\rm n})$ and tangential $(B_{\rm t})$ components of magnetic fields measured by the magnetic pick-up coils and poloidal magnetic fluxes $(\Psi_{\rm FL})$ measured by the flux loops tend to have residual drifts after calibrating the magnetic diagnostics (MDs). We train the neural networks with preprocessed, i.e., drift adjusted, magnetic signals. Therefore, we must be able to preprocess the signals in real time as well. Here, we introduce how we preprocess the magnetic signals in detail. The same preprocess is applied to all the training, validation and test data sets. Note that we do not claim that how we adjust the magnetic signals *corrects* the drifts completely.

1. Real-time drift adjustment with information obtained during the initial magnetization stage

To adjust the signal drifts, we deem *a priori* that the signals drift linearly in time [63–65]. Of course, nonlinear drift may well exist in the signals. However, we need to come up with a very simple and fast solution to adjust the drifts in real time with the limited amount of information. One can consider such linearization in time as taking up to the first order of Taylor expanded drifting signals. Therefore, we take the drifting components of the signals (y_i^m) from various types (the magnetic pick-up coils or the flux loops) of MDs to follow:

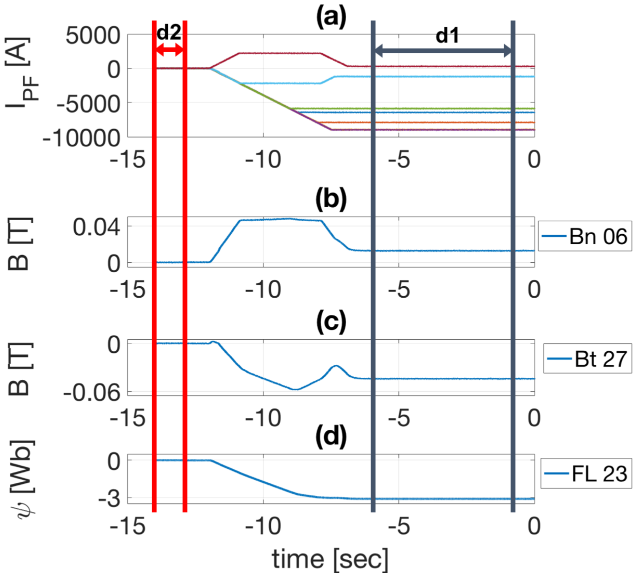

FIG. 19. An example of temporal evolutions of (a) currents in the PF coils, (b) normal and (c) tangential components of magnetic fields measured by the magnetic pick-up coils, respectively, and (d) poloidal flux measured by one of the flux loops during the initial magnetization stage, i.e., $t < 0$, for a typical KSTAR discharge. Information from the time interval d1 (d2) is used to estimate a_i^m (b_i^m).

$$y_i^m = a_i^m t + b_i^m, \qquad \text{(A1)}$$

where t is the time. a_i^m and b_i^m are the slope and the offset, respectively, of a drift signal for the $i^{\rm th}$ magnetic sensor of a type m (magnetic pick-up coils or flux loops). Then, our goal simply becomes finding a_i^m and b_i^m for all i and m of interests before a plasma starts or the blip time $(t = 0)$ so that y_i^m can be subtracted from the measured magnetic signals in real-time, i.e., preprocessing the magnetic signals for the neural networks.

We use two different time intervals during the initial magnetization stage, i.e., before the blip time, for every plasma discharge to find a_i^m and b_i^m, sequentially. Figure 19 shows an example of temporal evolutions of currents in the poloidal field (PF) coils, $B_{\rm n}$ and $B_{\rm t}$ and poloidal magnetic flux up to the blip time $(t = 0)$ of a typical KSTAR discharge.

During the time interval d1 in Figure 19, all the magnetic signals must be constant in time because there are no changes in currents of all the PF coils as well as there are no plasmas yet that can change the magnetic signals.

Therefore, any temporal changes in a magnetic signal during d1 can be regarded as due to a non-zero a_i^m. With the knowledge of a_i^m from d1 time interval, we obtain the value of b_i^m using the fact that all the magnetic signals must be zeros during the time interval d2 because there are no sources of magnetic fields, i.e., all the currents in the PF coils are zeros.

