Investigating pedestal dependencies at JET using an interpretable neural network architecture

PAPER • OPEN ACCESS

Investigating pedestal dependencies at JET using an interpretable neural network architecture

To cite this article: A. Gillgren *et al* 2025 *Nucl. Fusion* **65** 056033

OPEN ACCESS
IOP Publishing | International Atomic Energy Agency
Nuclear Fusion
Nucl. Fusion **65** (2025) 056033 (20pp)
https://doi.org/10.1088/1741-4326/adcbc2

Investigating pedestal dependencies at JET using an interpretable neural network architecture

A. Gillgren[1,*], A. Ludvig-Osipov[1,2], D. Yadykin[1], P. Strand[1] and JET contributors[a]

[1] Chalmers University of Technology, Gothenburg, Sweden
[2] United Kingdom Atomic Energy Authority, Culham Centre for Fusion Energy, Abingdon, United Kingdom of Great Britain and Northern Ireland

E-mail: andreas.gillgren@chalmers.se

Received 14 January 2025, revised 27 March 2025
Accepted for publication 11 April 2025
Published 24 April 2025

Abstract

We present NeuralBranch, an interpretable neural network framework. In this work, we use it specifically to predict the pedestal from key engineering parameters in tokamak fusion experiments. The main goal is to uncover intricate relationships that traditional power scalings, with their limited expressive capacity, fail to capture. A secondary objective is to provide a transparent alternative to current opaque, black-box machine learning models used to predict the pedestal in integrated modeling frameworks. By using the proposed method, we obtain a novel global overview of several intricate dependencies in the JET pedestal database. For instance, while both input power and plasma current are positively correlated with pedestal top pressure and temperature, NeuralBranch reveals an attenuating interaction. This means that increasing power weakens the impact that current has on pedestal pressure and temperature, and vice versa. Further investigation of this interaction may be important to avoid overestimating pedestal stored energy at future machines like ITER when using established power scalings. We also identify an amplifying interaction between plasma current and triangularity, where higher triangularity amplifies the effect of plasma current on pedestal density, and vice versa. In addition to these findings, NeuralBranch matches the accuracy of black-box neural networks, with R^2 values as high as 0.88. This demonstrates that interpretability, with its associated benefits, can be achieved without sacrificing accuracy, making NeuralBranch a promising alternative for pedestal predictions.

Keywords: fusion, tokamak, pedestal, interpretable, machine learning, ai, NeuralBranch

(Some figures may appear in colour only in the online journal)

[a] See Maggi *et al* 2024 (https://doi.org/10.1088/1741-4326/ad3e16) for JET contributors.
[*] Author to whom any correspondence should be addressed.

1. Introduction

The most studied magnetic confinement device is the tokamak, which is also the basis for ITER, a large fusion experiment designed to demonstrate fusion on a reactor scale. Despite that tokamak plasma physics has been extensively explored over the years, there are still aspects that are not fully understood. One of these aspects is the High Confinement Mode (H-Mode) [1], the baseline scenario planned for the operation of ITER. H-Mode involves intricate non-linear physics that are difficult to simulate from first principles. In this mode, turbulent energy transport is suppressed in a narrow region at the outer edge of the plasma when the power delivered to the plasma exceeds a certain threshold. This suppression leads to a steep pressure gradient near the edge, creating what is known as the pedestal. The level of the plasma pressure, density, and temperature just inside this region are referred to as the pedestal values. The pedestal plays an important role for the global energy confinement of the plasma, and it is therefore essential to understand the key parameters determining its properties.

An important factor for guiding the theoretical investigation of H-mode physics are the pedestal characteristics observed in experiments. Historically, studies have involved parameter scans, where specific tokamak or plasma parameters are varied while others remain constant [2–15]. This approach has been valuable for understanding how the pedestal depends on different parameters. However, such scans typically involve only a limited number of tokamak pulses, which may produce results that are only applicable to specific scenarios.

Another traditional approach for studying pedestal characteristics, as well as other confinement-related properties, is to use curve-fitting techniques like multivariate power scalings [4, 16–18]. In contrast to parameter scans, power scalings allow for the simultaneous analysis of how the pedestal depends on multiple tokamak parameters across large datasets. This helps mitigate the limitation of relying on data from only a few pulses, and indeed, power scalings have proven useful for identifying general trends. However, because power scalings are intentionally made simple for easier interpretation, they are rather restrictive and cannot capture more intricate relationships, such as certain interaction effects between the inputs of the model.

Recently, machine learning models, such as neural networks, have been employed to predict the pedestal from key tokamak parameters [19, 20]. This has resulted in more accurate predictions compared to simpler models, indicating that more intricate relationships exist between the pedestal and the tokamak parameters. However, these machine learning models have not been interpretable. In this context, lack of interpretability in machine learning models refers to the difficulty of extracting the learned relationships between the input and output parameters in a human-comprehensible form. This is problematic because it not only limits our ability to gain insights from the data but also undermines trust in the predictions, as the reasoning behind the model predictions remains unclear. This drawback of feedforward dense neural networks (referred to as regular neural networks in this paper) is commonly known as the black-box problem.

Fortunately, recent advancements in models that provide interpretability, often termed glass-box models, have demonstrated success across various applications. For instance, symbolic regression [21, 22] refers to a framework that automatically finds mathematical expressions that best describe the relationship between the inputs and the output. Another example is the Neural Additive Model (NAM) [23], which employs separate neural networks for the different input parameters, the model output being the sum of the outputs of the individual networks. In this framework, interpretability is achieved by graphically visualizing the output of each individual network versus its inputs. As glass-box models provide the overarching patterns learned across the full dataset they are trained on, they are said to exhibit *global* interpretability.

While glass-box models have been applied in certain areas of fusion energy and plasma physics, such as symbolic regression for confinement time scaling [24], their potential have not been widely explored in the sub-field of pedestal research.

1.1. Scope of work

The main goal of this work is to use an interpretable machine learning method to uncover intricate relationships between the pedestal and key tokamak parameters, beyond what power scalings can capture. Here, we analyze a large dataset, the Joint European Torus (JET) pedestal database [4], to provide a comprehensive overview of the relationships such that the results are not limited to a few pulses. To achieve our goal, we introduce a neural network based framework that we call *NeuralBranch*, which utilizes visualizations to facilitate global interpretability. While we acknowledge that NeuralBranch is inspired by NAMs, it overcomes two limitations that NAMs exhibit: (1) the enforced summation of network outputs, and (2) the restriction to pairwise interactions due to the parallel arrangement of individual networks in NAMs. Hence, another goal of this paper is to introduce, to our knowledge, a novel glass-box framework, which is applicable to different fields beyond that of pedestal physics and fusion energy.

In addressing the two objectives mentioned above, we will consequently also address an additional aim, which is to develop interpretable models capable of predicting the pedestal within integrated modeling simulations of tokamak plasmas. Specifically, we seek to offer a transparent alternative to existing black-box pedestal models used in integrated modeling frameworks, such as the European Transport Simulator (ETS) [25, 26].

The outline of the paper is as follows: we initially present the NeuralBranch methodology using a constructed toy example. Then, we apply the framework to the pedestal data, where results are presented, first for the pedestal pressure, and then for the density and temperature. The appendix includes additional details about the proposed framework, along with a comparison to the local interpretability method SHapley Additive exPlanation (SHAP) [27] applied to a Random Forest, which demonstrates why global interpretability methods like NeuralBranch are preferable to local interpretability methods for this application.

2. Methodology demonstration

2.1. Visualizing neural network mappings

As a first step in demonstrating the method proposed in this paper, we here show how interpretability can be facilitated by visualizing neural network mappings.

Consider a toy dataset generated with the equation

$$y = A\sin(10x) \tag{1}$$

where y is an output parameter of interest, and where A and x are two independent input parameters randomly sampled from a uniform distribution between 0 and 1. A neural network can be trained to predict y from A and x, as illustrated in figure 1. Here, $\hat{y}$ represents the prediction of the output, in contrast to the actual output y.

Assume that post training, the neural network is able to make accurate predictions. Also pretend that the true underlying equation (1) is unknown, and that our goal is to investigate the relationship between the parameter of interest y and the inputs x and A. Usually, interpreting the overarching mapping of a neural network is a non-trivial task due to the many connecting nodes in a neural network (the black-box problem). Fortunately, in this case, having only two input parameters enables another method for interpretation: visualization. Post training, we can parse the data through the model and plot the predicted output $\hat{y}$ as a function of the inputs for the full dataset. For instance, $\hat{y}$ can be plotted versus x with A dictating the color, as in figure 2. Here we can see that the neural network has learned a sine-relationship between $\hat{y}$ and x, where A affects the amplitude. This is in full agreement with the original equation (1). Note that the trainable parameters of the network do not need to be analyzed to interpret the model, as all we need for the visualization are the predicted output $\hat{y}$, along with the inputs x and A. In essence, the visualization technique facilitates a qualitative approach to investigating parameter dependencies.

Of course, with only two input parameters, it is not even necessary to use a neural network to investigate the relationship between y and the two inputs. We could simply plot the true output y versus x and A directly. However, the principle of visualizing neural networks serves as the cornerstone for the method we propose in the next section, which is why its introduction and emphasis are warranted here.

2.2. Interpretable branch-based architecture

Here, we demonstrate the main idea behind our interpretable framework NeuralBranch by considering a new toy data set generated with the equation

$$y = A\sin(10x) + B. \tag{2}$$

Similarly to the previous example, A, x and B are inputs that are randomly and independently sampled from a uniform distribution between 0 and 1. Assume again that our goal is to find the relationship between y and the three input parameters. A neural network could be trained to predict y from A, x, and B. However, with three inputs, visualizing the dependencies becomes more challenging, as it would require a 3D plot along with a color indicator. With additional parameters, interpreting the visualization of a regular neural network becomes even more challenging, if not infeasible.

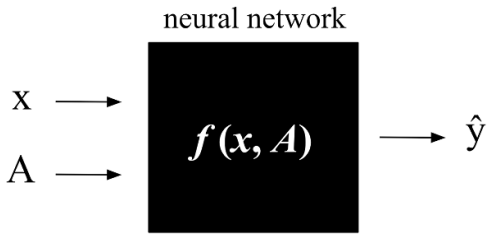


Figure 1. An illustration of a neural network designed to predict the parameter y from x and A. f represents the functional mapping of the network.

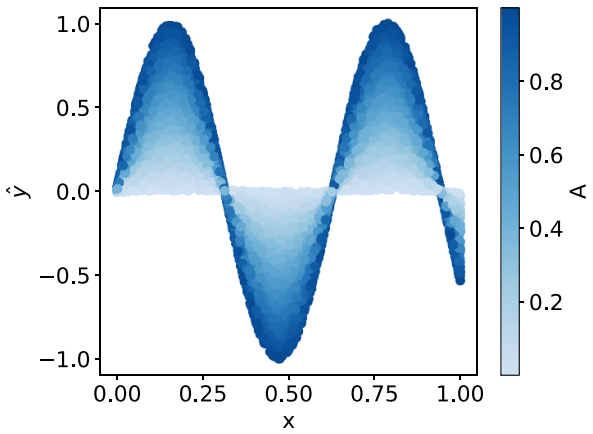


Figure 2. A visualization of the functional mapping learned by the neural network that predicts y from x and A, trained on a data set generated with equation (1).

To address this issue, our proposed NeuralBranch approach splits the neural network architecture into individual networks that we call *neural branches*, each handling only two input parameters and one output parameter. An example of such architecture is illustrated in figure 3. By only allowing two parameters to be parsed through each branch, visualization is enabled, and thus global interpretability is achieved for the full model. Additionally, since each neural branch essentially is a dense neural network, high expressive capacity is maintained.

By training a NeuralBranch model with the same architecture as illustrated in figure 3, on a data set generated with equation (2), a model that makes accurate predictions of y is achieved. For the interpretation, data is parsed through the model, and the predictions of neural branch 1 and neural branch 2 are visualized in figures 4(*a*) and (*b*) respectively. In summary, these plots show that neural branch 1 has learned to calculate the first term in equation (2), and that neural branch 2 effectively performs the addition operation in equation (2). Note that for this example, it is necessary to analyze the intermediate parameter z to grasp the relationship between the predicted output $\hat{y}$ and the inputs x and A. We also emphasize

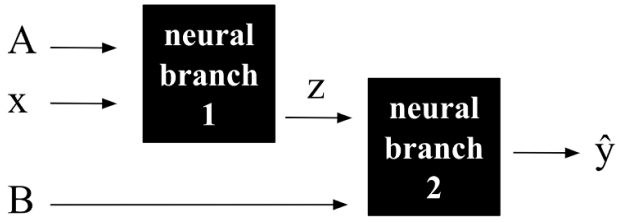


Figure 3. An example of a NeuralBranch model. Each neural branch (boxes) contains dense neural network layers. In this case, neural branch 1 calculates an intermediate value z from the input parameters A and x. This intermediate value is then forwarded to neural branch 2, which calculates the predicted output $\hat{y}$ from z and B.

that even though the neural branches in this model essentially are individual neural networks, they are trained together as one model. This means that prior knowledge of the intermediate parameter z is not required. In summary, by using the NeuralBranch model, we have achieved a fully transparent picture of the relationships between the output y and the three inputs A, x, and B.

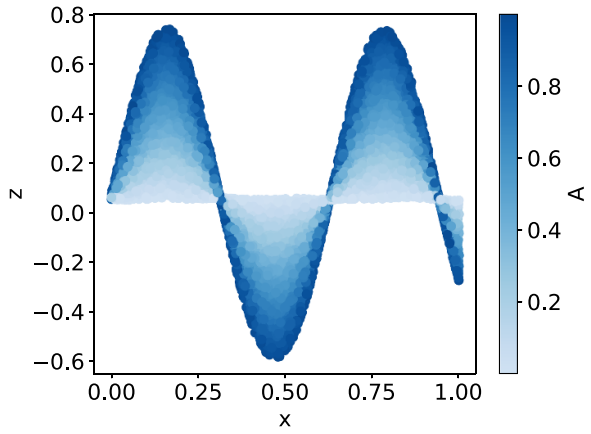
Allowing neural branches to be connected not only in parallel but also serially enables interactions between more than two parameters, addressing one of the limitations of NAMs mentioned in the introduction. Additionally, because a neural branch is always present at the end of the model in NeuralBranch, the other limitation of NAMs, enforced summation of network outputs, is also relaxed.

