PAPER • **OPEN ACCESS**

Optimisation of physics-informed neural network architecture and training for tokamak equilibrium reconstruction

To cite this article: Novella Rutigliano *et al* 2026 *Plasma Phys. Control. Fusion* **68** 045002

Plasma Physics and Controlled Fusion

OPEN ACCESS

RECEIVED
3 December 2025

REVISED
11 March 2026

ACCEPTED FOR PUBLICATION
19 March 2026

PUBLISHED
1 April 2026

PAPER

Optimisation of physics-informed neural network architecture and training for tokamak equilibrium reconstruction

Novella Rutigliano[1,*], **Andrea Murari**[2,3], **Pasquale Gaudio**[1], **Michela Gelfusa**[1] and **Riccardo Rossi**[1]

[1] Dipartimento di Ingegneria Industriale, Università degli Studi di Roma 'Tor Vergata', Via del Politecnico 1, 00133 Rome, Italy
[2] Consorzio RFX (CNR, ENEA, INFN, Università di Padova, Acciaierie Venete SpA), Corso Stati Uniti 4, 35127 Padova, Italy
[3] Istituto per la Scienza e la Tecnologia dei Plasmi, CNR, Padova, Italy
[*] Author to whom any correspondence should be addressed.

E-mail: novella.rutigliano@uniroma2.it

Keywords: physics-informed neural network, neural network optimisation, tokamak, multi-diagnostics equilibrium reconstruction, inverse problem

Abstract

Reconstructing the plasma state is a central challenge in nuclear fusion experiments, as it is essential for understanding and predicting plasma behaviour. Physics-informed neural networks (PINNs), especially when combined with a multi-diagnostic approach, offer powerful advantages for addressing this problem. PINNs embed the governing physical laws directly into the learning process through differential equation constraints, enabling them to integrate sparse or noisy measurements while maintaining physical consistency. This makes them particularly suitable for equilibrium reconstruction, where they can incorporate diagnostic data as boundary conditions and naturally enforce the structure of the magnetohydrodynamic equations. Moreover, the use of multiple diagnostics helps over-constrain the system, reducing uncertainties and mitigating the ill-posedness characteristic of the plasma core region. Starting from results obtained in previous works [1] where the capabilities of multi-diagnostics equilibrium reconstruction through PINNs were demonstrated, in this work we perform several parametric studies to optimise both the neural network architecture and the training procedure. We examine the impact of automatically adjusting the relative weights between data and physics losses during training, the role of specialised physics-based network's layers informed by prior knowledge or plasma state hypotheses, the choice of hidden layers' activation function, and the benefits of initialising training from a pre-trained network. These analyses provide guidelines for designing the most effective neural network and training strategy for specific plasma conditions.

1. Introduction

In magnetic confinement nuclear fusion solving the plasma state is an inverse problem aiming at inferring plasma internal property fields (such as temperature, density, magnetic flux and magnetic field) from external measurements, which are the problem's boundary conditions.

In tokamaks, the interpretation of many diagnostics requires the solution of inverse problems, particularly to derive local internal information from external measurements. Some examples are tomography [1–6], the determination of current density profile [7], the determination of free parameters for transport models [8, 9], some measurements interpretation [10], equilibrium reconstruction [11–14], and many others.

This work is an advancement of a methodology developed through physics-informed neural networks (PINNs) for solving the equilibrium reconstruction inverse problem in a tokamak. Equilibrium reconstruction is usually formalised through the set of partial differential equations (PDEs) denoted as magnetohydrodynamics (MHD) equations (see section 2).

PINNs were for the first time presented and used by Raissi *et al* in 2019 [15] as PDEs solvers. Indeed, in recent years, neural networks have become a powerful tool for modelling complex systems, particularly in contexts where large amounts of data are available and traditional model-based approaches face limitations. Transformations operated by neural networks are non-linear. For this reason, thanks to the universal approximation theorem, if the network is characterised by enough parameters θ, then it can be considered deep and it can approximate any function $F(x,\theta) = y$, where y is the target and x is the input space [16].

However, the direct application of purely data-driven neural networks to physical problems often suffers from poor generalisation, lack of interpretability, and the need for extensive training data, which are not always accessible in plasma physics.

By a computational point of view, as many PDEs, magnetohydrodynamic equations are traditionally solved through numerical methods (finite volumes, finite differences, finite elements). However, the established techniques rely on the domain discretisation with an impact on the results accuracy: for example, in finite difference method, partial derivatives are approximated, and the step size directly determine the accuracy and the computational cost of the method.

To overcome these limitations, the concept of PINNs has emerged as a novel paradigm that integrates physical laws directly into the learning process. Instead of treating the governing equations as external constraints applied after training, PINNs embed them into the network's loss function, guiding the optimisation of the model parameters.

PINNs core idea is to embed additional terms that enforce the satisfaction of governing physical laws in the neural network loss function. By embedding these constraints directly into the training process, PINNs extend the capabilities of standard machine learning approaches, opening the way to applications that were previously out of reach. A notable advantage is their ability to generalise beyond the available training data: physics-based loss functions act as a form of regularisation, ensuring that the model respects fundamental constraints even in regions with sparse or missing observations [17, 18]. Since their introduction, PINNs have proven to be highly effective for addressing both forward and inverse problems [19]. They have been applied successfully to areas such as numerical simulations of the Navier–Stokes equations, tomographic reconstruction, and multi-physics modelling [20–22]. However, PINNs should not be seen as simple replacements for classical solvers. Instead, they are characterised by several interesting features, including mesh-less approach, domain decomposition, and flexible handling of coupled physical processes [23–25]. Moreover, PINNs can incorporate experimental data, which is often noisy, alongside physics-based information such as numerical simulations. This hybrid capability of combining data-driven learning with physics-driven constraints allows for direct modelling of PDEs and enhances the processing of diagnostic signals. Such integration has already enabled innovative results, including super-resolution and improved data assimilation [23, 26–28].

In nuclear fusion plasma physics and tokamaks inverse problems, PINNs have already been used for facing several problems [21, 29–35], even if they are still considered a quite young methodology.

An advanced inversion methodology for equilibrium reconstruction based on PINNs has been developed in [29, 36], demonstrating the capability of PINNs to reconstruct plasma equilibria by constraining the solution with a high-fidelity physics-based model of the interferometer–polarimeter system. This approach accounts not only for the non-linearity between Faraday rotation and Cotton–Mouton phase shift (the so-called 'full Stokes' treatment), but also for the hot-plasma assumption, which will be particularly relevant in future tokamaks such as ITER and DEMO. However, the focus in [29] was the modelling of the diagnostics to include their non-linearities, whereas a detailed optimisation and parametric analysis of the model was out of the scope of that work.

Therefore, this work aims to provide a systematic and parametric analysis of the networks, to describe and assess possible alternatives, and to discuss under which conditions one solution may be more efficient than another. More specifically, the work presents the following four main studies:

- A description of the adaptive weighting scheme and its comparison with fixed weighting.
- An investigation of different neural network architectures, with a focus on hard and soft physics constraints.
- An investigation of various activation functions for the hidden layers.
- An assessment of the effects of pre-trained neural networks compared to randomly initialised ones.

The next section introduces the general methodology, i.e., the concept behind the PINNs, their training process, and loss formulation. More detailed information related to each parametric analysis is provided in the corresponding results sections. Finally, the discussion and conclusions are presented in the last section.

2. Methodology

In this Section, before going into the details of every methodological parametric analysis conducted, a general introduction to TokaLab, the synthetic scenario used for the analysis, to the physical equations used and to the PINN training algorithm are provided.