Summarizing our procedure, (1) we first obtain the slopes a_i^m based on the fact that all the magnetic signals must be constant in time during d1 time interval, and then (2) find the offsets b_i^m based on the fact that all the magnetic signals, after the linear drifts in time are removed based on the knowledge of a_i^m, must be zeros during d2 time interval.

2. Bayesian inference

Bayesian probability theory [51] has a general form of

$$p\left(\mathcal{W}|\mathcal{D}\right)=\frac{p\left(\mathcal{D}|\mathcal{W}\right)p\left(\mathcal{W}\right)}{p\left(\mathcal{D}\right)}, \tag{A2}$$

where $\mathcal{W}$ is a (set of) parameter(s) we wish to infer, i.e., a_i^m and b_i^m for our case, and $\mathcal{D}$ is the measured data, i.e., measured magnetic signals during the time intervals of d1 and d2 in Fig. 19. The *posterior* $p\left(\mathcal{W}|\mathcal{D}\right)$ provides us probability of having a certain value for $\mathcal{W}$ given the measured data $\mathcal{D}$ which is proportional to a product of *likelihood* $p\left(\mathcal{D}|\mathcal{W}\right)$ and *prior* $p\left(\mathcal{W}\right)$. Then, we use the *maximum a posterior* (MAP) to select the value of $\mathcal{W}$. The *evidence* $p\left(\mathcal{D}\right)$ (or marginalized *likelihood*) is typically used for a model selection and is irrelevant here as we are only interested in estimating the parameter $\mathcal{W}$, i.e., a_i^m and b_i^m.

We estimate values of the slope a_i^m and the offset b_i^m based on Equation (A2) in two steps as described above:

$$\text{Step (1)}\;:\;p(a_i^m|\vec{\mathcal{D}}^m_{i,d1})\propto p(\vec{D}^m_{i,d1}|a_i^m)p(a_i^m), \tag{A3}$$

$$\text{Step (2)}\;:\;p(b_i^m|\vec{\mathcal{D}}^m_{i,d2},a_i^{m*})\propto p(\vec{D}^m_{i,d2}|b_i^m,a_i^{m*})p(b_i^m), \tag{A4}$$

where $\vec{\mathcal{D}}^m_{i,d1}$ ($\vec{\mathcal{D}}^m_{i,d2}$) are the time series data from the i^{th} magnetic sensor of a type m (magnetic pick-up coils or flux loops) during the time intervals of d1 (d2) as shown in Fig. 19. a_i^{m*} is the MAP, i.e., the value of a_i^m maximizing the *posterior* $p(a_i^m|\vec{\mathcal{D}}^m_{i,d1})$. Since we have no prior knowledge on a_i^m and b_i^m, we take *priors*, $p(a_i^m)$ and $p(b_i^m)$, to be uniform allowing all the real numbers. Note that a correct $p(a_i^m)$ would be equal to $1/\left[\pi\left(1+(a_i^m)^2\right)\right]$ [66], but we sacrifice rigor to obtain a fast solution. Furthermore, the *posterior* for b_i^m should, rigorously speaking, be obtained by marginalizing over all possible a_i^m, i.e., $p(b_i^m|\vec{\mathcal{D}}^m_{i,d2})=\int p(b_i^m|\vec{\mathcal{D}}^m_{i,d2},a_i^m)p(a_i^m|\vec{\mathcal{D}}^m_{i,d1})da_i^m$. Again, as we are interested in real-time application, such a step is simplified just to use a_i^{m*}.

