

# [ChinaVis 2021] Visualization of plasma shape in the LHD-type helical fusion reactor, FFHR, by a deep learning technique

Kunqi Hu, Koji Koyamada, Hiroaki Ohtani, Takuya Goto and Junichi Miyazawa

**Abstract**—A magnetic field is used to confine the plasma to achieve controlled fusion. Therefore, since the movement of the plasma follows magnetic field lines, a plurality of magnetic field lines is calculated from electromagnetic field simulation results in a fusion reactor. Because of the complicated distribution of magnetic field lines in three-dimensional (3D) space, existing analysis measures which are mostly based on two-dimensional poloidal plasma cross-sections are unsatisfactory for domain experts. To solve this problem, we propose a technique for reconstructing a regular scalar field from the magnetic field lines. First, on poloidal plasma cross-sections, intersection points of magnetic field lines are used to make annotations of learning the plasma shape. Then, a deep neural network is built to approximate the scalar field that represents the probability of the existence of magnetic field lines. Consequently, a 3D model of plasma shape has managed to be constructed by applying the marching cubes method. The effectiveness of the proposed method is demonstrated by comparing it with the conventional method and domain experts' reviews.

**Index Terms**—Visualization, Deep learning, Plasma shape, 3D model

## 1 INTRODUCTION

Nuclear fusion is an ideal means to obtain clean energy by combining two or more atomic nuclei. There are several ways to achieve controlled nuclear fusion, among which the use of magnetic fields is the most possible method for confining the high-density and high-temperature plasma.

There are two basic ways to generate twisted magnetic fields confining the plasma at the center of the vessel. One is the combination of two types of magnetic fields that are respectively generated from toroidal field coils and the poloidal field due to plasma current, such as Tokamak [4, 16]. The other way is to directly use helical coils that have a complicated structure, such as Stellarator [8]. The Large Helical Device (LHD)-type helical fusion reactor, FFHR is the second type. This reactor designed by the National Institute for Fusion Science (NIFS) has helical coils to generate twisted magnetic field lines [19].

### 1.1 FFHR structure

The schematic view of the fundamental design for an FFHR is depicted in Fig. 1 [20]. From the plasma confinement center to the outside, are the blankets, a pair of helical coils, supporting the structure with inner and outer vertical field coils, and cryostat. In addition, the FFHR has a cyclic symmetry structure. The symmetrical angle is 36 degrees of the toroidal angle.

The poloidal cross-section of the FFHR at the horizontally elongated plasma cross-section is shown in Fig. 2 [5]. In the limited space between the helical coils and plasma, blanket modules will be installed in the immediate vicinity of the plasma to achieve the most effective use of neutron energy generated by the fusion reaction. The gap between the two blankets is arranged for the online removal of impurities from the plasma. These impurities are neutralized and evacuated at the divertor so that the nuclear reactor can continue operating.

- Kunqi Hu is with Kyoto University. E-mail: hu.kunqi.24w@st.kyoto-u.ac.jp.
- Koji Koyamada is with Kyoto University. E-mail: koyamada.koji.3w@kyoto-u.ac.jp.
- Hiroaki Ohtani is with the National Institute for Fusion Science. E-mail: ohtani.hiroaki@nifs.ac.jp.
- Takuya Goto is with the National Institute for Fusion Science. E-mail: goto.takuya@nifs.ac.jp.
- Junichi Miyazawa is with the National Institute for Fusion Science. E-mail: miyazawa@LHD.nifs.ac.jp.

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxx/TVCG.201x.xxxxxxx

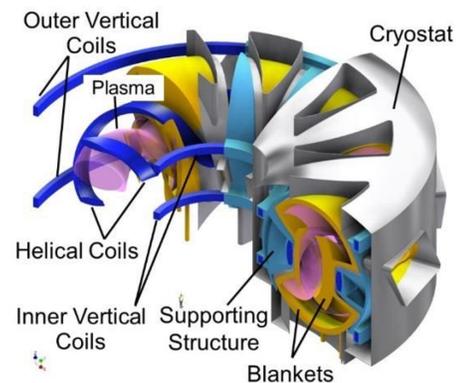


Fig. 1. Schematic of the FFHR-d1 series fusion reactor.

### 1.2 Plasma shape and magnetic field lines

The plasma is confined in the magnetic trap and moves along the magnetic field lines. Hence, the plasma shape can be obtained according to the distribution of magnetic field lines traced in a magnetic vector field inside the vessel of the FFHR. In general, the plasma shape is analyzed on discrete 2D poloidal cross-sectional shapes.

The poloidal plasma cross-section is located at the center of the poloidal cross-section of the FFHR, as shown in Fig. 2 [5]. Specifically, a poloidal plasma cross-section is composed of intersection points of each magnetic field line and a vertical plane at a certain toroidal angle. These points consist of three parts: the flux surface in an ellipse with light pink color, stochastic field lines surrounding the flux surface in purple, and edge surface layers with the divertor legs that extend to the divertor plates.