2.3. Determining which inputs to assign to each neural branch

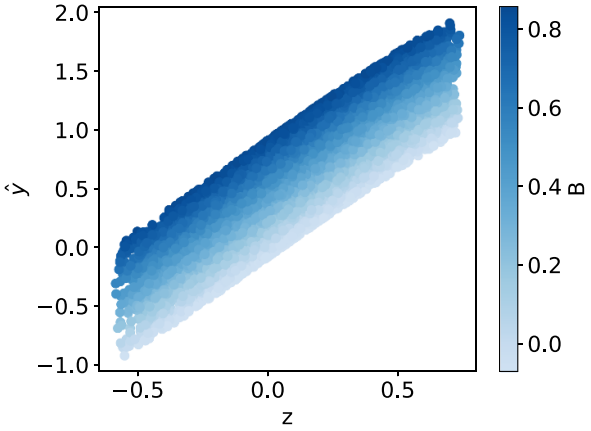
We have not yet explained an essential step in the NeuralBranch framework: how to determine, without prior knowledge of the data, which inputs to assign to each neural branch. In the current version of the framework, our approach is to test all possible ways to split the inputs into the different branches, and to select the configuration that leads to the highest prediction accuracy, as it best reflects the data. For instance, in the example based on equation (2), the split $[[A,x],[B]]$, which is illustrated in figure 3, results in the most accurate model. This is expected, as perfectly reflecting equation (2) would be impossible if B, together with A or x, were first reduced to z in neural branch 1.

An important detail about our approach is that we test all possible input split configurations 3 times, and we use only the highest scoring iteration for the evaluation of each split. This strategy ensures that no configuration is overlooked due to rare instances where the trainable weights of the models converge to a local optimum, resulting in a lower score.

More details regarding the assignment of inputs to the different neural branches are presented in the appendix. This includes a description of how the process is performed in steps for cases with more than three input parameters. The appendix also includes a description of the hyperparameters used in the training of the models in this work, as well as a description of how we handle cases where no split configurations yield



(a) Visualization of the mapping of neural branch 1 in Figure 3.



(b) Visualization of the mapping of neural branch 2 in Figure 3.

Figure 4. Visualizations of the two neural branches in figure 3.

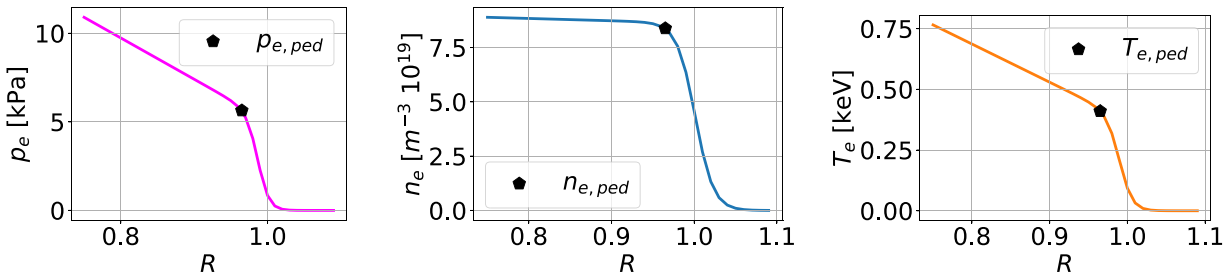
an accuracy that is considered close enough to the benchmark accuracy obtained with a regular neural network. For instance, this can occur when one or several input parameters are not easily separable.

For the pedestal related results presented later in this paper, we only show the final NeuralBranch architecture for each prediction case, along with the visualization of the neural branches.

2.4. Concluding remarks about NeuralBranch

We conclude this introduction to the NeuralBranch framework by highlighting two aspects that have not yet been mentioned:

- Since a NeuralBranch model is transparent, it is easier to detect irregular or overly complicated patterns that indicate overfitting, compared to black-box models.
- We expect NeuralBranch to be useful for cases with relatively few ($\approx< 10$) input parameter. This is because more inputs necessitate the incorporation of more neural branches, which makes the interpretation process more comprehensive as the branches depend on one another.



a) Electron pressure p_e b) Electron density n_e c) Electron temperature T_e

Figure 5. Sketches of electron pressure, density, and temperature profiles in the plasma edge region. Here, the radial coordinate R is normalized, with $R = 1$ representing the last closed flux surface (LCFS). The pedestal structure is illustrated in the profiles, with the pedestal top values $p_{e,\text{ped}}$, $n_{e,\text{ped}}$ and $T_{e,\text{ped}}$ indicated at the top of the steep regions. These are the three values we predict in this work.

3. Pedestal data set

In this work we are focusing on predicting the pedestal top values in the JET pedestal database to perform our investigation. Specifically, we predict the pre-ELM pedestal top electron pressure $p_{e,\text{ped}}$, density $n_{e,\text{ped}}$ and temperature $T_{e,\text{ped}}$, which are illustrated in figure 5. Here, *pre-ELM* refers to the pedestal top values just before the ELM (Edge Localized Mode) crash.

The JET pedestal database was created in prior work [4] by acquiring experimental High Resolution Thomson Scattering (HRTS) profiles [28] of electron temperature and density. These profiles were specifically acquired during quasi-steady states lasting at least 0.5 seconds, ensuring that only minor fluctuations due to ELMs were present. During these intervals, key tokamak and plasma parameters, such as β_N and the line-integrated density, were verified to remain constant. Additionally, to reflect conditions near the ELM stability limit, only the measured profiles in the phase representing 70-99% of the time interval before each ELM crash were included. Moreover, since pressure is a product of temperature and density, pressure profiles could also be acquired. More details about how curve-fitting techniques were used to obtain pedestal values from discrete measurements, as well as other aspects of the database, are thoroughly explained in [4]. In the following segment, we focus on the aspects that are specific to our work.

For our investigation, we only consider JET pulses performed with the ITER-like wall (ILW). Additionally, pulses involving kicks, impurity seeding, RMPs, and pellets have been excluded as these are techniques that can affect the pedestal. Furthermore, we focus exclusively on deuterium pulses, as the dataset is dominated by this ion species. The dataset is distributed across different divertor strike point configurations as follows: V/V (11%), V/C (9%), V/H (48%), C/V (2%), and C/C (30%). In these acronyms, 'V' refers to a vertical target, 'C' to a corner target, and 'H' to a horizontal target. For instance, 'V/C' indicates a configuration where the inner strike point is on the vertical target and the outer strike point is on the corner target. Another feature of the database is that it is dominated by type-I ELMs.

Table 1. The full set of parameters that are considered in this work. Here, 'Ped.' is short for pedestal. The triangularity parameter represents the average of the upper and lower plasma triangularity. The power parameter represents the sum of the NBI power, ICRH power, and ohmic heating, minus the shine through power.

Parameter	Min	Max	Unit
Outputs			
Ped. pressure $p_{e,\text{ped}}$	0.80	13.43	kPa
Ped. density $n_{e,\text{ped}}$	1.85	10.57	10^{19}m^{-3}
Ped. temperature $T_{e,\text{ped}}$	0.15	1.48	keV
Considered inputs			
Separatrix density $n_{e,\text{sep}}$	0.60	6.61	10^{19}m^{-3}
Plasma current I_P	0.97	3.96	MA
Toroidal field B	0.97	3.68	T
Minor radius a	0.87	0.96	m
Elongation κ	1.60	1.82	—
Triangularity δ	0.18	0.46	—
Total power P_{tot}	3.40	35.09	MW
Plasma volume V_{tot}	70.01	79.76	m^3
Safety factor q_{95}	2.66	4.35	—
Fuel rate of main ion Γ	0.07	7.28	10^{22} e s^{-1}
Effective ion charge Z_{eff}	1.00	3.50	–

In total, the data set in this work, which is a subset of the full JET pedestal dataset based on the selections described in this section, consists of 1043 entries from 852 different pulses ranging between JET pulse 81 768 to 98 004.

3.1. Input parameters

We consider the main engineering parameters listed in table 1 as potential input parameters for predicting $p_{e,\text{ped}}$, $n_{e,\text{ped}}$, and $T_{e,\text{ped}}$ respectively. Note that the three outputs are included in the table to show the range of these parameters, but they are not considered as potential input parameters in predicting one another. Note also that the separatrix density $n_{e,\text{sep}}$, which can be considered a plasma parameter rather than an engineering parameter, is included as a potential input. This is because, as shown in previous work [4], the separatrix density serves as

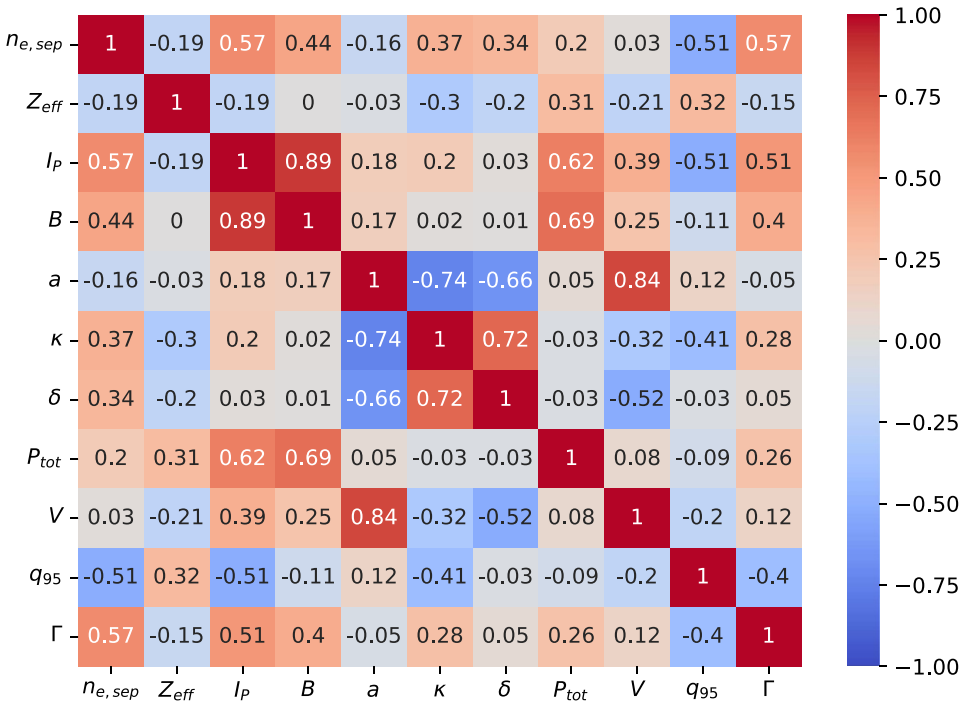


Figure 6. Pearson correlation matrix for the 11 input parameters considered in this work.

a more appropriate proxy for neutral pressure than the fueling parameter, whose effect can vary significantly depending on factors such as the divertor strike point configuration, and also the poloidal location of the fueling [29]. However, as also discussed in [30], there may be scenarios where relying on prior knowledge of $n_{e,\mathrm{sep}}$ for pedestal predictions is not preferable. Therefore, in this paper, we examine cases where $n_{e,\mathrm{sep}}$ is both included and excluded as a potential input parameter.

The inputs in table 1 are not fully independent, as illustrated in the Pearson correlation matrix in figure 6. The most obvious correlations are, for instance, the positive correlation between the magnetic field B and the plasma current I_P, the positive correlation between the minor radius a and the plasma volume V, and the negative correlation between a and the elongation κ. Some of these parameters are naturally correlated due to how they are defined, but also due to how the experiments need to be run to ensure stable plasmas. The correlations are however important to consider when analyzing the results presented in later sections. For instance, results attributed to I_P might potentially also implicitly be attributed to B due to the strong correlation.

4. Evaluation metric

As the main idea of this paper is to train different pedestal models to shed light on parameter relationships, we need a metric to evaluate how well the models fit the data, essentially to get an idea of how valid the results are. For this purpose, we employ the R^2 metric to quantify prediction accuracy. It is defined as

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \tag{3}$$

where y_i represents the true outputs, $\hat{y}_i$ represents the predictions, and $\bar{y}$ represents the mean value of the true outputs. In the best case, the predicted values exactly match the true values, resulting in $R^2 = 1$. Conversely, lower R^2 values indicate worse model performance. For instance, a poorly performing model that always simply predicts the mean value $\bar{y}$ corresponds to $R^2 = 0$.

The main reason for using the R^2 metric is its easily interpretable accuracy range of 0 to 1, as well as its use in previous studies related to fits of experimental pedestal data. However, since the dataset used in this work may not be identical from those in previous studies, a comparison of the exact R^2 values presented here and those presented in other work is not advisable. Instead, we encourage comparisons of the trends and the relative differences in R^2 that we present here.

4.1. Validation set evaluation

As high capacity models exhibit the risk of overfitting, we evaluate all neural network based models using a 5-fold cross validation method. This applies to regular neural networks trained for benchmarking, as well as the NeuralBranch models. The cross-validation approach allows us to assess the accuracy on the full dataset while ensuring that no individual model is evaluated on the same data it is trained on. That said, after the evaluation, we train the final NeuralBranch models

from scratch on the full dataset to ensure that no interesting data points are accidentally missed in the investigation of parameter relationships.

