Neural-Network-Based Free-Boundary Equilibrium Solver to Enable Fast Scenario Simulations

Zibo Wang, *Student Member, IEEE*, Xiao Song, Tariq Rafiq, and Eugenio Schuster, *Member, IEEE*

***Abstract*— A numerical free-boundary equilibrium (FBE) solver, based on finite-difference and Picard-iteration methods, has been recently developed on a rectangular grid to compute the poloidal-flux distribution in tokamaks. An accelerated version of this computationally intensive numerical solver, named FBE-Net, has been developed in this work by leveraging the physics-informed neural network (PINN) method. Within this framework, the neural-network (NN) component employs a fully connected multilayer perceptron (MLP) architecture. Critically, the underlying physical constraints are defined by the Grad–Shafranov (G-S) equation, ensuring the NN-based solver adheres to essential governing principles. FBE-net is trained on a dataset generated by the numerical solver, which serves as a source of ground truth. The inputs for FBE-Net are the plasma current, the normalized beta, and the coil currents, while the outputs are the poloidal-flux map and a set of flux-averaged equilibrium parameters. When compared to the numerical solver, the NN-based solver displays a significant increase in computational efficiency without notably sacrificing accuracy.**

***Index Terms*— Free-boundary equilibrium (FBE) solver, Grad–Shafranov (G-S) equation, physics-informed neural network (PINN).**

I. Introduction

IN MAGNETIC confinement fusion devices such as tokamaks and stellarators, plasma equilibrium is crucial for a wide range of applications. It is not only essential for physics analysis, including the study of magnetohydrodynamic (MHD) instability, transport, and turbulence but also plays a pivotal role in scenario development. These scenarios encompass diverse plasma shapes, including circular, noncircular, and configurations with X-points where the toroidal and poloidal magnetic fields undergo significant changes. Plasma equilibrium is vital for both addressing issues in existing tokamaks and designing new machines, impacting the entire discharge, from startup and ramp-up to the flat-top phase and ultimately, the controlled plasma termination in the ramp-down phase.

It is thus of paramount importance to develop solvers to address the complexities of the plasma equilibrium, primarily governed by the well-known Grad–Shafranov (G-S) equation [1], [2]. The solvers are dominantly categorized into two distinct types, i.e., fixed-boundary and free-boundary equilibrium (FBE) solvers. Fixed-boundary equilibrium solvers [3], [4], [5] make two key assumptions. First, they prescribe the shape of the plasma cross section, such as a circular or elongated shape, and its position within the vacuum vessel. Second, they express the right-hand side (RHS) of the G-S equation, i.e., the toroidal plasma current density (J_ϕ), either in a linear form [5] or as a constant [3], [4]. Thus, the G-S equation can be analytically solved due to this simple assumption on its dependence on the toroidal plasma current density. Fixed-boundary equilibrium solvers are widely used in studies of MHD instability and particle and heat transport, where simple plasma equilibrium states (e.g., prescribed plasma boundary) are required. In contrast, FBE solvers [6], [7] are designed to solve the G-S equation when the plasma boundary is not predefined but rather determined as part of the solution process. The FBE solver becomes indispensable when seeking to comprehend the dynamics of plasmas with evolving boundaries, especially in scenarios involving external coil currents as actuators for shape control. The inherent nonlinearity in the RHS of the G-S equation demands numerical solutions, often involving iterative processes. As a result, FBE solvers are more computationally demanding than fixed-boundary equilibrium solvers since they demand intensive numerical calculations on spatial grids.

To accelerate the computation process of an FBE solver, it becomes both practical and essential to devise a fast surrogate solver using neural network (NN) methodologies. Recently, NN-based FBE solvers and reconstructions algorithms [8], [9], [10], [11] have been developed to predict the poloidal-magnetic-flux rapidly while maintaining impressive accuracy. Notably, studies such as [8] and [10] incorporate governing physical principles or constraints into their NN models. By integrating constraints, they can align the predictions with the underlying physics (i.e., the G-S equation) while also demonstrating some level of robustness when extrapolating beyond the training data. Most of the previous work, including [9], [10], [11], has focused on the development of NN-based surrogate models for reconstruction algorithms like EFIT [12] where data from from flux loops and magnetic probes are used as inputs to the model. Building on prior knowledge, two types of NN-based FBE solvers are proposed in this work for EAST [13]. These two NN-based FBE solvers, which are named FBE-net1 and FBE-net2, do not use any type of magnetic measurements as inputs. FBE-net1 is based on the multilayer perceptron (MLP), while FBE-net2 incorporates the innovative architecture of physics-informed NNs (PINNs)

Manuscript received 4 October 2023; revised 12 February 2024; accepted 21 February 2024. Date of publication 19 March 2024; date of current version 9 December 2024. This work was supported by U.S. Department of Energy under Contract DE-SC0010537. The review of this article was arranged by Senior Editor R. Chapman. *(Corresponding author: Zibo Wang.)*

The authors are with the Department of Mechanical Engineering and Mechanics, Lehigh University, Bethlehem, PA 18015 USA (e-mail: zibo.wang@lehigh.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TPS.2024.3375284.