With Equation (A1), we model *likelihoods*, $p(\vec{D}^m_{i,d1}|a_i^m)$ and $p(\vec{D}^m_{i,d2}|b_i^m,a_i^{m*})$, as Gaussian:

$$p(\vec{D}^m_{i,d1}|a_i^m)=\frac{1}{\sqrt{(2\pi)^L|\sigma^m_{i,d1}|}}\times\exp\left(-\frac{\sum\limits_{t_l\in d1}^{L}\left[a_i^m(t_l-t_0)-\left(D^m_{i,d1}(t_l)-\left\langle D^m_{i,d1}(t_0)\right\rangle\right)\right]^2}{2(\sigma^m_{i,d1})^2})\right), \tag{A5}$$

$$p(\vec{D}^m_{i,d2}|b_i^m,a_i^{m*})=\frac{1}{\sqrt{(2\pi)^K|\sigma^m_{i,d2}|}}\times\exp\left(-\frac{\sum\limits_{t_k\in d2}^{K}\left[b_i^m-\left(D^m_{i,d2}(t_k)-a_i^{m*}t_k\right)\right]^2}{2(\sigma^m_{i,d2})^2}\right), \tag{A6}$$

which simply state that noises in the measured signals follow Gaussian distributions. Here, $\sigma^m_{i,d1}$ and $\sigma^m_{i,d2}$ are the experimentally obtained noise levels for the i^{th} magnetic sensor of a type m (magnetic pick-up coils and flux loops) during the time intervals of d1 and d2 in Figure 19, respectively. t_l and t_k define the actual time intervals of d1 and d2, i.e., $t_l\in[-6,-1]$ sec and $t_k\in[-14,-13]$ sec with L and K being the numbers of the data points in each time interval, respectively. t_0 can be any value within the d1 time interval, and we set $t_0=-2$ sec in this work. $\left\langle D^m_{i,d1}(t_0)\right\rangle$, removing the offset effect to obtain only the slope, is the time averaged value of $D^m_{i,d1}(t)$ for $t\in[t_0-0.5,t_0+0.5]$ sec. We use the time averaged value to minimize the effect of the noise in $D^m_{i,d1}(t)$ at $t=t_0$.

With our choice of uniform distributions for *priors* in Equations (A3) and (A4), MAPs for a_i^m and b_i^m, which we denote them as a_i^{m*} and b_i^{m*}, coincide with the maximum *likelihoods* which can be analytically obtained by maximizing Equations (A5) and (A6) with respect to a_i^m and b_i^m, respectively:

$$a_i^{m*}=\frac{\sum\limits_{t_l\in d1}^{L}\left[\left(D^m_{i,d1}(t_l)-\left\langle D^m_{i,d1}(t_0)\right\rangle\right)(t_l-t_0)\right]}{\sum\limits_{t_l\in d1}^{L}\left[t_l-t_0\right]^2}, \tag{A7}$$

$$b_i^{m*}=\frac{1}{K}\sum_{t_k\in d2}^{K}\left[D^m_{i,d2}(t_k)-a_i^{m*}t_k\right]. \tag{A8}$$

Now, we have attained simple algebraic equations based on Bayesian probability theory which can provide us values of the slope a_i^m and the offset b_i^m before the blip time, i.e., before $t=0$.

Since the required information (a_i^m and b_i^m) to adjust drifts in the magnetic signals is obtained before every discharge starts, we can preprocess the magnetic signals in real time. This is how we have adjusted the drift signals shown in Figure 2.

Appendix B: Image relevant figures of merit - PSNR and MSSIM

In Section IV, we used two image relevant figures of merit, namely PSNR (peak signal-to-noise ratio) [60, 61] and MSSIM (mean structural similarity) [62], to examine performance of the developed neural networks. Although these figures of merit are widely used and well known, we present short descriptions of PSNR and MSSIM for the sake of readers' convenience. Notice that we treat a reconstructed magnetic equilibrium as an image whose dimension (a number of pixels) is set by the spatial grid points.