### 1.3 Existing problems in plasma shape analysis

From Fig. 2 [5], one can notice that the space between the inboard blanket and edge surface layers is extremely narrow. Additionally, paths that extend to the divertor for impurities must also be accurately designed. Previously, the shape of the component in the reactor was designed considering the structure of magnetic field lines in a 2D cross-section, at each discrete toroidal angle. However, the magnetic field line has a 3D structure, observing a poloidal cross-section of the plasma shape and the FFHR in 2D lacks an efficient approach to achieve a global view, in addition to being time-consuming and error-prone. Furthermore, magnetic confinement causes gyration motion

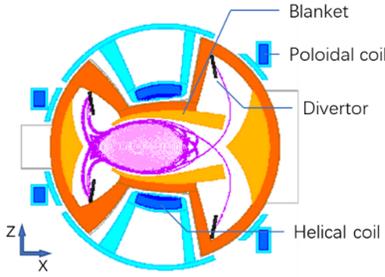


Fig. 2. Poloidal cross-section of the FFHR at horizontally elongated plasma cross-section.

of the plasma, which cannot be observed directly from magnetic field lines. This spread region, also known as Larmor radius, is an existing problem when studying the plasma shape using magnetic field lines. Above all, operating three-dimensionally with the plasma is required to optimize a 3D shape design of the component in the FFHR.

In summary, there are three requirements for a useful plasma shape observation approach from a 3D space, which are as follows.

- R1. The approach should enable the expression of a plasma existence region without any arbitrariness from the magnetic field lines data.
- R2. The Larmor radius should be included.
- R3. The approach should enable the detection of interferences between the plasma shape and the 3D design data of the components in the FFHR.

To meet the above requirements, a closed curved surface is considered to represent the last closed flux surface (LCFS) of the flux surface and edge surface layers with the divertor legs, respectively. The reasons for this are as follows. First, to check for interferences, it is unnecessary to consider those magnetic field lines that are buried under other lines. Rendering all lines in one scene will cause occlusion and cost significant computation resources. Second, compared with separated lines, a continuous surface can provide a clear boundary so that it becomes simple to inspect the interference of the plasma and components in the FFHR. Finally, the Larmor radius can be embedded into the preprocessing stage when generating the surface.

However, general surface construction methods are unlikely to meet R1. Due to the helical magnetic field, magnetic field lines are twisted and entwined with each other. Regardless of whether the connectivity information of a line has been provided, the relative position information between two lines is extremely complex. Therefore, it is difficult to know which lines are contributing to surface generation.

To tackle the aforementioned challenge, we propose using a deep neural network (DNN) to learn the representation of the plasma shape from the magnetic field lines. We aim to reconstruct a regular scalar field from the magnetic field lines. The intersection points on each poloidal plasma cross-section are used to make annotations according to the connection length of a magnetic field line to which they belong. Afterward, these labeled points will be inputted into a DNN as training material. Hence, the trained DNN can label each point from its new position from the scalar field. Then, the scalar field can represent the different parts of the plasma shape by collecting the points with the same label. Since the scalar field has been structured, the magnetic flux intensity at any position of this scalar field is easily obtained. The Larmor radius then can be calculated. The representation of a Larmor radius is then embedded in the scalar field. With the scalar field, the marching cubes (MC) method [13] can be used to generate an isosurface for different parts of the plasma shape.

To summarize our approach, the contributions of this research are as follows:

- We propose a DNN to reconstruct a scalar field that is friendly for surface generation.
- The Larmor radius is embedded in the scalar value of the scalar field. Consequently, the spread region representing the Larmor radius is included in the isosurface.
- Compared with topological reconstruction methods, using a DNN is time-saving. The scale of the scalar field can be customized according to users' demands.

## 2 RELATED WORKS

### 2.1 Visual analysis in 3D space

#### 2.1.1 Curve-based methods

Watanabe, T., et al. [21] studied magnetic field lines in the edge surface layer surrounding the exterior of the LCFS. They also plotted a pair of helical coils in one scene. All field lines terminate at footprints in the divertor plates. The magnetic field lines are classified depending on the connection length with different colors. The results showed that the connection length of the magnetic field lines closely surrounds the LCFS. However, the results are composed of thousands of streamlines that lack recognizability.

NIFS developed a virtual reality (VR) visualization system in which the space relationship between the plasma shape and components in the vessel of the LHD could be analyzed by using a CAVE system. In this VR visualization, simulation results of the equilibrium plasma in the LHD and the objective description of the LHD vessel device are represented simultaneously in the VR space [14]. The plasma pressure is shown with the isosurface, whereas the magnetic field lines and particle trajectories are displayed by curves.

#### 2.1.2 Surface-based methods

To the best of our knowledge, a few studies have outlined the plasma shape directly using 3D data. Paul, E. J., et al. [15] obtained the surface expression of the plasma shape with the specified winding surface geometry of a stellarator. They focused on numerical calculation rather than on generating surface using simulation data.

In our previous research [9], we studied the plasma shape by generating a 3D model through magnetic field lines, all of which were directly integrated into a scalar field. The MC method was used to detect an isosurface of this scalar field. The scalar data of each point in this scalar field were described as the number of magnetic field lines within a cube composed of seven other neighboring points. Furthermore, the Larmor radius was represented with a cluster of parallel lines surrounding each original magnetic field line so that the Larmor radius can become an integrated part of the 3D model. However, deciding the value of a point by counting the number of magnetic field lines is not an effective method for smoothing the generated surface. At the same time, interpolation is used to calculate the position of a triangular mesh so that a margin exists between the lines and generated surface. This margin will result in errors in the final interference check.

### 2.2 Visual analysis supported by deep learning

With the explosive growth of modern deep learning techniques, a growing number of researchers have been investigating the capabilities of DNNs to address different types of scientific visualization problems.