Digital Object Identifier 10.1109/TPS.2024.3375284

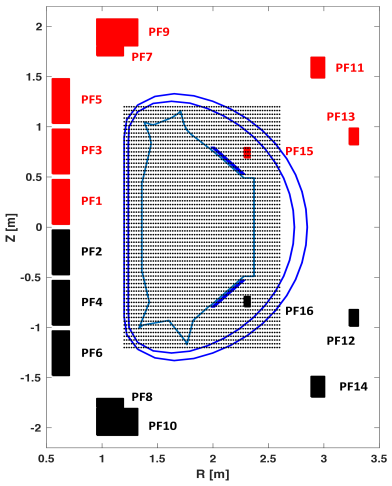


Fig. 1. Rectangular grid used for the numerical equilibrium solver.

[14] by incorporating the G-S equation directly into its loss function. As a difference from previous recent work, the proposed NN-based FBE solvers draw their ground truth data from a MATLAB-based numerical FBE solver [15]. The NN-based solver is designed to operate in direct mode [7], where the inputs are the plasma current I_p, the normalized beta β_p, and 16 poloidal coil currents. It is ultimately designed to be integrated into Control Oriented Transport SIMulator (COTSIM). In COTSIM, a 2-D equilibrium solver is used to provide flux-averaged equilibrium parameters, which are required by the transport equations for the poloidal magnetic flux, heat, particle, and momentum. Since the NN-based solver can run much faster than the original numerical FBE solver, its integration in COTSIM would facilitate testing of integrated equilibrium + transport feedback-control algorithms as well as fast scenario planning by model-based feedforward optimization [16].

This article is organized as follows. The collection, generation, and division of the dataset used for NN training are presented in Section II. Both the structures of the NNs and the training procedures are introduced in Section III. The performance of the NN-based solver is assessed in Section IV. Conclusions and possible future work are stated in Section V.

II. Dataset Generation and Data Processing

To achieve a robust understanding of plasma dynamics, generating a precise and varied dataset is of paramount importance. The dataset used to train the NN-based solver, which is instrumental in establishing a foundational ground truth, is collected from the execution of a MATLAB-based numerical equilibrium solver [15]. The solver uses a wide range of input conditions to generate 8455 feasible equilibrium states for the plasma. Specifically, data are created by manipulating parameters such as the plasma current (I_p), poloidal beta (β_p), and 16 poloidal field (PF) coils within a defined spectrum, ensuring comprehensive coverage of possible scenarios. The value of I_p is selected within the range of 0.32–0.5 MA at 0.01 MA intervals, while β_p values are chosen from the interval 0.2 to 0.8 with a step size of 0.01. The maximum, minimum, mean, and standard deviation values for all the PF coils are presented in Table I.

A. Numerical Equilibrium Solver

Recently, a numerical FBE solver [15] using the finite difference method (FDM) has been developed entirely within the MATLAB environment. The G-S equation is solved on a rectangular grid by leveraging the Picard iteration method [17]. The numerical FBE solver typically takes a maximum of 10 s to converge to an equilibrium solution. This solver has been cross-verified with fixed-boundary solvers [5], free-boundary solvers (i.e., FEEQS.M [18]), and compared with experimental equilibrium reconstruction data from codes like EFIT. In this section, a concise overview of the numerical equilibrium solver is offered, explaining its operational mechanics and highlighting its distinctive attributes as an essential tool for generating the required dataset.

In the cylindrical coordinate system (R, ϕ, Z), the ϕ component is deemed negligible due to the assumption of toroidal axisymmetry in tokamaks. The FBE equation is derived from the force balance equation in axisymmetric tokamak geometry

$$\Delta^* \psi(R, Z) = -\mu_0 R J_\phi(R, Z) \tag{1}$$

where ψ is the poloidal stream function and μ_0 is the permeability in vacuum. The toroidal current density J_ϕ depends on different regions

$$J_\phi(R, Z) = \begin{cases} Rp'(\psi) + \dfrac{ff'(\psi)}{\mu_0 R}, & \text{in plasma area } \Omega_{pl} \\ \dfrac{I_k}{S_k}, & \text{in coil } k \\ 0, & \text{elsewhere} \end{cases} \tag{2}$$

where $p(\psi)$ is plasma kinetic pressure, f is the diamagnetic function $f(\psi) \equiv RB_\phi$, B_ϕ is the toroidal magnetic field, I_k and S_k are the current and cross section of external conductor coil k, and the prime symbol denotes the gradient with respect to ψ. The operator Δ^* is defined as

$$\Delta^* \equiv R\frac{\partial}{\partial R}\left(\frac{1}{R}\frac{\partial}{\partial R}\right) + \frac{\partial^2}{\partial Z^2}. \tag{3}$$

As shown in Fig. 1, the 2-D cross section is defined in a rectangular computational domain (R_i, Z_j)

$$R_i = R_{\min} + \delta R \times (i-1), \quad Z_j = Z_{\min} + \delta Z \times (j-1). \tag{4}$$

The small-size steps in the R- and Z-directions are defined as

$$\delta R = \frac{R_{\max} - R_{\min}}{N_R - 1}, \quad \delta Z = \frac{Z_{\max} - Z_{\min}}{N_Z - 1}$$

where N_R and N_Z are the number of grid points in the R- and Z-directions, respectively ($i = 1, \ldots, N_R$, $j = 1, \ldots, N_Z$). They are both set as 65 in this work. The derivatives of the 2-D (R, Z) G-S operator Δ^* in (3) can be approximated as

$$\begin{aligned} \left.\frac{\partial \psi}{\partial R}\right|_{j,i} &\simeq \frac{\psi_{j,i+1} - \psi_{j,i-1}}{2\delta R} \\ \left.\frac{\partial^2 \psi}{\partial R^2}\right|_{j,i} &\simeq \frac{\psi_{j,i+1} - 2\psi_{j,i} + \psi_{j,i-1}}{(\delta R)^2} \\ \left.\frac{\partial^2 \psi}{\partial Z^2}\right|_{j,i} &\simeq \frac{\psi_{j+1,i} - 2\psi_{j,i} + \psi_{j-1,i}}{(\delta Z)^2}. \end{aligned} \tag{5}$$