2.1. Plasma equilibrium reconstruction

In nuclear fusion, plasma equilibrium reconstruction is based on the ideal, steady-state, static (negligible velocity) single fluid MHD equations, that can be written as:

$$\begin{aligned} \nabla \cdot \mathbf{B} &= 0 \\ \mu_0 \boldsymbol{J} &= \nabla \times \boldsymbol{B} \\ \boldsymbol{J} \times \boldsymbol{B} &= \nabla p \end{aligned} \tag{1}$$

where B is the magnetic field, $\boldsymbol{J}$ the plasma density current and p the plasma pressure. Reconstructing the plasma equilibrium means to solve an inverse problem where the plasma quantities of interest (density current, magnetic field and pressure) are inferred from measurements, behaving as boundary conditions in the inverse problem.

We define the system's frame of reference using the following cylindric geometric coordinates: R (radial), z (vertical), and ϕ (toroidal).

As the toroidal symmetry assumption ($\partial/\partial\phi = 0$) is typically assumed valid in tokamaks, the MHD equations can be written in their scalar form as:

$$\begin{aligned} \nabla \cdot \boldsymbol{B} = 0 &\rightarrow \frac{\partial B_R}{\partial R} + \frac{\partial B_z}{\partial z} = 0 \\ \mu_0 \boldsymbol{J} = \nabla \times \boldsymbol{B} &\rightarrow \begin{bmatrix} J_R = -\frac{1}{\mu_0}\frac{\partial B_\phi}{\partial z} \\ J_\phi = \frac{1}{\mu_0}\left(\frac{\partial B_R}{\partial z} - \frac{\partial B_z}{\partial R}\right) \\ J_z = \frac{1}{\mu_0}\left(\frac{B_\phi}{R} + \frac{\partial B_\phi}{\partial R}\right) \end{bmatrix} \\ \boldsymbol{J} \times \boldsymbol{B} = \nabla p &\rightarrow \begin{bmatrix} J_\phi B_z - J_z B_\phi = \frac{\partial p}{\partial R} \\ J_R B_z - J_z B_R = 0 \\ J_R B_\phi - J_\phi B_z = \frac{\partial p}{\partial z} \end{bmatrix}. \end{aligned} \tag{2}$$

Under the introduced assumptions, the aforementioned equations can be manipulated to derive the Grad–Shafranov equation [37, 38] that is the standard equation used for equilibrium reconstruction in tokamaks:

$$\Delta^* \Psi = -\mu_0 (2\pi R)^2 \frac{\mathrm{d}p}{\mathrm{d}\Psi} - (2\pi)^2 F \cdot \frac{\mathrm{d}F}{\mathrm{d}\Psi} \tag{3}$$

where Ψ is the poloidal flux (sometimes the reduced poloidal flux $\psi = \Psi/2\pi$ is used), F is equal to $B_\phi R$ and the elliptical operator $\Delta^* \Psi = \frac{\partial^2 \Psi}{\partial z^2} - \frac{1}{R}\frac{\partial \Psi}{\partial R} + \frac{\partial^2 \Psi}{\partial R^2}$ has been introduced.

In this work we mainly use the MHD equations in the form presented through the equations (1) and (2). This should be considered the standard except for the analyses in which the use of the Grad–Shafranov equation is explicitly mentioned.

2.2. Scenario presentation: TokaLab geometry and diagnostics

In this work a synthetic scenario is used for which the ground truth is known allowing for a direct and simple evaluation of algorithm performances in reconstructing target fields.

TokaLab is an open access virtual tokamak for education and research [39]. It is designed to provide a flexible geometry easily modifiable. In this work, TokaLab standard scenario is used for most of the

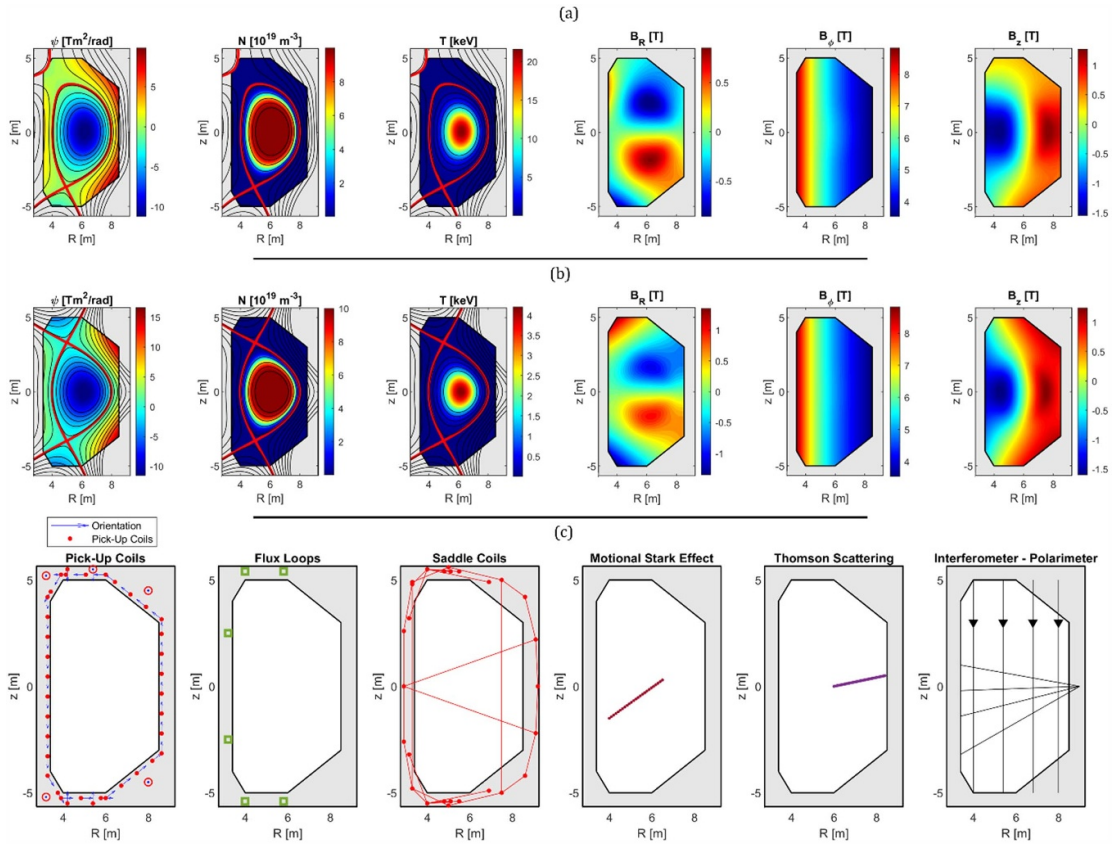

Figure 1. (a) TokaLab standard scenario fields (SN), (b) TokaLab double null (DN) scenario fields, (c) TokaLab diagnostics schematic representation. PU = pick up coils, FL = flux loops, SL = saddle loops, TS = Thomson scattering, IP = interferometer–polarimeter, MSE = motional Stark effect.

parametric analyses. Since for the transfer learning analysis (section 3.4) the double null (DN) configuration is mentioned, in table 1 the geometric parameters for both the single null (SN) standard scenario and the DN are reported. These scenario configurations and TokaLab diagnostics set are shown in figure 1.

$$J_{\phi,\text{profile}}(\psi_n) = \left(\beta_0 \frac{R}{R_0} + \left((1-\beta_0)\frac{R_0}{R}\right)\right)\left(1 - \left(\frac{\psi_n - \psi_{n,\text{peak}}}{1-\psi_{n,\text{peak}}}\right)^{\alpha_1}\right)^{\alpha_2}$$
$$J_\phi = J_{\phi,\text{profile}} \frac{I_p}{\int J_{\phi,\text{profile}} \mathrm{d}S}. \tag{4}$$

TokaLab equilibria have been obtained using a finite difference algorithm to solve the Grad–Shafranov equation (3), where the plasma boundary (last closed flux surface, LCFS) is imposed using the parametric function detailed in [40]. In the numerical simulation, a pressure of 100 Pa at the LCFS is imposed.