Han, Jun and his crew members [7] built a FlowNet for clustering and selection of streamlines and stream surfaces, but each line or surface was inputted orderly and independently. Guo, Li, et al. [6] built a novel deep learning framework that produces coherent spatial super-resolution of 3D vector field data. It helps on phase of data saving and simulation. Qi, Charles R., et al. [17] built a PointNet for 3D classification and segmentation on unordered point sets in 3D. The segmentation result is presented using different colors for different point groups. Mescheder, Lars, et al. [12] built occupancy networks, which implicitly represent the 3D surface as the continuous decision

Table 1. Datasheet of a point

Parameter	Value (unit)	Data type
Magnetic field line index	$n_m$	Integer
Toroidal angle index	$n_T$	
Radial coordinate	$R(\text{m})$	Double float
Toroidal angle	$\phi_T$ (degrees)	
X coordinate	$X(\text{m})$	
Y coordinate	$Y(\text{m})$	
Z coordinate	$Z(\text{m})$	
Magnetic field line connection length	$L_m(\text{m})$	
Radial component of magnetic field	$B_r(\text{T})$	
Axial component of magnetic field	$B_z(\text{T})$	
Azimuthal component of magnetic field	$B_t(\text{T})$	
Strength of magnetic field	$B(\text{T})$	

boundary of a DNN classifier. The result proved the high performance of DNNs on learning unstructured data.

### 3 PLASMA SHAPE IN 3D SPACE

In this section, first, we will introduce the source of magnetic field lines that were used in our research. The reason why we reconstructed a scalar field depending on DNNs is explained subsequently. Then, to construct a dataset for DNNs, annotation, labeling, and normalization were applied. Afterward, we will introduce the structure of the DNN, which is built using the TensorFlow library developed by Google and the representation of the Larmor radius. Finally, the MC method is presented.

#### 3.1 Magnetic field lines data

The magnetic vector field is calculated under static conditions. The precise structure of the magnet coils, vacuum vessel, and blankets are input to establish the 3D modeling of the inner environment of the FFHR for finite element calculation [10]. The magnetic field tracing code (MGTRC) [1] is used for tracing the magnetic field lines. At this time, 799 magnetic field lines are stored. Each line is given an index. All lines consist of almost  $4.8 \times 10^6$  points. A point belonging to a magnetic field line is stored in the following format, as shown in Table 1.

From the first line of content is the magnetic field line index that indicates which magnetic field line this point belongs to. The second parameter is the toroidal angle index. Information about a point will be recorded for certain degrees. The third parameter is the distance between the projection of a point onto the  $X - Y$  plane and the origin of coordinates. This third parameter is called radial distance, and the fifth and sixth parameters are  $X$  and  $Y$  coordinates, respectively. With the  $Z$  coordinate, the position of a point in 3D space is established. The eighth parameter indicates the length of a magnetic field line. The last four parameters are three components and the strength of the magnetic field at this point. These three components belong to an independent simple torus coordinate. This information will be used to calculate the Larmor radius at this point. Fig. 3 shows three magnetic field lines with different indices in their respective colors. Fig. 4 shows all magnetic field lines.

In Fig. 4, it illustrates why it is unsuitable to simply use the streamline visualization method. The critical point is to detect the interference with the components in the reactor. This means that lines located at the outer layer are likely required to be valued. However, as shown in the figure, different magnetic field lines overlap another because of the twisted magnetic field. It is difficult to distinguish the outer lines from those relatively close to the center. This results in many more magnetic field lines being rendered than were necessary. The situation not only causes occlusion but also places a burden on the computation. The representation of the Larmor radius is another difficult problem. Since the Larmor radius is a radius of the charged particle gyromotion centered at the magnetic field line on the plane whose normal vector is parallel to the magnetic field line, the Larmor radius, that is, the gyromotion, will be represented as a tube or a ring around the field streamlines. In this case, the magnetic field line is hidden by the tube,

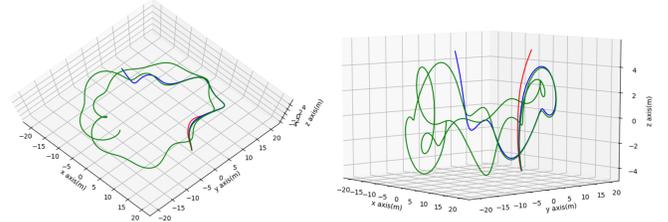


Fig. 3. Three magnetic field lines with different indices in respective color. The shortest one in red with index 10, the medium one in blue with index 214, the longest one in green with index 280.

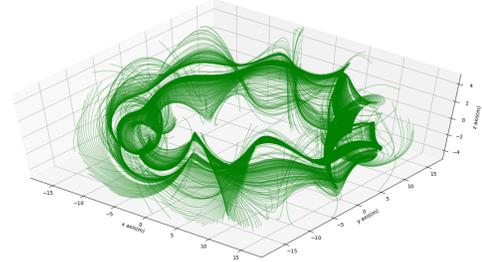


Fig. 4. Rendering of all the magnetic field lines with transparency.

and then, the visualization of the field line will incur significantly more computing resources. This is why we attempt to use the stream surface to represent the plasma shape.

Because a magnetic field line is traced as its 3D footprints, all magnetic field lines can also be considered an unordered point cloud. Since the information about magnetic flux intensity has been assigned to each point, it can also be known as a scalar field if the association of a point's location information and its magnetic flux intensity are considered. A point from this unordered scalar field can be expressed in the following way:

$$P(x, y, z | B_r, B_z, B_t) \quad (1)$$

The next step is the ordering of this scalar field. In this way, many classic surface generation methods that address the ordering of scalar fields, such as the MC method, can be used with ease.

