Investigating characteristics of the growth rates from QuaLiKiz using an interpretable surrogate model

RESEARCH ARTICLE | MAY 21 2025

Investigating characteristics of the growth rates from QuaLiKiz using an interpretable surrogate model

Special Collection: Papers from the 5th International Conference on Data-Driven Plasma Science

A. Gillgren ; E. Fransson; A. Ludvig-Osipov ; W. Enström ; L. Flyckt; M. Green ; M. Kvartsén ; Y. Liljegren; E. Olsson ; A. Orthag ; H. Wennberg ; P. Strand

Phys. Plasmas 32, 052306 (2025)
https://doi.org/10.1063/5.0261456

Physics of Plasmas ARTICLE pubs.aip.org/aip/pop

Investigating characteristics of the growth rates from QuaLiKiz using an interpretable surrogate model

Cite as: Phys. Plasmas **32**, 052306 (2025); doi: 10.1063/5.0261456
Submitted: 30 January 2025 · Accepted: 7 May 2025 ·
Published Online: 21 May 2025

View Online Export Citation CrossMark

A. Gillgren,[1,a)] E. Fransson,[2] A. Ludvig-Osipov,[1,3] W. Enström,[1] L. Flyckt,[1] M. Green,[1] M. Kvartsén,[1] Y. Liljegren,[1] E. Olsson,[1] A. Orthag,[1] H. Wennberg,[1] and P. Strand[1]

AFFILIATIONS

[1]Chalmers University of Technology, Gothenburg, Sweden
[2]Aix-Marseille Université, Marseille, France
[3]UKAEA (United Kingdom Atomic Energy Authority), Culham Campus, Abingdon, Oxfordshire OX14 3DB, United Kingdom

Note: This paper is part of the Special Topic on Papers from the 5th International Conference on Data-Driven Plasma Science.
[a)]**Author to whom correspondence should be addressed:** andreas.gillgren@chalmers.se

ABSTRACT

We present an interpretable, machine learning-based surrogate model for the eigenvalue solver in QuaLiKiz, a model that simulates turbulent transport in fusion plasmas. The aim is to exploit prediction transparency to gain insight into the anticipated behavior of QuaLiKiz-based surrogates and the underlying eigenvalue solver, a task that is more challenging when using black-box surrogate models. Specifically, we focus on predicting the growth rate of turbulence driving ion temperature gradient instabilities computed by QuaLiKiz for the normalized poloidal wavenumber $k_\theta \rho_s = 0.325$. We split the task into a classification task, to determine whether the growth rate is positive (unstable mode) or not, and a growth rate prediction task, knowing the mode is unstable. The dataset used is a QuaLiKiz dataset based on JET pulses. The method used is the *NeuralBranch* method, a neural network-based method that reveals how the inputs of the models, in this case plasma parameters, impact the output. Results show that NeuralBranch models outperform linear models and match dense neural networks (traditional black-box models) in accuracy while being interpretable. By analyzing the NeuralBranch models, we identify parameter dependencies that cannot be captured by linear models. For instance, the models indicate that the stabilizing effect of ExB shear on the growth rate is suppressed at low magnetic shear, which can be attributed to how ExB shear influences the eigenfunction width in QuaLiKiz. In summary, this work demonstrates how interpretable methods can shed light on the behavior of surrogates and their underlying counterpart, thus enhancing both model credibility and understanding.

I. INTRODUCTION

Numerical simulations have been key in advancing magnetic confinement fusion research and are utilized in design of reactors and reactor components,[1–3] experimental campaigns,[4–7] and improving our understanding of physical processes in plasmas.[8,9] Such simulations often involve several coupled physical models as a consequence of extreme temporal, spatial, and temperature ranges defining processes in fusion reactors. For example, a multiphysics simulation of tokamak plasma core typically involves models for auxiliary heating (Ohmic, radio frequency, neutral-beam injection), for magnetohydrodynamics (MHD) equilibrium, and for plasma transport (neoclassical and turbulent).[10–15] However, as many of the commonly used first-principles-based models involve a set of equations that need to be solved numerically, simulations often require significant computational effort. Even reduced first-principles-based models typically used in integrated modeling amount to significant computational costs when called repeatedly during simulations of plasma evolution.

In recent years, effort has been made to alleviate this problem through the development of machine learning-based surrogate models.[16–20] These surrogates are trained on datasets generated by computationally demanding models, such that once a surrogate is trained, it mimics the behavior of the original model at a fraction of the numerical cost. This efficiency is achieved because a forward pass through a relatively simple machine learning model is much faster than solving

the original model numerically. For instance, Ref. 21 presents a surrogate model that is approximately 10^4 times faster than the original model it mimics. Additionally, an advantage of surrogate models is the automatic differentiation capability, enabling sensitivity analysis and gradient-based resolution of non-linearities.[22,23]

However, one drawback of most present-day surrogate models is that they are black boxes. In other words, they are often built using machine learning architectures that do not provide a straightforward way to extract the learned relationships between the inputs and the output of the model in an easily interpretable form. Fortunately, as this is a common challenge in the field of artificial intelligence and machine learning, development has been made in the field of explainable and interpretable AI.[24–29] While interpretable models have primarily been shown to provide insights into empirical data, there are several reasons why enabling interpretability would also benefit surrogate models:

- While the system of equations defining a specific model is known, the exact analytic solution remains unknown, which is why numerical methods are used. An interpretable surrogate model trained to replicate the original model could offer a transparent view of how the solution depends on the inputs. In other words, interpretability can enhance the understanding of the behavior of the surrogate model and the original model.
- Even when there is a general understanding of the behavior of a model, interpretability remains important for validating that the surrogate behaves as expected. This is important for building trust in machine learning-based surrogates.
- If an interpretable surrogate makes an outlier prediction, it is significantly easier to backtrack and investigate the cause compared to the black-box case.

Motivated by these benefits, this work aims to develop an interpretable surrogate model for the eigenvalue solver of the quasi-linear model QuaLiKiz.[30,31] Specifically, the aim is to predict the growth rates of turbulent transport driving instability modes, which is a key output of the QuaLiKiz eigenvalue solver. A previous study has demonstrated the feasibility of creating accurate surrogates for QuaLiKiz and its eigenvalue solver,[32] both for classifying whether the growth rate is positive (indicating an unstable mode) and for calculating the actual growth rate when the mode is unstable. However, as these surrogates have been black-box models, we here build on these prior efforts by developing interpretable surrogates for both the stable/unstable classification sub-task and the growth rate prediction sub-task. The dataset we use is a priorly created QuaLiKiz dataset based on experimental values from JET pulses.[21] Moreover, we focus on the growth rate of the ion temperature gradient (ITG)-mode instability at a specific normalized poloidal wavenumber, $k_\theta \rho_s = 0.325$, with $\rho_s = c_s/\Omega_c$, where c_s is the ion sound speed and Ω_c is the ion cyclotron frequency. This is the wavenumber where the ITG-mode, which is the primary contributor to turbulent transport in the core,[33,34] often has its maximum growth rate in the dataset. The interpretable machine learning method we employ is the *NeuralBranch* method,[35] which is a recently developed neural network based method that enables global interpretability.

The goals of our work can be summarized as follows:

- To demonstrate how an interpretable surrogate model can be created as an alternative to opaque black-box surrogate models.
- To analyze our models, which is enabled by interpretability, to shed light on how the ITG growth rate depends on the inputs of the QuaLiKiz eigenvalue solver. This analysis is intended to inform users of QuaLiKiz-based surrogates about the model behavior they may anticipate, particularly in the context of the ITG mode.
- As a secondary goal, we discuss similarities/differences between the behavior of our models and established theory in the context of the ITG mode.

Additionally, in Appendix B, we include a brief analysis involving an alternative output: the ratio of the growth rate to the associated real frequency, as this ratio plays a role in determining whether turbulence is classified as strong or weak.[36]

II. QUALIKIZ DATASET

In this work, we use a priorly created dataset[21] generated with the model QuaLiKiz,[30] where the input parameters in the data are based on experimental values from JET pulses. As mentioned, QuaLiKiz is a quasi-linear model, meaning that it first solves a linear dispersion relation to obtain the eigenvalues of instabilities that lead to turbulent transport. The solution to this eigenvalue problem is the growth rates, γ, and associated real frequencies, ω_r (i.e., the imaginary and real part of the eigenvalue) for the two fastest growing instabilities at 18 different wavenumbers. After the eigenvalue problem is solved numerically, the linear eigenvalues are connected to saturated quantities, such as the electrostatic potential, to obtain the properties of the turbulent transport. In this work, we are focusing on the part of QuaLiKiz that is responsible for calculating the eigenvalues of the instabilities, and as mentioned, we specifically target the growth rate at a specific normalized poloidal wavenumber, $k_\theta \rho_s = 0.325$. In this work, we regard this wavenumber as a reasonable representative of the ITG-wavenumber spectrum for the purpose of investigating input-to-output dependencies, given that the growth rate curve in the database generally follows a negative quadratic trend. This arises as for low $k_\theta \rho_s$, the diamagnetic drift frequency is proportional to the wavenumber, and the growth rate is in turn proportional to the diamagnetic drift. For the higher wavenumbers, in the ion-scales, the growth rate is reduced by finite Larmor radius effect, which as seen in Ref. 37 scales as $(k_\theta \rho_s)^2$.

A. Characteristics of QuaLiKiz

QuaLiKiz is based on electrostatic assumption, using a ballooning representation and $s - \alpha$ geometry.[38] The instabilities captured by QuaLiKiz are the ion temperature gradient (ITG) mode, trapped electron mode (TEM), and electron temperature gradient (ETG) mode. As QuaLiKiz is electrostatic, electromagnetic instabilities such as the kinetic ballooning and micro tearing modes are not captured. This simplifies the process of distinguishing ITG modes in the dataset. Specifically, a negative real frequency represents motion in the ion drift direction, usually associated with the ITG mode, and a positive real frequency represents motion in the electron drift direction, usually associated with the TEM and ETG mode. Hence, it is possible to exclude all instabilities except the ITG mode in the dataset by only checking the sign of the real frequency. However, a caveat to this is that during special circumstances the ITG mode is known to move in the electron drift

direction. These occasions were however deemed too few to motivate an attempt to identify and include such special cases. In other words, our study only concerns ITG modes that move in the ion drift direction.

B. Input parameters

The input parameters of QuaLiKiz consist of plasma parameters that are normalized and dimensionless. For the simulations performed to generate the dataset, light impurities with a charge of less than $10|e|$, where e is the elementary charge, were coalesced into one "light impurity species." Similarly, impurities with a higher charge than $10|e|$ were coalesced into one "heavy impurity species." Hence, the simulations were performed with four species, one main ion species, two impurity species and electrons which makes a total of 33 input parameters. However, by using constraints such as quasi-neutrality and certain assumptions due to the availability of data for the JET pulses, the number of input parameters can be reduced to 15. The assumptions used are as follows:

- The effective charge, Z_{eff}, is radially constant throughout the plasma, i.e., $\nabla Z_{eff} = 0$.
- $T_i = T_{imp}$, as the widely available diagnostics measure the temperature of impurity ion species, implying $R/L_{T_i} = R/L_{T_{imp}}$.
- The main ion is deuterium, with $Z_i = 1$ and $A_i = 2$.