TABLE I
MAXIMUM, MINIMUM, MEAN, AND STANDARD DEVIATION VALUES FOR 16 PF COILS

	PF1 kA/turn	PF2 kA/turn	PF3 kA/turn	PF4 kA/turn	PF5 kA/turn	PF6 kA/turn	PF7 kA/turn	PF8 kA/turn	PF9 kA/turn	PF10 kA/turn	PF11 kA/turn	PF12 kA/turn	PF13 kA/turn	PF14 kA/turn	PF15 kA/turn	PF16 kA/turn
Max	−2.9	−4.2	−1.3	−5.7	12.7	8.6	3.1	3.0	−1.6	1.2	7.6	1.5	13.4	0.8	2.8	0.1
Min	−15.0	−16.9	−16.2	−19.2	−3.1	−0.1	−0.8	1.0	−8.7	−6.2	−14.7	−10.5	−19.7	−15.9	−14.9	−7.6
Avg.	−8.2	−9.5	−8.3	−11.3	4.7	3.8	1.0	1.9	−5.5	−2.8	−1.5	−3.2	−4.7	−6.1	−4.5	−3.7
SD.	2.2	2.3	3.2	2.6	2.8	1.5	0.7	0.4	1.3	1.7	4.5	1.9	5.2	2.9	2.8	1.4

TABLE II
LIST OF INPUTS AND OUTPUTS OF NN-BASED SOLVERS

	Symbol	Explanation
Input	I_p	Plasma current (A)
	β_p	Poloidal beta
	I_{coil}	Coil currents (16 coils in EAST)
Output: FBE-net1	$\hat{F}$	Geometric factor [19]
	$\hat{G}$	Geometric factor [19]
	$\hat{H}$	Geometric factor [19]
	$\psi(R,Z)$	Poloidal flux $(Wb \cdot rad^{-1})$
Output: FBE-net2	$J_\phi(R,Z)$	Toroidal current density (MA/m^2)
	$\psi(R,Z)$	Poloidal flux $(Wb \cdot rad^{-1})$

The $\Delta^*\psi$ term in (1) is rewritten based on the approximations in (5)

$$\frac{1}{(\delta Z)^2}\psi_{j-1,i} + \left(\frac{1}{(\delta R)^2} + \frac{1}{2R_i(\delta R)}\right)\psi_{j,i-1} - 2\left(\frac{1}{(\delta R)^2} + \frac{1}{(\delta Z)^2}\right)\psi_{j,i} + \left(\frac{1}{(\delta R)^2} - \frac{1}{2R_i(\delta R)}\right)\psi_{j,i+1} + \frac{1}{(\delta Z)^2}\psi_{j+1,i} = -\mu_0 R_i J_{\phi_{j,i}} \tag{6}$$

in which $\psi_{j,i}$ denotes $\psi(R_i, Z_j)$ and $J_{\phi j,i}$ represents $J_\phi(R_i, Z_j)$. Furthermore, (6) can be simplified as

$$A\psi = b \Leftrightarrow \psi = A^{-1}b \tag{7}$$

where A is a constant matrix and the matrix b depends on J_ϕ.

It should be emphasized that the boundary of J_ϕ is excluded from (6) by choosing $i \in (1, N_R)$ and $j \in (1, N_Z)$. The boundary is assumed to adhere to a Dirichlet boundary condition [6]. A Picard iteration is implemented as

$$\frac{1}{(\delta Z)^2}\psi^n_{j-1,i} + \left(\frac{1}{(\delta R)^2} + \frac{1}{2R_i(\delta R)}\right)\psi^n_{j,i-1} - 2\left(\frac{1}{(\delta R)^2} + \frac{1}{(\delta Z)^2}\right)\psi^n_{j,i} + \left(\frac{1}{(\delta R)^2} - \frac{1}{2R_i(\delta R)}\right)\psi^n_{j,i+1} + \frac{1}{(\delta Z)^2}\psi^n_{j+1,i} = -\mu_0 R_i J^n_{\phi_{j,i}}\left(\psi^{n-1}_{j,i}\right) \tag{8}$$

where n represents the iteration number. Given an initial assumption for $J^0_{\phi j,i}$, the value of $\psi^0_{j,i}$ can be derived from (8). Subsequently, $J^1_{\phi j,i}$ is updated based on (2), i.e., $J^1_{\phi j,i} = J_\phi(\psi^0_{j,i})$. The process iteratively computes the numerical solutions for $J^n_{\phi j,i}$ and $\psi^n_{j,i}$ until the convergence criterion, specifically $\|\psi^n_{j,i} - \psi^{n-1}_{j,i}\| < 10^{-4}$, is satisfied.