5. Selecting input parameters

Before using the NeuralBranch framework, we investigate which parameters in table 1 that actually are important for achieving high prediction accuracy when predicting the pedestal top values, such that we may only include those inputs. This is important for several reasons. First, a machine learning model has no incentive to learn meaningful relationships between the output and inputs that do not help minimize the loss function, in our case, inputs that do not improve prediction accuracy. Second, we reduce the risk of including parameters that are too strongly correlated such that they do not provide any unique relevant information. While this may not be a significant problem when the main goal is to achieve accurate predictions, it can complicate the analysis of parameter relationships, which is a key objective of this paper. Lastly, the NeuralBranch framework is easier to implement and analyze with fewer input parameters.

The method we use to identify the most important inputs for each output is the Sequential Forward Selection (SFS) approach [31]. This approach can be summarized as follows: we start with an empty input set, and then we iteratively add the best performing input until accuracy stops improving. Here, we use regular neural networks for the modeling to ensure that the results are not influenced by limited expressive capacity. Note however that new inputs are not added during the actual training process. Instead, a new neural network is trained each time a new input is added. Moreover, we use Sequential Backward Selection (SBS) to check if the results are the same as when using SFS. In SBS, which essentially is the opposite to SFS, one starts with all inputs and iteratively removes the input that leads to the lowest reduction in accuracy when removed. For all cases presented in this paper, the results from SFS are consistent with the result from SBS. We motivate the use of greedy approaches like SFS and SBS due to their efficiency compared to testing all possible input combinations. This reduces the number of evaluations from 2047 to just 65. Additionally, just as in the NeuralBranch methodology, we train each input parameter setup 3 times where the best scoring model is used for evaluation. As mentioned, we want to ensure that no input setups are overlooked due to rare cases where the trainable weights of the models converge to a local optimum, resulting in a lower score.

5.1. Input parameters for predicting $p_{e,\mathrm{ped}}$

In this work, we predict the three outputs separately, and we begin our analysis with the pedestal pressure $p_{e,\mathrm{ped}}$. To avoid potential confusion, all results related to $p_{e,\mathrm{ped}}$, including input parameter selection and NeuralBranch results, are presented before we address results for $n_{e,\mathrm{ped}}$ and $T_{e,\mathrm{ped}}$.

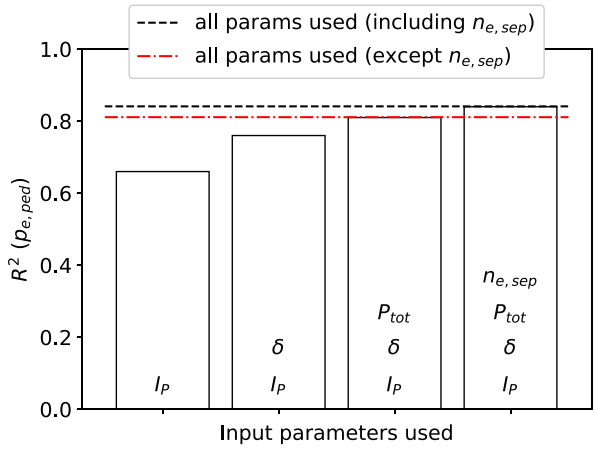


Figure 7. The result of the Sequential Forward Selection (SFS) approach when selecting inputs for predicting $p_{e,\mathrm{ped}}$. The input parameters in each bar indicate which parameters are used in each model. The lines are used as references for when all 11 input parameters in table 1 are included (all 10 when $n_{e,\mathrm{sep}}$ is excluded).

Figure 7 presents the results of the SFS method used to identify the key input parameters for predicting $p_{e,\mathrm{ped}}$. At each step, only the parameter that yields the highest prediction accuracy is shown. For example, when limited to one input, I_P provides the highest accuracy. As additional parameters are introduced incrementally, δ, followed by P_{tot} and $n_{e,\mathrm{sep}}$ provide the highest accuracy. Notably, using these four inputs together yields approximately the same accuracy as the full set of inputs listed in table 1 ($R^2 = 0.84$). When $n_{e,\mathrm{sep}}$ is excluded as a potential input, we observe that the three other parameters, I_P, δ and P_{tot}, yield the same accuracy as when all other inputs are included ($R^2 = 0.81$).

Based on these findings, we consider two input parameter sets for the remainder of the analysis related to $p_{e,\mathrm{ped}}$, one excluding $n_{e,\mathrm{sep}}$ {I_P, δ, P_{tot}}, and one including $n_{e,\mathrm{sep}}$ {I_P, δ, P_{tot}, $n_{e,\mathrm{sep}}$}. These inputs are largely consistent with the ones in the Cordey scaling [18] and other recent power-law fits predicting pedestal stored energy [4], which is directly proportional to pedestal pressure. The minor differences include the choice of shaping parameters and the inclusion of ion mass and tokamak major radius dependencies in the Cordey scaling, which are irrelevant in this work as we only include deuterium data from the JET tokamak.

The results presented in this section do not necessarily imply that other parameters are always unimportant for predicting $p_{e,\mathrm{ped}}$, as the result may be influenced by correlations between the input parameters and other characteristics of the specific data set used in this work.

5.2. Unimportance of fueling Γ on $p_{e,\mathrm{ped}}$

One example of a parameter that likely has impact on $p_{e,\mathrm{ped}}$ even though it does not appear in the SFS selection results is the fueling Γ. Specifically, previous parameter scans indicate that pedestal pressure is influenced by fueling [3, 7]. However, the unimportance of Γ that we observe is similarly reflected in

the power scaling for pedestal stored energy in [4], where it is partly explained by Γ being an imperfect proxy for neutral pressure compared to $n_{e,\mathrm{sep}}$. For instance, it is possible that the relationship between Γ and $p_{e,\mathrm{ped}}$ varies across the dataset, depending on factors beyond the other input parameters considered in this work. This would make Γ appear unimportant on a large scale data set while a variation of the pedestal would be identifiable in a Γ-scan when considering just a few pulses.

6. Pedestal pressure $p_{e,\mathrm{ped}}$: results

6.1. Power scaling for $p_{e,\mathrm{ped}}$

In addition to the NeuralBranch method, we first fit power scalings to predict $p_{e,\mathrm{ped}}$. Although this method and the key results presented here are not novel in the field of pedestal physics, as demonstrated in [4, 18], we include it to establish a baseline for comparison with the NeuralBranch method.

All coefficients in the power scalings in this paper are determined by performing linear regression on the data transformed into logarithmic space. However, after finding the optimal coefficients that minimize the sum of squared residuals, the R^2-value for each fit is evaluated on the original, untransformed data.

When considering the case where $n_{e,\mathrm{sep}}$ is not included as an input parameter, we obtain the following result

$$p_{e,\mathrm{ped}} = 1.25 I_P^{1.05} \delta^{0.46} P_{\mathrm{tot}}^{0.41} \tag{4}$$

which yields $R^2 = 0.78$ both when fitted and evaluated on the entire data set and when assessed using cross-validation. The fit shows a positive relationship between $p_{e,\mathrm{ped}}$ and all three input parameters, which is coherent with previous scalings for the pedestal stored energy [4, 18] and scans for the pedestal pressure [3, 7].

When considering the case where $n_{e,\mathrm{sep}}$ is included as an input parameter, we obtain the following result

$$p_{e,\mathrm{ped}} = 1.78 I_P^{1.34} \delta^{0.57} P_{\mathrm{tot}}^{0.33} n_{e,\mathrm{sep}}^{-0.21} \tag{5}$$

which yields $R^2 = 0.79$ both when fitted and evaluated on the entire data set and when assessed using cross-validation. Overall, we observe similar trends compared to the former power scaling, with the addition that $n_{e,\mathrm{sep}}$ is negatively correlated with $p_{e,\mathrm{ped}}$. This is also coherent with other scalings for the pedestal stored energy [4].

While both these power scalings demonstrate reasonable accuracy, there is a subtle yet noticeable difference when compared to the corresponding neural networks ($R^2 = 0.81$ for the neural network without $n_{e,\mathrm{sep}}$ and $R^2 = 0.84$ for the neural network with $n_{e,\mathrm{sep}}$) that were used in the input parameter selection test. This indicates that the neural networks have incorporated a more nuanced pattern in the data, which the power scalings are unable to fully capture due to their restrictive form, thus motivating the use of the NeuralBranch framework.

6.2. NeuralBranch for $p_{e,\mathrm{ped}}$

Here, we apply the NeuralBranch framework to predict $p_{e,\mathrm{ped}}$. We use the method as described in section 2 and present the final architecture for the two cases, one where $n_{e,\mathrm{sep}}$ is included as an input, and one where $n_{e,\mathrm{sep}}$ is not included.

Figure 8 shows the NeuralBranch model and associated visualizations when $n_{e,\mathrm{sep}}$ is not included as an input parameter. This model yields $R^2 = 0.81$ when evaluated using cross-validation, matching the accuracy of the corresponding neural network and thus showing a slight improvement over the power scaling ($R^2 = 0.78$). The visualizations show, much like the corresponding power scaling, that all three inputs P_{tot}, I_P, and δ generally are positively correlated with $p_{e,\mathrm{ped}}$. The NeuralBranch model has however also learned a more nuanced pattern, which can be summarized as:

- In figure 8(b), we see that the positive contribution from P_{tot} on $p_{e,\mathrm{ped}}$ diminishes as I_P increases (z is proportional to $p_{e,\mathrm{ped}}$). In more detail, when I_P is high, we see that higher P_{tot} is required to increase $p_{e,\mathrm{ped}}$ further. This is an example of an attenuating interaction between I_P and P_{tot} in relation to $p_{e,\mathrm{ped}}$. This particular phenomenon, where two inputs are individually positively correlated with the output, but exhibit an attenuating interaction, cannot be captured with a power scaling.

We now turn to the other case, where $n_{e,\mathrm{sep}}$ is included as an input parameter when predicting $p_{e,\mathrm{ped}}$. Figure 9 shows the final NeuralBranch architecture and the associated visualizations for this case. This model yields $R^2 = 0.84$ when evaluated using cross-validation, which like the previous case, matches the neural network and exceeds the accuracy of the corresponding power scaling ($R^2 = 0.79$). By analyzing the visualizations, we see that this model effectively has learned the same patterns as the previous NeuralBranch model for $p_{e,\mathrm{ped}}$, with the addition that it here includes the negative contribution from $n_{e,\mathrm{sep}}$.

6.3. Discussion of NeuralBranch results for $p_{e,\mathrm{ped}}$

The attenuating interaction between I_P and P_{tot} in relation to $p_{e,\mathrm{ped}}$ has been hinted at in previous work. For instance, in [4], it is observed that the relationship between P_{tot} and $p_{e,\mathrm{ped}}$ is weaker at higher I_P. The proposed potential explanation in [4] is that higher I_P experiments typically involve increased fueling rates, which can lead to a reduction in pedestal pressure. However, when $n_{e,\mathrm{sep}}$ is included, which incorporates the impact of fueling, we still observe the interaction between I_P and P_{tot}. This implies that the attenuating interaction between I_P and P_{tot} is not necessarily a consequence of higher I_P being accompanied with higher fueling (higher $n_{e,\mathrm{sep}}$), since the inclusion of $n_{e,\mathrm{sep}}$ should be able to correct for such embedded biases, at least to some extent. Therefore, we recommend conducting further targeted investigations, potentially informed by the findings presented here, to examine the

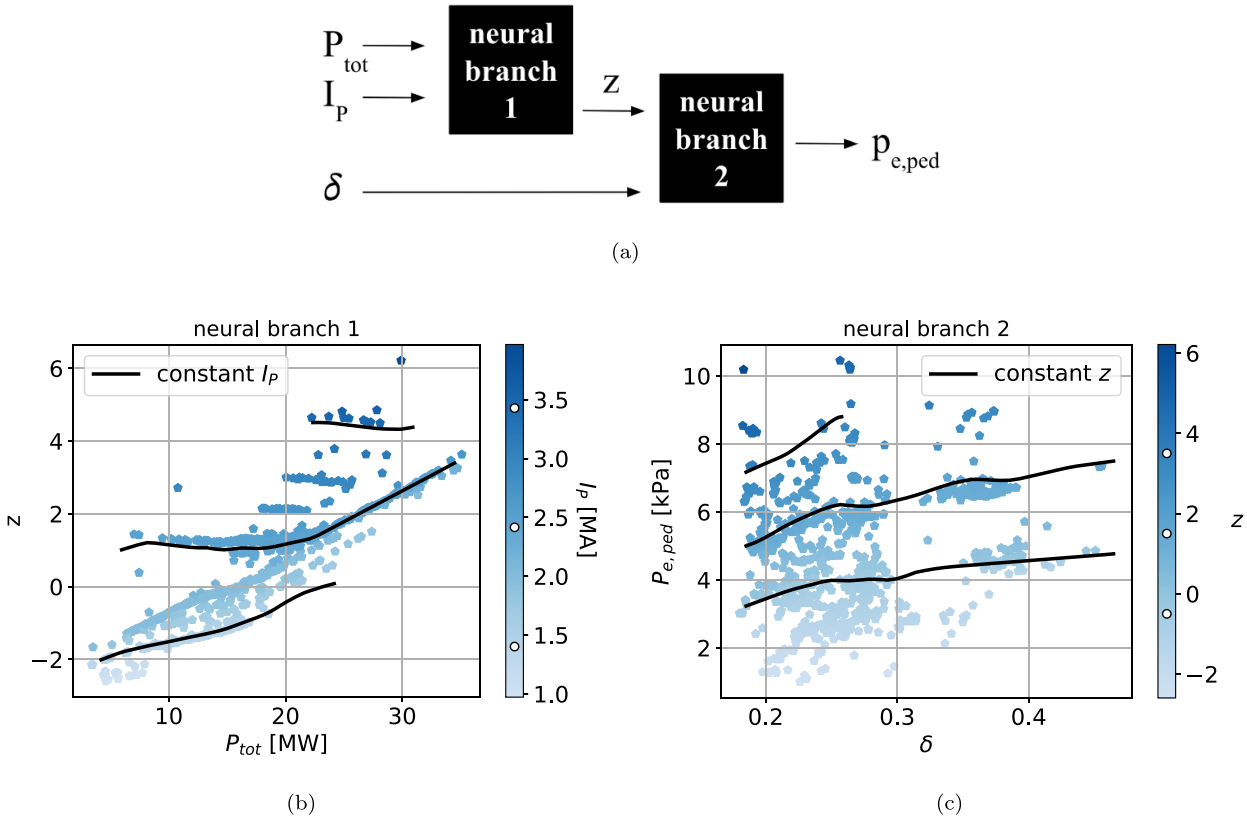


Figure 8. The final architecture (*a*) and visualizations (*b*), (*c*) of the NeuralBranch model that predicts $p_{e,\mathrm{ped}}$ ($n_{e,\mathrm{sep}}$ not included as a potential input). The scatter points in (*b*), (*c*) are obtained by parsing the full data set thought the model. Here, the output prediction of each branch is plotted on the *y*-axis versus one of its inputs on the *x*-axis, with the other input dictating the brightness of the points. The black lines in (*b*), (*c*) represent model predictions where we scan over the input on the *x*-axis, with the other input, the one dictating the brightness, being constant. The white dots on the colorbars show at which constant values the predictive scans are performed. The visualizations (*b*), (*c*) enable interpretability regarding how the model predicts $p_{e,\mathrm{ped}}$ from the three inputs. For instance, (*a*) shows how P_{tot} and I_P influence the intermediate parameter z, and (*b*) shows how z and δ in turn influence the predictions of $p_{e,\mathrm{ped}}$.

interaction between I_P and P_{tot}. This is especially relevant in the context of the Cordey scaling and existing power scalings that predict pedestal stored energy, which currently suggest an amplifying interaction between I_P and P_{tot}, which according to the NeuralBranch model, is not correct.