1. Peak signal-to-noise ratio (PSNR)

PSNR is calculated as

$$\mathrm{PSNR} = 10 \times \log_{10} \left[\frac{\max \left(y^{\mathrm{Target}} \right)^2}{\frac{1}{M} \sum_{i=1}^{M} \left(y_i^{\mathrm{Target}} - y_i^{\star} \right)^2} \right], \quad \text{(B1)}$$

where y_i is the value of either ψ or $\Delta^* \psi$ at the i^{th} position of the spatial grid (analogous to a pixel value of an image), and M for the total number of the grid points, i.e., either $286 (= 22 \times 13)$ or $4225 (= 65 \times 65)$ depending on our choice for reconstructing an equilibrium. $\max(\cdot)$ operator selects the maximum value of an argument, and y^{Target} is an array containing 'pixel' values of a reference EFIT 'image', that is a reconstructed magnetic equilibrium. $y^{\star}$ is also an array, and depending on whether we wish to compare the off-line EFIT result with either rt-EFIT result or nn-EFIT result, we select the corresponding values.

2. Mean structural similarity (MSSIM)

MSSIM is calculated as

$$\mathrm{MSSIM} = \frac{\left(2\mu_{y^{\mathrm{Target}}} \mu_{y^{\star}} + C_1 \right) \left(2\sigma_{y^{\mathrm{Target}} y^{\star}} + C_2 \right)}{\left(\mu^2_{y^{\mathrm{Target}}} + \mu^2_{y^{\star}} + C_1 \right) \left(\sigma^2_{y^{\mathrm{Target}}} + \sigma^2_{y^{\star}} + C_2 \right)}, \quad \text{(B2)}$$

where $\mu_{y^{\mathrm{Target}}}$ and $\mu_{y^{\star}}$ are the mean values of y^{Target} and $y^{\star}$, respectively. Here, y^{Target} and $y^{\star}$ mean the same as in Section B 1. $\sigma^2_{y^{\mathrm{Target}}}$ and $\sigma^2_{y^{\star}}$ are the variances of y^{Target} and $y^{\star}$, respectively; while $\sigma_{y^{\mathrm{Target}} y^{\star}}$ is the covariance between y^{Target} and $y^{\star}$. C_1 and C_2 are used to prevent a possible numerical instability, i.e., denominator being zero, and set to be small numbers. Following [62], we have $C_1 = 10^{-4}$ and $C_2 = 9 \times 10^{-4}$.

REFERENCES

[1] L. L. Lao, H. S. John, R. D. Stambaugh, A. G. Kellman, and W. Pfeiffer, Nuclear Fusion **25**, 1611 (1985).

[2] J. P. Freidberg, *Ideal Magnetohydrodynamics* (Plenum Press, New York, 1987).

[3] L. L. Lao, H. E. S. John, Q. Peng, and J. R. Ferron, Fusion Science and Technology **48**, 968 (2005).

[4] D. P. O'Brien, L. L. Lao, E. R. Solano, M. Garribba, T. S. Taylor, J. G. Cordey, and J. J. Ellis, Nuclear Fusion **32**, 1351 (1992).

[5] S. A. Sabbagh, S. M. Kaye, J. Menard, F. Paoletti, M. Bell, R. E. Bell, J. M. Bialek, M. Bitter, E. D. Fredrickson, D. A. Gates, A. H. Glasser, H. Kugel, L. L. Lao, B. P. LeBlanc, R. Maingi, R. J. Maqueda, E. Mazzucato, D. Mueller, M. Ono, S. F. Paul, M. Peng, C. H. Skinner, D. Stutman, G. A. Wurden, W. Zhu, and N. R. Team, Nuclear Fusion **41**, 1601 (2001).

[6] Q. Jinping, W. Baonian, L. L. Lao, S. Biao, S. A. Sabbagh, S. Youwen, L. Dongmei, X. Bingjia, R. Qilong, G. Xianzu, and L. Jiangang, Plasma Science and Technology **11**, 142 (2009).

[7] Y. 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, J. Kim, H. L. Yang, K. I. You, Y. S. Bae, J. Chung, M. Kwon, Y. K. Oh, W. C. Kim, J. Y. Kim, S. G. Lee, H. K. Park, H. Reimerdes, J. Leuer, and M. Walker, Nuclear Fusion **51**, 053001 (2011).

[8] J. R. Ferron, M. L. Walker, L. L. Lao, H. E. S. John, D. A. Humphreys, and J. A. Leuer, Nuclear Fusion **38**, 1055 (1998).

[9] D. Van Houtte and E. T. SUPRA, Nuclear Fusion **33**, 137 (1993).