After solving the Grad–Shafranov equation through the finite difference method, all the fields of interest are then derived from the MHD relations (equations (1) and (2)). For what concerns the plasma density and temperature, the single fluid assumption is made ($N_e = N_i = N$ and $T_e = T_i = T$, where N_e, T_e, N_i and T_i are the electron density, electron temperature, ion density and ion temperature, respectively). Under this approximation, the plasma pressure is $p = 2eNT$, where e the electron unity charge (the unit of temperature is the eV).

The electron density is subsequently calculated using a typical radial function on the normalised flux surface (ψ_n):

$$N_e(\psi_n) = (N_{e,\text{axis}} - N_{e,\text{LCFS}})(1 - \psi_n^{\alpha_1})^{\alpha_2} + N_{e,\text{LCFS}} \tag{5}$$

where $N_{e,\text{axis}}$ is the electron density on the magnetic axis while $N_{e,\text{LCFS}}$ is the density on the LCFS. These quantities are introduced in table 1. It follows that the temperature is equal to:

$$T_e = \frac{p}{2eN_e}. \tag{6}$$

Table 1. TokaLab scenarios parameter for the single null (SN) and the double null (DN). The J_ϕ parameters are quantities used to simulate the plasma toroidal density current (4). The relation used is a modified version of a standard method [40] which allows to displace the peak. The notation [29] is adopted to indicate unitless numbers (dimensionless quantities).

	SN	DN
Major radius R_0 [m]	6	6
Minor radius a [m]	2	2
Vertical position z_0 [m]	0	0
Elongation—upper k_1 [1]	1.7	1.7
Elongation—bottom k_2 [1]	2	1.7
Triangularity—upper d_1 [1]	0.5	0.5
Triangularity—bottom d_2 [1]	0.5	0.5
Angle—upper left $\gamma_{n,1}$ (rad)	0	$\pi/3$
Angle—upper right $\gamma_{p,1}$ [rad]	0	$\pi/6$
Angle—bottom left $\gamma_{n,2}$ [rad]	$\pi/3$	$\pi/3$
Angle—bottom right $\gamma_{p,2}$ [rad]	$\pi/6$	$\pi/6$
J_ϕ parameter 1—β_0 [1]	0.5	0.5
J_ϕ parameter 2—α_1 [1]	2	2
J_ϕ parameter 3—α_2 [1]	2	2
J_ϕ parameter 4—$\psi_{n,peak}$ [1]	0	0
Plasma current I_p [MA]	12	12
Toroidal magnetic field B_ϕ [T]	5	5
Electron density in $\psi_n = 0$ $N_{e,axis}$ $[m^{-3}]$	10^{20}	10^{20}
Electron density at the separatrix $N_{e,\,LCFS}$ $[m^{-3}]$	10^{17}	10^{17}
$N_e(\psi_n)$ parameter 1—γ_1 [1]	6	6
$N_e(\psi_n)$ parameter 2—γ_2 [1]	3	3

Table 2. Number of diagnostics and measurements uncertainties used in each case. (The acronyms stand for: PU = pick-up coils, FL = flux loops, SL = saddle loops, TS = Thomson scattering, LID = interferometry line integrated density, POL-Far = polarimetry—Faraday rotation, POL-CM = polarimetry—Cotton–Mouton phase shift. MSE = motional Stark effect).

	PU	FL	SL	TS	LID	POL-Far	POL-CM	MSE
# measurements	42	6	30	60	8	8	8	25
σ	0.05 T	0.05 Wb rad^{-1}	0.05 Wb rad^{-1}	(N_e) 10^{18}m^{-3}	10^{18}m^{-2}	3.5 mrad	3.5 mrad	1.7 mrad

Once all the synthetic plasma quantities are computed, it is possible to model the diagnostics and, so, to simulate the measurements.

The simulated plasma fields of interest are shown figures 1(a) and (b) (respectively for the standard SN and DN scenarios).

For what concerns TokaLab diagnostics shown in figure 1(c), all the details of the modelling are reported in appendix A. The loss terms for both physics and data are provided in appendix B.

To simulate the diagnostics realistically in the synthetic scenario, normally distributed random noise has been added to the ideal measurements. Both the location and the noise uncertainties for the TokaLab diagnostics are Joint European Torus-inspired [41–43]. Table 2 displays the number and the set of diagnostics used for the PINN algorithm. The acronyms for the various diagnostics are defined in the table caption.

2.3. PINN training algorithm

The schematic representation of the PINN training algorithm is shown in figure 2(b), while panel (a) shows the reference neural network architecture of this work.

Referring first to the reference network architecture (figure 2(a)), the inputs for the PINN are the coordinates R and z, in which the plasma needs to be assessed. The network predicts four quantities: the potential vector A_ϕ, the toroidal magnetic field component, B_ϕ, the electron density N_e and the electron temperature T_e. The experimental measurements of these quantities have numerical values differing of orders of magnitude and therefore they are normalised, as required to ensure a proper training. The network outputs are then rescaled to provide physically meaningful estimates. In more detail, let us define the vector of outputs (network predictions) y made of four components and y_4, one for each network