TABLE III
DETERMINE HYPERPARAMETERS VIA METHODICALLY SCANNING

Nodes per hidden layer	258
Number of batch size	150
Number of epochs	200
Activation function	tanh, ReLU
Rate of learning	0.01

B. Data Normalization

The proper scaling of input and output data, as detailed in Table II, is crucial for optimizing the NN training efficiency. An attempt to train the NN model based on unnormalized data may inadvertently impede the learning progression or in extreme cases, cause it to stop entirely because of issues like gradient explosion. Such problems arise when the hidden units become saturated [20], causing a loss of clarity and making it challenging to effectively train the NN-based solver. To address these problems often arising during training, this work makes use of the normalization approach delineated in [21]. This normalization approach ensures data scaling within the [0.01, 1] range, enhancing numerical stability and preventing vanishing gradient issues due to the sensitivity of activation functions to very small inputs. This method not only makes the training process better, but it also keeps the advantage of being able to easily switch back the predictions to their original scale.

III. NN MODEL AND TRAINING

This section explores the details of the chosen NN model and its training approach for the NN-based solver. Initially, a comprehensive examination of the foundational MLP, named FBE-net1, is presented, highlighting its architectural design and performance characteristics. Following this, a novel adaptation is presented, integrating the G-S equation as a constraint in the loss function, in line with the PINN method. This second NN-based solver is called FBE-net2.

A. Fully Connected MLP

The normalized dataset detailed in Section II-B is partitioned randomly into three distinct sets. Specifically, 80% is allocated for training purposes (that is, for NN model training), 10% is designated for validation (essentially to optimize hyperparameter values), while the remaining portion serves as the testing set (employed to evaluate the accuracy of the network's predictions).

The NN architecture adopted in this investigation follows the MLP design, featuring three hidden layers. The hyperparameters, which are tuned by methodical scanning, are presented in Table III. Each layer is structured to optimize data flow and learning capacity, with a designated number of neurons to address the complexity of the problem. To handle nonlinearities and improve the model's ability to capture intricate relationships in the dataset, activation functions such as hyperbolic tangent (tanh) and rectified linear unit (ReLU) are employed. Additionally, regularization techniques, including dropout [22], are incorporated to enhance the model's robustness and prevent overfitting.

During the training phase, the model is trained in 200 epochs, with each epoch consisting of a batch size of 150. The nodes per hidden layer are set at 258, and a learning rate of 0.01 is applied to optimize the training process and manage overfitting. This customized MLP architecture serves as the foundation for the advanced plasma equilibrium computations explored in this study.

IV. NN-Based Solver Evaluation and Discussion

A. Integrate G-S Equation as Constraint

The role of the loss function is crucial in the implementation of PINNs, serving as a bridge between data-driven modeling and the dominant physics governing the phenomenon. The loss function ϵ is defined as

$$\epsilon = \lambda_1 \mathcal{L}_1 + \lambda_2 \mathcal{L}_2 + \lambda_3 \mathcal{L}_3 \tag{9}$$

$$\mathcal{L}_1 = \frac{1}{N_d} \sum_{i=1}^{N_d} \left(J_\phi^{\mathrm{NN}} - J_\phi^{\mathrm{DT}}\right)^2 \tag{10}$$

$$\mathcal{L}_2 = \frac{1}{N_p} \sum_{i=1}^{N_p} \left(R^2 \nabla \left(\frac{\nabla \psi (J_\phi^{\mathrm{NN}})}{R^2} \right) + \mu_0 R J_\phi^{\mathrm{DT}} \right)^2 \tag{11}$$

$$\mathcal{L}_3 = \frac{1}{N_d} \sum_{i=1}^{N_d} \left(J_{\phi,\mathrm{out}}^{\mathrm{NN}} - J_{\phi,\mathrm{out}}^{\mathrm{DT}}\right)^2 \tag{12}$$

where $(\cdot)^{\mathrm{NN}}$ is predicted by the NN, $(\cdot)^{\mathrm{DT}}$ is predicted by the original numerical solver (ground truth data), and $(\cdot)_{\mathrm{out}}$ is the data-point on the outermost contour of J_ϕ. The parameters λ_1, λ_2, and λ_3 are the weights of each loss function.

In this design, the G-S equation is directly integrated as a constraint within the loss function using the PyTorch library [23]. PyTorch, a leading deep learning tool, facilitates the seamless integration of plasma physics knowledge with state-of-the-art computational techniques, enhancing the predictions of the NN-based solver. It is noteworthy that J_ϕ is chosen as the target feature instead of ψ. While setting the target feature as either ψ or J_ϕ is equivalent, the implementation with J_ϕ is preferred for its earlier development.

Following the training procedures delineated in Section III, the performance of the NN-based solver is evaluated using the testing dataset. This includes comparing predictions from the numerical equilibrium solver with those generated by the NN-based solver for specific discharges in the testing dataset.

Fig. 2 displays log-scale histograms detailing the regression outcomes for predictions, notably representing Fig. 2(a) $\psi(R, Z)$, (b) $\hat{F}$, (c) $\hat{G}$, and (d) $\hat{H}$. It shows that most data points cluster along the diagonal line, especially for the three geometric factors $\hat{F}$, $\hat{G}$ and $\hat{H}$ [19]. This clustering suggests that the geometric factors predicted by the NN exhibit significant congruence with the results derived from the numerical FBE solver. This alignment is further substantiated by the correlation coefficients (R^2) approaching unity, as showcased in the upper-left corner. A comparative analysis presented in Fig. 3 juxtaposes the calculations from the numerical FBE solver against the predictions of the NN-based solver, revealing a generally harmonious alignment, with minor exceptions. Within the Python computational context, FBE-net1 boasts a commendable efficiency, yielding geometric factor predictions in an average span of roughly 1 ms, a pace that markedly outstrips the numerical FBE solver by several orders of magnitude.