Here, T_i is the main ion temperature, T_{imp} is the impurity temperature, R is the tokamak major radius, and L_X is the gradient length, $L_X := -(\frac{\partial}{\partial \rho_{tor}} \ln X)^{-1}$, where X is a plasma profile such as the densities and temperatures. ρ_{tor} is a flux label defined as

$$\rho_{tor} := \sqrt{\frac{\psi_{tor}(r)}{\psi_{tor}(a)}}, \tag{1}$$

where ψ_{tor} is the toroidal magnetic flux and a is the minor radius. The full set of the 15 input parameters in the dataset is presented in Table I. Here, $\tau = T_i/T_e$, where T_e is the electron temperature.

C. Data specifications and selection

The original dataset was based on 2135 JET pulses, both from quasi-steady-state and transient scenarios, and a total of 12 328 time windows were selected. In certain cases, all necessary data were not available and assumptions were applied.

- The Z_{eff} contribution of the light impurity did not exceed 0.2 if insufficient impurity information is provided.
- $M_{tor} = R/L_{u_{tor}} = \gamma_E = 0$ if no plasma rotation measurements are available.
- $T_i = T_{imp} = T_e$ if no ion temperature measurements are available.
- $Z_{eff} = 1.25$ if no line-integrated effective charge measurements are available.

The extracted experimental data were used to populate the dataset at nine equidistant radial positions between $\rho_{tor} = 0.1$ and 0.9. Note that the experimental data were only used for the input parameters, as the output growth rate for all cases is computed by QuaLiKiz. Furthermore, the dataset was expanded beyond the experimental values in the parameters $\{R/L_{n_e}, R/L_{T_e}, R/L_{T_i}, \hat{s}, \gamma_e\}$. This means that additional entries were generated by varying these five parameters while keeping the other inputs coherent with the experimental values. Additionally, if a data entry had rotation data available, an identic data entry was created but with zero rotation.

All these procedures resulted in a total number of data entries in the original dataset being roughly 3.7×10^7. However, in this work, we have selected, from the original dataset, a random sample of 30 000 entries for the training set, and a random sample of 30 000 entries for the test set used for evaluation. Here, a check was made to ensure that data entries with extreme values (outliers) were excluded. Additionally, to ensure an unbiased evaluation, it was made sure that no entry was included in both the training set and the test set. The decision to use only a fraction of the full dataset was driven by the need to enhance the efficiency of the investigation process, which required numerous training iterations. That being said, we did not observe a significant drop in prediction accuracy when using the smaller dataset for training compared

TABLE I. The full set of input parameters in the QuaLiKiz dataset used in this work.

Dimensionless parameter	Associated physical parameter	Dataset min	Dataset max	Description
ρ_{tor}	r	0.10	0.93	Flux label
q	q	0.79	3.99	Safety factor
$\hat{s}$	∇q	−0.48	3.99	Magnetic shear
R/L_{T_e}	∇T_e	−4.97	24.99	Normalized electron temperature gradient
Z_{eff}	Z_{eff}	1.00	3.97	Effective charge
$\log_{10}(\nu^*)$	n_e	−1.499	0.49	Collisionality
R/L_{n_e}	∇n_e	−4.97	9.98	Normalized electron density gradient
τ	T_{imp}	0.500	1.75	Ion and electron temperature ratio
R/L_{T_i}	∇T_{imp}	−4.99	19.98	Normalized ion temperature gradient
$R/L_{n_{imp,light}}$	$\nabla n_{imp,light}$	−5.02	10.03	Normalized light impurity density gradient
$N_{imp,light}$	$n_{imp,light}$	0.0002	0.049	Light impurity density
α	B_0	−0.047	1.499	Normalized pressure gradient
M_{tor}	Ω_{tor}	−0.048	0.99	Rotation Mach number
$R/L_{u_{tor}}$	$\nabla \Omega_{tor}$	−0.99	4.98	Normalized rotation gradient
γ_E	$\nabla^2(n_i T_i)$	−1.49	0.49	ExB shearing rate

to when using a large dataset consisting of 3×10^6 entries, and the parameter distributions were comparable in both cases.

In both the training set and the test set, approximately 45% of the entries are accompanied by a positive growth rate (unstable mode) at the specific wavenumber we are considering. Consequently, since only unstable entries were considered for the growth rate prediction, approximately 13 500 entries were included for this sub-task.

D. Correlations among the inputs

While most of the input parameters presented in Table I are relatively disentangled, they are not fully independent. In some cases, there are natural connections between parameters. For instance, the magnetic shear $\hat{s}$ is the normalized gradient of the safety factor q, and the light impurity density $N_{imp,light}$ affects the effective charge Z_{eff}.

For a more comprehensive overview, Fig. 1 shows the Pearson correlation matrix of the training set for all input parameters, which, for instance, suggests that magnetic shear $\hat{s}$ and the radial position, ρ_{tor}, have the strongest correlation (0.81). This is natural because of the usual monotonically increasing safety factors in JET plasmas. Specifically, we expect to have a flat safety factor profile and therefore magnetic shear close to 0 near the center of the plasma. Moreover, we observe non-negligible correlations between ρ_{tor} and normalized gradients since these are larger toward the edge of the plasma, i.e., higher ρ_{tor}.

However, except for ρ_{tor}, which is not even a plasma parameter but rather a representative of the radial position, the parameters generally do not exhibit correlations at levels that raise concern. This is an important consideration for the analysis of the parameter relationships of the models presented in this work. As will be seen, ρ_{tor} is not a parameter that we include in our models as it is shown to not be critical for the predictions, which further alleviates potential concerns related to the correlations where ρ_{tor} is involved.

III. ION TEMPERATURE GRADIENT MODE

The ITG mode has been studied extensively the past decades, both analytically and numerically.[37,39–41] Therefore, it has several known characteristics, which we describe in this section to establish a foundation for comparing them with the findings from the models we train in this work.

The ITG mode is destabilized by a high background ion temperature gradient which drives the instability, and it exists as slab- and toroidal-like versions. The slab-like mode exists due to the background temperature causing a change in the temperature perturbation that moves in phase with the perturbed ExB drift. The toroidal-like mode arises due to the ∇B and curvature drifts, connecting to the background temperature gradients on the bad curvature side, i.e., the low-field side of the tokamak. These two versions of the ITG mode are separated by the shear length scale, L_s, with gradient scale lengths that play a key role theoretically. The limits can be described as $\hat{s}/q \ll 1$ for the slab limit and $\hat{s}/q \gg 1$ for the toroidal limit. To date, the toroidal mode has been the cause of the higher turbulent transport in experiments.[34]

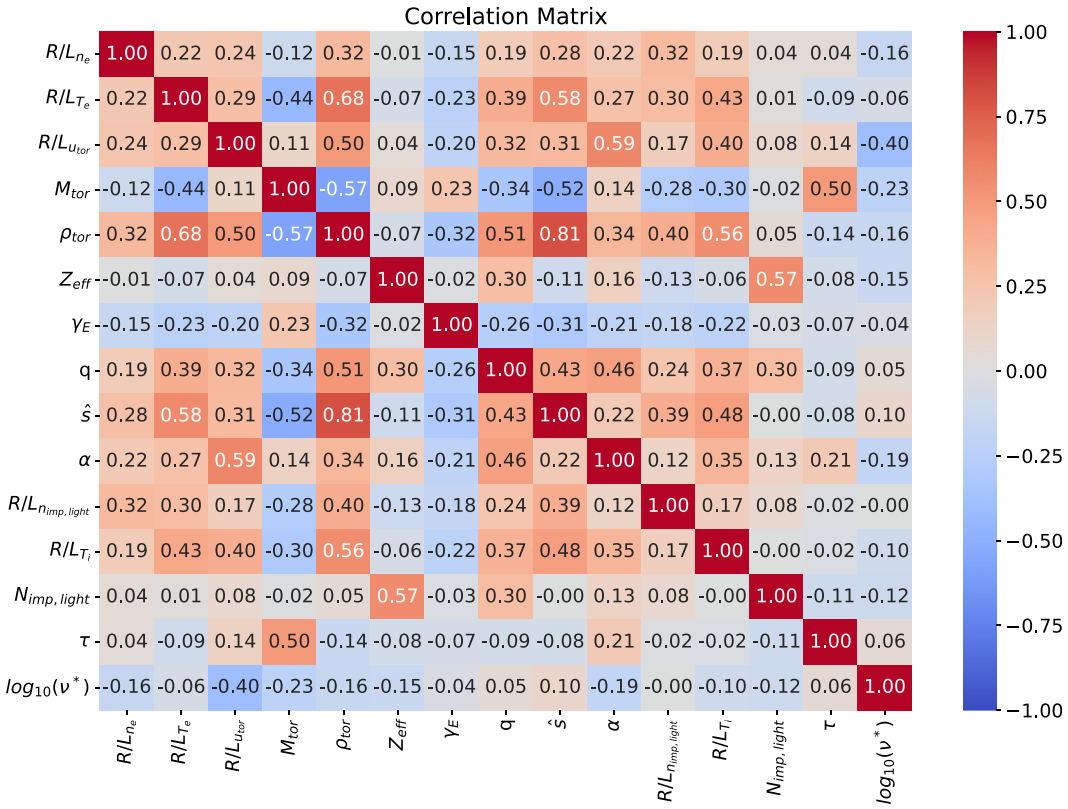

FIG. 1. Pearson correlation matrix for all 15 input parameters in the QuaLiKiz dataset used in this work.

A. Ion temperature gradient mode threshold

An important feature of ITG mode is the existence of an instability threshold. Specifically, under a certain critical normalized ion temperature gradient $R/L_{T_i}|_{crit}$, the mode is stable. Analytical expressions have been derived for the threshold, here by Jenko *et al.*,[42]

$$R/L_{T_i}|_{crit} = \max\left((1+\tau)\left(\frac{4}{3}+1.91\hat{s}/q\right),\ 0.8R/L_n\right). \quad (2)$$

Note that in Ref. 42, the analysis was performed with only one ion species, and therefore, the electron and ion normalized density were always the same. In our dataset, we use the electron normalized density gradient R/L_{n_e} as an input, which varies slightly from the ion normalized gradient because of the low impurity content. However, in the argumentation regarding Eq. (2), we will assume the density gradient being roughly the same as the normalized electron density gradient.

The first argument of the max statement was derived by combining the formulas for slab-like ITG mode by Romanelli[40] with the $\frac{4}{3}$ term, and the toroidal-like mode by Hahm and Tang[43] (the $1.91\hat{s}/q$-term). These were derived in the flat density profile limit, $a/L_n \to 0$. In this limit, the threshold is determined by kinetic effects associated with the magnetic drift frequency.[43,44]

Based on Eq. (2), we can anticipate beforehand how the models we train in this work should behave. First, a higher normalized ion temperature gradient R/L_{T_i} should yield a higher probability of an unstable mode as it drives the ITG mode. Second, the parameters τ, $\hat{s}/q$, and R/L_n ought to have an important role deciding the stable/unstable regions. All of these three parameters increase the threshold according to (2), meaning that higher values of these parameters should lead to a higher chance of predicting a stable mode.