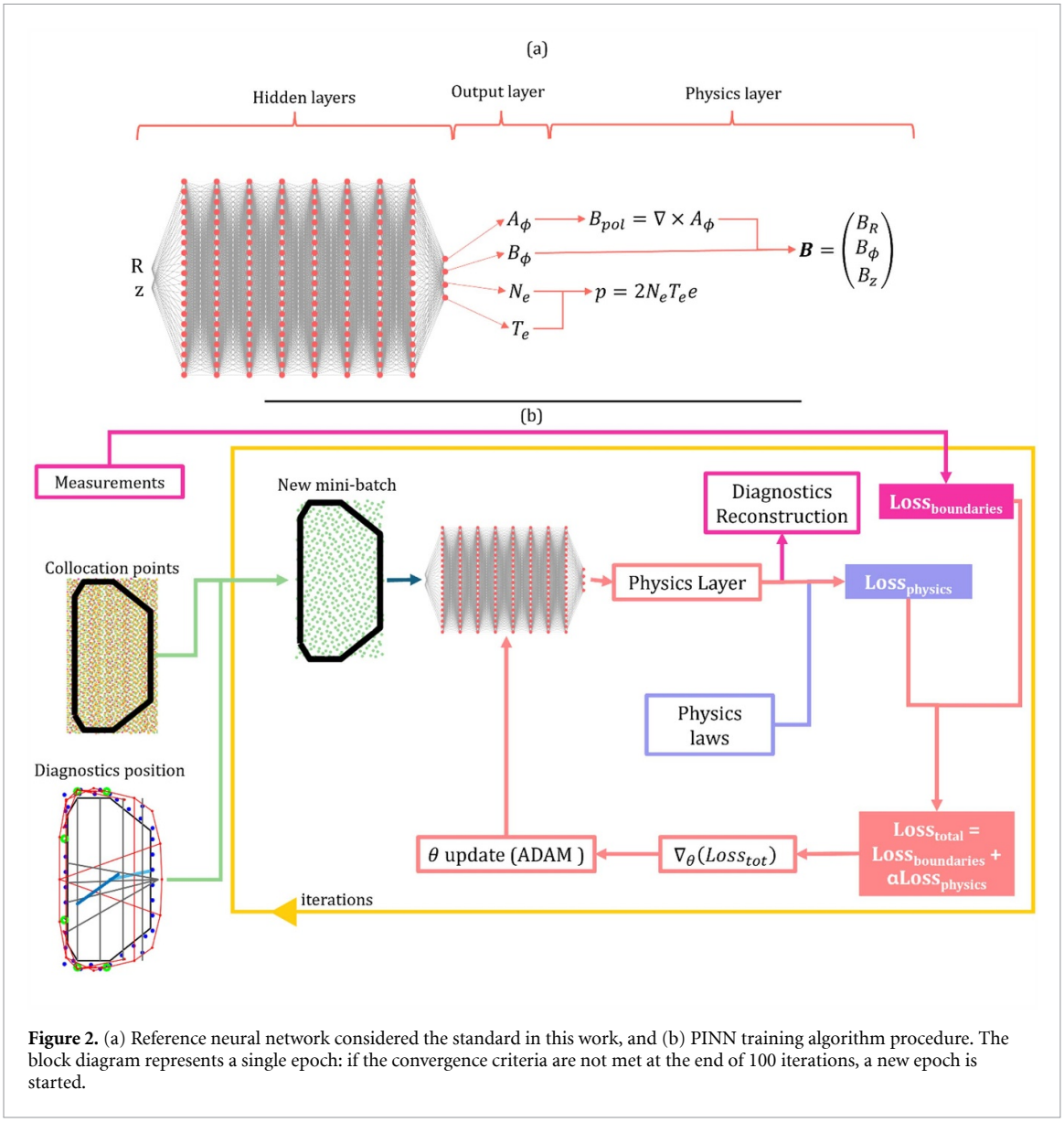

Figure 2. (a) Reference neural network considered the standard in this work, and (b) PINN training algorithm procedure. The block diagram represents a single epoch: if the convergence criteria are not met at the end of 100 iterations, a new epoch is started.

output. Plasma quantities predicted from the network are obtained as reported below:

$$\begin{bmatrix} A_\phi \\ B_\phi \\ N_\mathrm{e} \\ T_\mathrm{e} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 10^{19} \\ 10^{3} \end{bmatrix} \left\{ Y \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} + Y \begin{bmatrix} 0 \\ 0 \\ \frac{e^{y_3}}{1+e^{0.9y_3}} \\ \frac{e^{y_4}}{1+e^{0.9y_4}} \end{bmatrix} \right\}. \tag{7}$$

The second term of the sum ensures the non-negativity of the electron density and temperature (y_3 and, respectively).

After this manipulation of the network outputs, a physics layer is introduced to compute the poloidal magnetic field $\boldsymbol{B_{pol}} = (B_R, B_z)$ from the vector potential A_ϕ and a second physics layer is used to evaluate the plasma pressure $p = 2N_\mathrm{e}eT_\mathrm{e}$.

Regarding the architecture, the reference neural network is a fully-connected network with 8 hidden layers with 20 neurons each. The activation function in the reference scenario is the tanh for all hidden layers. The iterative approach through which the network parameters are opportunely tuned is called 'training': its algorithm is schematically represented in figure 2(b). The tuning of the network's parameters relies on the minimisation of a loss term, based on the residuals, the differences between the estimates and the data, and on the quality of fit of the physical equations.

To determine the loss term associated to the data it is necessary to evaluate the diagnostics values given the reconstructed fields in order to compare the reconstructed measurements with the original ones. In our model, the term 'loss boundary' refers to the sum of the losses that constrain the PDE

problem and are characterised by uncertainty. A complete list of the contributions to the boundary loss is provided in appendix A: Diagnostics Modelling. The loss term of the physics ($\text{Loss}_{\text{physics}}$) is evaluated on the so-called collocation points (generated following a Sobol' distribution [44, 45]). The two main terms $\text{Loss}_{\text{boundaries}}$ and $\text{Loss}_{\text{physics}}$ are then summed up:

$$\text{Loss}_{\text{total}} = \text{Loss}_{\text{boundaries}} + \alpha \text{Loss}_{\text{physics}} \tag{8}$$

where α is a suitable weighting factor. The impact of the selection of the hyperparameter α is discussed in section 3.1. The loss functions formulation is the core aspect of the algorithm setting. The definition of all the loss terms is discussed in appendix B: losses formulation. In the same way, the description of the diagnostic models is provided in appendix A: diagnostics modelling.

Coming back to the training scheme, at each iteration (yellow box in figure 2(b)), the points used for the reconstruction if the diagnostics (their location) and a subset (called mini-batch) of the collocation points are taken as the neural network input. The network returns the predicted plasma quantities and the loss terms and the gradient of the loss terms with respect to the network parameters ($\nabla_\theta \text{Loss}_{\text{total}}$) are calculated.

The network parameters are updated following the direction that minimises the $\text{Loss}_{\text{total}}$ (this is called the 'descend gradient approach') through the ADAM algorithm [46] and a new iteration begins. Each epoch is composed of 100 iterations. At the end of each epoch, an evaluation of the satisfaction of the stop condition is made. If the criteria are reached, the training is considered complete. The stop condition in this work is the maximum number of epochs, namely 5000. Other stop conditions can be set defining a desired accuracy or a threshold for the loss gradient.

The minibatch size of the collocation points is equal to 1000, and therefore 100 000 collocations points have been generated using the Sobol sequence. A decay learning rate (the step size of the parameters' update) has been used:

$$\text{Learning Rate} = \frac{\text{LearningRate}_0}{1 + \text{iteration LearningDecayRate}} \tag{9}$$

where $\text{LearningRate}_0 = 10^{-3}$ and $\text{LearningDecayRate} = 10^{-4}$.

3. Results

3.1. Adaptive weighting scheme

The loss weighting, or balancing, plays a fundamental role in determining the accuracy of a PINN-based solver. For the total loss, we introduced the following specific formulation:

$$\text{Loss}_{\text{total}} = \frac{\text{Loss}_{\text{boundaries}} + \alpha \text{Loss}_{\text{physics}}}{1 + \alpha} \tag{10}$$

in order to avoid unbalanced losses between successive epochs and to smooth the convergence of the neural network with respect to the standard formulation of PINNs given in (8). The choice of α determines the relative importance of the physics loss with respect to the data loss. Unless differently specified, α is adaptive, meaning that its value changes to steer the network towards a better result.

In this paper, the total loss ($\text{Loss}_{\text{total}}$) is consistently computed with equation (10). Most studies on adaptive weighting focus on balancing $\text{Loss}_{\text{boundaries}}$ and $\text{Loss}_{\text{physics}}$ for ideal or noise free boundaries [47–49]. However, in equilibrium reconstruction tasks, the measurements are typically affected by non-negligible noise, which plays an important role in determining the quality of the reconstruction.

If α is too small, the PINN will disregard the physical constraints and overfit the diagnostics, with error propagation affecting the entire reconstruction. Conversely, if the physics weight is too large, the neural network may converge to a trivial solution of the physical equations (for example, all fields equal to zero), corresponding to a problem with no boundary conditions. Conceptually, these two limiting cases can be described as follows:

$$\begin{aligned} \text{Loss}_{\text{total}\,\alpha\to 0} &= \lim_{\alpha\to 0} \frac{\text{Loss}_{\text{boundaries}} + \alpha \text{Loss}_{\text{physics}}}{1+\alpha} \to \text{Loss}_{\text{boundaries}} \\ \text{Loss}_{\text{total}\,\alpha\to \infty} &= \lim_{\alpha\to \infty} \frac{\text{Loss}_{\text{boundaries}} + \alpha \text{Loss}_{\text{physics}}}{1+\alpha} \to \text{Loss}_{\text{physics}}. \end{aligned} \tag{11}$$

Therefore, it is essential to find a suitable trade-off for α, which typically requires a computationally intensive parametric analysis. On the other hand, as shown in this work, a fixed α limits the accuracy of the PINN, regardless of its value, and therefore an adaptive scheme should be implemented.