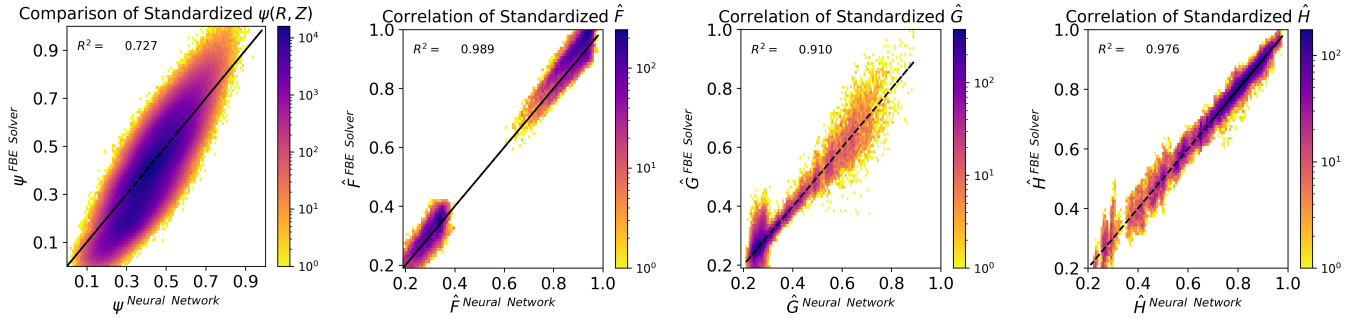
TABLE IV
Performance Summary: NN Solvers' MSE and Speed

FBE-net1				FBE-net2			
	MSE	R^2	Speed		MSE	R^2	Speed
$\hat{F}$	$1.75e^{-4}$	0.989	$< 1ms$				
$\hat{G}$	$6.26e^{-4}$	0.910	$< 1ms$				
$\hat{H}$	$5.74e^{-3}$	0.976	$< 1ms$				
				J_ϕ	$5.11e^{-3}$	0.974	$< 5ms$
ψ	0.15	0.727	$< 5ms$	ψ	$4.12e^{-6}$	0.982	$< 5ms$

As shown in Figs. 2(a) and 3(a), the predictions from FBE-net1 associated with the ψ map necessitate further refinements. In response to this, FBE-net2 has been formulated to deliver enhanced results for ψ predictions. By incorporating the G-S equation as a constraint, the NN-based solver for $J_\phi(R, Z)$ exhibits a commendable correlation with the reference data, as depicted in Fig. 4. Consequently, the results from FBE-net2 more closely align with the baseline data, as shown by the contour illustration in Fig. 5. Then $J_\phi(R, Z)$ is converted to $\psi(R, Z)$ by applying (7), where the graphic result is shown in Fig. 6. FBE-net2 significantly outperforms its predecessor, FBE-net1, in its predictive capabilities, aligning closely with the reference dataset and achieving a much lower mean squared error (MSE).

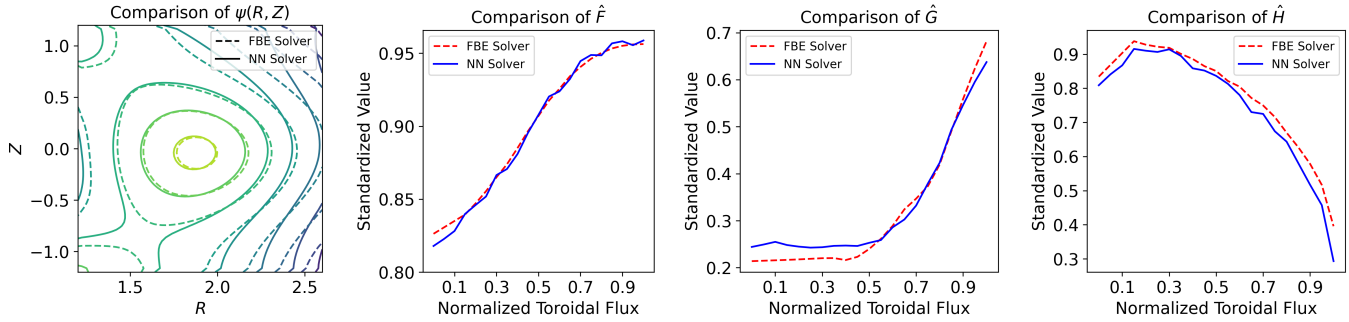
In Table IV, a summary of the average correlations and MSE between the predictions from the numerical equilibrium solver and the NN-based solver is provided. The results suggest that the PINN-based surrogate model introduced here serves as a competent substitute for the conventional numerical equilibrium solver. This is supported by the fact that the correlation coefficient R^2 approaches 1 for the testing dataset as shown in Table IV.

While two neural NN-based solvers are developed, their deployment strategy is evaluated to ensure the highest level of accuracy and computational efficiency. The primary task involves the prediction of two critical parameters: the geometry factors and the ψ map. For the geometry factor prediction, FBE-net1 demonstrates prowess in providing both rapid and precise outcomes. Conversely, for the prediction of the ψ map, FBE-net2 emerges as superior. Comparative analyses reveal



(a) (b) (c) (d)

Fig. 2. Histograms of regression results for shots in the testing dataset using FBE-net1. (a) $\psi(R, Z)$ map, (b) $\hat{F}$, (c) $\hat{G}$, and (d) $\hat{H}$.