B. Ion temperature gradient mode growth rate

We now continue with theoretical descriptions regarding the actual growth rate of the ITG mode given the scenario that the mode is unstable. First, gradients far beyond the critical gradient threshold are expected to result in a larger growth rate. Therefore, we can anticipate that the parameters discussed in Sec. III A will also play a significant role in training the growth rate prediction model. To understand these parameters in more detail, and to understand which other parameters might be important, we analyze two analytical expressions derived in Ref. 45 in the fluid limit. The expression for the slab limit is

$$\gamma^2_{slab} = \frac{\tau}{Z_{eff}}\frac{n\omega^*_{pe}k_\theta d_{eff}c_{eff}}{2L_s}, \quad (3)$$

and the interchange limit

$$\gamma^2_{interchange} = \frac{\left(f_t+\frac{\tau}{Z_{eff}}\right)n^2\omega^*_{pe}\omega_{de}}{f_p}, \quad (4)$$

where γ_{slab} and $\gamma_{interchange}$ are the growth rates for the two limits, n being the toroidal wavenumber, $\omega_{de} = -\frac{k_\theta T_e}{eB}(\cos\theta + \hat{s}\theta\sin\theta)$ being the vertical drift frequency, $\omega^*_{pe} = -\frac{k_\theta T_e}{eB}\frac{1}{L_p}$ being the diamagnetic frequency associated with the pressure gradient, $c_{eff} = T_e/m_p$, m_p is the proton mass, $d_{eff} = \frac{f_p}{f_t}\frac{T_e}{n_e}\sum_i \frac{n_i Z_i^2}{T_i}\delta_i + \frac{4T_e m_p}{e^2B^2}$, $f_{t,p}$ being the trapped and passing particles fractions, δ being the banana width, θ being the poloidal angle, and B being the magnetic field.

While we observe that more parameters influence the growth rate than the critical threshold, we choose to emphasize those parameters that appear in both the critical threshold (2) and the growth rate expressions (3) and (4). For instance, the magnetic shear dependency in the slab limit (3) comes from the shear length scale, as $L_s = Rq/\hat{s}$. For the interchange limit (4), it enters through ω_{de}. Thus, we have the dependencies, $\gamma^2_{slab} \sim \hat{s}$, and $\gamma^2_{interchange} \sim (\cos\theta + \hat{s}\theta\sin\theta)$. The ion and electron temperature ratio τ dependency is explicit in the equations, $\gamma^2_{slab} \sim \tau$, and $\gamma^2_{interchange} \sim (f_t + \tau/Z)$. The normalized density and ion temperature gradients enter through ω^*_{pe} and the pressure length scale for both expressions for the growth rate. Hence, the growth rates increase with the normalized gradients. In total, we note that only one out of the four parameters in the critical gradient expression (2) has the same sign of the dependency in the growth rate cases, namely, the normalized ion temperature gradient with its destabilizing effect (increasing the growth rate). This is not surprising as it is the drive of the instability. The other three parameters $\hat{s}$, τ, and a/L_n are destabilizing (increasing the growth rate) rather than stabilizing like in the case for the critical threshold. This is an indication of increased stiffness for higher values of these parameters.

The theoretical concepts discussed in this section are revisited at the end of the results section for each prediction sub-task to connect the findings of our machine-learning based models to the theory.

IV. METHOD

A. NeuralBranch framework

In this work, we use the *NeuralBranch* method[35] to enable interpretability when creating surrogate models for the two sub-tasks. This approach splits dense neural networks into separate sub-networks of dense layers, each handling only two input parameters and one output parameter. In the rest of this work, we refer to these sub-networks as neural branches. By limiting each neural branch to two inputs, the output can be visualized as a function of the two inputs in a plot, for example, with the inputs on the x and y axes and the output represented by color. This visualization enables a qualitative interpretation of the relationship between the inputs and outputs of each neural branch. In practice, this is achieved by first training the model, and then by parsing the inputs of the full test set through the model, which gives arrays of all inputs and outputs of the neural branches. Note also that since we only need the two inputs and the output of each neural branch to plot the parameter relationships, there is no need to analyze the internal weights of the hidden nodes when using this approach.

A complete NeuralBranch model includes multiple neural branches arranged such that the output of a neural branch either is connected to the input of another neural branch, or set as the final output of the model. This is illustrated in Fig. 2. In this example, the output $\hat{y}$ is predicted from three input parameters: x_1, x_2, and x_3. The first neural branch takes x_1 and x_2 as inputs and produces an intermediate parameter z. The second neural branch takes z and x_3 as inputs and outputs the final prediction $\hat{y}$. By visualizing how z depends on x_1 and x_2, and subsequently how $\hat{y}$ depends on z and x_3, the dependencies of $\hat{y}$ on x_1, x_2, and x_3 can be fully interpreted. As each neural branch contains dense layers of neural network nodes, they allow for the learning of complex mappings. Furthermore, all neural branches are trained

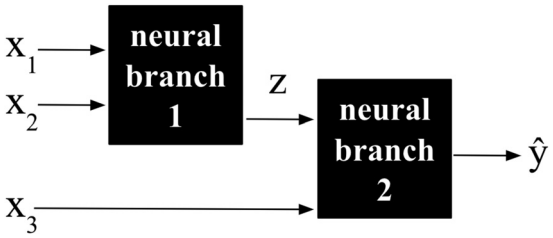

FIG. 2. An illustrative example of a NeuralBranch model architecture. Each neural branch, which are illustrated with boxes, includes dense neural network layers. For cases with more than three input parameters, more branches are required.

together as a single model, eliminating the need for prior assumptions about the intermediate parameter z.

An important aspect of the NeuralBranch method is the selection of which input parameters that should be allocated to which neural branch. For instance, without prior knowledge, we cannot be certain that x_1 and x_2 in this example is the appropriate choice of parameters to be passed to neural branch 1. Consequently, all possible pairings of inputs must be explored, and the configuration that minimizes prediction error is selected as it best reflects the data. The process for selecting neural branch configurations and handling cases with more than three input parameters is detailed in Ref. 35. In the results sections, we provide the final NeuralBranch architecture and visualizations for the most accurate configuration. We refer to Appendix A for more details on training specifications, such as hyperparameters used in this work.

We also want to acknowledge that the NeuralBranch method is inspired by Neural Additive Models (NAMs),[24] which also enable interpretability by splitting the neural network into sub-networks that process at most two input parameters each, and by visualizing the learned mappings. However, in NAMs, the sub-networks operate in parallel, and the complete model output is the sum of the outputs from the individual sub-networks. While this simplifies interpretation by removing dependencies between sub-networks, it also introduces limitations. For example, the NeuralBranch model in Fig. 2 allows for complex interactions among all three input parameters: x_1 and x_2 may interact in neural branch 1, and x_3 may interact with z, representing the contribution from x_1 and x_2, in neural branch 2. Such interactions involving three or more parameters would not be possible in a NAM with two sub-networks. Additionally, the output of the NeuralBranch model is not restricted to being the sum of neural branch outputs. Nevertheless, NAMs may still be advantageous in scenarios where the data patterns align with their constraints, particularly in cases with a large number of important input parameters. This is because the sub-networks in NAMs are independent on one another, making the complete model easier to interpret when many sub-networks are included.

B. Evaluation metrics

To evaluate the performance of all classifier models and growth rate prediction models presented in this work, the balanced F-score F_1 and the coefficient of determination R^2 are used, respectively.

The balanced F-score is defined by

$$F_1 = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}, \tag{5}$$

where precision is the ratio between the number of correctly predicted positive results and the number of all samples predicted to be positive (including false positives), and recall is the ratio of the number of correctly predicted positive results and the number of all samples with positive ground truth value. The closer the F_1 score of a model is to 1 (the maximum attainable value), the more accurate the model is.

The coefficient of determination is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{N_{\text{test}}} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{N_{\text{test}}} (y_i - \bar{y})^2}, \tag{6}$$

where N_{test} is the size of a test dataset, y_i are the ground truth values, $\hat{y}_i$ are the corresponding model predictions, and $\bar{y}$ is the mean of the ground truth values. As in the case of F_1, the closer the R^2 score of a model is to 1 (the maximum attainable value), the more accurate the model is. The R^2-metric suits cases where the output is continuous, which is the case for the prediction of the growth rate on the unstable entries in the dataset.

C. Selecting input parameters

While the QuaLiKiz dataset used in this work includes 15 input parameters, they are not all equally important for achieving high prediction accuracy. This is demonstrated in this section with a test where the goal is to find which inputs need to be included to achieve high prediction accuracy, and consequently which can be excluded without reducing the accuracy. Excluding input parameters that do not improve prediction accuracy is important for two main reasons in our work. First, a machine learning model is not encouraged to learn relationships related to input parameters that do not reduce the loss function. Second, using fewer input parameters results in fewer neural branches in a NeuralBranch model, which simplifies the interpretation process.

To perform the input selection test, we employ a sequential forward selection (SFS) approach,[46] which can be summarized in the following steps:

1. A dense neural network is used to predict the given output when all inputs are included. This provides a benchmark value for the prediction accuracy.
2. Another neural network is trained to predict the output, but now only one input parameter is allowed. The goal here is to find, out of all inputs, which one provides the highest prediction accuracy when used alone. This input is then fixed as the first input parameter.
3. Gradually introduce one additional input parameter at a time. The parameter that produces the highest accuracy, when combined with the already fixed parameters, is selected as the next input to be fixed. Note that a new neural network is trained from scratch each time a new input parameter is added.
4. Repeat step 3 until the prediction accuracy is considered close enough to the benchmark score.

By incrementally fixing one input parameter at a time, we conserve computational resources, as this approach significantly reduces the number of parameter combinations compared to testing all possible configurations (120 vs 32 767). Moreover, the reason for using

dense neural networks for the SFS approach, rather than simpler models, is to avoid the results being impacted by insufficient expressive capacity.

To demonstrate the input selection method using SFS in practice, we here apply it to the classification sub-task, and the result is presented in Fig. 3. We observe that the input parameter R/L_{T_i} produces the highest accuracy when only one input is allowed. Specifically, R/L_{T_i} alone yields $F_1 = 0.82$ when predicting whether the mode is stable or not. This is not surprising considering that ITG modes are, as mentioned, driven by ion temperature gradients. As additional parameters are introduced incrementally, we see that $\hat{s}$ followed by τ and R/L_{n_e} lead to a prediction accuracy which is approximately the same as the benchmark value when all 15 inputs are included ($F_1 \approx 0.89$). Therefore, for the classification model in our work, we use these four most important inputs. However, it is important to note that the input selection test presented here does not imply that the other parameters are universally unimportant in all scenarios. Instead, the results indicate that, on average, only the four input parameters shown in Fig. 3 are important for the stable/unstable classification applied to the specific dataset used.

Since we are initially focusing on the classification sub-task, the input selection test and all other results for the growth rate prediction sub-task are presented later in Sec. VI.

V. STABLE/UNSTABLE CLASSIFICATION: RESULTS

In this section, we present the remaining results related to the stable/unstable classification sub-task. At the end of this section, we discuss the results in relation to the ITG stability theory.