In a previous study [50], we demonstrated that for noisy data there exists an adaptive scheme that generally provides very good performance without the need for parametric analysis. The only soft requirement is that the uncertainty of the measurements is known. If this is the case, one can assume that the data loss has a statistical target. Suppose we have M measurements $y_{b,j}$, each characterised by an ideal value $y_{0,j}$ and affected by an uncertainty σ_j. The predicted values of the PINNs, $y_{p,j}$ must be compared with the measurements (since the ideal values are unknown). Writing the loss in the form of the normalised mean squared error and assuming that the predicted values converge towards the ideal ones (the boundary conditions or measurements), we obtain:

$$\mathrm{Loss}_{\mathrm{boundaries}} = \frac{1}{M}\sum \frac{\left(y_{p,j} - y_{b,j}\right)^2}{\sigma_j^2} = \frac{1}{M}\sum \frac{\left(y_{p,j} - y_{o,j} - \varepsilon_j\right)^2}{\sigma_j^2} \to \frac{1}{M}\sum \frac{\varepsilon_j^2}{\sigma_j^2} \to 1 \tag{12}$$

where ε_j is the noise affecting $y_{b,j}$ (the difference between the real noisy value and the ideal one: $y_{b,j} = y_{0,j} + \varepsilon_j$).

Therefore, the ideal target of the data loss is 1 ($\mathrm{Loss}_{\mathrm{target}}$). If the data loss drops below one, this indicates that the difference between the measured and predicted values is statistically smaller than the noise, meaning that the PINN is overfitting. Conversely, a much larger value implies that the predictions deviate significantly from the statistical distribution of the measurements. However, for reasons of convergence speed and to account for possible offsets and outliers, a target value of 1.5 has been found to be a good alternative, especially when a multi diagnostic approach is used.

Since the target of the data loss is known, the idea is to implement an adaptive scheme that aims to reach this value and, once achieved, to increase the weight of the physical loss. The adaptive scheme implemented here is based on the following system of equations:

$$\begin{aligned}
\left(\frac{\Delta\alpha}{\alpha}\right)_{\mathrm{epoch}} &= 0.1\tanh\left(-10\left(\frac{\mathrm{Loss}_{\mathrm{boundaries}}\left(\mathrm{epoch}\right)}{\mathrm{Loss}_{\mathrm{target}}} - 1\right)\right) \\
\alpha_{\mathrm{epoch}} &= \alpha_{\mathrm{epoch}} - 1\left(1 + \left(\frac{\Delta\alpha}{\alpha}\right)_{epoch}\right) \\
\alpha_{\mathrm{epoch}} &= \max\left(\alpha_{\mathrm{epoch}}, \alpha_{\mathrm{min}}\right)
\end{aligned} \tag{13}$$

where $\alpha_{\mathrm{min}} = 0.01$.

A hyperbolic tangent has been selected as the adaptive function because it exhibits a steep gradient, which is useful to avoid stagnation when updating α.

A parametric investigation was performed using different fixed weights and the adaptive weighting scheme in the framework of the TokaLab scenario introduced in section 2 and following the training procedure described in this section and in section 2. Diagnostic models and loss functions are described in appendices A and B. Figure 3 (left) shows the boundary loss versus the physics loss using the static weighting scheme for four different values of α. The case with the lowest value of α (0.1) reveals that the PINN attempts to minimise the data error below the ideal limit of one, indicating overfitting. It reaches a data loss of about 0.7 and cannot decrease further because of the unsatisfied physics equations imposed by the physics loss (α is not exactly zero), the PINN architecture (which implements some physical constraints implicitly, as mentioned in the methodology section and detailed later), and the multi diagnostic constraints.

By increasing α to 100, the performance improves, but the network still exhibits partial overfitting (minimum boundary loss around 0.85), and the physics loss remains at approximately 2×10^{-3}. To select the most suitable model, the Pareto front approach is graphically useful. It is the distribution of physics and loss boundaries, forming a 'frontier' of potential optimal compromises [51]. In a multi-objective optimisation problem, a Pareto front represents the set of solutions that are theoretically equally valid but representing different trade-offs between the desired goals (minimising the residuals on physics equations or on boundary conditions).

Using a higher weight (5×10^3), the PINN reaches a lower point on the Pareto front, with a data loss between 1 and 1.5 and a physics loss of about 6×10^{-5}. Increasing α further ($\alpha = 2 \times 10^5$) causes the PINN to converge to one of the infinite trivial solutions, completely disregarding the measurements, with a data loss of about 100, meaning that the predictions deviate from the target by roughly ten times their standard deviation.

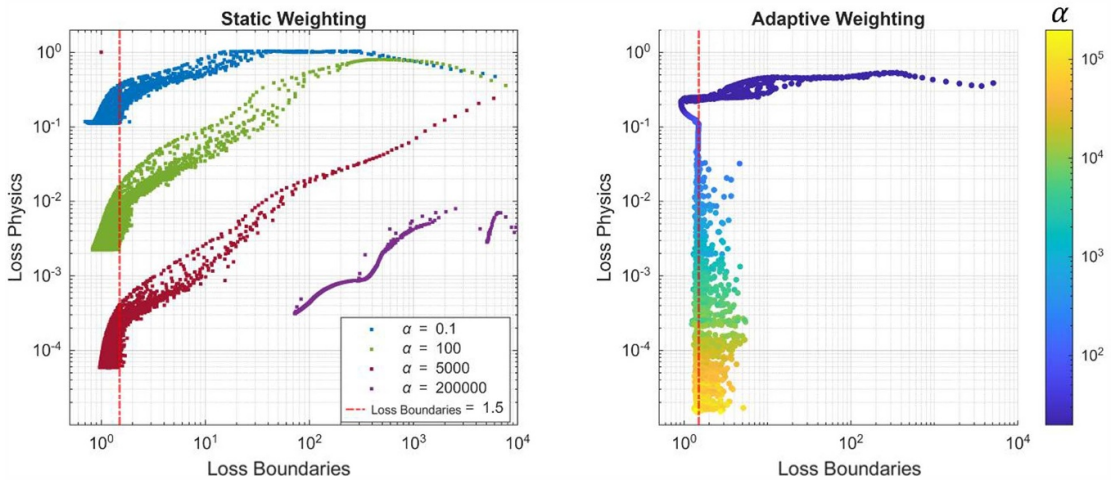

Figure 3. Loss boundaries vs loss physics with static weighting (left) using different values of α and with adaptive weighting (right). On the right plot the colour code represents graphically the value of α.

Table 3. Mean squared error normalised by the variance $(1 - R^2)$ between target and reconstructed fields using the PINN with different weighting schemes.