Regarding the reconstruction of a new ordered scalar field, there are various methods based on the topological relationship between each point from the original unordered scalar field, such as clustering and interpolation. However, both are time-consuming when dealing with a large number of points. In this regard, a DNN is the best candidate for handling huge amounts of data.

#### 3.2 Data processing

After observing the magnetic field line data, the points at the divertor legs were found to be insufficient as learning data. Compared with the number of points at the flux surface, the weight of these points at the divertor legs is decreased. Eventually, the feature at the divertor legs will be ignored or considered as noise by the DNN. The possible solution is using congregated points from each symmetrical area. We obtained 36 pieces of Poincare plots, each of which has 1 degree

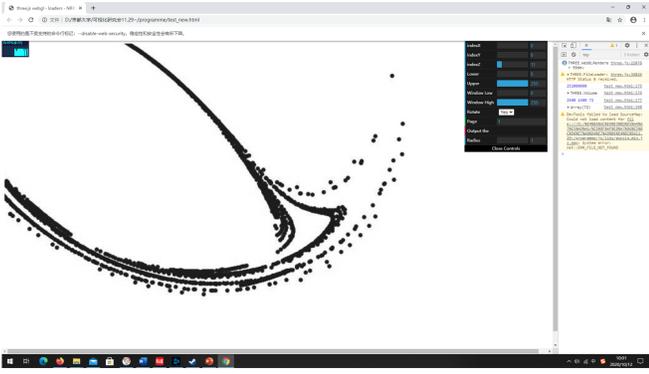


Fig. 5. User interface of the annotation tool. A Poincaré plot without flux surface point data being uploaded to it.

shifting along the toroidal direction. To reconstruct the most basic ordered scalar field for surface generation, these Poincaré plots are expanded to 90 degrees of the toroidal angles. The transformation rule is as follows:

- For a point from a Poincaré plot, we obtain its  $R$  and  $\phi_T$  from Table 1.
- Based on the toroidal angle  $\phi_T$  of this point, new toroidal angles are assigned as  $\phi_T + 36$  degrees,  $\phi_T + 72$  degrees. If  $\phi_T + 72$  degrees is greater than 90 degrees, it will be ignored.
- Calculating the new coordinates of  $(x_1, y_1)$ ,  $(x_2, y_2)$  (if exists) with new toroidal angles on the  $X - Y$  plane.

We did not intend to expand the Poincaré plots to 360 degrees for three reasons. First, as the expanded area has the same symmetry structure as the first 36 degrees, it cannot provide additional plasma shape information. Second, not expanding the Poincaré points would save considerable time in the adjustment phase of DNN optimization. Finally, the NIFS had been working on developing a new MGTRC, more point data at the divertor legs will be expected. Therefore, in this stage, we only considered the plasma shape from 0 to 90 degrees of the toroidal angle, which is also known as a quarter of the entire plasma shape.

### 3.2.1 Points annotation

To detect the shape of edge surface layers and the divertor legs, some assisting points are necessary. These assisting points are marked using a points annotation tool [2]. This procedure is completed before expanding the Poincaré plots to 90 degrees.

The user interface of the annotation tool is shown in Fig. 5. To use this annotation tool, a Poincaré plot is first transformed into an image. After uploading an image of a Poincaré plot to it, the points from the Poincaré plot will be displayed in the main area of the annotation tool. Once a user double clicks the mouse at the main area, the location of the mouse relative to the image of a Poincaré plot will be recorded. The assisting points are taken averagely at the vicinity outside the edge surface layers and the divertor legs. Then, location information will be downloaded as a comma-separated Values file. The above process was followed for all 36 pieces of images.

### 3.2.2 Labeling and normalization

To find the outermost closed curved surface of the flux surface and the edge surface layers, respectively, labeling is required to indicate which place a point is from the Poincaré plot. Points within the flux surface are labeled as “2”. At the same time, points within the edge surface layers but outside the flux surface are labeled as “1”. Additionally, the assisting points taken manually from the annotation tool are labeled as “0”. Labeling is processed soon after expanding the Poincaré plots. For

each Poincaré plot, it has almost 1900, 11700, and 49200 points with labels “0”, “1” and “2”. Because the assisting points are made up of a single layer, there are fewer of them. Additionally, there are 90 times as many points of a Poincaré plot in total. After finishing the labeling, the sequence of the points is disrupted and stored.

With respect to data normalization, at the input side,  $\phi_T$  has a different unit from the other variables. However, on the output side, since an output activation function has been tailored to the distribution of the output variables, it is not necessary to rescale the output dataset.

## 3.3 Network structure

A fully connected DNN is built in this study. Specifically, we customize the loss function. It is introduced in the next part. The structure, selection of a loss function, and an optimizer, and other information of this DNN are as follows:

- Structure: This DNN has five layers, with numbers of nodes 3-512-128-16-3 along the forward direction. The first and the last 3 are the numbers of the nodes of the input and output layers, respectively.
- Activation function: Rectified linear unit activation is used, except for the last layer of the DNN. The last layer uses softmax activation, which normalizes the result into a probability distribution in the interval  $(0, 1)$ , and the components of the result will add up to 1.
- Loss function: The sum of a sparse categorical cross-entropy (SCC) and a customized part.
- Optimizer: RMSProp.
- Metrics: Accuracy, which indicates the accuracy of evaluation on the validation dataset.
- Dataset: The training and validation datasets are 90 and 10 percentages of points from the unordered scalar field. The test dataset comprises uniformed points created for constructing the ordered scalar field. Based on the scale of the plasma shape, the size of the ordered scalar field is both  $0 \sim 24$  meters on the  $X$  and  $Y$  coordinates, and  $-7 \sim 7$  meters on the  $Z$  coordinate. The default distance between each point on the  $X$ ,  $Y$ , and  $Z$  coordinate is 0.1 meters. Note that the distance of each point could be changed according to a user’s requirement. If a user wants a higher resolution result, he/She can set the distance of each point to be smaller than 0.1 meters. The input dataset format is as follows:

$$P(x, y, z | label) \quad (2)$$

The output format is

$$P(x_1, y_1, z_1 | p_0, p_1, p_2) \quad (3)$$

where  $(x_1, y_1, z_1)$  is a new point from the ordered scalar field and  $p_0$ ,  $p_1$ , and  $p_2$  represent a probability distribution, indicating which point group the new point probably belongs to. The sum of this probability distribution equals 1, which is exactly how the softmax activation works. Additionally, the highest probability from  $p_0$ ,  $p_1$ , and  $p_2$  is chosen and assigned to the new point.

Fig. 6 shows points with label “1” from the new ordered scalar field in 3D space.

## 3.4 Customized loss function

A customized loss function is composed of an SCC and an average of dot product results. We reference the loss error strategy proposed by Raissi, M., et al. [18] in their physics-informed neural network to replace the mean squared error. Considering a point, along with its Cartesian coordinate, the partial derivatives of this point are obtained from the DNN. Then, we calculate the dot product between the magnetic flux vector and partial derivatives. The average of all dot products

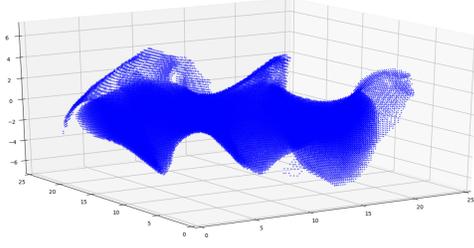


Fig. 6. Points with label “1” from the ordered scalar field have been selected and rendered in blue.

should be close to 0. The DNN is trained to follow a forward direction for each magnetic field line. The customized loss function is as follows:

$$Loss = L_{SCC} + \frac{1}{n} \sum_i^n \left| \left( \frac{\partial f}{\partial x_i}, \frac{\partial f}{\partial y_i}, \frac{\partial f}{\partial z_i} \right) \cdot (B_{x_i}, B_{y_i}, B_{z_i}) \right| \quad (4)$$

where,  $n$  represents the number of input points and  $f$  represents the DNN.

### 3.5 Larmor radius representation

The Larmor radius would improve precision when predicting the interferences of the plasma shape and the components in the FFHR. A charged particle performs gyration motion along the magnetic field lines. The radius of the gyration motion depends on the velocity of the particle and the strength of the magnetic field. Assume that  $v_{\perp}$  is a velocity that is perpendicular to the magnetic field. it can be calculated as follows:

$$v_{\perp} = v_{th} = \sqrt{\frac{2kT}{m}} \quad (5)$$

where  $v_{th}$ ,  $k$ ,  $T$ , and  $m$  represent the thermal velocity, the Boltzmann constant, the temperature and the mass of a charged particle, respectively. In our research,  $kT$  is the energy with 3.5 MeV. In this case, the particle moving to the divertor legs will collide with the vessel and lose its energy. Hence, 3.5 MeV is considered the least energy that a particle should keep. The particle is an alpha particle produced in the process of nuclear fusion, which has a mass four times of a proton,  $6.645 \times 10^{-27}$  kg. Consequently, the Larmor radius, denoted by  $r_L$ , can be calculated as follows:

$$r_L = \frac{mv_{\perp}}{qB} \quad (6)$$

where  $q$  and  $B$  represent the electric charge of the alpha particle, which is twice that of a proton,  $3.204 \times 10^{-19}$  C, and the intensity of the magnetic field at that point, respectively.

Since we only focus on the Larmor radius of points at the outline, we choose points with label “1”. Subsequently, the Larmor radius of all points with label “1” is calculated. Afterward, the distance between this point and surrounded points will be reordered as an updated scalar value. The surrounded points are found by judging whether a point is inside the sphere with the center of this point and radius equal to this point’s Larmor radius. The judgment equation is as follows:

$$V = r_L - \sqrt{i^2 + j^2 + k^2} \quad (7)$$

where  $V$  represents the updated scalar value of a surrounded point, and  $i$ ,  $j$ , and  $k$  represent integer multiples with positive or negative of the default distance, either 0. The integer is also defined as the loops start from 1. The maximum integer is determined at a certain loop, all updated scalar values of surrounded points are smaller than 0. For example, in loop one, with the default as 0.1 meters, the relative

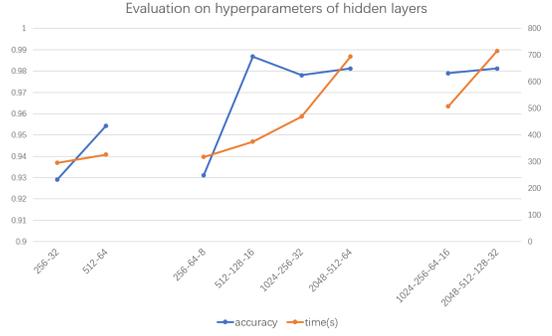


Fig. 7. The training time and accuracy on an average of different hidden layer settings. The blue lines show the time(s) difference. The orange lines show the training accuracy.