A. Linear classification model

Before using the neural network-based NeuralBranch model, we first train a linear classification model. The goal is to create a simple model that can be compared with the NeuralBranch model. Since the output is binary in classification tasks, a sigmoid function σ is applied to a linear combination of the inputs. The coefficients in this linear model are found through the minimization of the binary cross-entropy when exposing the model to the training dataset, and the result is

$$\text{output} = \sigma(0.96R/L_{T_i} - 1.85\hat{s} - 3.68\tau + 0.20R/L_{n_e} - 1.20), \quad (7)$$

where σ is the sigmoid function. Here, an output that is larger than 0.5 means that the mode is predicted as unstable, which means that the inputs with positive coefficients in (7) have a destabilizing effect on the mode when increased. Additionally, this linear model yields $F_1 = 0.84$ when evaluated on the test set, which is a competitive result considering that the neural network used in the input selection test yields $F_1 = 0.89$. There is however a non-negligible discrepancy in the F_1-values, which motivates the use of the NeuralBranch model to investigate what is causing this difference. Nevertheless, by analyzing the coefficients in (7), we note that increased R/L_{T_i} and R/L_{n_e} overall have a destabilizing effect on the mode and that increased $\hat{s}$ and τ overall have a stabilizing effect on the mode.

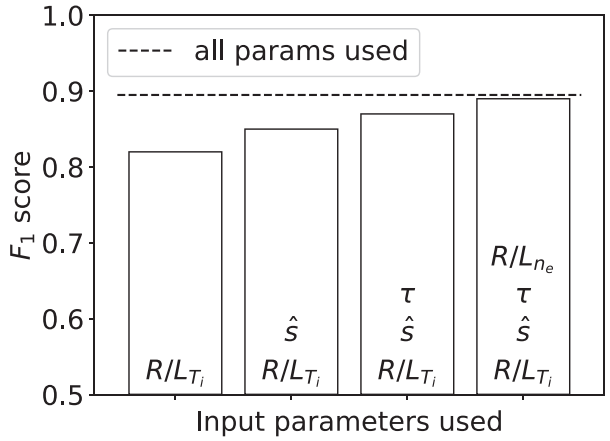

FIG. 3. Result of the SFS method used to find the most important inputs for the stable/unstable classifier. The leftmost bar shows which parameter achieves the highest F_1-score when only one input is allowed, and the subsequent bars show which parameters that are the highest scoring as additional parameters are allowed incrementally. The dashed line indicates the score when all 15 inputs are used.

B. NeuralBranch model for the classification

We now turn to the main classification model in our work, which is the NeuralBranch model for this sub-task. Figure 4 shows the final architecture and the visualizations of this model. In the visualizations, the output of the neural branches is indicated by the color, and contour lines are used to highlight constant values of the output. This NeuralBranch model yields $F_1 = 0.89$, which matches the neural network used in the input selection test and exceeds the linear classification model ($F_1 = 0.84$). We have summarized conclusions of our interpretation of the NeuralBranch model in Table II. As seen in the table, the model reveals patterns and interactions that linear models cannot fully capture, which explains the higher accuracy in the NeuralBranch model. In Sec. V C, details regarding how we performed the interpretation are described.

C. Details on the interpretation process of the NeuralBranch classifier

When interpreting the visualizations of the NeuralBranch classifier (Fig. 4), we analyze how the output of the neural branches depends on their inputs, and also how the intermediate values z_1 and z_2 propagate through the model. For instance, we can see that higher τ increases z_2 in neural branch 2, and that higher z_2 in turn increases the stability threshold in neural branch 3. This reasoning applies for how we conclude on how each input affects the output. Moreover, the approximately straight equidistant lines in neural branch 2 tells us that there is an additive (non-interactive) behavior between τ and z_1, which represents the contribution from $\hat{s}$ and R/L_{n_e}. In contrast, neural branch 1 reveals an interactive pattern as higher R/L_{n_e} appears to shift the impact that $\hat{s}$ has on z_1.

D. Discussion of classifier results in relation to theory

We now use the theory outlined in Sec. III as a starting point to be compared with the behavior of the presented classifiers (and QuaLiKiz as this is the model our models attempt to mimic).

First, we observe that the four input parameters that turned out to be important for achieving high prediction accuracy, see Fig. 3, are the ones in Eq. (2) for the critical threshold. Second, for the simple linear model, see Eq. (7), we can notice that the dependencies of the

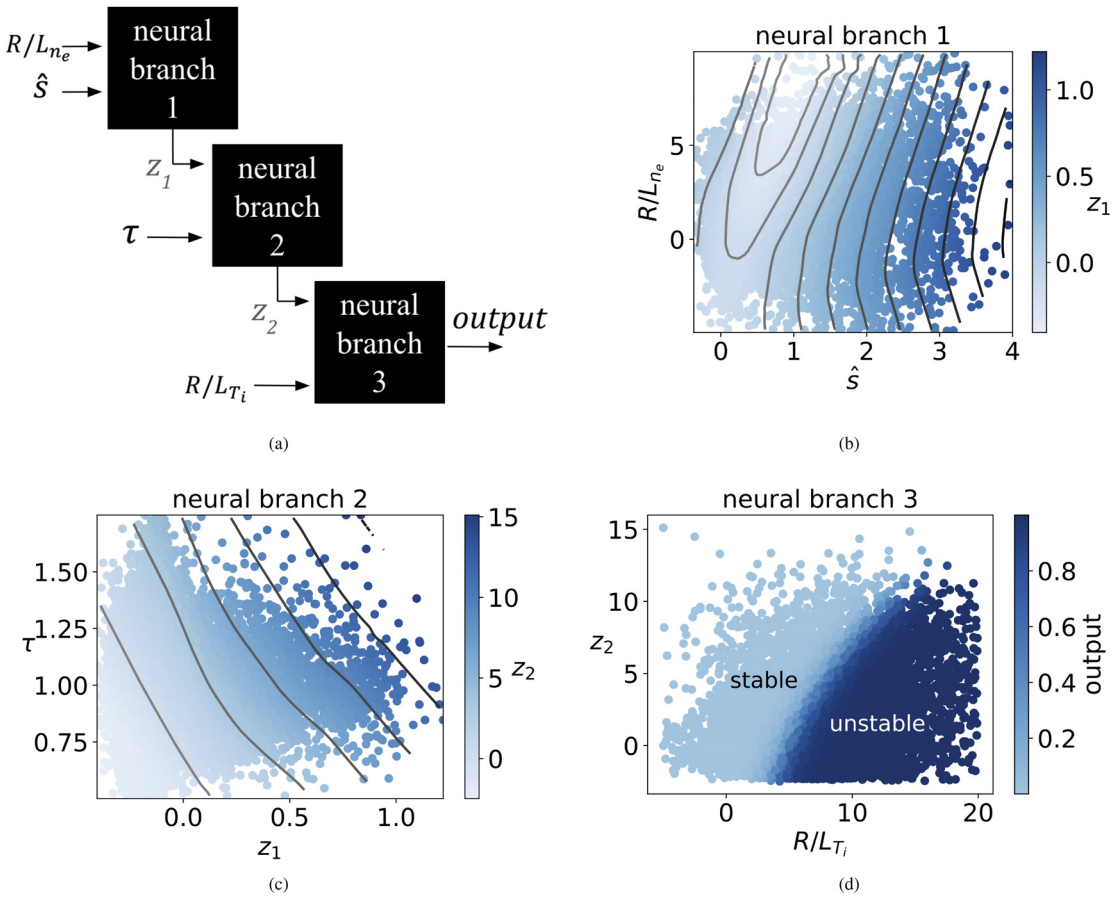

FIG. 4. The architecture (a) and visualizations (b)–(d) of the NeuralBranch model for the classification task. In the visualizations, we have plotted the output of each neural branch in color. The contour lines in (b) and (c) indicate where the output of the neural branches are constant. The scatter points in the visualizations are obtained through parsing the full test set through the model post-training.

normalized ion temperature gradient R/L_{T_i}, magnetic shear $\hat{s}$, and the ion and electron temperature ratio τ are qualitatively the same as in Eq. (2). An increase in R/L_{T_i} gives an unstable mode, and increases in $\hat{s}$ and τ gives a stable mode. However, the normalized gradient R/L_{n_e} has the opposite dependency compared to the critical threshold equation. Specifically, for the linear model fit on the data, it is destabilizing but for the theoretical critical threshold equation it is stabilizing.

Turning to the results of the NeuralBranch model, see Table II, $\hat{s}$ and τ increase the critical threshold, which is in accordance with the analytical critical threshold. The more interesting part is the interplay between R/L_{n_e} and $\hat{s}$ displayed in Fig. 4(b). Specifically, according to the analytical expression the normalized density gradient ought to increase the critical threshold, however as we noticed this is only true for low magnetic shear according to the NeuralBranch model. Additionally, according to the analytical expression in Eq. (2), the importance of the normalized density gradient should increase when the magnetic shear is low, due to the max-statement in the equation. However, this is not an obvious trait of neural branch 2 displayed in Fig. 4(b).

The mentioned discrepancies, and the positive scaling for the density gradient for the linear model, see Eq. (7), might be due to the influence of the TEM. Specifically, while all unstable data entries used in this work are associated with ITG-modes, some are likely coupled to the TEM. Since the TEM is driven by the density gradient,[47] an increase in the density gradient would destabilize the ITG-TEM coupled cases, thus explaining the positive scaling observed in Eq. (7).

VI. GROWTH RATE PREDICTION: RESULTS

We now move on from the classification of stable/unstable modes to the growth rate prediction sub-task. Here, all models are only trained and evaluated on the unstable entries in the dataset, and the output, which is the actual growth rate of the instability, is a continuous parameter. Therefore, all models now have a linear output node and are evaluated using the R^2-metric instead of the F_1-metric. We first present the input parameter selection test, and then a linear growth rate prediction model, followed by the final NeuralBranch model for this sub-task. The section ends with a discussion of the results in relation to the theory regarding ITG growth rates outlined in Sec. III.

A. Input parameter selection for growth rates

Figure 5 shows the result of the SFS method used to find the most important input parameters for the growth rate prediction. Much like

TABLE II. Summary of input-to-output behavior of the NeuralBranch model for the classification sub-task, based on interpretation of Fig. 4.

Input parameter	Impact on classification	Interactions with other parameters
R/L_{T_i}	Higher R/L_{T_i} leads to unstable output.	The other inputs affect the critical value of R/L_{T_i} where the output becomes unstable.
$\hat{s}$	Higher $\hat{s}$ leads to stable output as it increases the R/L_{T_i} threshold. However, the impact of $\hat{s}$ is not linear as it has a minimum destabilizing effect at $0 < \hat{s} < 1$.	The most destabilizing $\hat{s}$ gradually goes from $\hat{s} \approx 0$ to $\hat{s} \approx 1$ as R/L_{n_e} increases.
τ	Higher τ leads to stable output as it increases the R/L_{T_i} threshold.	No interactions with remaining inputs ($\hat{s}$ and R/L_{n_e}).
R/L_{n_e}	Higher R/L_{n_e} generally leads to unstable output as it lowers the R/L_{T_i} threshold.	At the very lowest $\hat{s}$ values, we observe a deviation from the general trend as here higher R/L_{n_e} stabilizes the output.

in the classification task, R/L_{T_i} is the highest scoring parameter when only one input is allowed, and it is followed by a similar succession of parameters: $\hat{s}$, γ_E, R/L_{n_e}, and τ. Note that the exception, namely, γ_E which was unimportant for the classification task, now appears before R/L_{n_e} and τ for the growth rate prediction task. Another difference compared to the classification is that these first few parameters are not fully sufficient to reach the same prediction accuracy compared to when all 15 input parameters are used. For instance, including the five most important parameters yields $R^2 = 0.85$ while including all 15 inputs yields $R^2 = 0.93$. This is an indication that while all parameters in the dataset are not necessarily important for the classification, they are more important when considering the finer nuances in the growth rate prediction. However, we choose to focus the analysis on the five most important parameters and examine their impact on the output growth rate, namely, R/L_{T_i}, $\hat{s}$, γ_E, R/L_{n_e}, and τ. This selection is partly motivated by the fact that, with the exception of γ_E, these are the same parameters used in the classification model. Additionally, while we acknowledge that the following observation is somewhat subjective, the relative increase in R^2 per parameter in Fig. 5 appears to be weaker after the fifth parameter compared to prior parameters. That said, in a surrogate model where maximizing prediction accuracy is the top priority, we recommend including all 15 input parameters from the dataset. The following analysis is simply intended to inform on how the five most important inputs, that are responsible for 91% of the prediction accuracy, impact the output growth rate.