A detailed theoretical examination of the relationships learned by the models is beyond the scope of this work. Nonetheless, it is worth noting that one proposed mechanism by which increased P_{tot} might enhance pedestal stability involves its influence on the Shafranov shift [32], which subsequently affects the magnetic shear and contributes to the stabilization of ballooning modes [33]. This mechanism exhibits a similarity to the stabilization of ballooning modes by increased plasma current, which also modifies the magnetic shear [34]. A plausible hypothesis that would explain the attenuating interaction seen in figure 8(*b*) is that when either I_P or P_{tot} already stabilizes specific ballooning modes, the influence of the other parameter diminishes, necessitating a more substantial increase in the second parameter to achieve further stabilization of the pedestal. However, this remains speculative and requires further investigation to validate.

7. Pedestal density $n_{e,\mathrm{ped}}$: results

In this section, we present results related to the pedestal density $n_{e,\mathrm{ped}}$, including selection of input parameters, power scaling fits, and the NeuralBranch method.

7.1. Input parameters for $n_{e,\mathrm{ped}}$

Figure 10 presents the results of the SFS algorithm used to identify the key input parameters for predicting $n_{e,\mathrm{ped}}$, both when $n_{e,\mathrm{sep}}$ is included and excluded as an input parameter. We observe that when $n_{e,\mathrm{sep}}$ is included, it is the first parameter in the sequence. By adding I_P and δ as inputs, a similar level of accuracy is achieved ($R^2 = 0.88$) as when all potential input parameters in table 1 are used ($R^2 = 0.89$). Therefore, for the case where $n_{e,\mathrm{sep}}$ is included, we focus on $\{n_{e,\mathrm{sep}}, I_P, \delta\}$ as inputs. The significance of $n_{e,\mathrm{sep}}$ in predicting $n_{e,\mathrm{ped}}$ has also been highlighted in recent studies. For example, a recent model combining neutral penetration with pedestal transport to predict pedestal density supports this finding [30].

In the case where $n_{e,\mathrm{sep}}$ is excluded, we observe that I_P, followed by δ, Γ, and P_{tot} achieve approximately the

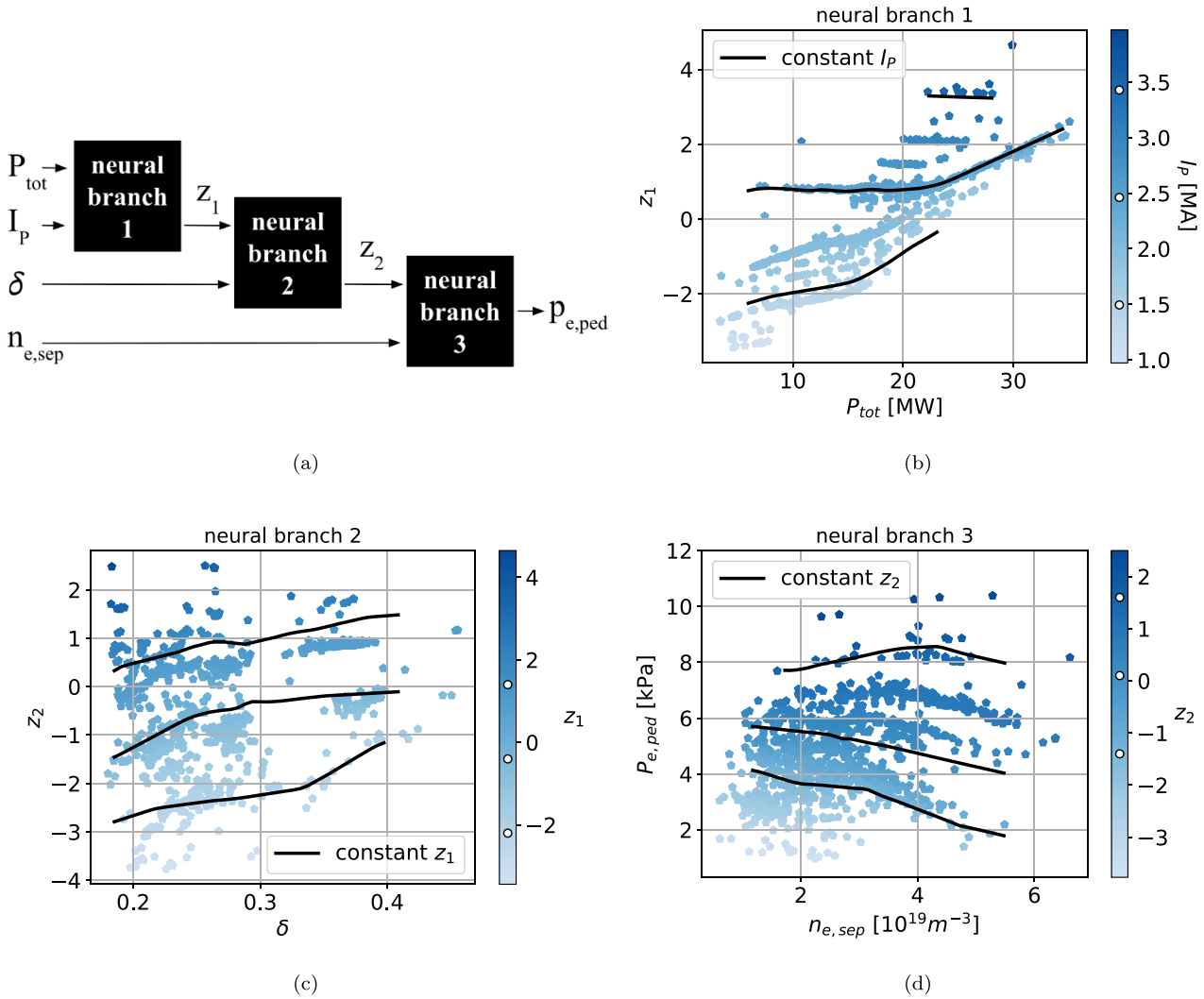


Figure 9. The final architecture (*a*) and visualizations (*b*), (*c*), (*d*) of the NeuralBranch model that predicts $p_{e,\mathrm{ped}}$ ($n_{e,\mathrm{sep}}$ included as input). See the caption of figure 8, or the demonstration example in section 2, for instructions regarding how to interpret the visualizations.

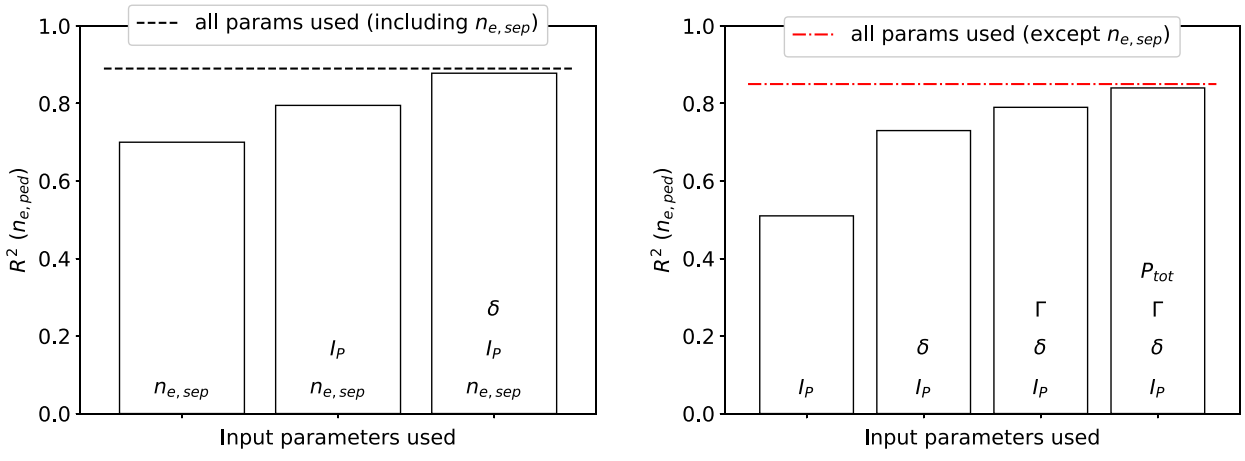


Figure 10. The result of the SFS input parameter selection test when predicting $n_{e,\mathrm{ped}}$, both when $n_{e,\mathrm{sep}}$ is included (*a*) and excluded (*b*) as a potential input parameter. The lines are used as references for when all 11 input parameters in table 1 are included (all 10 when $n_{e,\mathrm{sep}}$ is excluded).

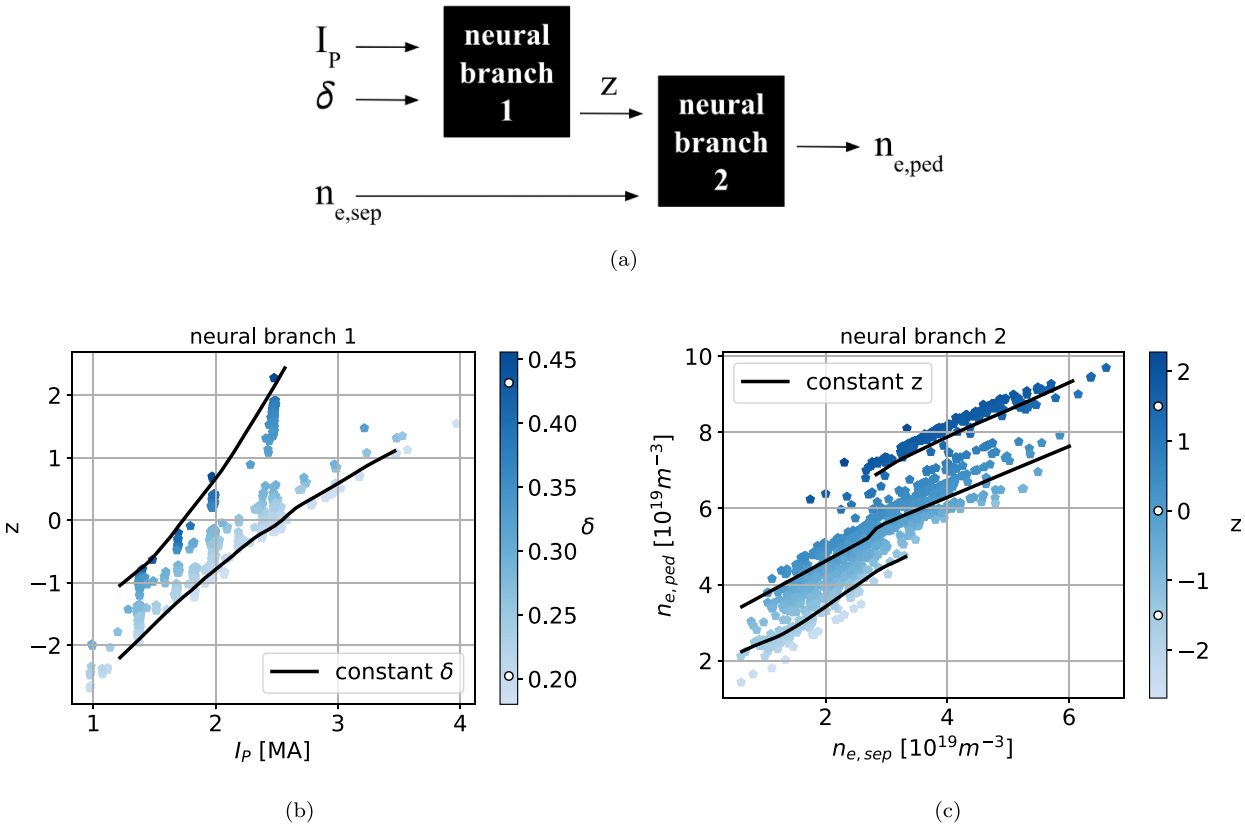


Figure 11. The final architecture (*a*) and visualizations (*b*), (*c*) of the NeuralBranch model that predicts $n_{e,\text{ped}}$. This is for the case where $n_{e,\text{sep}}$ is included as an input. See the caption of figure 8, or the demonstration example in section 2, for instructions regarding how to interpret the visualizations.

same accuracy ($R^2 = 0.84$) as when all input parameters from table 1, except $n_{e,\text{sep}}$, are used. In other words, when $n_{e,\text{sep}}$ is not considered, Γ and P_{tot} play a more significant role in producing high accuracy. Therefore, the second input set we consider for predicting $n_{e,\text{ped}}$ is: $\{I_P, \delta, \Gamma, P_{\text{tot}}\}$. This is the same set of input parameters that is used in a recent power scaling predicting the pedestal density [4], with the exception that the ion mass is excluded in our model since we only consider deuterium data.