	$\psi\ \left[\frac{\text{Wb}}{\text{rad}}\right]$	$N_e\ [m^{-3}]$	T_e [eV]	$B_R\ [T]$	$B_\phi\ [T]$	$B_z\ [T]$
$\alpha = 0.1$	2.80×10^{-2}	3.43×10^{-2}	2.19×10^{-1}	7.08×10^{-2}	2.01×10^{-2}	7.85×10^{-2}
$\alpha = 100$	6.06×10^{-3}	1.40×10^{-2}	1.79×10^{-2}	2.14×10^{-2}	2.50×10^{-4}	7.84×10^{-3}
$\alpha = 5 \times 10^3$	8.40×10^{-4}	5.83×10^{-4}	5.23×10^{-3}	3.15×10^{-3}	1.45×10^{-5}	1.19×10^{-3}
$\alpha = 2 \times 10^5$	3.97	4.99×10^{-1}	6.93×10^{-1}	3.71	1.02×10^{-2}	4.17
Adaptive α	1.92×10^{-3}	1.81×10^{-3}	3.99×10^{-3}	2.86×10^{-3}	1.98×10^{-5}	1.83×10^{-3}

Figure 3 (right) shows the evolution of the losses during training using the adaptive scheme, with the colour code indicating the weight value α. At the beginning, the weight is small, and the PINN focuses on minimising the boundary loss, which initially falls below one. Once the boundary loss goes below 1.5 (the target loss), the weight starts to increase, giving greater importance to the physics. As a result, the boundary loss increases, since the overfitted component (the noise) is inconsistent with the physics. The weight continues to increase throughout the training, allowing the PINN to reduce the physics loss while ensuring that the boundary loss remains below the target. At the end of training for a weight around 2×10^5, the PINN reaches the lowest physics loss, and a boundary loss close to the target value of 1.5.

This is a significant result because the PINN has successfully solved the inverse problem with a very large physics weight, whereas the fixed scheme would have converged to an incorrect solution. This outcome demonstrates that the adaptive scheme can explore solutions that are inaccessible with static weighting. It should also be noted that these results have been obtained with a maximum of 5000 epochs; the adaptive weighting scheme would likely increase α further, potentially achieving an even lower physics loss. Table 3 quantifies the reconstruction error, expressed as a mean squared error normalised by the variance $\left(1 - R^2\right)$ between the reconstructed and target fields, showing that the best results are achieved with the adaptive weighting scheme and the static scheme at $\alpha = 5 \times 10^3$. The metric $1 - R^2$ is calculated as:

$$1 - R^2 = \frac{\sum_{j=1}^{N} \left(y_{\text{predicted},j} - y_{\text{target},j}\right)^2}{\sum_{j=1}^{N} \left(y_{\text{target},j} - \overline{y_{\text{target}}}\right)^2} \tag{14}$$

where the subscript 'predicted' refers to the values predicted by the trained network and the subscript 'target' refers to the original TokaLab scenario's quantity.

3.2. Neural network architecture

3.2.1. Architectures description

In this section, three architectures are investigated. In a PINN, physics can be imposed as a hard or a soft constraint. In general, hard constraints are strict requirements that must be satisfied by a solution to be considered valid. Soft constraints are more flexible requirements, and their violation is allowed,

but there is a penalty associated with the degree of such a violation. In the context of PINNs the hard constraints are typically implemented directly in the network's architecture while soft constraints are included in the loss functions.

For example, let us consider a PINN that reconstructs only the MHD equilibrium quantities (excluding the density and temperature profiles). In this case, the equations involved are the divergence of the magnetic field, Ampere's law, and the pressure balance. This problem can be addressed in different ways. One approach is to train the neural network to predict directly the seven quantities of interest, namely B_R, B_ϕ, B_z, J_R, J_ϕ, J_z and p. In this case, the physics loss of the PINN must include all seven equations, which requires appropriate balancing of the physics loss terms [52, 53]. Here, the physics is considered only in the loss function, which must be minimised, and not in the neural network architecture.

An alternative approach is to use a different formulation, such as the one described in the methodology section. By predicting the vector potential and calculating the magnetic field as its curl, one automatically imposes that the divergence of the magnetic field is exactly satisfied:

$$\nabla \cdot B = \nabla \cdot (\nabla \times A) = 0. \tag{15}$$

Similarly, the Ampere's law can be used to evaluate the plasma current density ($\mu_0 \boldsymbol{J} = \nabla \times \boldsymbol{B}$), ensuring that Ampere's law is also exactly fulfilled. In this situation, only three scalar equations (the pressure balance equations) must be minimised, and these have the same dimensions. The number of predicted quantities is therefore reduced from seven to four, the three components of the vector potential and the pressure. In this approach, part of the physics is imposed as a hard constraint, because it is always exactly satisfied (see |figure 4(a)).

While these considerations have been tested and discussed in previous works [29, 36], in this study we investigate whether alternative architectures can provide better or faster results.

For this purpose, two additional PINN architectures are introduced: the Grad Shafranov network (GS-Net) and the Hybrid Grad Shafranov network (H-Net), shown in figures 4(b) and (c). The concept behind the GS-Net is based on the flux function. In ideal steady state toroidally symmetric equilibria, both the pressure and the quantity F ($F = RB_\phi$) are functions of the poloidal flux:

$$F = f(\psi) \text{ and } p = g(\psi) \tag{16}$$

where f and g represent generic functions of ψ. Under certain conditions, the density and temperature can also be considered as functions of the poloidal flux:

$$p(\psi) = N(\psi)\, T(\psi) = g(\psi). \tag{17}$$

The basic idea of the Grad–Shafranov network is therefore to impose automatically that the predicted density and temperature are general functions of ψ. This leads to a simplification of the neural network, as it only needs to determine a simple relation between ψ and the density and temperature profiles. However, in a tokamak it is often useful to distinguish several regions with different physical properties. In particular, one may consider the plasma inside the LCFS, the plasma outside the LCFS (i.e., the scrape-off layer, SOL), and the surrounding material structures (here referred to as the wall). While p and F depend on the poloidal flux ψ within the LCFS, their behaviour differs in the other regions. In the wall, the pressure vanishes and F remains constant. In the region between the LCFS and the material boundary, the density and temperature may vary according to the characteristics of the SOL, divertor conditions, and other factors. As a result, the neural network must be able to distinguish whether a given point lies inside or outside the LCFS. Unfortunately, this classification depends on the equilibrium itself and cannot be imposed automatically. To address this, the GS-Net introduces an additional predicted quantity, denoted as λ. Here, $\lambda = 1$ indicates plasma inside the LCFS, while $\lambda = 0$ corresponds to plasma outside the LCFSThe GS-Net parameterises p and F as functions of the poloidal flux ψ. As a consequence, the architecture cannot represent independent plasma parameters in the region between the LCFS and the wall, where multiple spatial locations may share the same value of ψ. In our implementation, density and temperature are therefore set to zero in this region. While this is clearly an unsatisfactory constraint for advanced applications aiming to reconstruct the SOL regions accurately, such considerations fall outside the scope of the current work. Figure 4(b) shows the Grad–Shafranov network architecture.

Regarding the loss terms, both the physics and data losses are essentially the same as in the standard formulation (equations (B.4)–(B.14)), with the only difference being that the regularisation term used to impose the smoothness of the density and temperature on the magnetic surfaces is implemented as a