B. Linear growth rate prediction model

In this sub-task, no sigmoid function is needed to be applied to the linear function since the output now is continuous. Moreover, here the fitting coefficients are found by minimizing the mean squared error (mse) of the model rather than the binary cross-entropy. This gives the result

$$\begin{aligned}\text{growth rate} = {} & 0.051 R/L_{T_i} - 0.081\hat{s} - 0.003\gamma_E \\ & + 0.014 R/L_{n_e} - 0.102\tau - 0.062,\end{aligned} \tag{8}$$

which yields $R^2 = 0.72$. This is a noticeable drop compared to the neural network used in the input selection test ($R^2 = 0.85$), which motivates the use of the NeuralBranch model to investigate the more complicated parameter relationships that are causing this difference.

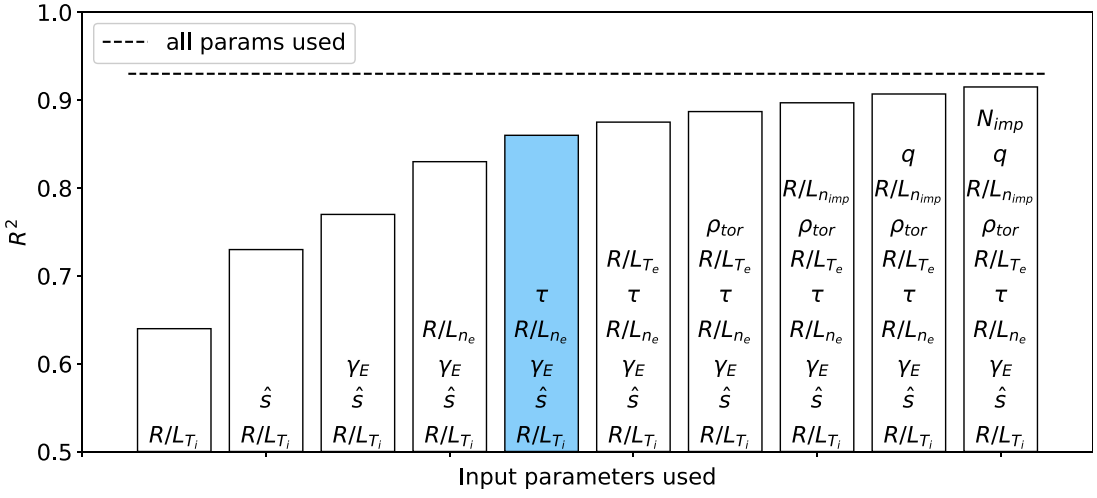

FIG. 5. Result of the SFS method used to find the most important inputs for the growth rate prediction sub-task. The blue bar indicates which parameters we decided to include in the final analysis. Note that $N_{imp,light}$ and $R/L_{n_{imp,light}}$ are written as N_{imp} and $R/L_{n_{imp}}$ to make them small enough to fit in the bars.

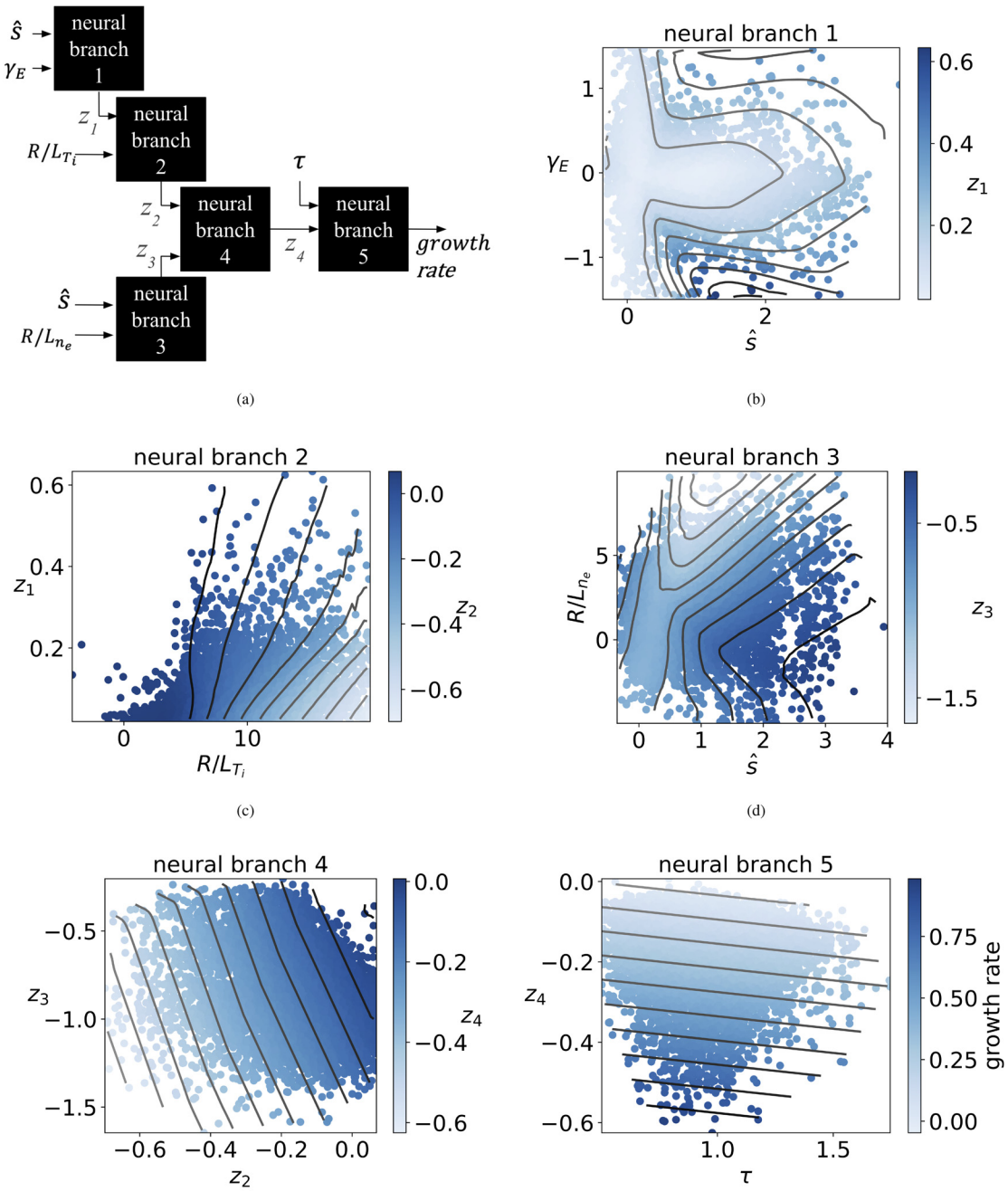

FIG. 6. The architecture (a) and visualizations (b)–(f) of the NeuralBranch model for the growth rate prediction task. In the visualizations, we have represented the output of each neural branch in color. The contour lines indicate where the output of each neural branch is constant. The scatter points in the visualizations are obtained through parsing the full test set through the model post-training.

For instance, we can conclude that the linear model likely fails to accurately capture the way γ_E contributes to the growth rate. This is because in the input selection test, where neural networks are used, it is shown to be the third most important parameter, but in (8), the growth rate is almost independent of γ_E considering the small accompanied coefficient and the data range of γ_E. Nevertheless, by analyzing the other coefficients in (8), we find that higher R/L_{T_i} and R/L_{n_e} overall lead to a higher growth rate, and that higher $\hat{s}$ and τ (and γ_E although the coefficient is small in relation to the parameter range) overall lead to a lower growth rate.

TABLE III. Summary of the input-to-output behavior of the NeuralBranch model for the growth rate prediction sub-task, based on interpretation of Fig. 6.

Input parameter	Impact on growth rate	Interactions with other parameters
R/L_{T_i}	Higher R/L_{T_i} increases the growth rate.	Higher $\lvert\gamma_E\rvert$ weakens the impact of R/L_{T_i} on the growth rate, but only when $\hat{s}$ is large enough ($\hat{s} \gtrsim 0.5$).
$\hat{s}$	Higher $\hat{s}$ generally decreases the growth rate. However, as in the classification sub-task, the impact of $\hat{s}$ is not linear as its impact reaches a minimum at $0 < \hat{s} < 1$.	As in the classification case, the value of $\hat{s}$ that minimizes its impact on the output shifts from $\hat{s} \approx 0$ to $\hat{s} \approx 1$ as R/L_{n_e} increases. $\hat{s}$ also interacts with γ_E as discussed below.
γ_E	Higher $\lvert\gamma_E\rvert$ decreases the growth rate. However, this only occurs at specific conditions as described in the interaction column.	The impact of γ_E is completely suppressed at low $\hat{s}$ ($\lesssim$0.5). Moreover, the impact of γ_E is slightly weaker at lower R/L_{T_i}, even when $\hat{s}$ is sufficiently large.
R/L_{n_e}	Higher R/L_{n_e} increases the growth rate.	The impact of R/L_{n_e} on the growth rate is weakened at low $\hat{s}$.
τ	Higher τ decreases the growth rate.	No interactions with other inputs.

C. NeuralBranch model for growth rate predictions

Figure 6 shows the final architecture and the visualizations of the NeuralBranch model that predicts the growth rate, which yields $R^2 = 0.84$. This model almost matches the corresponding neural network used in the input selection test ($R^2 = 0.85$) and exceeds the linear classification model ($R^2 = 0.72$). We have summarized conclusions of our interpretation of the model in Table III. As in the classification case, we are able to identify several non-linear and interactive patterns, which explains why the NeuralBranch model outperforms the corresponding linear model. In Sec. VI D, we describe certain details regarding how we performed the interpretation for this sub-task.