7.2. Power scaling for $n_{e,\text{ped}}$

When $n_{e,\text{sep}}$ is included as an input parameter, we obtain the following power scaling

$$n_{e,\text{ped}} = 3.71 n_{e,\text{sep}}^{0.46} I_P^{0.58} \delta^{0.42} \tag{6}$$

which achieves $R^2 = 0.85$, both when evaluated on the entire data set and when evaluated using cross-validation. The fit shows a positive correlation between all three input parameters and the pedestal density, which is coherent with previous findings [4].

When considering the other case, where $n_{e,\text{sep}}$ is excluded as an input parameter, we obtain

$$n_{e,\text{ped}} = 10.59 I_P^{1.15} \delta^{0.68} P_{\text{tot}}^{-0.26} \Gamma^{0.11} \tag{7}$$

which achieves $R^2 = 0.77$ both when evaluated on the entire data set and when evaluated using cross-validation. Like the previous power scaling, the positive and negative exponents are coherent with findings in previous work [4].

Similar to the pedestal pressure case, the power scalings for $n_{e,\text{ped}}$ demonstrate reasonable accuracy. However, there is again a subtle yet noticeable difference compared to the corresponding neural networks used in the input parameter selection test ($R^2 = 0.88$ for the neural network with $n_{e,\text{sep}}$ and $R^2 = 0.85$ for the neural network without $n_{e,\text{sep}}$). Therefore, we use the NeuralBranch method in the next section to investigate what is causing this difference.

7.3. NeuralBranch for $n_{e,\text{ped}}$

Figure 11 shows the final NeuralBranch architecture and associated visualizations when predicting $n_{e,\text{ped}}$ (here, $n_{e,\text{sep}}$ is included as an input parameter). This model yields $R^2 = 0.88$ when evaluated using cross-validation, matching the accuracy of the corresponding neural network and thus showing a slight improvement over the power scaling ($R^2 = 0.85$). Much like the power scaling, the NeuralBranch model suggests a positive relationship between $n_{e,\text{ped}}$ and the three inputs I_P, $n_{e,\text{sep}}$, and δ. Additionally, the NeuralBranch model indicates two more nuances patterns:

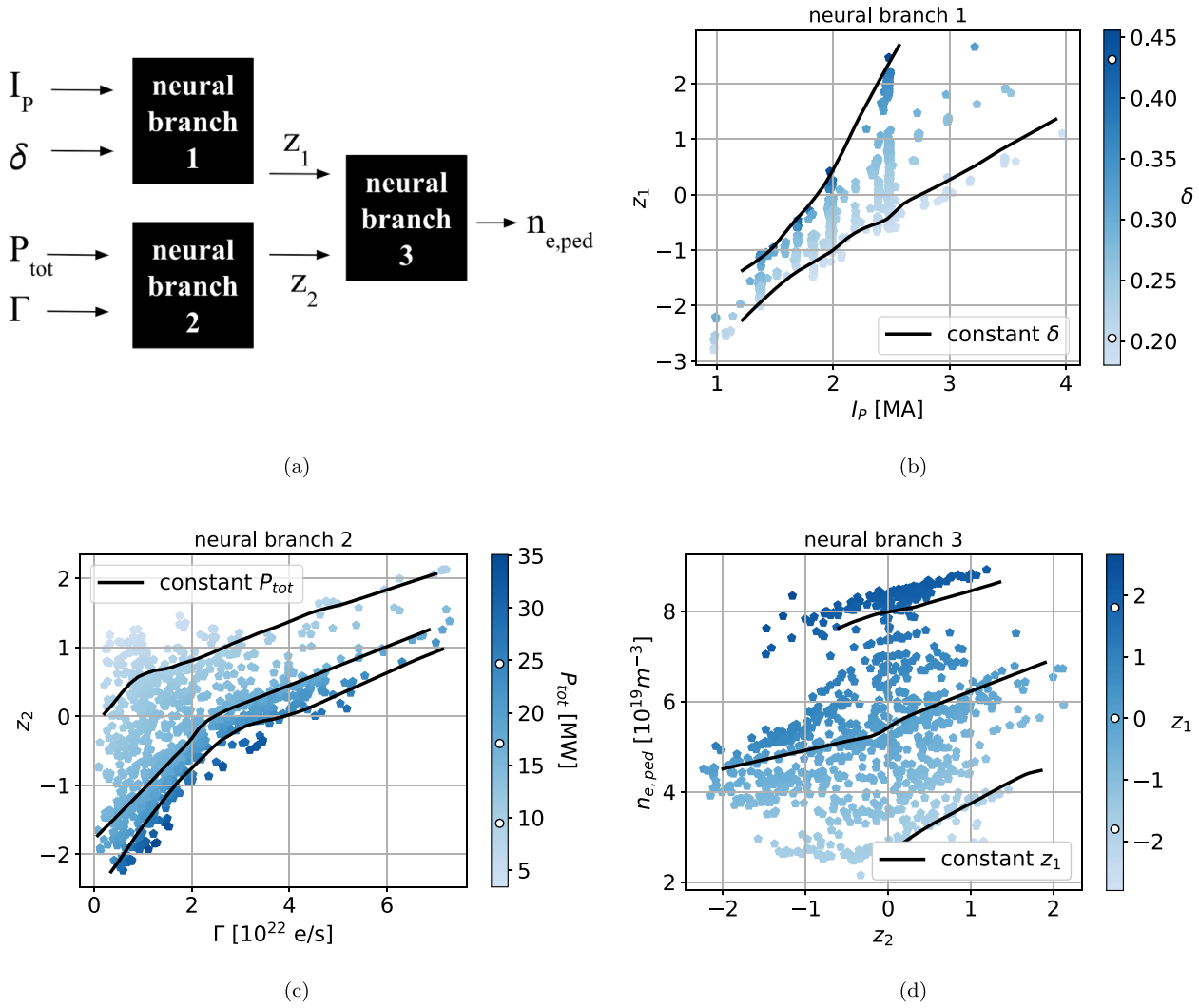


Figure 12. The final architecture (a) and visualizations (b)–(d) of the NeuralBranch model that predicts $n_{e,\text{ped}}$ ($n_{e,\text{sep}}$ excluded as input). See the caption of figure 8, or the demonstration example in section 2, for instructions regarding how to interpret the visualizations.

- In figure 11(b), we observe an amplifying interaction between I_P and δ. This means that higher δ amplifies the positive relationship between I_P and $n_{e,\text{ped}}$, and vice versa, that higher I_P amplifies the positive relationship between δ and $n_{e,\text{ped}}$. Note that we arrive at this conclusion as the intermediate parameter z is approximately proportional to $n_{e,\text{ped}}$ in this case.
- In figure 11(c), we see that the linear slope between $n_{e,\text{sep}}$ and $n_{e,\text{ped}}$ is overall unaffected by z, which represents the contribution from I_P and δ. This may explain why the NeuralBranch model achieves slightly higher accuracy than the power scaling, as the latter assumes an amplifying interaction between $n_{e,\text{sep}}$ and the other two inputs, which according to the NeuralBranch model, is incorrect.

We now turn to the other case, where $n_{e,\text{sep}}$ is excluded as an input parameter when predicting $n_{e,\text{ped}}$. Figure 12 shows the final NeuralBranch architecture and the associated visualizations for this case. This model yields $R^2 = 0.84$ when evaluated using cross-validation, which like the previous case, matches the neural network and exceeds the accuracy of the corresponding power scaling ($R^2 = 0.77$). The NeuralBranch model generally agrees with the positive and negative coefficients in the corresponding power scaling, but there are also nuances found by the NeuralBranch model worth mentioning:

- In figure 12(b), we observe the same amplifying interaction between I_P and δ that we also observed in the previous NeuralBranch model for $n_{e,\text{ped}}$.
- In figure 12(c), we observe that the negative contribution from P_{tot} on $n_{e,\text{ped}}$ saturates at high P_{tot}. We also see that the relationship between Γ and $n_{e,\text{ped}}$ is stronger at lower Γ. We arrive at these conclusions as the intermediate parameter z_2 is proportional to $n_{e,\text{ped}}$.
- The NeuralBranch model suggests that there is no interaction between the I_P/δ-branch and the P_{tot}/Γ-branch. This might contribute to the power scaling being less accurate, since the power scaling function essentially consists of one interaction term that involves all inputs.

7.4. Discussion of NeuralBranch results for $n_{e,\mathrm{ped}}$

As previously mentioned, this study does not aim to provide a comprehensive theoretical explanation of the results. However, we note that the interaction between I_P and δ may arise from how these parameters interdependently affect magnetic shear and other key properties that influence pedestal stability, as further detailed in [35]. Furthermore, this type of amplifying interaction, where the two parameters also are individually positively correlated with the output, is one of the few interaction patterns that a power scaling can capture. This explains why the power scaling for $n_{e,\mathrm{ped}}$ provides a high accuracy for the case when $n_{e,\mathrm{sep}}$ is included, seemingly by a fortunate coincidence regarding the interaction type.

8. Pedestal temperature $T_{e,\mathrm{ped}}$: results

8.1. Input parameters for $T_{e,\mathrm{ped}}$

Figure 13 presents the results of the SFS algorithm used to identify the key input parameters for predicting $T_{e,\mathrm{ped}}$, both when $n_{e,\mathrm{sep}}$ is included and excluded as an input parameter. We observe that when $n_{e,\mathrm{sep}}$ is included, the three inputs P_{tot}, $n_{e,\mathrm{sep}}$ and I_P are sufficient to achieve the same accuracy as when all input parameters in table 1 are used ($R^2 = 0.82$).

For the case where $n_{e,\mathrm{sep}}$ is not considered as an input, we observe that P_{tot}, Γ, I_P and δ are necessary to achieve the same level of accuracy as when all other input parameters are used. We do however notice a significant drop in R^2 when $n_{e,\mathrm{sep}}$ is excluded, going from 0.82 to 0.68. Specifically, although δ and Γ somewhat compensates for the absence of $n_{e,\mathrm{sep}}$, they are not sufficient to describe the loss of relevant information embedded in $n_{e,\mathrm{sep}}$. Nevertheless, based on these results, we consider two input parameter sets when predicting $T_{e,\mathrm{ped}}$, namely the first set: $\{P_{\mathrm{tot}}, n_{e,\mathrm{sep}}, I_P\}$, and the second set: $\{P_{\mathrm{tot}}, \Gamma, I_P, \delta\}$. These are overall the same input parameters that have been used in previous power scalings for the pedestal temperature [4].

8.2. Power scalings for $T_{e,\mathrm{ped}}$

Fitting a power scaling for the pedestal temperature $T_{e,\mathrm{ped}}$ with $n_{e,\mathrm{sep}}$ included as an input yields

$$T_{e,\mathrm{ped}} = 0.18 n_{e,\mathrm{sep}}^{-0.60} P_{\mathrm{tot}}^{0.48} I_P^{0.54} \tag{8}$$

which achieves $R^2 = 0.74$. The result indicates that I_P and P_{tot} are positively correlated with $T_{e,\mathrm{ped}}$, and that $n_{e,\mathrm{sep}}$ is negatively correlated with $T_{e,\mathrm{ped}}$, which is consistent with previous power scalings [4]. The accuracy is however notably lower compared to the corresponding neural network ($R^2 = 0.82$), which again motivates the use of the NeuralBranch method.

When $n_{e,\mathrm{sep}}$ is excluded as an input parameter, we obtain

$$T_{e,\mathrm{ped}} = 0.07 P_{\mathrm{tot}}^{0.68} \Gamma^{-0.18} I_P^{0.03} \delta^{-0.22} \tag{9}$$

which yields $R^2 = 0.56$, a notably lower accuracy compared to the corresponding neural network from the input selection test ($R^2 = 0.68$). Nevertheless, we observe that the two new parameters, Γ and δ, contribute negatively to $T_{e,\mathrm{ped}}$, which is also consistent with previous studies [4]. We additionally observe that the dependence on I_P effectively has disappeared compared to the previous power scaling. A possible explanation mentioned in previous work is that there is not a sufficient spread in the I_P data [4]. However, our results contradict this, as all our high-capacity models need to include I_P to achieve the highest possible accuracy, meaning that its distribution is wide enough to be informative to the models. A more likely explanation for this particular case is that I_P contributes to $T_{e,\mathrm{ped}}$ in a more complicated way than what can be captured with the power scaling.

8.3. NeuralBranch for $T_{e,\mathrm{ped}}$

Figure 14 shows the NeuralBranch architecture and associated visualizations when predicting $T_{e,\mathrm{ped}}$ (here $n_{e,\mathrm{sep}}$ is included as an input parameter). This model yields $R^2 = 0.82$ when evaluated using cross-validation, matching the accuracy of the corresponding neural network and thus showing an improvement over the power scaling ($R^2 = 0.74$). The NeuralBranch model generally agrees with the coefficients of the power scaling, but it has also learned a significant interaction pattern:

- In figure 14(c), we observe that the impact that I_P has on $T_{e,\mathrm{ped}}$ is greatly impacted by P_{tot} and $n_{e,\mathrm{sep}}$. Specifically, when P_{tot} is high and $n_{e,\mathrm{sep}}$ is low (resulting in high z), the impact of I_P on $T_{e,\mathrm{ped}}$ diminishes. Similarly, the impact that P_{tot} and $n_{e,\mathrm{sep}}$ has on $T_{e,\mathrm{ped}}$ weakens at high I_P. This is effectively the same attenuating interaction that we observed when predicting $p_{e,\mathrm{ped}}$, the main difference being that $n_{e,\mathrm{sep}}$ is included in the interaction here, potentially as $n_{e,\mathrm{sep}}$ is more important when predicting $T_{e,\mathrm{ped}}$.