(a) (b) (c) (d)

Fig. 3. Prediction results for shots in the testing dataset using FBE-net1. (a) $\psi(R, Z)$ map, (b) $\hat{F}$, (c) $\hat{G}$, and (d) $\hat{H}$.

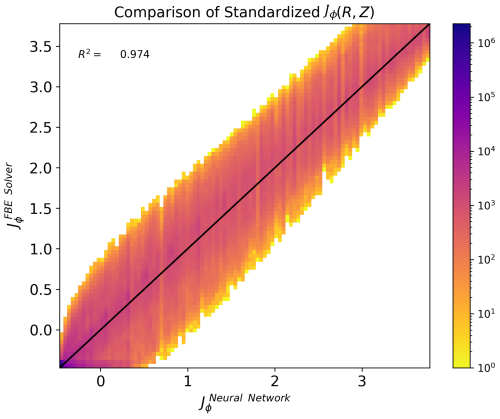


Fig. 4. Histogram representation of regression results for $J_\phi(R, Z)$ derived from FBE-net2 for shots in the testing dataset.

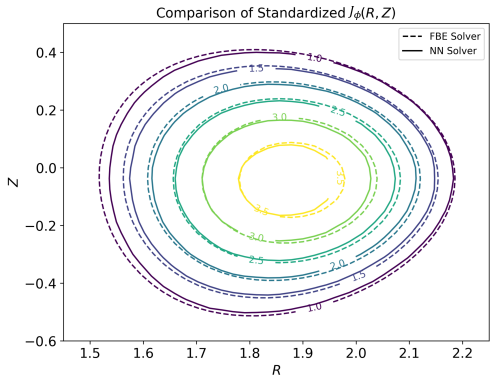


Fig. 5. $J_\phi(R, Z)$ map calculated from FBE-net2. The solid line represents contours generated by FBE-net2, with the dotted line representing those produced by the numerical FBE solver.

that, while both solvers operate with comparable speeds, FBE-net2 substantially outperforms FBE-net1 in terms of precision for ψ map predictions. Thus, by harnessing the strengths of each solver, high levels of accuracy for the respective prediction targets are maintained, without compromising on computational efficiency.

B. Predictions With Inputs Outside the Training Dataset

To evaluate the robustness and generalization capabilities of FBE-net2, assessments are conducted using inputs beyond the training dataset. As an illustrative example, the inputs I_p and β_p are chosen as 0.52 MA (exceeding the training dataset's maximum by 4%) and 0.90 (surpassing the dataset's maximum by 12.5%), respectively. Fig. 7(a) compares the prediction made by FBE-net2 for these inputs with the solution provided by the FBE solver. The contours generated by FBE-net2 resemble in shape those of the FBE solver, although they show discrepancies in the exact values (predictions by FBE-net2 are consistently smaller in value). The MSE map in Fig. 7(b) provides a quantitative measure of these discrepancies. Remarkably, the MSE is kept under an order of magnitude of 10^{-3}. In contrast, FBE-net1 not only struggles to produce meaningful contours when using inputs outside its training set as shown in Fig. 7(c) but also shows an MSE two orders of magnitude higher ($\sim 10^{-1}$) as shown in Fig. 7(d).

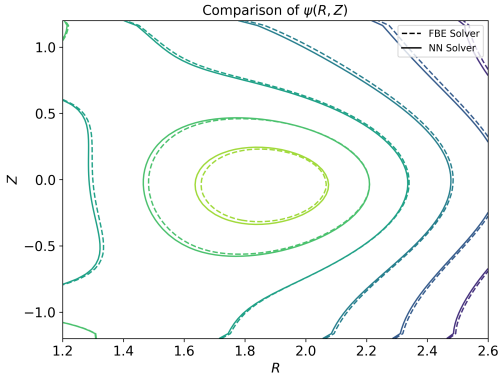


Fig. 6. $\psi(R, Z)$ map calculated from FBE-net2. The solid line represents contours generated by FBE-net2, with the dotted line representing those produced by the numerical FBE solver.

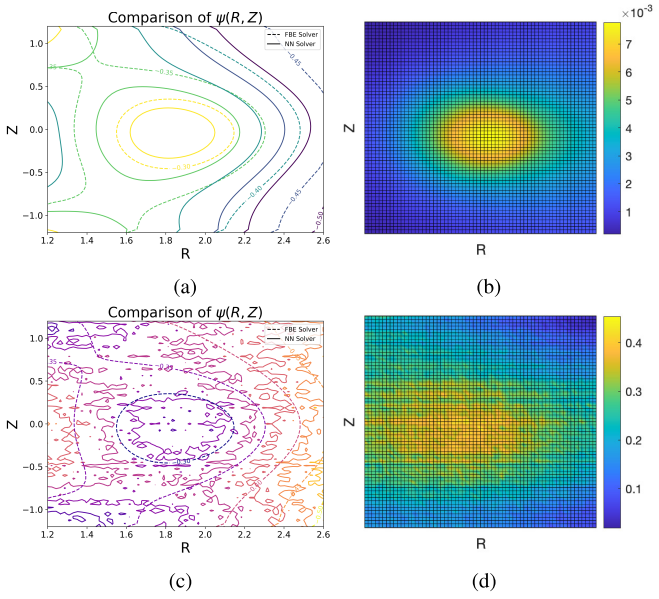


Fig. 7. $\psi(R, Z)$-contour comparison and MSE maps for predictions with inputs outside the training dataset. (a) $\psi(R, Z)$ contour comparison for FBE-net2, (b) MSE for FBE-net2, (c) $\psi(R, Z)$ contour comparison for FBE-net1, and (d) MSE for FBE-net1.