D. Details on the interpretation process of the NeuralBranch growth rate prediction

As in the classification case, our interpretation is based on how the output of each neural branch depends on its two inputs, and on how the intermediate z-values propagate through the model. For instance, both neural branch 4 and neural branch 5 indicate straight equidistant contour lines of approximately the same inclination angle, which means that both these neural branches can be thought of as weighted addition operators. Therefore, in this case, it is straightforward to analyze how the output from prior neural branches propagate through the rest of the model. We also note that $\hat{s}$ needs to be present in two neural branches in order to achieve high prediction accuracy, as $\hat{s}$ interacts with different parameters in different ways. One of these interactions is indicated in neural branch 3, where we observe a similar interaction that was observed in the classification model, namely, that higher R/L_{n_e} shifts the impact that $\hat{s}$ has on z_3. The other strong interaction effect is observed in neural branch 1, where low $\hat{s}$ completely suppresses the impact that γ_E has on z_1. We also observe an interaction in neural branch 2, where the contour lines have a steeper inclination angle as R/L_{T_i} increases. This is an indication of a slight interaction effect between R/L_{T_i} and z_1, where z_1 represents the contribution from $\hat{s}$ and γ_E.

E. Predicted vs dataset values

In addition to the previously mentioned prediction accuracy of the NeuralBranch model $R^2 = 0.84$, Fig. 7 presents the predicted vs actual growth rate values in the test set, providing a more comprehensive overview. It can be observed that while most predictions are accurate, some exhibit significant errors. This is not surprising, as 10 out of the 15 input parameters are excluded in this model. Speculatively, although these parameters are less important on average across the dataset, they may hold significance for specific instances. That said, as was seen in the input selection test, the choice of excluding some of the input parameters is not the only reason for why the model is imperfect. Specifically, the R^2 value still deviates from 1 when all inputs are included (0.93), which is also the case in a previous study where a growth rate surrogate model for QuaLiKiz is created.[32]

F. Discussion of growth rate prediction results in relation to theory

We now discuss and compare the results of the growth rate prediction models with the corresponding analytical expressions in Sec. III. We limit the discussion to the five parameters that were most

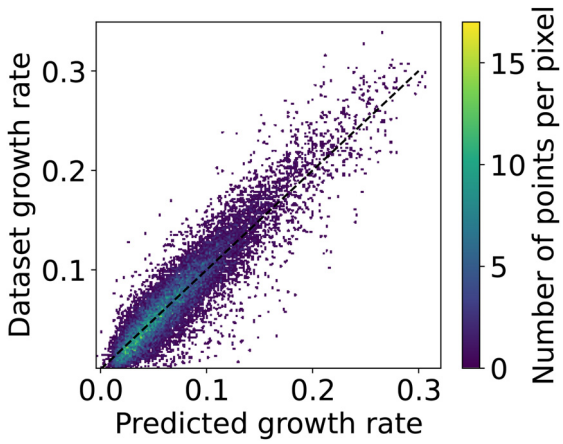

FIG. 7. The predicted vs dataset growth rate values for the NeuralBranch model that predicts the growth rate given that the mode is unstable. The brightness of the points indicate how many points there are in each pixel in the scatterplot. The dashed line represents the perfect prediction line.

important for the predictions, and therefore included in the models, namely, R/L_{T_i}, $\hat{s}$, γ_E, R/L_{n_e}, and τ.

As mentioned in Sec. III, since the deviation from the critical gradient affects the growth rate, it is not surprising that the parameters important for stable/unstable classification also were important for the growth rate prediction (all five except γ_E were important for the classification).

When analyzing the linear fit for growth rate predictions (8), we observe the same signs for the coefficients of the parameters compared to the classification case (7). This is partly in contradiction to the analytical expression for the growth rates in Eqs. (3) and (4). Specifically, both higher τ and higher $\hat{s}$ reduces the growth rate in the linear fit (and also in the NeuralBranch model), but this is not the case in the analytical expressions. A possible explanation is that the unstable entries in the dataset might be close to the critical threshold, which is likely due to the dataset being based on experimental data from JET pulses. In more detail, turbulent transport in tokamaks is stiff, meaning that, above the critical threshold, a small increase in the normalized gradients leads to a large increase in the fluxes. Hence, it is difficult to raise the gradients of the profiles far above the critical threshold. However, this is only a partial explanation for the observed coefficients, as the dataset also includes expansions in the gradient parameters beyond their critical values.

By more closely analyzing the coefficients in the linear models, we observe that τ is not as destabilizing in the growth rate prediction case compared to the classification (critical threshold) case. Specifically, the coefficient for τ is significantly stronger in relation to all other parameters in the classification case. This is an indication that the dependency that our models find in relation to τ might be a mixture of the analytical expression for the critical threshold and the analytical expressions in the limit far from the threshold.

We now continue to a discussion about the NeuralBranch model for the growth rate, which, as demonstrated, have identified parameters patterns that are too complicated for a linear model to capture. For instance, the model shows that the stabilization effect of ExB-shearing γ_E is absent at low magnetic shear $\hat{s}$. This can be explained by how QuaLiKiz handles the related eigenfunction. Specifically, as seen in Ref. 31, the eigenfunction in QuaLiKiz is a shifted Gaussian

$$\tilde{\phi} \sim \phi_0 \exp\left(-\frac{x - x_0}{2w^2}\right), \tag{9}$$

where ϕ is the electrostatic potential, w is the width, x is the distance from the mode surface, and x_0 is the shift. The ExB-shearing enters primarily through its impact on x_0, which can be seen in the derivations for w and x_0 in Ref. 31. Additionally, through investigating expressions for w, we can see that the magnetic shear dependency for the width is $w \sim 1/\hat{s}$. Hence, the eigenfunction is wider for lower shear, and therefore the shift x_0 (and consequently ExB shearing) is less impactful. The wider eigenfunction at low magnetic shear is expected as the ITG instability is created at the low field side and carried to the high field side along the magnetic field lines. Specifically, for high magnetic shear the structure of the instability is torn apart and therefore rather localized on the low field side. For a low magnetic shear the instability survives and get extended. This is not compatible with QuaLiKiz strong ballooning assumption,[48] which requires the mode to be localized. Hence, the low magnetic shear domain is more difficult to accurately represent with QuaLiKiz. A study for low magnetic shear has been performed and it was found that QuaLiKiz is valid down to magnetic shear 0.1.[45] Note that this study was performed with a different expression for the width of the eigenfunctions.

The other main pattern indicated by the NeuralBranch model, that is too complicated for the linear model to capture, is the interplay between a/L_{n_e} and $\hat{s}$ in Fig. 6(d). This is effectively the same pattern that was found in the classifier in Fig. 4(b), which again might be due to the influence of the TEM as previously discussed. Another possible explanation for the behavior at low shear that we also see in the classifier could also be, as discussed, related to how low shear impacts the eigenfunction.

VII. SUMMARY, CONCLUSION, AND FUTURE WORK

In this work, we used the NeuralBranch framework, an interpretable neural network framework, to create surrogate models for the growth rates from the QuaLiKiz eigenvalue solver. Our initial focus was on developing an interpretable surrogate model to classify whether the growth rate is positive (unstable mode) or not. Then, we developed an interpretable surrogate model to predict the actual growth rate given that the mode is unstable. The goal was to take advantage of the interpretability of these models to investigate how the classification and growth rate depend on the most significant input parameters, thereby providing insight into the model behavior users may anticipate when employing QuaLiKiz-based surrogates. Moreover, since our models were trained on QuaLiKiz data, they not only provide insight into the likely behavior of QuaLiKiz-based surrogates, but also, to some extent, into the behavior of the QuaLiKiz eigenvalue solver itself, something that was not feasible with previous black-box surrogate models. We limited the study to include only the ITG-mode growth rate at a specific normalized poloidal wavenumber, namely, $k_\theta \rho_s = 0.325$. As a secondary objective, we compared the patterns found in our models with analytical expressions from theory.

As a preparatory step, we investigated which input parameters were most important for each sub-task. For the classification, using only R/L_{T_i}, $\hat{s}$, τ, and R/L_{n_e} provided the same accuracy as when all 15 inputs were included, hence we limited the analysis to these four parameters. However, in the growth rate sub-task, no input could be removed without slightly reducing the accuracy. Nevertheless, we chose to focus the analysis of this sub-task on the five most important input parameters, namely, R/L_{T_i}, $\hat{s}$, γ_E, R/L_{n_e}, and τ, which together accounted for 91% of the prediction accuracy.

When proceeding to the NeuralBranch models, we observed that they outperformed linear models and matched black-box neural networks in accuracy, while simultaneously revealing how the most important input parameters affect both classification and growth rate predictions. For the growth rate prediction sub-task, we want to emphasize that the NeuralBranch model matched a black-box neural network that used only the same five inputs, and not a black-box neural network using all 15 available inputs.

The parameter dependencies indicated by the NeuralBranch models were summarized in Table II for the classification and in Table III for the growth rate prediction. In general, the models indicated several intricate parameter dependencies, one example being how the stabilizing effect of γ_E on the growth rate is completely suppressed at low $\hat{s}$, possibly due to the widening of the eigenfunction in QuaLiKiz when the magnetic shear is low.

While the main goal of providing a transparent overview of the most essential parameter relationships has been achieved, we

acknowledge that these findings may not be universally applicable. Specifically, the results presented here are valid only for the dataset used, which is based on JET pulses, and may also be somewhat influenced by correlations among the input parameters. Additionally, as this analysis focused on the growth rate at only one wavenumber, slight quantitative differences should be expected when considering other wavenumbers at the ion scale. That said, when doing further testing in predicting for a few other wavenumbers across the ion scale, we found the same qualitative patterns. Furthermore, while the NeuralBranch models presented are generally accurate, there are instances where they exhibit significant prediction errors. As such, the findings presented in this work should be interpreted as general trends rather than a perfect representation of the true behavior of the QuaLiKiz eigenvalue solver.

Future work could involve analyses similar to those presented in this study, but applied to data based on experimental values from other tokamaks or to data for other instability types, such as the ETG mode or the TEM. In a broader context, future research may also explore the application of interpretable machine learning methodologies to other models subject to surrogate modeling. This could be relevant even for computationally inexpensive models. While surrogate models offer less advantage in terms of computational speed for such cases, their interpretability could still provide deeper insights into the underlying behavior of these models.

In summary, as demonstrated, methods that enable interpretability can assist in providing deeper insights into the behavior of models like the QuaLiKiz eigenvalue solver and assist in making machine learning-based surrogate models more transparent and, therefore, more trustworthy.

ACKNOWLEDGMENTS

This work is a continuation of two B.Sc. thesis projects by Flyckt *et al.*[49] and Enström *et al.*[50] both conducted at Chalmers University of Technology under the supervision of A. Gillgren and A. Ludvig-Osipov. Additionally, the authors would like to thank Aaron Ho for valuable discussions and for providing the QuaLiKiz dataset. Moreover, 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. Finally, the authors would like to thank the Swedish Research Council who are funding this work under the diary No. 2020-05465.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

A. Gillgren: Conceptualization (lead); Data curation (lead); Formal analysis (equal); Investigation (lead); Methodology (lead); Project administration (lead); Software (lead); Supervision (equal); Validation (lead); Visualization (lead); Writing – original draft (lead). **E. Fransson:** Formal analysis (lead); Investigation (equal); Methodology (equal); Writing – original draft (equal). **A. Ludvig-Osipov:** Conceptualization (supporting); Methodology (supporting); Project administration (supporting); Supervision (equal); Writing – review & editing (lead). **W. Enström:** Investigation (equal); Methodology (supporting); Software (equal); Writing – review & editing (supporting). **L. Flyckt:** Investigation (equal); Methodology (supporting); Software (equal); Writing – review & editing (supporting). **M. Green:** Investigation (equal); Methodology (supporting); Software (equal); Writing – review & editing (supporting). **M. Kvartsén:** Investigation (equal); Methodology (supporting); Software (equal); Writing – review & editing (supporting). **Y. Liljegren:** Investigation (equal); Methodology (supporting); Software (equal); Writing – review & editing (supporting). **E. Olsson:** Investigation (equal); Methodology (supporting); Software (equal); Writing – review & editing (supporting). **A. Orthag:** Investigation (equal); Methodology (supporting); Software (equal); Writing – review & editing (supporting). **H. Wennberg:** Investigation (equal); Methodology (supporting); Software (equal); Writing – review & editing (supporting). **P. Strand:** Funding acquisition (lead); Resources (lead); Supervision (equal).