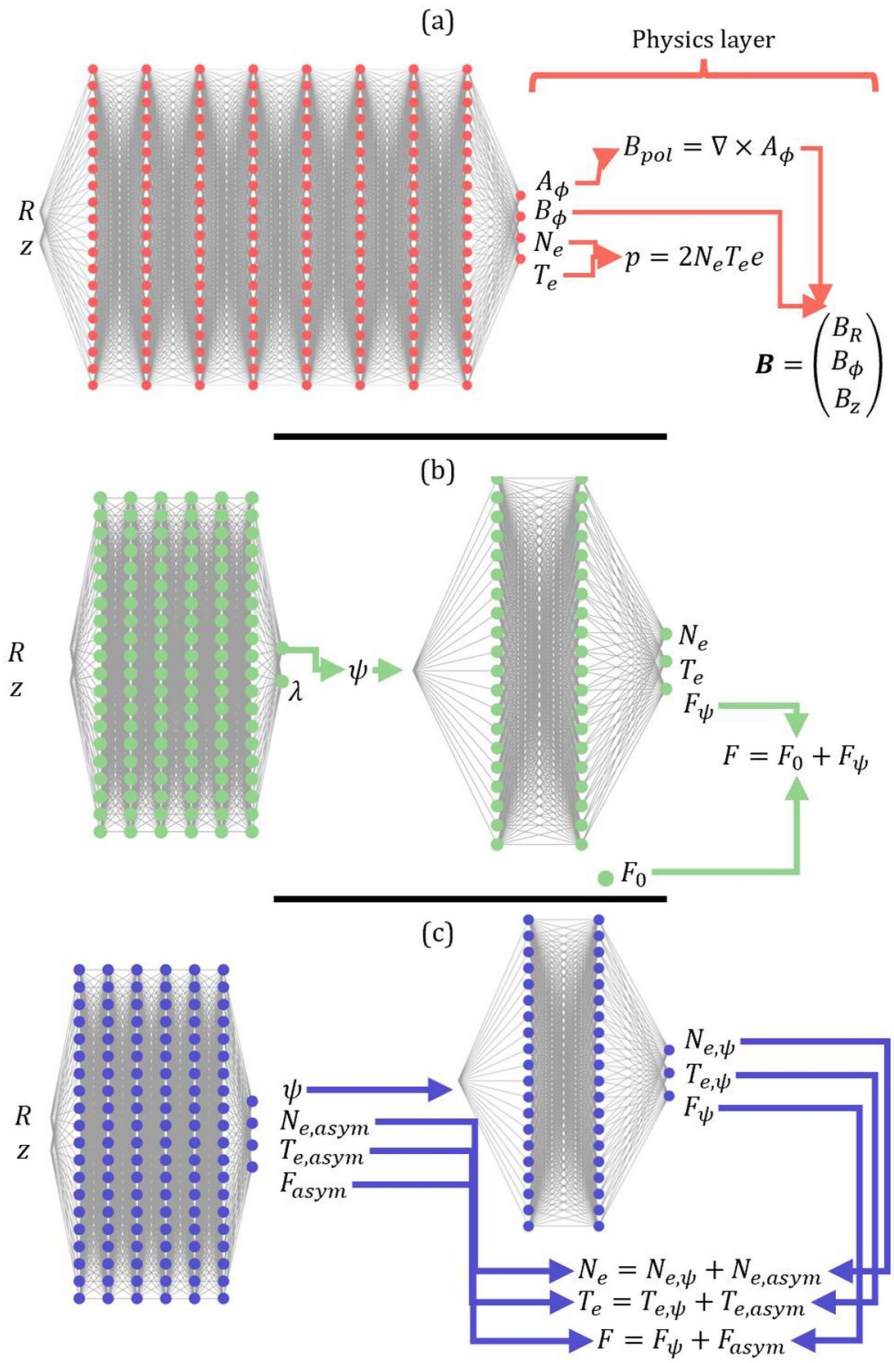

Figure 4. Schematic neural network architectures investigated: (a) MHD-Net, (b) GS-Net, and (c) H-Net.

hard constraint (the subscript p refers to 'predicted'):

$$\text{Loss}_{\text{physics}} = \text{Loss}_{\text{MHD}} + \text{Loss}_{\text{vacuum,GS-Net}} + \text{Loss}_{\text{wall,GS-Net}} \tag{18}$$

where:

$$\text{Loss}_{\text{vacuum,GS-Net}} = \frac{1}{N_{\text{coll}}} \sum \left[\frac{p_e^2}{p_0^2} + \frac{J_R^2}{J_0^2} + \frac{J_\phi{}^2}{J_0^2} + \frac{J_z^2}{J_0^2} \right] (1 - i_p)(1 - \lambda) \tag{19}$$

$$\text{Loss}_{\text{wall,GS-Net}} = \frac{1}{N_{\text{wall}}} \sum \left[\frac{N_e(R_i, z_i)^2}{N_{e,0}^2} \lambda + \frac{T_e(R_i, z_i)^2}{T_{e,0}^2} \lambda \right] \tag{20}$$

where N_{wall} are the 100 points in which the wall has been discretised, (R_w, z_w) the coordinates of a point belonging to the wall and $N_{e,0}$ and $T_{e,0}$ are two normalisation factors equal to 10^{20}m^{-3} and $2 \times 10^4\text{eV}$, respectively. The parameter i_p denotes if a point is inside ($i_p = 1$) or outside ($i_p = 0$) the wall.

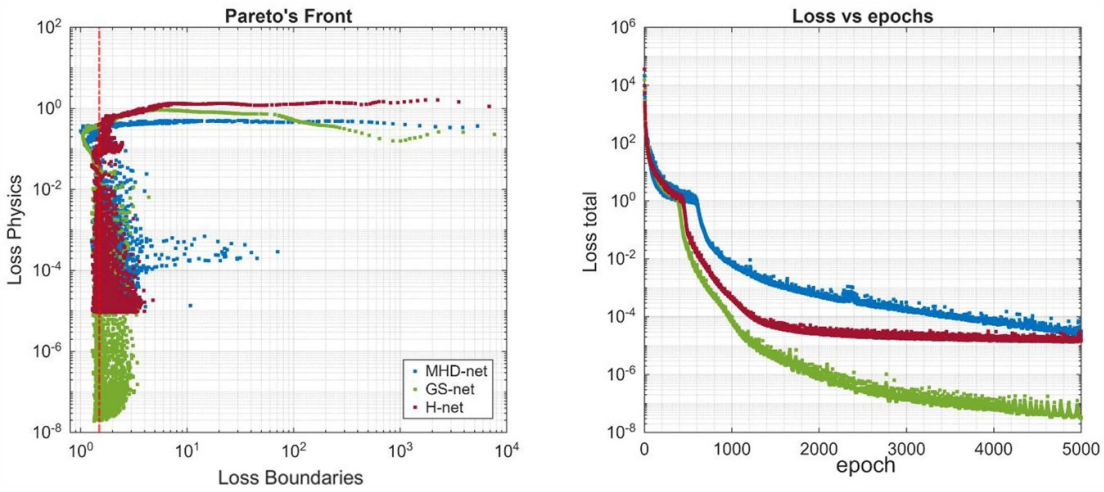

Figure 5. Pareto's front (left) and total loss during the training (right) of the PINN using the three architectures (MHD-net, GS-net and H-net).

In many cases, however, tokamak plasmas exhibit poloidal asymmetries, typically induced by impurities. As a result, the density and temperature experience poloidal variations and are no longer functions of the poloidal flux. This leads to difficulties in the reconstruction of the equilibrium, such as the appearance of artefacts.

Since these asymmetries are usually of limited amplitude value, another solution has been explored: the Hybrid H-Net, which combines the Grad–Shafranov and MHD-Net architectures. The first network predicts the poloidal flux and three quantities, $N_{e_{\rm asym}}$, $T_{e_{\rm asym}}$ and $F_{\rm asym}$. A second network then takes the poloidal flux as input and predicts the quantities N_{e_ψ}, T_{e_ψ} and F_ψ. The density, temperature, and F quantity are then computed as the sum of the asymmetric and symmetric components:

$$\begin{aligned} N_{\rm e}(R,z) &= N_{e_\psi}(R,z) + N_{e_{\rm asym}}(R,z) \\ T_{\rm e}(R,z) &= T_{e_\psi}(R,z) + T_{e_{\rm asym}}(R,z) \\ F(R,z) &= F_\psi(R,z) + F_{\rm asym}(R,z). \end{aligned} \tag{21}$$

The losses are the same as those of the MHD-Net and are therefore not reported here. The H-Net core idea is to simplify the prediction task by dividing it into two parts: one branch predicts the flux-dependent plateau portion, and a second branch is dedicated exclusively to correcting the symmetric component to accommodate any asymmetries. As already mentioned, the diagnostics models and loss functions used are reported in appendix A: diagnostics modelling and appendix B: losses formulation.