We now continue to figure 15, which shows the NeuralBranch model that predicts $T_{e,\mathrm{ped}}$ when $n_{e,\mathrm{sep}}$ is excluded as an input. This model yields $R^2 = 0.66$ when evaluated using cross-validation, which almost matches the corresponding neural network ($R^2 = 0.68$) and exceeds the accuracy of the corresponding power scaling ($R^2 = 0.56$). Overall, the NeuralBranch model agrees with the coefficients in the power scaling (except for the general non-dependence of I_P in the power scaling). Moreover, this NeuralBranch model has learned the same attenuating interaction pattern that was seen in the previous NeuralBranch model for $T_{e,\mathrm{ped}}$. The main difference is that Γ effectively replaces $n_{e,\mathrm{sep}}$ in this case.

8.4. Discussion of NeuralBranch results for $T_{e,\mathrm{ped}}$

To avoid repetition, we refer to the pedestal pressure section for the discussion related to the attenuating interaction pattern involving I_P and P_{tot}. However, we note that previous findings

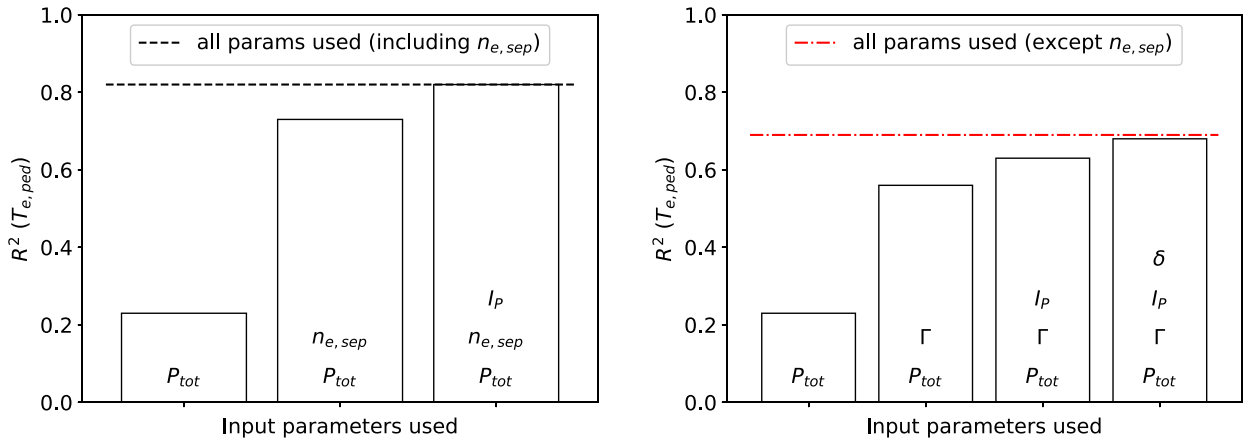


Figure 13. The result of the SFS input parameter selection test when predicting $T_{e,\mathrm{ped}}$, both when $n_{e,\mathrm{sep}}$ is included (*a*) and excluded (*b*) as a potential input parameter. The lines are used as references for when all 11 input parameters in table 1 are included (all 10 when $n_{e,\mathrm{sep}}$ is excluded).

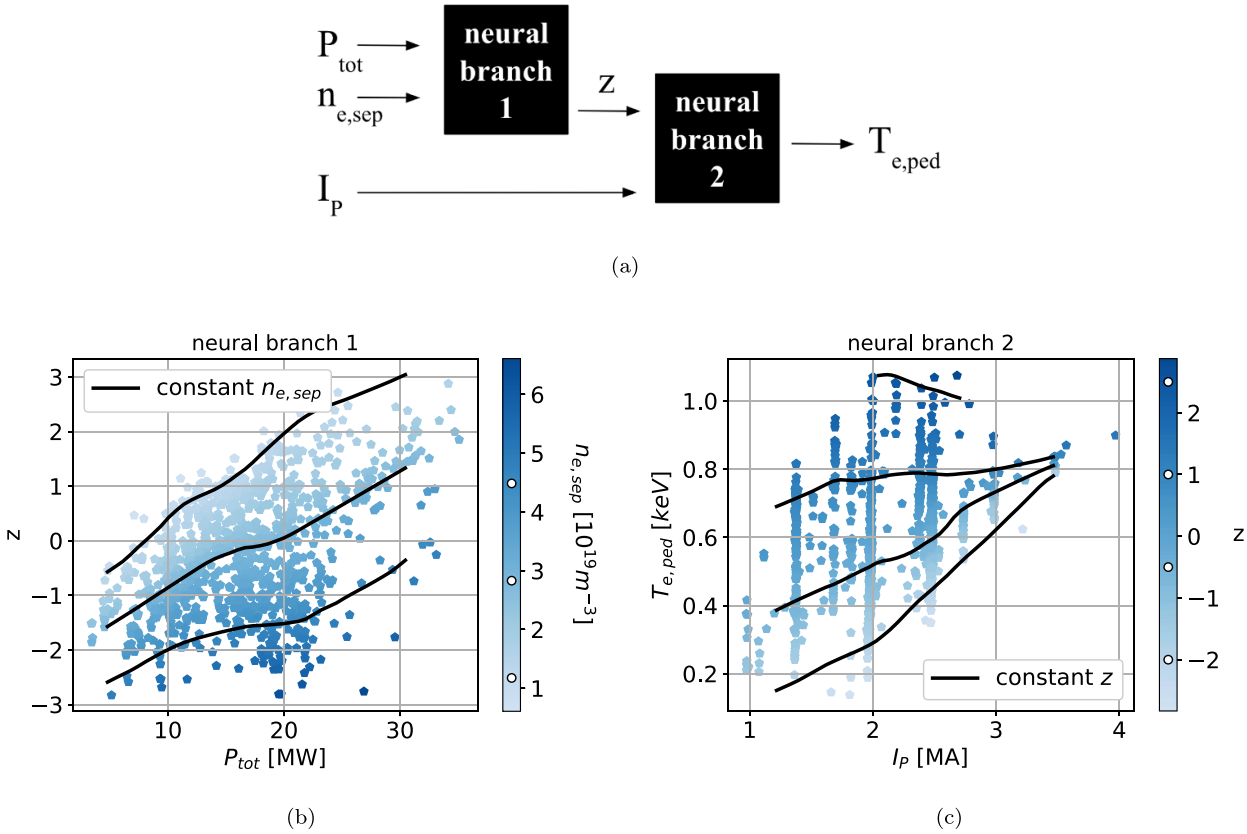


Figure 14. The architecture (*a*) and visualizations (*b*), (*c*) of the NeuralBranch model that predicts $T_{e,\mathrm{ped}}$ from $n_{e,\mathrm{sep}}$, I_P, and P_{tot}. See the caption of figure 8, or the demonstration example in section 2, for instructions regarding how to interpret the visualizations.

specific to $T_{e,\mathrm{ped}}$ suggest that the maximum $T_{e,\mathrm{ped}}$ is independent with respect to I_P, and that the relationship between $T_{e,\mathrm{ped}}$ and P_{tot} is weaker at higher I_P [4]. This is coherent with the attenuating interaction in our NeuralBranch models.

We also note that the main interaction in the NeuralBranch models for $T_{e,\mathrm{ped}}$ involves three parameters, first $n_{e,\mathrm{sep}}$, I_P and P_{tot}, and then Γ replaces $n_{e,\mathrm{sep}}$. It is therefore not certain that the interaction would have been as easily identified with other interpretable models that are best suited for pairwise interactions, such as NAMs.

Additionally, the NeuralBranch models support the hypothesis that power scalings are too simple to capture how I_P

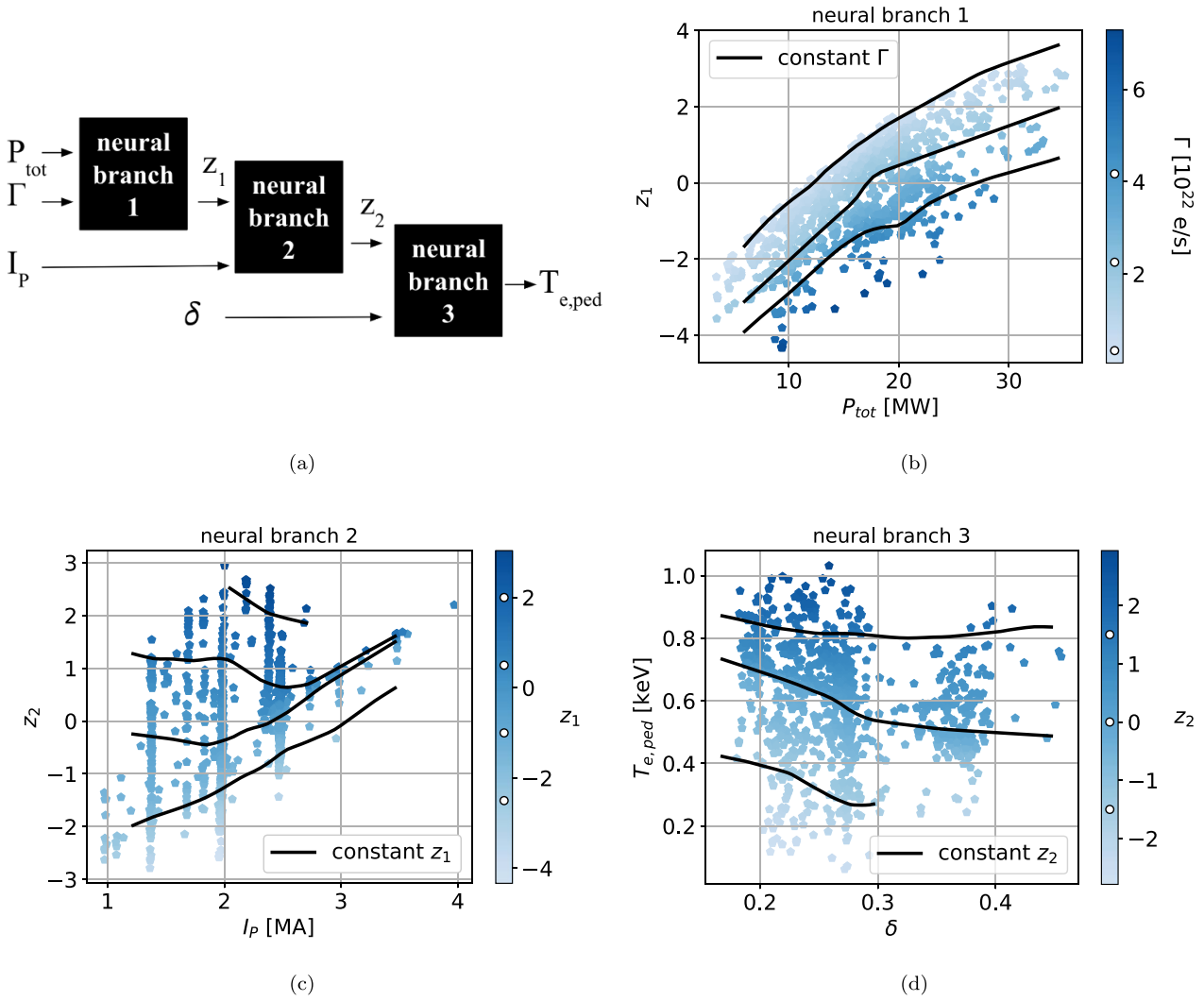


Figure 15. The final architecture (*a*) and visualizations (*b*)–(*d*) of the NeuralBranch model that predicts $T_{e,\mathrm{ped}}$ ($n_{e,\mathrm{sep}}$ excluded as input). See the caption of figure 8, or the demonstration example in section 2, for instructions regarding how to interpret the visualizations.

contributes to $T_{e,\mathrm{ped}}$, simply because the NeuralBranch models reveal a pattern that power scalings indeed cannot capture.

9. Summary of prediction accuracy of models

Table 2 presents a summary, mainly focusing on the accuracy of the various models trained in this study. As noted previously, the power scalings demonstrate relatively high accuracy, and higher-capacity models, such as the NeuralBranch models and neural networks, yield an improvement. However, the relatively high accuracy of the power scalings does not necessarily imply that power scalings are adequate for capturing all pedestal-related dependencies, as R^2 values are strongly influenced by the distribution of the data. As an example, the NeuralBranch models have indicated interaction patterns mostly related to I_P, and while its distribution is wide enough to be informative to the models, it is noteworthy that $\approx 70\%$ of the data is in the range 1.9-2.5 MA. Hence, it is plausible that the difference in R^2 between the power scalings and the high-capacity models would have been larger if the data and exploration space was even more spread out across different I_P values.

Nevertheless, the other clear trend in table 2 is that the NeuralBranch models match the accuracy of black-box neural networks, suggesting that interpretability can be achieved without compromising prediction accuracy for these cases.

In addition to the R^2 values, figure 16 shows the predicted values versus the database values for the different NeuralBranch models, which provides a more comprehensive picture of the accuracy. It is noteworthy that although the models demonstrate reasonable accuracy, there are still instances of significant error. This indicates that, as expected, the models capture general trends within the database rather than providing a complete representation of pedestal physics.

Table 2. Summary of models in this work. Here, NN is short for neural network, and the R^2-values in the parentheses represent those where $n_{e,\mathrm{sep}}$ is not included as an input.