DATA AVAILABILITY

The data that support the findings of this study are openly available in zenodo at https://zenodo.org/records/7418108, Ref. 54.

APPENDIX A: TRAINING SPECIFICATIONS

We here present the machine learning details, including hyperparameters used when training the NeuralBranch models in this work.

- Activation function in all hidden nodes: ReLU.
- Activation function in output node of classification models: sigmoid.
- Activation function in output node of growth rate prediction models: linear.
- Number of hidden layers in each branch: 3.
- Number of nodes in each hidden layer: 64.
- Optimizer: Adam, with a learning rate of 0.001.
- Loss function for classifier: binary cross-entropy.
- Loss function for growth rate prediction model: mean squared error (mse).
- Batch size: 256.
- Epochs: 100, although this is not always reached as we implement early stopping when the loss function on a temporary validation set stops to decrease, with a patience of 10 epochs.
- Data normalization method: MinMax scaling, range [0,1], applied to the inputs, and also the output in the growth rate prediction sub-task.

These hyperparameters were chosen based on early-stage hyperparameter searches conducted during this work, as well as on those used in previous studies related to QuaLiKiz surrogate modeling.[21,32] Additionally, the dense neural networks used in this work to provide benchmark values for model performance and identify the most important input parameters share the same

hyperparameters as the NeuralBranch models. There is however a difference, namely, that since the NeuralBranch models consists of several branches of dense layers, these get more nodes in total compared to the dense neural networks. This might seem unfair when comparing the accuracy of NeuralBranch models and dense neural networks, but through testing we observe that additional nodes and layers do not lead to an improved performance for the dense neural networks. The number of nodes in the NeuralBranch models was not reduced to match the total number of nodes in the dense neural networks because we wanted to avoid making prior assumptions about the complexity of the functions each neural branch needs to learn. However, while the large total number of nodes might typically raise concerns about overfitting, the interpretability of the NeuralBranch models allows for straightforward detection of irregular or overly complex patterns, which are indicative of overfitting.[51] As demonstrated, the NeuralBranch models in this work learn patterns that are regular and not overly complex, leading us to conclude that overfitting is not a significant concern. That said, some of the branches presented, especially those that indicate simple learned patterns, could likely have been reduced further in terms of the number of nodes without sacrificing accuracy.

An additional note about the training process is that each model configuration was trained from scratch five times, and the best score among the five iterations is selected as the evaluation score for that configuration. This applies both to the NeuralBranch models when searching for the most accurate configuration and for neural networks that are used to find the most important input parameters. This strategy ensures that no configuration is overlooked due to rare instances where the trainable weights of the models converge to a local optimum,[51] resulting in a lower score.

APPENDIX B: WEAK VS STRONG TURBULENCE

In this work, we have focused on disentangling the dependencies of machine-learning-based surrogate models for the growth rate from the eigenvalue solver in QuaLiKiz. However, there are also metrics other than the growth rate that can be insightful regarding the impact on the turbulence; the ratio of the growth rate over the associated real frequency may determine if we have strong or weak turbulence,[36] and the comparison of the ratio of the growth rate over the poloidal wavenumber between ion and electron scales has been used to determine the impact of the electron temperature gradient mode.[52] In this section, we address one of these additional metrics, namely, the ratio of the growth rate over the real frequency.

There are two different states of turbulence which can be classified as weak turbulence (WT) and strong turbulence (ST).[53] An important difference between them is that the fluxes for WT scales with the electrostatic potential squared, $Q \propto |\phi|^2$, and the fluxes for the ST scale as the electrostatic potential, $Q \propto |\phi|$. Therefore, it is of interest to investigate which parameters that change the turbulence from one state to the other.

As mentioned, it has been shown that the ratio of the growth rate over the real frequency, $\gamma/|\omega_r|$, can indicate which state the turbulence is in, small values indicate WT and large values indicate ST. Hence, we have performed an additional input parameter selection for regression models predicting $\gamma/|\omega_r|$ using SFS, similar to the analysis in Sec. VI A. The five most prominent input parameters are presented in Fig. 8, and

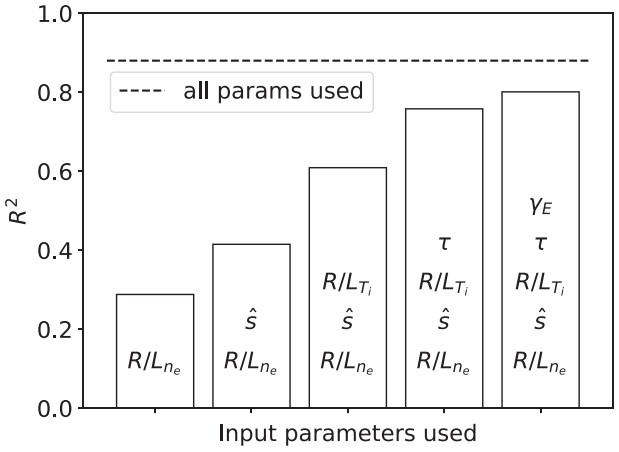

FIG. 8. Result of the SFS method used to find the most important inputs when predicting the ratio of the growth rate over the real frequency $\gamma/|\omega_r|$.

these are the same parameters as for the regression model for the growth rate. However, the normalized ion temperature gradient R/L_{T_i} is no longer the parameter which gives the highest accuracy when only one input parameter is allowed. This indicates that the real frequency has a positive dependency on R/L_{T_i}, since if both the growth rate and real frequency scale positively with R/L_{T_i}, it follows that R/L_{T_i} becomes less impactful on the ratio $\gamma/|\omega_r|$ compared to the case where we only predict the growth rate.

The relationship between the real frequency and R/L_{T_i} is linear in the large R/L_{T_i} limit. This can be derived by solving for the real frequency to the first-order perturbations in the equations from Ref. 45 (the cited article only displays result for the zeroth order). This is also in accordance with Ref. 37. Specifically, the linear dependency is found in a finite Larmor radius effects term. As previously shown in Sec. III B, the growth rate, both in the slab and interchange limits, is proportional to the square root of the normalized ion temperature gradient in the limit far from the critical threshold. Hence, $\gamma/|\omega_r| \sim 1/\sqrt{R/L_{T_i}}$ in this limit. However, close to the critical threshold the growth rate has a very strong dependency on the normalized ion temperature gradient.

When fitting a simple linear regression model to predict $\gamma/|\omega_r|$, we find poor performance ($R^2 = 0.41$). Because of the low score, we find this model too unreliable to analyze its fitting coefficients. However, we also used the NeuralBranch framework to predict $\gamma/|\omega_r|$. This model achieved $R^2 = 0.79$, which matched a black-box neural network. The architecture of the model was exactly the same as for the NeuralBranch model predicting the growth rate presented in the Fig. 6. Moreover, except that R/L_{T_i} had a weaker impact on the output, the impact of the other parameters, namely, R/L_{n_e}, $\hat{s}$, τ, and γ_E, were qualitatively the same as in the growth rate NeuralBranch model, see Table III. In this case, where R/L_{T_i} has less influence on the output compared to the growth rate prediction scenario, it is reasonable that the higher-capacity models significantly outperform the linear model. This is because the linear model can no longer rely as heavily on capturing a relatively simple relationship between the output and R/L_{T_i}. In other words, capturing the more complicated patterns that are related to the other parameters becomes more important when predicting $\gamma/|\omega_r|$.

In summary, this additional investigation suggests that similar parameter patterns that are present when predicting the growth rate alone also are important for predicting $\gamma/|\omega_r|$, which in turn suggests that these patterns have influence on the weak/strong turbulence limit.

REFERENCES

[1]X. Litaudon, F. Jenko, D. Borba, D. V. Borodin, B. J. Braams, S. Brezinsek, I. Calvo, R. Coelho, A. J. H. Donné, O. Embréus, D. Farina, T. Görler, J. P. Graves, R. Hatzky, J. Hillesheim, F. Imbeaux, D. Kalupin, R. Kamendje, H.-T. Kim, H. Meyer, F. Militello, K. Nordlund, C. Roach, F. Robin, M. Romanelli, F. Schluck, E. Serre, E. Sonnendrücker, P. Strand, P. Tamain, D. Tskhakaya, J. L. Velasco, L. Villard, S. Wiesen, H. Wilson, and F. Zonca, "EUROfusion-theory and advanced simulation coordination (E-TASC): Programme and the role of high performance computing," Plasma Phys. Controlled Fusion **64**, 034005 (2022).

[2]J. Garcia, R. Dumont, J. Joly, J. Morales, L. Garzotti, T. Bache, Y. Baranov, F. Casson, C. Challis, K. Kirov *et al.*, "First principles and integrated modelling achievements towards trustful fusion power predictions for JET and ITER," Nucl. Fusion **59**, 086047 (2019).

[3]J. Mailloux, N. Abid, K. Abraham, P. Abreu, O. Adabonyan, P. Adrich, V. Afanasev, M. Afzal, T. Ahlgren, L. Aho-Mantila *et al.*, "Overview of JET results for optimising ITER operation," Nucl. Fusion **62**, 042026 (2022).

[4]H.-T. Kim, F. Auriemma, J. Ferreira, S. Gabriellini, A. Ho, P. Huynh, K. Kirov, R. Lorenzini, M. Marin, M. Poradzinski *et al.*, "Validation of D–T fusion power prediction capability against 2021 JET D–T experiments," Nucl. Fusion **63**, 112004 (2023).

[5]T. Luda, C. Angioni, M. Dunne, E. Fable, A. Kallenbach, N. Bonanomi, P. Schneider, M. Siccinio, G. Tardini, ASDEX Upgrade Team, and EUROfusion MST1 Team, "Integrated modeling of ASDEX Upgrade plasmas combining core, pedestal and scrape-off layer physics," Nucl. Fusion **60**, 036023 (2020).

[6]D. Fajardo, C. Angioni, R. Dux, E. Fable, U. Plank, O. Samoylov, G. Tardini, and ASDEX Upgrade Team, "Full-radius integrated modelling of ASDEX Upgrade L-modes including impurity transport and radiation," Nucl. Fusion **64**, 046021 (2024).

[7]V. Ostuni, J. Artaud, G. Giruzzi, E. Joffrin, H. Heumann, and H. Urano, "Tokamak discharge simulation coupling free-boundary equilibrium and plasma model with application to JT-60SA," Nucl. Fusion **61**, 026021 (2021).