[10] A. Ekedahl, L. Delpech, M. Goniche, D. Guilhem, J. Hillairet, M. Preynas, P. Sharma, J. Achard, Y. Bae, X. Bai, C. Balorin, Y. Baranov, V. Basiuk, A. Bécoulet, J. Belo, G. Berger-By, S. Brémond, C. Castaldo, S. Ceccuzzi, R. Cesario, E. Corbel, X. Courtois, J. Decker, E. Delmas, X. Ding, D. Douai, C. Goletto, J. Gunn, P. Hertout, G. Hoang, F. Imbeaux, K. Kirov, X. Litaudon, R. Magne, J. Mailloux, D. Mazon, F. Mirizzi, P. Mollard, P. Moreau, T. Oosako, V. Petrzilka, Y. Peysson, S. Poli, M. Prou, F. Saint-Laurent, F. Samaille, and B. Saoutic, Nuclear Fusion **50**, 112002 (2010).

[11] S. Itoh, K. N. Sato, K. Nakamura, H. Zushi, M. Sakamoto, K. Hanada, E. Jotaki, K. Makino, S. Kawasaki, H. Nakashima, and A. Iyomasa, Plasma Physics and Controlled Fusion **41**, A587 (1999).

[12] H. Zushi, S. Itoh, K. Hanada, K. Nakamura, M. Sakamoto, E. Jotaki, M. Hasegawa, Y. Pan, S. Kulkarni, A. Iyomasa, S. Kawasaki, H. Nakashima, N. Yoshida, K. Tokunaga, T. Fujiwara, M. Miyamoto, H. Nakano, M. Yuno, A. Murakami, S. Nakamura, N. Sakamoto, K. Shinoda, S. Yamazoe, H. Akanishi, K. Kuramoto, Y. Matsuo, A. Iwamae, T. Fuijimoto, A. Komori, T. Morisaki, H. Suzuki, S. Masuzaki, Y. Hirooka, Y. Nakashima, and O. Mitarai, Nuclear Fusion **43**, 1600 (2003).

[13] B. Saoutic, Plasma Physics and Controlled Fusion **44**, B11 (2002).

[14] H. Park, M. Choi, S. Hong, Y. In, Y. Jeon, J. Ko, W. Ko, J. Kwak, J. Kwon, J. Lee, J. Lee, W. Lee, Y. Nam, Y. Oh, B. Park, J. Park, Y. Park, S. Wang, M. Yoo, S. Yoon, J. Bak, C. Chang, W. Choe, Y. Chu, J. Chung, N. Eidietis, H. Han, S. Hahn, H. Jhang, J. Juhn, J. Kim, K. Kim, A. Loarte, H. Lee, K. Lee, D. Mueller, Y. Na, Y. Nam, G. Park, K. Park, R. Pitts, S. Sabbagh, and G. Y. and, Nuclear Fusion **59**, 112020 (2019).

[15] B. Wan, Y. Liang, X. Gong, N. Xiang, G. Xu, Y. Sun, L. Wang, J. Qian, H. Liu, L. Zeng, L. Zhang, X. Zhang, B. Ding, Q. Zang, B. Lyu, A. Garofalo, A. Ekedahl, M. Li, F. Ding, S. Ding, H. Du, D. Kong, Y. Yu, Y. Yang, Z. Luo, J. Huang, T. Zhang, Y. Zhang, G. Li, T. Xia, and and, Nuclear Fusion **59**, 112003 (2019).

[16] J.-K. Park, Y. Jeon, Y. In, J.-W. Ahn, R. Nazikian, G. Park, J. Kim, H. Lee, W. Ko, H.-S. Kim, N. C. Logan, Z. Wang, E. A. Feibush, J. E. Menard, and M. C. Zarnstroff, Nature Physics **14**, 1223 (2018).

[17] F. Reimold, M. Wischmeier, M. Bernert, S. Potzel, A. Kallenbach, H. Mller, B. Sieglin, and U. S. and, Nuclear Fusion **55**, 033004 (2015).