While FBE-net2 does lose prediction accuracy as it uses inputs outside the training set, as expected, it does so in a much less dramatic fashion than FBE-net1. The capability to handle data beyond its training set while maintaining a tolerable bounded prediction error is a sample of robustness and adaptability. However, this assessment should not be interpreted by any means as a recommendation to use FBE-net2 outside the training set.

V. Conclusion and Future Work

In this work, two fast FBE solvers have been developed by leveraging the capabilities of NN techniques. FBE-net1 is designed based on MLP, while at the core of the development of FBE-net2 lies the strategic utilization of the G-S equation as a constraint. The incorporation of this physics-based constraint significantly increases the prediction accuracy of the $\psi(R, Z)$ map by effectively combining physical principles with state-of-the-art computational methodologies. The accelerated performance of the NN-driven solver and its demonstrated robustness position it as an optimal solution for equilibrium-transport integration within COTSIM. This integration offers a range of applications, including feedforward optimization, feedback control simulations, and real-time plasma control.

Future work will aim at improving the NN-based solver's performance and leveraging it in control-oriented equilibrium + transport simulations. First, expanding the training data's scope to cover a broader range of conditions and diverse tokamak configurations will create a more universally applicable model. Second, incorporating coil-connection constraints such as the usual anti-series connection of the PF15 and PF16 coils [24] will more accurately reflect operation conditions at EAST. Third, refining the NN framework's robustness and efficiency through hyperparameter tuning and deep NN topology exploration, by leveraging techniques like Evolutionary Algorithms [25] and Meta-learning/AutoML [26], [27], can significantly enhance prediction performance. Lastly, integrating the NN-based FBE solver into COTSIM will enable fast simulations combining equilibrium and transport solvers for studies demanding regulation of the plasma boundary by active shape control.

References

[1] H. Grad and H. Rubin, "Hydromagnetic equilibria and force-free fields," *J. Nucl. Energy*, vol. 7, nos. 3–4, pp. 284–285, Sep. 1958.
[2] V. Shafranov, "On magnetohydrodynamical equilibrium configurations," *Sov. Phys. JETP*, vol. 6, no. 3, pp. 545–554, 1958.
[3] S. B. Zheng, A. J. Wootton, and E. R. Solano, "Analytical tokamak equilibrium for shaped plasmas," *Phys. Plasmas*, vol. 3, no. 3, pp. 1176–1178, Mar. 1996.
[4] A. J. Cerfon and J. P. Freidberg, "'One size fits all' analytic solutions to the Grad–Shafranov equation," *Phys. Plasmas*, vol. 17, no. 3, Mar. 2010, Art. no. 032502.
[5] L. Guazzotto and J. P. Freidberg, "Simple, general, realistic, robust, analytic tokamak equilibria. Part 1. Limiter and divertor tokamaks," *J. Plasma Phys.*, vol. 87, no. 3, Jun. 2021, Art. no. 905870303.
[6] Y. M. Jeon, "Development of a free-boundary tokamak equilibrium solver for advanced study of tokamak equilibria," *J. Korean Phys. Soc.*, vol. 67, no. 5, pp. 843–853, Sep. 2015.
[7] H. Heumann et al., "Quasi-static free-boundary equilibrium of toroidal plasma with CEDRES++: Computational methods and applications," *J. Plasma Phys.*, vol. 81, no. 3, Jun. 2015, Art. no. 905810301.
[8] B. P. van Milligen, V. Tribaldos, and J. A. Jiménez, "Neural network differential equation and plasma equilibrium solver," *Phys. Rev. Lett.*, vol. 75, no. 20, pp. 3594–3597, Nov. 1995.
[9] B. Wang, B. Xiao, J. Li, Y. Guo, and Z. Luo, "Artificial neural networks for data analysis of magnetic measurements on east," *J. Fusion Energy*, vol. 35, no. 2, pp. 390–400, Apr. 2016.
[10] S. Joung et al., "Deep neural network Grad–Shafranov solver constrained with measured magnetic signals," *Nucl. Fusion*, vol. 60, no. 1, Jan. 2020, Art. no. 016034.
[11] J. T. Wai, M. D. Boyer, and E. Kolemen, "Neural net modeling of equilibria in NSTX-U," *Nucl. Fusion*, vol. 62, no. 8, Aug. 2022, Art. no. 086042.
[12] L. L. Lao, H. St. John, R. D. Stambaugh, A. G. Kellman, and W. Pfeiffer, "Reconstruction of current profile parameters and plasma shapes in tokamaks," *Nucl. Fusion*, vol. 25, no. 11, pp. 1611–1622, Nov. 1985.
[13] S. Wu, "An overview of the EAST project," *Fusion Eng. Design*, vol. 82, nos. 5–14, pp. 463–471, Oct. 2007.
[14] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, pp. 686–707, Feb. 2019.
[15] X. Song, B. Leard, and E. Schuster, "MATLAB-based free-boundary equilibrium solver for fast control-oriented predictions," *Bull. Amer. Phys. Soc.*, vol. 2022, p. BP11-006, Oct. 2022.