coordinates of a candidate point are the combination of the  $i$ ,  $j$ , and  $k$  from the list “-0.1, 0, 0.1”. If a  $V$  is larger than 0, it means this candidate point is in the sphere, which also means that it needs to enter loop two. In loop two, the relative coordinates of a candidate point will be the combination of the  $i$ ,  $j$ , and  $k$  from the list “-0.2, -0.1, 0, 0.1, 0.2”. If in this loop, all updated scalar values are smaller than 0. The Larmor radius calculation will be stopped. Then, the absolute coordinates and scalar value of a surrounded point within the Larmor radius can be decided as:

$$P(x_2, y_2, z_2 | V), \begin{cases} x_2 = x_1 + i \\ y_2 = y_1 + j \\ z_2 = z_1 + k \end{cases} \text{ for } i, j, k \text{ in a list} \quad (8)$$

Since a point will be involved in multiple spheres, the largest of all scalar values calculated at such a point is retained. In this way, the sphere boundary generated on the basis of the largest value of this point can hold any other sphere boundary.

### 3.6 The MC method

In the MC algorithm [11], the threshold is set as 0 and 1.5, respectively. To prevent the effect of updated scalar values of Larmor radius, we apply the MC method twice with different point groups. When the threshold is 0, it generates triangular elements at the exact position of the boundary of any sphere. When the threshold is 1.5, triangular elements are generated between the edge surface layers and LCFS.

## 4 EXPERIMENT

In this section, we first discuss the selection of hyperparameters of the DNN. Then, we compare the results using an SCC and a customized loss function. The final part presents the visualization result of the plasma shape. The program is compiled with Python version 3.6. The part of deep learning is run on a GPU for mobile, which mode is “RTX 2070 MAX-Q”.

### 4.1 Selection of hyperparameters

We first evaluate the model performance of different hidden layers. The number of hidden layers ranges from 2 to 4, and a layer has nodes of powers of 2. The training time and accuracy in average are shown in Fig. 7. The training time increases gradually with the complexity of hidden layers, where accuracy remains constant when the first hidden layer is greater than 1024. Given the calculation time of the partial derivatives, which is proportional to the number of nodes and accounts for approximately half of the training time, it was decided to use hidden layers in 512-128-16.

We then evaluate a suite of different batch sizes from 500 to 3000 in steps of 500. For each batch size, we also change the number of epochs from 10 to 40 in steps of 10. Hence, we have a total of 24 results. The comparison of different batch sizes and training epochs is shown in Fig.

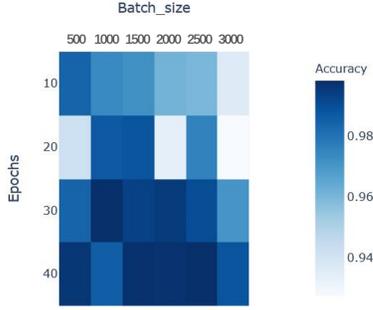


Fig. 8. Heatmap with different combinations of batch sizes and epochs. The color of block indicates the accuracy under certain combinations. The darker the color, the better the effect of the combination.

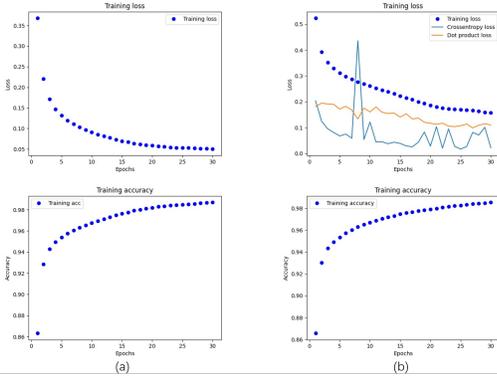


Fig. 9. Training loss and training accuracy of the DNN with an SCC loss function (a) and a customized loss function (b), respectively. In (b), it also records the changing of cross-entropy and dot product loss individually.

8 with the heatmap. The density of the color denotes accuracy. The darker the color, the better the effect of the combination.

It can be seen from Fig. 8 that when the epoch is 10, accuracy decreases gradually from 500. When the epoch is 20, the result is unstable. For epochs 30 and 40, the accuracy remains at an average high level. Additionally, we can see that the batch size has little effect on accuracy at line 40. Considering time efficiency, we adopt the combination of 30 epochs and 2000 batch size.

## 4.2 Loss function performance

Based on the above setting, the DNN is trained with an SCC loss function and a customized loss function as a comparison. Fig. 9 shows the training processes. Both loss functions decrease gradually and are nearly identical. After rendering two plasma 3D models, as shown in Fig. 10, it demonstrates that the customized loss function can bring a complete structure.

## 4.3 Visualization of plasma shape

The rendering of the plasma shape model uses an open-source graphing library called “Plotly” [3]. We used “Plotly” to check the plasma shape model developed with the proposed method. The visualization result is shown in Fig. 11. The arrangement of the figures is at a viewpoint of  $X - Y - Z$ ,  $X - Y$ ,  $Y - Z$ ,  $X - Z$  respectively, from the top left following a clockwise direction.

The model accurately reproduces the complicated plasma shape structure. Compared with rendering using line objects, it clearly distinguished the two areas. Furthermore, because of the surface representation, it is possible to check the interferences directly between the

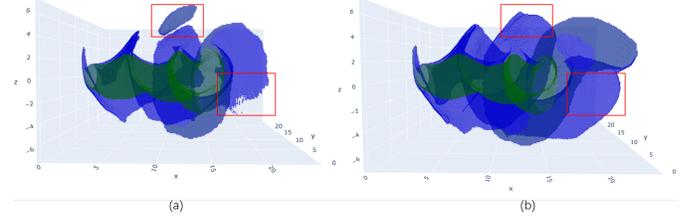


Fig. 10. Rendering of plasma 3D models. (a) For an SCC loss function. (b) For a customized loss function. The red frames show the difference in models. It proves that (b) is more complete.