[18] A. Jaervinen, C. Giroud, M. Groth, P. Belo, S. Brezinsek, M. Beurskens, G. Corrigan, S. Devaux, P. Drewelow, D. Harting, A. Huber, S. Jachmich, K. Lawson, B. Lipschultz, G. Maddison, C. Maggi, C. Marchetto, S. Marsen, G. Matthews, A. Meigs, D. Moulton, B. Sieglin, M. Stamp, and S. W. and, Nuclear Fusion **56**, 046012 (2016).

[19] Yue, X N, Xiao, B J, Luo, Z P, and Guo, Y, Plasma Physics and Controlled Fusion **55**, 085016 (2013).

[20] Huang, Yao, Xiao, B J, Luo, Z P, Yuan, Q P, Pei, X F, and Yue, X N, Fusion Engineering and Design , 1 (2016).

[21] O. Barana, A. Murari, P. Franz, L. C. Ingesson, and G. Manduchi, Review of Scientific Instruments **73**, 2038 (2002).

[22] Murari, A, Arena, P, Buscarino, A, Fortuna, L, Iachello, M, and contributors, JET-EFDA, Nuclear Inst. and Methods in Physics Research, A **720**, 2 (2013).

[23] M. Boyer, S. Kaye, and K. Erickson, Nuclear Fusion **59**, 056008 (2019).

[24] A. Murari, D. Mazon, N. Martin, G. Vagliasindi, and M. Gelfusa, IEEE Transactions on Plasma Science **40**, 1386 (2012).

[25] A. Murari, J. Vega, D. Mazon, D. Patan, G. Vagliasindi, P. Arena, N. Martin, N. Martin, G. Ratt, and V. Caloone, Nuclear Instruments and Methods in Physics Research Section A **623**, 850 (2010).

[26] P. Gaudio, A. Murari, M. Gelfusa, I. Lupelli, and J. Vega, Plasma Physics and Controlled Fusion **56**, 114002 (2014).

[27] Kates-Harbeck, Julian, Svyatovskiy, Alexey, and Tang, William, Nature **568**, 526 (2019).

[28] B. Cannas, A. Fanni, G. Pautasso, G. Sias, and P. Sonato, Nuclear Fusion **50**, 075004 (2010).

[29] A. Pau, A. Fanni, S. Carcangiu, B. Cannas, G. Sias, A. Murari, and F. R. and, Nuclear Fusion **59**, 106017 (2019).

[30] Meneghini, O, Luna, C J, Smith, S P, and Lao, L L, Physics of Plasmas **21**, 060702 (2014).

[31] O. Meneghini, S. P. Smith, P. B. Snyder, G. M. Staebler, J. Candy, E. Belli, L. Lao, M. Kostuk, T. Luce, T. Luda, J. M. Park, and F. Poli, Nuclear Fusion **57**, 086034 (2017).

[32] Citrin, J, Breton, S, Felici, F, Imbeaux, F, Aniel, T, Artaud, J F, Baiocchi, B, Bourdelle, C, Camenen, Y, and Garcia, J, Nuclear Fusion **55**, 092001 (2015).

[33] F. Felici, J. Citrin, A. A. Teplukhina, J. Redondo, C. Bourdelle, F. Imbeaux, O. Sauter, J. Contributors, and t. E. M. Team, Nuclear Fusion **58**, 096006 (2018).

[34] Matos, Francisco A, Ferreira, Diogo R, and Carvalho, Pedro J, Fusion Engineering and Design **114**, 18 (2017).

[35] D. R. Ferreira, P. J. Carvalho, H. Fernandes, and J. Contributors, Fusion Science and Technology **74**, 47 (2018), https://doi.org/10.1080/15361055.2017.1390386.

[36] D. Böckenhoff, M. Blatzheim, H. Hölbe, H. Niemann, F. Pisano, R. Labahn, T. S. Pedersen, and T. W.-X. Team, Nuclear Fusion **58**, 056009 (2018).

[37] B. Cannas, S. Carcangiu, A. Fanni, T. Farley, F. Militello, A. Montisci, F. Pisano, G. Sias, and N. Walkden, Fusion Engineering and Design (2019).