[16] X. Song, B. Leard, W. Zibo, R. Tariq, and E. Schuster, "Coupling of free-boundary-equilibrium and transport solvers to enable model-based scenario optimization and integrated control integrating shape and profile control for advanced-scenario development," in *Proc. IAEA Fusion Energy Conf.*, 2023, Paper no. TH-P6.
[17] E. Kreyszig, *Advanced Engineering Mathematics*, 8th ed. Hoboken, NJ, USA: Wiley, 1999.
[18] J. Blum, H. Heumann, E. Nardon, and X. Song, "Automating the design of tokamak experiment scenarios," *J. Comput. Phys.*, vol. 394, pp. 594–614, Oct. 2019.
[19] Y. Ou et al., "Towards model-based current profile control at DIII-D," *Fusion Eng. Design*, vol. 82, nos. 5–14, pp. 1153–1160, Oct. 2007.
[20] C. Yu, M. T. Manry, J. Li, and P. Lakshmi Narasimha, "An efficient hidden layer training method for the multilayer perceptron," *Neurocomputing*, vol. 70, nos. 1–3, pp. 525–535, Dec. 2006.
[21] Z. Wang, S. Morosohk, T. Rafiq, E. Schuster, M. D. Boyer, and W. Choi, "Neural network model of neutral beam injection in the EAST tokamak to enable fast transport simulations," *Fusion Eng. Design*, vol. 191, Jun. 2023, Art. no. 113514.
[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, Sep. 2014.
[23] N. Ketkar, J. Moolayil, N. Ketkar, and J. Moolayil, "Introduction to PyTorch," in *Deep Learning With Python: Learn Best Practices of Deep Learning Models With PyTorch*. Berkeley, CA, USA: Apress, 2021, pp. 27–91.
[24] B. Xiao et al., "EAST plasma control system," *Fusion Eng. Design*, vol. 83, nos. 2–3, pp. 181–187, 2008.
[25] T. Bäck and H.-P. Schwefel, "An overview of evolutionary algorithms for parameter optimization," *Evol. Comput.*, vol. 1, pp. 1–23, Dec. 1993.
[26] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015.
[27] P. Brazdil, C. G. Carrier, C. Soares, and R. Vilalta, *Metalearning: Applications to Data Mining*. Berlin, Germany: Springer, 2008.

Zibo Wang (Student Member, IEEE) received the B.S. degree in processing equipment and control system from East China University of Science and Technology, Shanghai, China, in 2012, and the M.S. degree in mechanical engineering and mechanics from the University of Pittsburgh, Pittsburgh, PA, USA, in 2018. He is currently pursuing the Ph.D. degree with the Plasma Control Laboratory, Department of Mechanical Engineering and Mechanics, Lehigh University, Bethlehem, PA, USA.

His research interests include cover optimal control, machine learning, and scenario planning.

Xiao Song received the B.E. degree in mineral engineering from Central South University, Hunan, China, in 2008, and the Ph.D. degree in applied mathematics from Université Côte d'Azur, Nice, France, in 2019.

From 2020 to 2021, he was an Associate Researcher with the Southwestern Institute of Physics, Chengdu, Sichuan, China. He is currently a Research Scientist with the Department of Mechanical Engineering and Mechanics, Lehigh University, Bethlehem, PA, USA. His research interests include optimal control, numerical solvers, advanced scenarios and integrated models in tokamaks.

Tariq Rafiq received the Master of Philosophy degree in plasma physics from Quaid-i-Azam University, Islamabad, Pakistan, in 1998, and the Ph.D. degree in electrical engineering from Chalmers University, Gothenburg, Sweden, in 2004.

Following this, he completed a Postdoctoral Fellowship with the University of Wisconsin, Madison, WI, USA. He is currently as a Research Associate Professor with the Department of Mechanical Engineering and Mechanics, Lehigh University, Bethlehem, PA, USA. He is a nuclear fusion theorist and modeler, specializing in the development of thermal, particle, and momentum transport models. His research interests include validating developed models against tokamak experiments, modeling scenarios, improving tokamak discharge performance, designing new experiments, and extrapolating results to future thermonuclear fusion devices like ITER and the fusion pilot plant.

Eugenio Schuster (Member, IEEE) received the B.Sc. degree in electronic engineering from the University of Buenos Aires, Buenos Aires, Argentina, in 1994, the B.Sc./M.Sc. degree in nuclear engineering from the Balseiro Institute, in 1998, and the M.Sc. and Ph.D. degrees in aerospace engineering from the University of California at San Diego, San Diego, CA, USA, in 2000 and 2004, respectively.

He is currently a Professor with the Department of Mechanical Engineering and Mechanics, Lehigh University, Bethlehem, PA, USA. He is an Expert in nuclear-fusion plasma control and leads the Lehigh University Plasma Control Laboratory. Prof. Schuster was a recipient of the National Science Foundation (NSF) CAREER Award for his work on "Nonlinear Control of Plasmas in Nuclear Fusion." He has been appointed as a Scientist Fellow in Plasma Control by the ITER Organization. Moreover, he has been designated by U.S. Department of Energy (DOE) as an expert member of the Integrated Operation Scenarios (IOS) Topical Group within the International Tokamak Physics Activity (ITPA), which he is the Chair. He has served as Leader of the Operations and Control Topical Group, U.S. Burning Plasma Organization (BPO) and as the Founder Chair of the Technical Committee on Power Generation within the IEEE Control Systems Society.