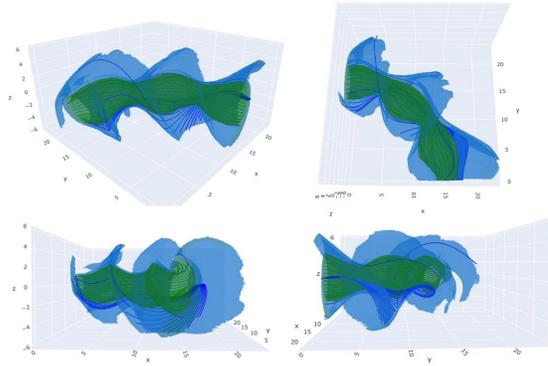


Fig. 11. Visualization result of a quarter of the plasma shape. It is rendered with Plotly, a graphing library. The surface represents the edge surface layers and part magnetic field lines inside it, which are in blue, and the LCFS and part magnetic field lines inside it are in green. From the top left in a clockwise direction, are the viewpoints at  $X - Y - Z$ ,  $X - Y$ ,  $Y - Z$ ,  $X - Z$ .

plasma shape and 3D design data of components in the FFHR. The rendering result of the plasma model and the fusion reactor model is shown in Fig. 12.

## 5 EVALUATION

In this section, the performance of the 3D model is evaluated by slicing the model along the toroidal direction. Furthermore, we would like to share the feedback from domain experts at the NIFS.

### 5.1 Three-dimensional model performance

To evaluate the accuracy of the 3D model, we chose to return to the 2D space by slicing the model along the toroidal direction. Then, original magnetic field lines at the same toroidal angle with the Larmor radius were added on those slices. Finally, we plot these slices to evaluate the performance of the plasma 3D model. The slices are shown in Fig. 13. From the left to right are the slices at 0 degree, 10 degrees, 30 degrees. The blue points represent the cross-section of the magnetic field lines, including the Larmor radius. The green points represent the outlines of the 3D model, which are also the boundary of the two areas of the plasma shape.

In Fig. 13, it can notice that one or two circles are neglected by the DNN when learning the representation of the plasma shape. This is because there are no other circles around the circles. It means that a few magnetic field lines pass at this place. Thus, the points labeled as “1” are few and assisting points labeled as “0” have taken advantage of this place. When new points with coordinates similar to these isolated points are obtained from the ordered scalar field and are to be labeled, the trained DNN will produce a label of “0” rather than “1” according

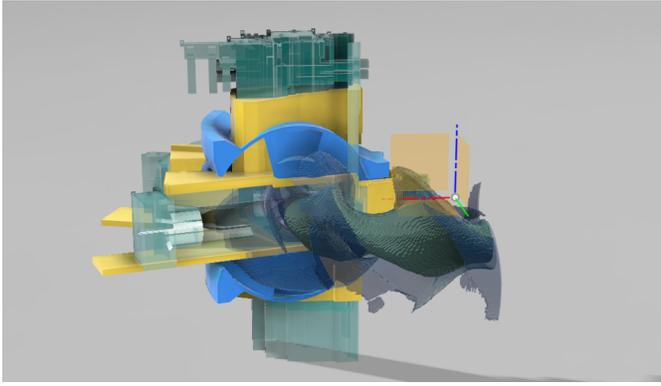


Fig. 12. Rendering result of the plasma model and the fusion reactor model created with graphics software Fusion360.

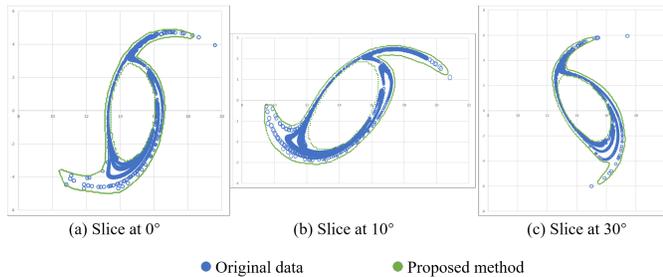


Fig. 13. Slices at (a) 0, (b) 10 and (c) 30 degrees, respectively. The green points indicate the slice of the model. The blue circles represent points from magnetic field lines with their Larmor radius.

to the probability distribution of point labels at this place. Despite the above issue, the good processing ability to the non-linear information makes DNN a good candidate for analyzing complex plasma shapes.

Compared with the approach in our previous paper [9], the DNN approach solves the existing integrity and margin problem. The DNN approach has fewer processing procedures on the Larmor radius. Furthermore, the DNN approach has less data volume than before but achieves high performance when describing the plasma shape, costing MC method less time.

## 5.2 Feedback from the domain experts

The result and relative data were sent to the domain experts at the NIFS for feedback. Since fusion reactor design is a long and complex process, obtaining a comprehensive and precise evaluation takes a long time. In this stage, we would like to share their perspectives and suggestions toward our approach.

The different areas of the plasma shape are divided according to the connecting lengths of magnetic field lines. To achieve high precision, the frequency of tracing should be increased. Additionally, to obtain a global view of the plasma shape, traced magnetic field lines are expected to have an even distribution. Consequently, the location of the starting points is set and changed frequently. After applying the DNNs, with only limited numbers of magnetic field lines, the representation of the plasma shape can be accurately described.