Model	R^2 ($p_{e,\mathrm{ped}}$)	R^2($n_{e,\mathrm{ped}}$)	R^2($T_{e,\mathrm{ped}}$)	Key patterns learned	Notes
Power scaling	0.79 (0.78)	0.85 (0.77)	0.74 (0.56)	simple positive and negative relationships	limited expressive capacity
Black-box NN	0.84 (0.81)	0.88 (0.84)	0.82 (0.68)	unclear due to opaqueness	no interpretability
NeuralBranch	0.84 (0.81)	0.88 (0.84)	0.82 (0.66)	P_{tot}/I_P-attenuation, I_P/δ-amplification	interpretable, matches NN accuracy

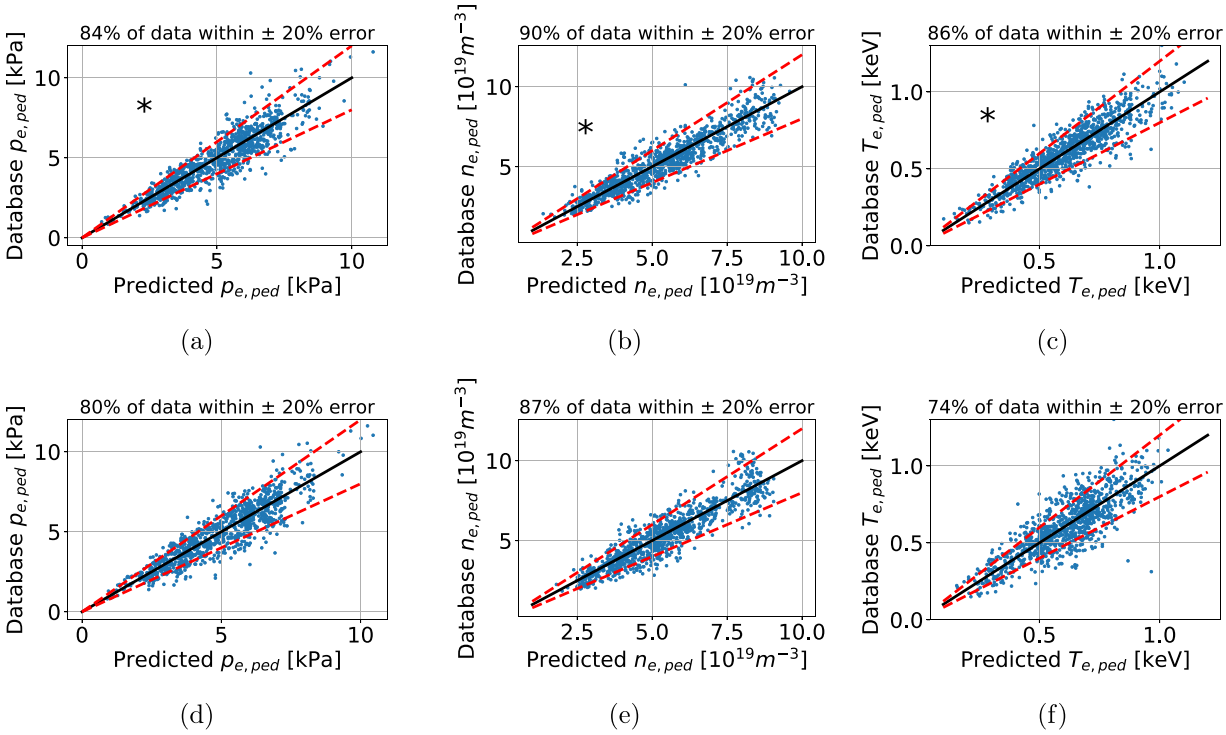


Figure 16. Predicted versus database values for the six NeuralBranch models in this work. The top row (*a*)–(*c*) shows the result for the models that include $n_{e,\mathrm{sep}}$ as an input, which is also indicated by the *, and the bottom row (*d*)–(*f*) shows the result for the cases where $n_{e,\mathrm{sep}}$ is excluded as an input. The black solid lines represent perfect prediction lines, and the red dashed lines represent $\pm$ 20% error margins.

10. Conclusions

In this work, we have used an interpretable machine learning method to investigate relationships between the pedestal and key tokamak parameters in the JET pedestal database. The main goal was to uncover general yet intricate relationships that prior power scalings [4, 18] have not been able to capture. A secondary goal was to develop a transparent alternative that maintains high predictive accuracy for future integrated modeling applications, addressing the opacity of previous black-box, machine learning-based models [19, 20]. For these purposes, we introduced a new interpretable framework, called NeuralBranch. Consequently, the third goal of this work was to outline the methodology associated with the framework.

In terms of the primary and secondary goal, we obtained NeuralBranch models that outperformed power scalings and matched the predictive accuracy of black-box neural networks while also being interpretable. Due to the interpretability, several intricate relationships related to the pedestal were uncovered. Specifically, while both the plasma current I_P and input power P_{tot} are individually positively correlated with the pedestal pressure $p_{e,\mathrm{ped}}$ and temperature $T_{e,\mathrm{ped}}$, the results suggest that I_P and P_{tot} exhibit an attenuating interaction. Investigating this interaction more thoroughly might be important to prevent overestimating pedestal stored energy in future high-power, high-current machines like ITER. Current power scalings, such as the Cordey Scaling [18], assume an amplifying interaction between I_P and P_{tot}, which could result in prediction errors without a deeper understanding of this effect. Additionally, we observed an amplifying interaction between the plasma current I_P and the triangularity δ when predicting the pedestal density $n_{e,\mathrm{ped}}$. While some of these interaction patterns have been hinted at in previous work [4], the NeuralBranch models in this work offer a novel global overview across the extensive JET pedestal database.

As we have presented results related to specific tokamak parameters, we emphasize that these findings may be influenced by correlations between the input parameters considered within the JET pedestal database. For example, the strong positive correlation between the toroidal magnetic field B and the

plasma current I_P raises the possibility that results attributed to I_P could, in part, be related to B, even though B was not an explicit input parameter in any model used.

Furthermore, we emphasize that the results are based on the analysis of learned model patterns, which, while more accurate than previous power scalings, are not without imperfections. Thus, our results should be interpreted as general trends, which we differentiate from a complete description of pedestal dependencies.

Future work may include theoretical investigations and targeted empirical investigations related to the interaction patterns that are indicated in this work. Specifically, since most of the patterns identified here involve interactions between I_P and other parameters, further analysis could explore how pedestal dependencies on various parameters evolve as I_P changes. Additionally, if large-scale pedestal data sets from other tokamaks become available, similar pedestal related analyses using NeuralBranch, or other interpretable models, could be conducted.

Finally, we recommend the use of interpretable pedestal models, such as those introduced in this study, instead of black-box machine learning models for integrated modeling applications. As demonstrated, prediction transparency, with its associated benefits, can be achieved without compromising prediction accuracy.

Acknowledgments

The authors would like to thank Lorenzo Frassinetti for valuable input prior to this work and for providing the pedestal dataset.

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200- EUROfusion). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

The work has been funded by the Swedish Research Council under the diary No. 2020-05465 and the EUROfusion Enabling Research Project ENR-MOD.01.FZJ 'Development of machine learning methods and integration of surrogate model predictor schemes for plasma-exhaust and PWI in fusion'.

Appendix A. Details on NeuralBranch methodology

A.1. Assigning inputs to neural branches

As mentioned in section 2, we assign inputs to the different neural branches in a NeuralBranch model by testing which input splits that lead to the highest accuracy. For cases with more than two input parameters, this search process occurs in subsequent steps, as illustrated in figure 17.

A.2. When inputs are not separable

It is not always the case that one can easily split the inputs such that they belong to only one neural branch. For instance, consider a new toy data set generated with the equation

$$y = (a+b)\sin(c+b)\,. \tag{10}$$

In this case, b affects both the amplitude and the argument of the sine-wave, and is therefore not easily separable. To achieve accurate predictions with NeuralBranch, b must be parsed through two branches, as illustrated in figure 18.

In this paper, we have seen that the inputs were separable for all scenarios. However, if in another application no split yield an accuracy sufficiently close to the benchmark (within a tolerance given by the user), one may first attempt allowing each input to be parsed to both branches, but only one at a time. If yet none of these setups yield a sufficient accuracy, it is possible to allow up to two input parameters to be parsed through both branches, and so on. Data sets with highly inseparable inputs are more difficult to interpret, and may limit the applicability of a NeuralBranch framework for such scenarios. This is however not only a challenge when considering NeuralBranch, but when considering fitting models for interpretation purposes in general.

Appendix B. Hyperparameters

The following hyperparameters were used for the regular neural networks and the NeuralBranch models in this work:

- Activation function in the hidden nodes: ReLU
- Activation function in the output node: Linear
- Activation function in the output node of each individual branch: Linear
- Number of hidden layers in the neural networks: 3
- Number of hidden layers in each neural branch in the NeuralBranch models: 3
- Number of nodes in each hidden layer in the neural networks: 40
- Number of nodes in each hidden layer in the NeuralBranch models: 40
- Optimizer: Adam, with a learning rate of 0.001
- Loss function: mean squared error (mse)
- Batch size: 64
- Epochs: 200, although this is not always reached as we implement early stopping using a temporary validation set with a patience of 30 epochs.
- Data normalization method: standard scaling, applied to both the inputs and the outputs.

Note that since each neural branch has the same number of nodes and hidden layers as the regular neural networks, the NeuralBranch models, which consist of multiple branches, have more nodes in total. While this might seem unfair, we

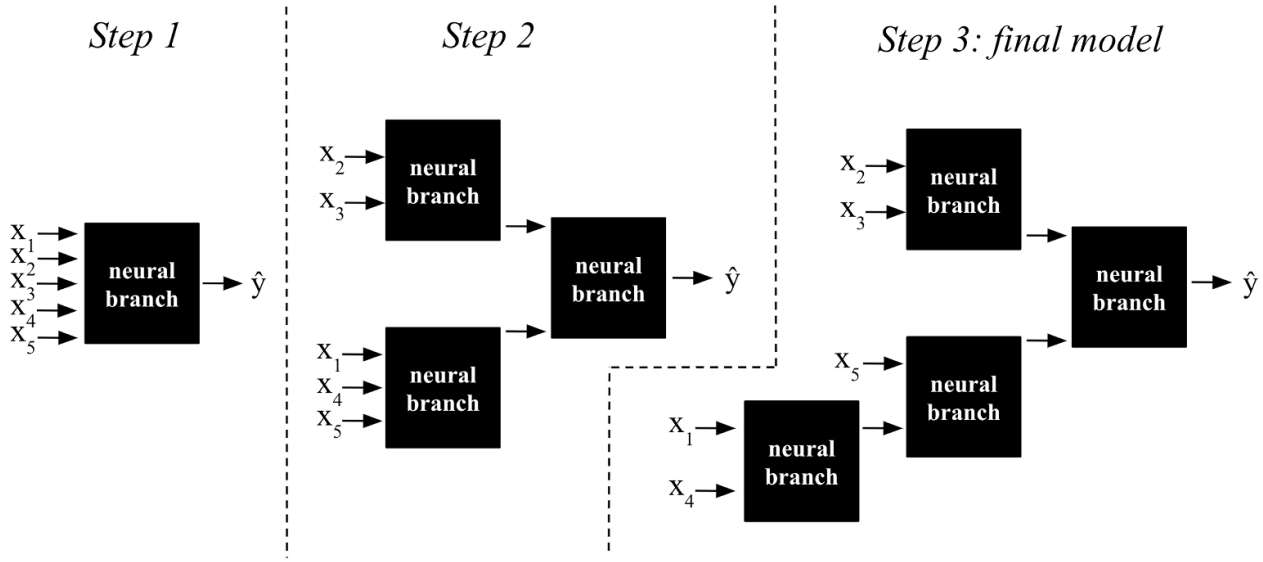


Figure 17. For cases with more than 3 input parameters, the search for the appropriate input assignment to the neural branches occurs in steps. In this example, there are five input parameters: x_1, x_2, x_3, x_4 and x_5. As a first step, a regular neural network is used to obtain a benchmark value for the accuracy. In step two, the input parameters are split into two branches. For this example, we find that the split $\{x_2,x_3\}$ $\{x_1,x_4,x_5\}$ leads to the highest accuracy after testing all possible splits. In step 3, another branch is added to the bottom branch from the previous step, such that only two parameters are parsed through each branch in the final model. Note that in step 3, x_2 and x_3 are locked as inputs to the upper branch since we know from step 2 that this is appropriate. In other words, step 3 focuses solely on exploring all possible ways to parse x_1, x_4 and x_5 through the two lower branches, and finding which setup leads to the highest accuracy.

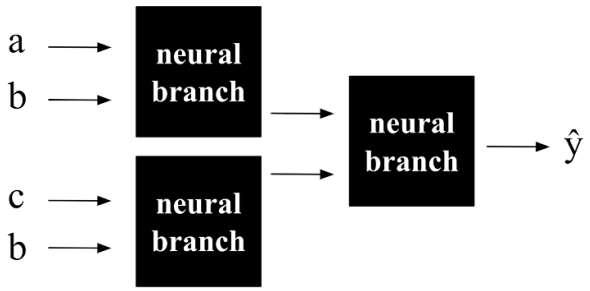


Figure 18. NeuralBranch model when predicting y from a, b, and c, trained on a data set generated with equation (10). Since b is not separable from the other input parameters, it needs to be parsed through two branches to accurately reflect the data.

have observed that the accuracy does not increase when giving the regular neural networks more nodes and layers. The reason for not using fewer nodes and layers in the neural branches is that we do not know *a priori* which of them need to represent more intricate functions. In normal circumstances, the increased number of nodes would lead to a greater concern regarding overfitting. However, as we can interpret the models, we can check if the models have learned irregular, overfitting-like, patterns. Indeed, when we analyze the neural branches used for the pedestal in this work, we do not observe signs of overfitting. That said, it would likely be possible to reduce the number of nodes and layers in some NeuralBranch models presented in this work, especially those that suggest simple learned patterns in the data.

Appendix C. Comparison to SHAP applied to random forest

We here highlight how global interpretability methods like NeuralBranch can be beneficial compared to local interpretability methods like SHAP [27]. For this demonstration, we use a Random Forest (including 100 trees) together with SHAP when predicting the pedestal temperature $T_{e,\mathrm{ped}}$ from the three inputs P_{tot}, $n_{e,\mathrm{sep}}$, and I_P. We use the same dataset that was used in the other methods presented in this work.