[8]F. Jenko and W. Dorland, "Nonlinear electromagnetic gyrokinetic simulations of tokamak plasmas," Plasma Phys. Controlled Fusion **43**, A141 (2001).

[9]J. Candy, E. Belli, and R. Bravenec, "A high-accuracy Eulerian gyrokinetic solver for collisional plasmas," J. Comput. Phys. **324**, 73–93 (2016).

[10]R. J. Hawryluk, "An empirical approach to tokamak transport," in *Physics of Plasmas Close to Thermonuclear Conditions*, edited by B. Coppi, G. G. Leotta, D. Pfirsch, R. Pozzoli, and E. Sindoni (Pergamon, 1981), pp. 19–46.

[11]G. Cenacchi and A. Taroni, "Jetto a free boundary plasma transport code," Report No. ENEA-RT-TIB–88-5 (ENEA, 1988).

[12]G. V. Pereverzev and P. Yushmanov, "ASTRA Automated system for transport analysis in a tokamak," Report No. IPP 5/98, 2002.

[13]M. Honda and A. Fukuyama, "Dynamic transport simulation code including plasma rotation and radial electric field," J. Comput. Phys. **227**, 2808–2844 (2008).

[14]D. P. Coster, V. Basiuk, G. Pereverzev, D. Kalupin, R. Zagórksi, R. Stankiewicz, P. Huynh, F. Imbeaux *et al.*, "The European transport solver," IEEE Trans. Plasma Sci. **38**, 2085–2092 (2010).

[15]M. Romanelli, G. Corrigan, V. Parail, S. Wiesen, R. Ambrosino, P. D. S. A. Belo, L. Garzotti, D. Harting, F. Köchl, T. Koskela *et al.*, "JINTRAC: A system of codes for integrated simulation of tokamak scenarios," Plasma Fusion Res. **9**, 3403023 (2014).

[16]O. Meneghini, S. P. Smith, P. B. Snyder, G. M. Staebler, J. Candy, E. Belli, L. Lao, M. Kostuk, T. Luce, T. Luda *et al.*, "Self-consistent core-pedestal transport simulations with neural network accelerated models," Nucl. Fusion **57**, 086034 (2017).

[17]K. L. van de Plassche, J. Citrin, C. Bourdelle, Y. Camenen, F. J. Casson, V. I. Dagnelie, F. Felici, A. Ho, S. Van Mulders, and J. Contributors, "Fast modeling of turbulent transport in fusion plasmas using neural networks," Phys. Plasmas **27**, 022310 (2020).

[18]A. Panera Alvarez, A. Ho, A. Järvinen, S. Saarelma, S. Wiesen, JET Contributors, and ASDEX Upgrade Team, "EuroPED-NN: Uncertainty aware surrogate model," Plasma Phys. Controlled Fusion **66**, 095012 (2024).

[19]S. Dasbach and S. Wiesen, "Towards fast surrogate models for interpolation of tokamak edge plasmas," Nucl. Mater. Energy **34**, 101396 (2023).

[20]O. Meneghini, G. Snoep, B. Lyons, J. McClenaghan, C. Imai, B. Grierson, S. Smith, G. Staebler, P. Snyder, J. Candy, E. Belli, L. Lao, J. Park, J. Citrin, T. Cordemiglia, A. Tema, and S. Mordijck, "Neural-network accelerated coupled core-pedestal simulations with self-consistent transport of impurities and compatible with ITER IMAS," Nucl. Fusion **61**, 026006 (2021).

[21]A. Ho, J. Citrin, C. Bourdelle, Y. Camenen, F. J. Casson, K. L. van de Plassche, H. Weisen, and J. Contributors, "Neural network surrogate of QuaLiKiz using JET experimental data to populate training space," Phys. Plasmas **28**, 032305 (2021).

[22]F. Felici, J. Citrin, A. Teplukhina, J. Redondo, C. Bourdelle, F. Imbeaux, O. Sauter, JET Contributors, and EUROfusion MST1 Team, "Real-time-capable prediction of temperature and density profiles in a tokamak using RAPTOR and a first-principle-based transport model," Nucl. Fusion **58**, 096006 (2018).

[23]J. Citrin, I. Goodfellow, A. Raju, J. Chen, J. Degrave, C. Donner, F. Felici, P. Hamel, A. Huber, D. Nikulin, D. Pfau, B. Tracey, M. Riedmiller, and P. Kohli, "TORAX: A fast and differentiable tokamak transport simulator in JAX," arXiv:2406.06718 (2024).

[24]R. Agarwal, L. Melnick, N. Frosst, X. Zhang, B. Lengerich, R. Caruana, and G. Hinton, "Neural additive models: Interpretable machine learning with neural nets," arXiv:2004.13912 (2020).

[25]S.-M. Udrescu and M. Tegmark, "AI Feynman: A physics-inspired method for symbolic regression," Sci. Adv. **6**, eaay2631 (2020).

[26]D. Angelis, F. Sofos, and T. Karakasidis, "Symbolic regression trends and perspectives," Artif. Intell. Phys. Sci. **30**, 3845–3865 (2023).

[27]S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)* (Curran Associates Inc., Red Hook, NY, 2017), pp. 4768–4777.

[28]M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision (ECCV)* (Springer, 2014), pp. 818–833.

[29]N. Nanda, L. Chan, T. Lieberum, J. Smith, and J. Steinhardt, "Progress measures for grokking via mechanistic interpretability," arXiv:2301.05217 (2023).

[30]C. Bourdelle, J. Citrin, B. Baiocchi, A. Casati, P. Cottier, X. Garbet, F. Imbeaux, and J. Contributors, "Core turbulent transport in tokamak plasmas: Bridging theory and experiment with QuaLiKiz," Plasma Phys. Controlled Fusion **58**, 014036 (2016).

[31]J. Citrin, C. Bourdelle, F. J. Casson, C. Angioni, N. Bonanomi, Y. Camenen, X. Garbet, L. Garzotti, T. Görler, O. Gürcan *et al.*, "Tractable flux-driven temperature, density, and rotation profile evolution with the quasilinear gyrokinetic transport model QuaLiKiz," Plasma Phys. Controlled Fusion **59**, 124005 (2017).

[32]E. Fransson, A. Gillgren, A. Ho, J. Borsander, O. Lindberg, W. Rieck, M. Åqvist, and P. Strand, "A fast neural network surrogate model for the eigenvalues of QuaLiKiz," Phys. Plasmas **30**, 123904 (2023).

[33]X. Garbet, P. Mantica, C. Angioni, E. Asp, Y. Baranov, C. Bourdelle, R. Budny, F. Crisanti, G. Cordey, L. Garzotti *et al.*, "Physics of transport in tokamaks," Plasma Phys. Controlled Fusion **46**, B557 (2004).

[34]J. Weiland, *Stability and Transport in Magnetic Confinement Systems* (Springer, New York, 2012).

[35]A. Gillgren, A. Ludvig-Osipov, D. Yadykin, and P. Strand, "Investigating pedestal dependencies at jet using an interpretable neural network architecture," Nucl. Fusion **65**, 056033 (2025).

[36]S. M. Vasil Bratanov and D. Hatch, "Transition from weak to strong turbulence in magnetized plasmas," New J. Phys. **21**, 043046 (2019).

[37]H. Nordman and J. Weiland, "Transport due to toroidal η_i mode turbulence in tokamaks," Nucl. Fusion **29**, 251 (1989).

[38]C. Bourdelle, X. Garbet, G. Hoang, J. Ongena, and R. Budny, "Stability analysis of improved confinement discharges: Internal transport barriers in Tore Supra and radiative improved mode in TEXTOR," Nucl. Fusion **42**, 892 (2002).

[39]B. Coppi and F. Pegoraro, "Theory of the ubiquitous mode," Nucl. Fusion **17**, 969 (1977).
[40]F. T. Romanelli, "Ion temperature-gradient-driven modes and anomalous ion transport in tokamaks," Phys. Fluids B **1**, 1018–1025 (1989).
[41]A. Di Siena, T. Görler, E. Poli, A. Bañón Navarro, A. Biancalani, R. Bilato, N. Bonanomi, I. Novikau, F. Vannini, and F. Jenko, "Nonlinear electromagnetic interplay between fast ions and ion-temperature-gradient plasma turbulence," J. Plasma Phys. **87**, 555870201 (2021).
[42]F. Jenko, W. Dorland, and G. Hammett, "Critical gradient formula for toroidal electron temperature gradient modes," Phys. Plasmas **8**, 4096–4104 (2001).
[43]T. Hahm and W. Tang, "Properties of ion temperature gradient drift instabilities in H-mode plasmas," Report No. PPPL-2565 [Princeton Plasma Physics Lab. (PPPL), Princeton, NJ, 1988].
[44]A. Casati, C. Bourdelle, X. Garbet, and F. Imbeaux, "Temperature ratio dependence of ion temperature gradient and trapped electron mode instability thresholds," Phys. Plasmas **15**, 042310 (2008).
[45]J. Citrin, C. Bourdelle, P. Cottier, D. Escande, Ö. D. Gürcan, D. Hatch, G. Hogeweij, F. Jenko, and M. Pueschel, "Quasilinear transport modelling at low magnetic shear," Phys. Plasmas **19**, 062305 (2012).
[46]A. Marcano-Cedeño, J. Quintanilla-Domínguez, M. G. Cortina-Januchs, and D. Andina, "Feature selection using sequential forward selection and classification applying artificial metaplasticity neural network," in *IECON 2010-36th Annual Conference on IEEE Industrial Electronics Society* (IEEE, 2010), pp. 2845–2850.
[47]J. Weiland, A. Jarmén, and H. Nordman, "Diffusive particle and heat pinch effects in toroidal plasmas," Nucl. Fusion **29**, 1810 (1989).
[48]C. D. Stephens, X. Garbet, J. Citrin, C. Bourdelle, K. L. van de Plassche, and F. Jenko, "Quasilinear gyrokinetic theory: A derivation of QuaLiKiz," J. Plasma Phys. **87**, 905870409 (2021).
[49]L. Flyckt, M. Green, E. Olsson, and A. Orthag, "Samband mellan input-och outputparametrar i QuaLiKiz-modellen," B.Sc thesis (Chalmers University of Technology, 2024).
[50]W. Enström, M. Kvartsén, Y. Liljegren, and H. Wennberg, "Konstruktion av tolkbara neurala nätverk för analys av QuaLiKiz-modellen," B.Sc thesis (Chalmers University of Technology, 2024).
[51]I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016).
[52]P. Mantica, N. Bonanomi, A. Mariani, P. Carvalho, E. Delabie, J. Garcia, N. Hawkes, T. Johnson, D. Keeling, M. Sertoli, G. Staebler, G. Szepesi, D. Taylor, A. Thorman, and J. Contributors, "The role of electron-scale turbulence in the jet tokamak: Experiments and modelling," Nucl. Fusion **61**, 096014 (2021).
[53]Y. Z. Zhang and S. M. Mahajan, "Correlation theory of a two-dimensional plasma turbulence with shear flow," Phys. Fluids B **5**, 2000–2020 (1993).
[54]H. Aaron (2021). "QuaLiKiz-v2.6.2 linear instability spectra based on JET experimental plasma profiles," Zenodo. https://zenodo.org/records/7418108