The surface-based method compensates for the shortcomings of the traditional method. With information regarding surface meshes, the detection of the interference can be easily accessed with a computer-aided design software application.

## 6 CONCLUSION AND FUTURE WORK

In this research, we studied the visualization method of the plasma shape in the FFHR. Because of the helical coils in this reactor, the simulation result of the magnetic field lines shows a complex structure. As existing methods are either inefficient or counterintuitive, we propose

the introduction of deep learning technology into the visualization of the plasma shape in 3D space. The distribution of magnetic field lines is fed to a DNN to learn the representation of the plasma shape. Then, a new ordered scalar field is reconstructed. The Larmor radius has also been integrated into the ordered scalar field. Finally, the MC method is used to discover the isosurface of this scalar field. This isosurface accurately represents the plasma shape with its Larmor radius.

In future work, we would like to explore new structures of the DNN that are specialized in plasma shape visualization. Instead of using the MC method, we would like to discover a possible way to generate the surface mesh directly from the DNN or a function that can fully describe the plasma shape. Furthermore, we would like to extend the application of DNN to other complex surface generations. We believe that the non-linear processing ability of the DNN will achieve great success in these fields.

## ACKNOWLEDGMENTS

This work is supported by the NIFS within the Fusion Engineering Research Project (FERP). This work is also performed with the support and under the auspices of the NIFS Collaboration Research program (NIFS19KNTS057, NIFS18KKG022 and NIFS17UFFF040), and some calculation is performed with the ‘‘Plasma Simulator’’ of NIFS.

## REFERENCES

- [1] Project webpage. 2010. <https://github.com/yasuhiro-suzuki/MGTRC>.
- [2] Points annotation tool webpage. 2019. <https://github.com/Zhongjiangan/>.
- [3] Plotly webpage. 2020. <https://plotly.com/python/>.
- [4] G. Ambrosino and R. Albanese. Magnetic control of plasma current, position, and shape in tokamaks: a survey of modeling and control approaches. *IEEE Control Systems Magazine*, 25(5):76–92, 2005.
- [5] T. Goto and A. Sagara. System design of the heliotron-type reactor fhr. In *2015 IEEE 26th Symposium on Fusion Engineering (SOFE)*, pp. 1–6. IEEE, 2015.
- [6] L. Guo, S. Ye, J. Han, H. Zheng, H. Gao, D. Z. Chen, J.-X. Wang, and C. Wang. Ssr-vfd: Spatial super-resolution for vector field data analysis and visualization. In *2020 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 71–80. IEEE Computer Society, 2020.
- [7] J. Han, J. Tao, and C. Wang. Flownet: A deep learning framework for clustering and selection of streamlines and stream surfaces. *IEEE transactions on visualization and computer graphics*, 26(4):1732–1744, 2018.
- [8] D. Hartmann. Stellarators. *Fusion science and technology*, 57(2T):46–58, 2010.
- [9] K. Hu, Q. Wang, K. Koyamada, H. Ohtani, T. Goto, and J. Miyazawa. Visualization of the plasma shape in a force free helical reactor, fhr. *Journal of Advanced Simulation in Science and Engineering*, 7(1):151–167, 2020.
- [10] X. Ji, N. Yanagi, T. Goto, H. Tamura, J. Miyazawa, A. Sagara, S. Wang, M. Qi, and Y. Song. Investigation on the magnetic field distortion by ferromagnetism of the blanket for the helical fusion reactor. *Fusion Engineering and Design*, 125:631–634, 2017.
- [11] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987.
- [12] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4460–4470, 2019.
- [13] T. S. Newman and H. Yi. A survey of the marching cubes algorithm. *Computers & Graphics*, 30(5):854–879, 2006.
- [14] H. Ohtani, Y. Tamura, A. Kageyama, and S. Ishiguro. Scientific visualization of plasma simulation results and device data in virtual-reality space. *IEEE Transactions on Plasma Science*, 39(11):2472–2473, 2011.
- [15] E. Paul, M. Landreman, A. Bader, and W. Dorland. An adjoint method for gradient-based optimization of stellarator coil shapes. *Nuclear Fusion*, 58(7):076015, 2018.
- [16] A. Pironti and M. Walker. Fusion, tokamaks, and plasma control: an introduction and tutorial. *IEEE Control Systems Magazine*, 25(5):30–43, 2005.
- [17] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the*

- IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017.
- [18] 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. *Journal of Computational Physics*, 378:686–707, 2019.
- [19] A. Sagara, H. Tamura, T. Tanaka, N. Yanagi, J. Miyazawa, T. Goto, R. Sakamoto, J. Yagi, T. Watanabe, S. Takayama, et al. Helical reactor design ffhr-d1 and c1 for steady-state demo. *Fusion Engineering and Design*, 89(9-10):2114–2120, 2014.
- [20] H. Tamura, T. Tanaka, T. Goto, J. Miyazawa, S. Masuzaki, T. Watanabe, N. Yanagi, A. Sagara, S. Ito, and H. Hashizume. Novel divertor design to mitigate neutron irradiation in the helical reactor ffhr-d1. *Fusion Engineering and Design*, 98:1629–1633, 2015.
- [21] T. Watanabe, Y. Matsumoto, M. Hishiki, S. Oikawa, H. Hojo, M. Shoji, S. Masuzaki, R. Kumazawa, K. Saito, T. Seki, et al. Magnetic field structure and confinement of energetic particles in the lhd. *Nuclear fusion*, 46(2):291, 2006.