Initially, we find that a tree depth of at least 6 is required to achieve the same accuracy as the corresponding neural network and NeuralBranch model ($R^2 = 0.82$), see table 2. This leads to every tree consisting of an average of 59 nodes (115 if leaf-nodes are included when counting). Hence, any individual tree is not easily interpretable on its own, which is why we need to resort to methods like SHAP.

In SHAPs, the goal is to assign an importance value (SHAP value) to each input for a specific prediction, helping us understand how the model makes decisions. This is achieved by analyzing how each input contributes to shifting the prediction from the baseline (expectation value of the output prediction).

While SHAP is primarily an effective tool for analyzing input importance in individual predictions, it can also be used to get a sense of global patterns by aggregating SHAP values across the given dataset. A common method is to analyze SHAP dependence plots, where the SHAP value of an input is plotted versus the actual value of the input. Here, pairwise interactions can be investigated by coloring the scatter points according to the value of another input.

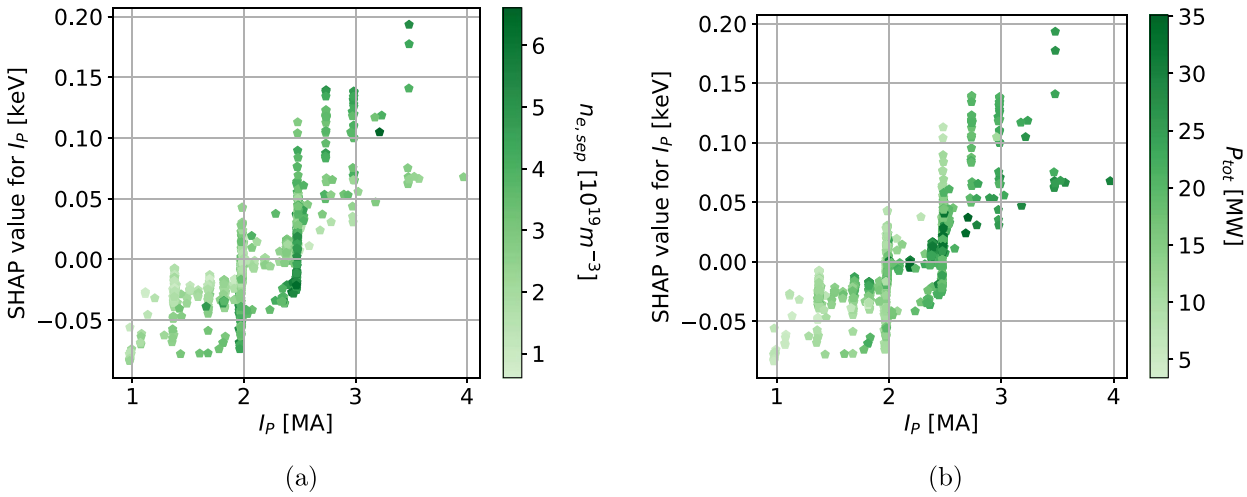


Figure 19. SHAP dependence plots for the Random Forest used to predict $T_{e,\mathrm{ped}}$. Here we show the dependence with respect to I_P, where (*a*) indicates the interaction between I_P and $n_{e,\mathrm{sep}}$, and (*b*) indicates the interaction between I_P and P_{tot}.

By applying SHAP to the Random Forest, we obtain two SHAP dependence plots for I_P that are shown in figure 19. Here, the color indicates the values of $n_{e,\mathrm{sep}}$ and P_{tot} respectively. Note that there are no black lines in these plots where we scan over the x-axis parameter. This is because the scanning method used in NeuralBranch cannot be applied in the same way here. Specifically, in NeuralBranch, even if there are more than two inputs in total, each neural branch only handles two inputs. This allows for a predictive scan performed for an individual neural branch, where one scans over one of its two inputs while keeping the other input constant. However, with our three-input Random Forest model, there is no obvious approach for handling the third parameter while scanning the first parameter with the second held constant. As an alternative approach, we attempted making contours of constant SHAP value in, for instance, I_P and $n_{e,\mathrm{sep}}$ space. However, the variation in the SHAP value imposed by P_{tot} led to irregular, discontinuous contours that offered no meaningful insight. Instead, they only made the plots more crowded and more difficult to interpret. For this reason, we chose to present the dependence plot without these contours.

Nevertheless, to compare the findings in the SHAP dependence plots with those found using NeuralBranch, we first recall that the corresponding NeuralBranch model (figure 14) suggests that there is a strong interaction effect between all three inputs. We can see hints of these patterns in the SHAP dependence plots, but they are not as clear as in NeuralBranch. For instance, NeuralBranch indicates that $T_{e,\mathrm{ped}}$ is independent on I_P when there is a combination of high P_{tot} and low $n_{e,\mathrm{sep}}$. In the SHAP dependence plots, we can only observe that the positive relationship between $T_{e,\mathrm{ped}}$ and I_P is weaker at low $n_{e,\mathrm{sep}}$, and, though less obvious, also slightly weaker at the highest P_{tot}.

In this case, where the main interaction involves three inputs, it is expected that SHAP dependence plots do not provide a fully detailed global picture, as they are limited to indicating only pairwise interactions. For instance, P_{tot} contributes to a vertical spread, even in the dependence plot where only $n_{e,\mathrm{sep}}$ indicates the color. Hence, P_{tot} creates what appears as noise, which makes it difficult to isolate and understand the relationship in detail. This limitation of SHAP becomes increasingly problematic as the number of interacting inputs increases, which is something that global interpretability frameworks like NeuralBranch can handle.

ORCID iDs

A. Gillgren https://orcid.org/0000-0002-3810-2913
A. Ludvig-Osipov https://orcid.org/0000-0002-7057-6414
P. Strand https://orcid.org/0000-0002-8899-2598

References

[1] Wagner F. *et al* 1982 Regime of improved confinement and high beta in neutral-beam-heated divertor discharges of the ASDEX tokamak *Phys. Rev. Lett.* **49** 1408–12
[2] Challis C.D. *et al* (JET Contributors) 2015 Improved confinement in JET high β plasmas with an ITER-like wall *Nucl. Fusion* **55** 053031
[3] Maggi C.F. *et al* (JET Contributors) 2015 Pedestal confinement and stability in JET-ILW ELMy H-modes *Nucl. Fusion* **55** 113031
[4] Frassinetti L. *et al* (JET contributors) 2020 Pedestal structure, stability and scalings in JET-ILW: the eurofusion JET-ILW pedestal database *Nucl. Fusion* **61** 016001
[5] Frassinetti L. *et al* (JET Contributors) 2023 Effect of the isotope mass on pedestal structure, transport and stability in D, D/T and T plasmas at similar βN and gas rate in JET-ILW type I ELMy H-modes *Nucl. Fusion* **63** 112009
[6] Urano H. 2014 Pedestal structure in H-mode plasmas *Nucl. Fusion* **54** 116001
[7] Dunne M.G. *et al* (The EUROfusion MST1 Team and The ASDEX-Upgrade Team) 2016 The role of the density profile in the ASDEX Upgrade pedestal structure *Plasma Phys. Control. Fusion* **59** 014017
[8] Dunne M.G. *et al* (The EUROfusion MST1 Team and The ASDEX Upgrade Team) 2016 Global performance enhancements via pedestal optimisation on ASDEX Upgrade *Plasma Phys. Control. Fusion* **59** 025010

[9] Stefanikova E. *et al* (JET contributors) 2018 Effect of the relative shift between the electron density and temperature pedestal position on the pedestal stability in JET-ILW and comparison with JET-C *Nucl. Fusion* **58** 056010
[10] Stefanikova E., Frassinetti L., Saarelma S., Perez von Thun C. and Hillesheim J.C. (JET Contributors) 2020 Change in the pedestal stability between JET-C and JET-ILW low triangularity peeling-ballooning limited plasmas *Nucl. Fusion* **61** 026008
[11] Beurskens M.N.A. *et al* 2014 Global and pedestal confinement in JET with a BE/W metallic wall *Nucl. Fusion* **54** 043001
[12] Field A.R. *et al* (JET Contributors) 2020 The dependence of exhaust power components on edge gradients in JET-C and JET-ILW H-mode plasmas *Plasma Phys. Control. Fusion* **62** 055010
[13] Leonard A.W., Casper T.A., Groebner R.J., Osborne T.H., Snyder P.B. and Thomas D.M. 2007 Pedestal performance dependence upon plasma shape in DIII-D *Nucl. Fusion* **47** 552
[14] Laggner F.M. , Wolfrum E., Cavedon M., Dunne M.G., Birkenmeier G., Fischer R., Willensdorfer M. and Aumayr F. (The EUROfusion MST1 Team and The ASDEX Upgrade Team) 2018 Plasma shaping and its impact on the pedestal of ASDEX Upgrade: edge stability and inter-ELM dynamics at varied triangularity *Nucl. Fusion* **58** 046008
[15] Kallenbach A., Beurskens M.N.A., Korotkov A., Lomas P., Suttrop W., Charlet M., McDonald D.C., Milani F., Rapp J. and Stamp M. (EFDA-JET Workprogramme contributors and ASDEX Upgrade Team) 2002 Scaling of the pedestal density in type-I ELMy H-mode discharges and the impact of upper and lower triangularity in JET and ASDEX Upgrade *Nucl. Fusion* **42** 1184
[16] ITER Physics Expert Group on Confinement and Transport, ITER Physics Expert Group on Confinement Modelling and Database and ITER Physics Basis Editors 1999 Chapter 2: plasma confinement and transport *Nucl. Fusion* **39** 2175
[17] Martin Y.R. and Takizuka T. (the ITPA CDBM H-mode Threshold Database Working Group) 2008 Power requirement for accessing the H-mode in ITER *J. Phys.: Conf. Ser.* 123 012033
[18] Cordey J.G. (for the ITPA H-Mode Database Working Group and the ITPA Pedestal Database Working Group) 2003 A two-term model of the confinement in Elmy H-modes using the global confinement and pedestal databases *Nucl. Fusion* **43** 670
[19] Gillgren A., Fransson E., Yadykin D., Frassinetti L. and Strand P. (JET Contributors) 2022 Enabling adaptive pedestals in predictive transport simulations using neural networks *Nucl. Fusion* **62** 096006
[20] Kit A., Järvinen A.E., Frassinetti L. and Wiesen S. (JET Contributors) 2023 Supervised learning approaches to modeling pedestal density *Plasma Phys. Control. Fusion* **65** 045003
[21] Udrescu S.-M. and Tegmark M. 2020 AI Feynman: a physics-inspired method for symbolic regression *Sci. Adv.* **6** eaay2631
[22] Angelis D., Sofos F. and Karakasidis T.E. 2023 Symbolic regression trends and perspectives *Artif. Intell. Phys. Sci.* **30** 3845–65
[23] Agarwal R., Melnick L., Frosst N., Zhang X., Lengerich B., Caruana R., and Hinton G. 2020 Neural additive models: interpretable machine learning with neural nets
[24] Murari A., Peluso E., Lungaroni M., Gelfusa M. and Gaudio P. 2015 Application of symbolic regression to the derivation of scaling laws for tokamak energy confinement time in terms of dimensionless quantities *Nucl. Fusion* **56** 026005
[25] Coster P., Basiuk V., Pereverzev G., Kalupin D., Zagorksi R., Stankiewicz R., Huynh P. and Imbeaux F. (Members of the Task Force on Integrated Tokamak Modelling) 2010 The european transport solver *IEEE Trans. Plasma Sci.* **38** 2085–92
[26] Kalupin D. *et al* (ITM-TF contributors and JET-EFDA Contributors) 2013 Numerical analysis of JET discharges with the european transport simulator *Nucl. Fusion* **53** 123007
[27] Lundberg S.M. and Lee S.-I. 2017 A unified approach to interpreting model predictions *Proc. 31st Int. Conf. on Neural Information Processing Systems, NIPS'17 (Red Hook, NY, USA, Curran Associates Inc)* pp 4768–77
[28] Pasqualotto R. , Nielsen P., Gowers C., Beurskens M., Kempenaars M., Carlstrom T. and Johnson D. (JET-EFDA Contributors) 2004 High resolution Thomson scattering for Joint European Torus (JET) *Rev. Sci. Instrum.* **75** 3891–3
[29] Coffey I. *et al* (JET EFDA Contributors) 2014 Effect of fueling location on pedestal and ELMs in JET *41st EPS Conf. on Plasma Physics*
[30] Saarelma S. *et al* (the ASDEX Upgrade Team, MAST-U team, STEP Team, JET Contributors and the Eurofusion Tokamak Exploitation Team) 2024 Density pedestal prediction model for tokamak plasmas *Nucl. Fusion* **64** 076025
[31] Marcano-Cedeão A., Quintanilla-Domínguez J., Cortina-Januchs M.G. and Andina D. 2010 Feature selection using sequential forward selection and classification applying artificial metaplasticity neural network *IECON 2010 - 36th Annual Conf. on IEEE Industrial Electronics Society* pp 2845–50
[32] Snyder P.B. *et al* 2007 Stability and dynamics of the edge pedestal in the low collisionality regime: physics mechanisms for steady-state ELM-free operation *Nucl. Fusion* **47** 961
[33] Connor J.W., Ham C.J. and Hastie R.J. 2016 The effect of plasma beta on high-*n* ballooning stability at low magnetic shear *Plasma Phys. Control. Fusion* **58** 085002
[34] Snyder P.B., Wilson H.R., Ferron J.R., Lao L.L., Leonard A.W., Osborne T.H., Turnbull A.D., Mossessian D., Murakami M. and Xu X.Q. 2002 Edge localized modes and the pedestal: a model based on coupled peeling-ballooning modes *Phys. Plasmas* **9** 2037–43
[35] Snyder P.B., Wilson H.R., Osborne T.H. and Leonard A.W. 2004 Characterization of peeling-ballooning stability limits on the pedestal *Plasma Phys. Control. Fusion* **46** A131