[38] Clayton, D J, Tritz, K, Stutman, D, Bell, R E, Diallo, A, LeBlanc, B P, and Podestà, M, Plasma Physics and Controlled Fusion **55**, 095015 (2013).

[39] J. B. Lister and H. Schnurrenberger, Nuclear Fusion **31**, 1291 (1991).

[40] E. Coccorese, C. Morabito, and R. Martone, Nuclear Fusion **34**, 1349 (1994).

[41] C. M. Bishop, P. S. Haynes, M. E. U. Smith, T. N. Todd, and D. L. Trotman, Neural Computing & Applications **2**, 148 (1994).

[42] Cacciola, Matteo, Greco, Antonino, Morabito, Francesco Carlo, and Versaci, Mario, in *Neural Information Processing* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2006) pp. 353–360.

[43] Y.-M. Jeon, Y.-S. Na, M.-R. Kim, and Y. S. Hwang, Review of Scientific Instruments **72**, 513 (2001).

[44] B. Wang, B. Xiao, J. Li, Y. Guo, and Z. Luo, Journal of Fusion Energy **35**, 390 (2016).

[45] B. P. van Milligen, V. Tribaldos, and J. A. Jiménez, Physical Review Letters **75**, 3594 (1995).

[46] P. Rodrigues and J. a. P. S. Bizarro, Phys. Rev. Lett. **95**, 015001 (2005).

[47] P. Rodrigues and J. a. P. S. Bizarro, Phys. Rev. Lett. **99**, 125001 (2007).

[48] G. Ludwig, P. Rodrigues, and J. P. Bizarro, Nuclear Fusion **53**, 053001 (2013).

[49] J. W. C. van Lint, S. P. Hoogendoorn, and H. J. van Zuylen, Transportation Research Part C: Emerging Technologies **13**, 347 (2005).

[50] S. Joung, J. Kim, S. Kwak, K.-r. Park, S. H. Hahn, H. S. Han, H. S. Kim, J. G. Bak, S. G. Lee, and Y. c. Ghim,

Review of Scientific Instruments **89**, 10K106 (2018).

[51] D. S. Sivia and J. Skilling, *Data Analysis: A Bayesian Tutorial* (Oxford: Oxford University Press, 2006).

[52] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning* (The MIT Press, 2006).

[53] S. G. Lee, J. G. Bak, E. M. Ka, J. H. Kim, and S. H. Hahn, Review of Scientific Instruments **79**, 10F117 (2008).

[54] S. Haykin, *Neural Networks and Learning Machines* (Pearson, 2008).

[55] X. Glorot and Y. Bengio, in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research, Vol. 9, edited by Y. W. Teh and M. Titterington (PMLR, Chia Laguna Resort, Sardinia, Italy, 2010) pp. 249–256.

[56] D. P. Kingma and J. Ba, CoRR **abs/1412.6980** (2014), arXiv:1412.6980.

[57] M. 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.

[58] R. Ge, F. Huang, C. Jin, and Y. Yuan, CoRR **abs/1503.02101** (2015), arXiv:1503.02101.

[59] L. Bottou, in *in COMPSTAT* (2010).

[60] Q. Huynh-Thu and M. Ghanbari, Electronics Letters **44**, 800 (2008).

[61] *JPEG vs. JPEG 2000: an objective comparison of image encoding quality* (2004).

[62] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, IEEE Transactions on Image Processing **13**, 600 (2004).

[63] E. J. Strait, J. D. Broesch, R. T. Snider, and M. L. Walker, Review of Scientific Instruments **68**, 381 (1997).

[64] Xia, Yu-Jun, Zhang, Zhong-Dian, Xia, Zhen-Xin, Zhu, Shi-Liang, and Zhang, Rui, Measurement Science and Technology **27**, 025104 (2015).

[65] Ka, E M, Lee, S G, Bak, J G, and Son, D, Review of Scientific Instruments **79**, 10F119 (2008).

[66] U. von Toussaint, Rev. Mod. Phys. **83**, 943 (2011).