3.2.2. Results

The first analysis compares the capability of the three architectures to reconstruct a standard simulated SN scenario. The evolution of the two loss components on the Pareto front and of the total loss as a function of the epoch is shown in figure 5. The results are consistent with theoretical expectations: the GS-Net converges the fastest, reaching a physics loss of about 2×10^{-8}, compared with the MHD-Net and the H-Net, which both reach values above 10^{-5}. The faster convergence is since the GS-Net learns a simpler function, since the density, temperature, and magnetic field are directly predicted as functions of the poloidal flux ψ. In contrast, both the MHD-Net and H-Net must predict four quantities that are not directly constrained but only constrained through the physics loss.

As previously mentioned, in realistic plasma scenarios there are often effects that lead to asymmetries, making the density and temperature no longer functions of the poloidal flux.

To assess the performance of the PINNs under these conditions, a test case featuring asymmetric density and temperature profiles has been simulated, while maintaining p and F as flux functions.

The simulation process involves two steps:

1. An asymmetry is introduced into the initial symmetric density profile.
2. The temperature profile is then derived from the pressure profile (obtained as the solution of the equilibrium equation) and the newly asymmetric density profile.

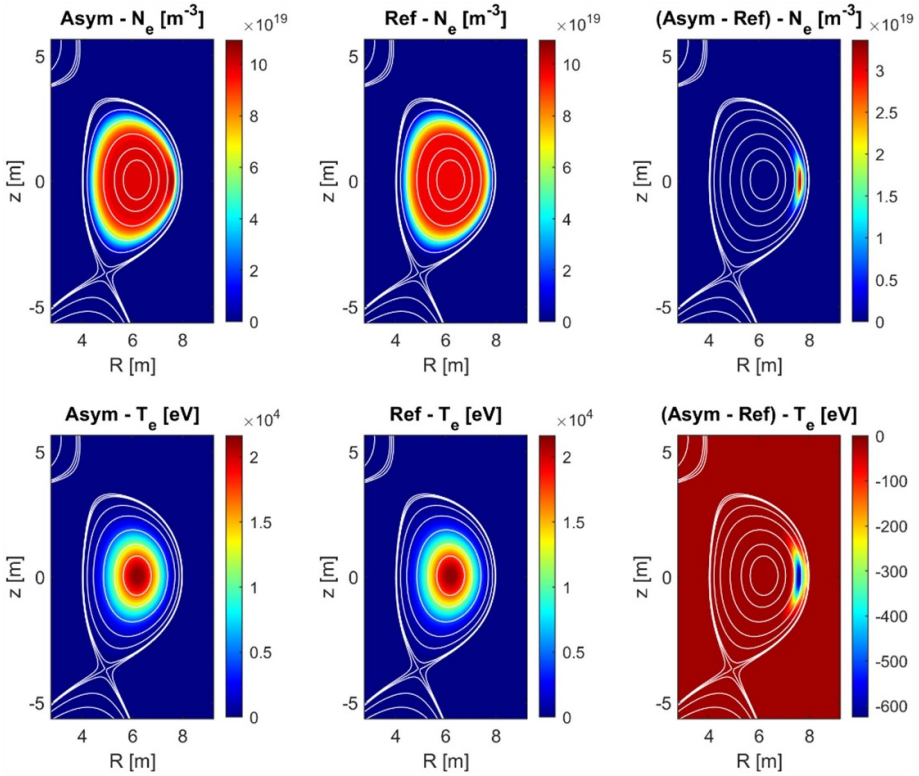

Figure 6. Asymmetric (asym) and reference (Ref) field of density and temperature and their difference (third column).

Figure 6 compares the asymmetric density and temperature fields with the reference ones. The simulated asymmetry is very small, both in spatial extension (localised on the low field side at $\psi \approx 0.7$) and in intensity (a peak difference of 3.5×10^{19} m^{-3} for density and -600 eV for temperature).

Using the three PINN architectures in the asymmetric case leads to the following results:

1. The GS-Net converges to a higher physics loss ($\sim 10^{-6}$) compared with the reference case ($\sim 10^{-7}$). In contrast to the previous case, α saturates to a fixed value ($\sim 3.8 \times 10^5$), indicating that an increase in the physics weight would also increase the data loss. This is expected when the physics model is inaccurate. Moreover, the GS-Net produces an artefact, in this case a ring (figure 7).
2. A similar but less pronounced behaviour is observed with the H-Net, suggesting that it primarily relies on the symmetric part of the network and struggles to represent the asymmetric part. This may be due to possible saturation in the non-symmetric component of the H-Net.
3. The MHD-Net performs better in this case. After 10 000 epochs, α is still increasing ($\sim 1.2 \times 10^6$). This occurs because, in this case, the 'incorrect physics' assumption (that density and temperature are flux functions) is imposed as a soft rather than a hard constraint. The asymmetry propagates through the reconstruction but with a smaller effect.

Figure 8 shows histograms of the error between the reconstructed and reference poloidal flux, electron density, and temperature fields using the MHD-Net and the GS-Net for both the symmetric and asymmetric cases. The results show that the GS-Net provides a better error distribution for the poloidal flux in the symmetric case, while in the asymmetric case the two PINNs behave similarly, with slightly smaller errors for the MHD-Net. For the density and temperature fields, the two PINNs perform similarly in the symmetric case, but in the asymmetric case the MHD-Net gives better predictions for both quantities.

These results are consistent with expectations. The GS-Net performs slightly better for symmetric cases, especially in terms of the magnetic field, since some relationships (p and F as functions of ψ) are strictly imposed. However, this advantage is lost in the asymmetric case, as the violation of the flux function assumption for density and temperature introduces errors that propagate to the magnetic field.

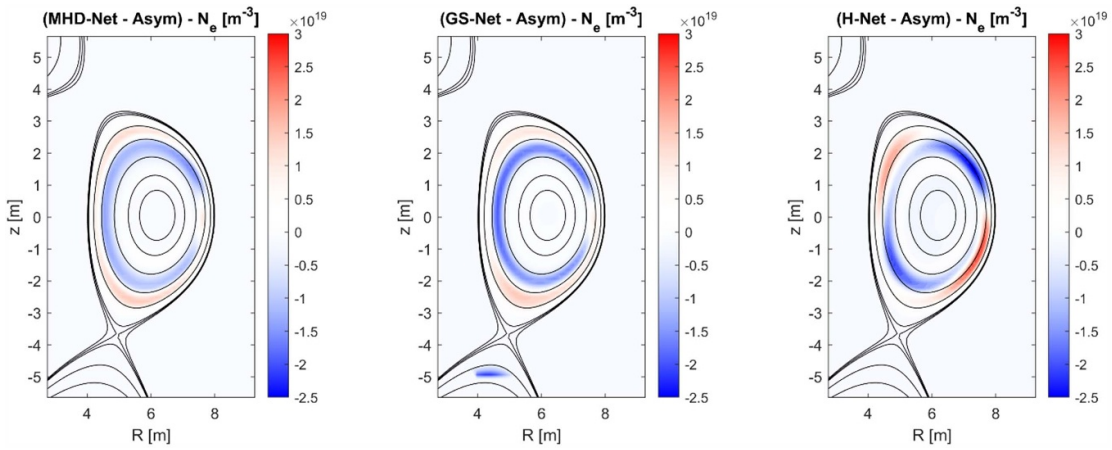

Figure 7. Error between reconstructed and target density for the asymmetric case using the MHD-Net, the GS-Net and the H-Net.

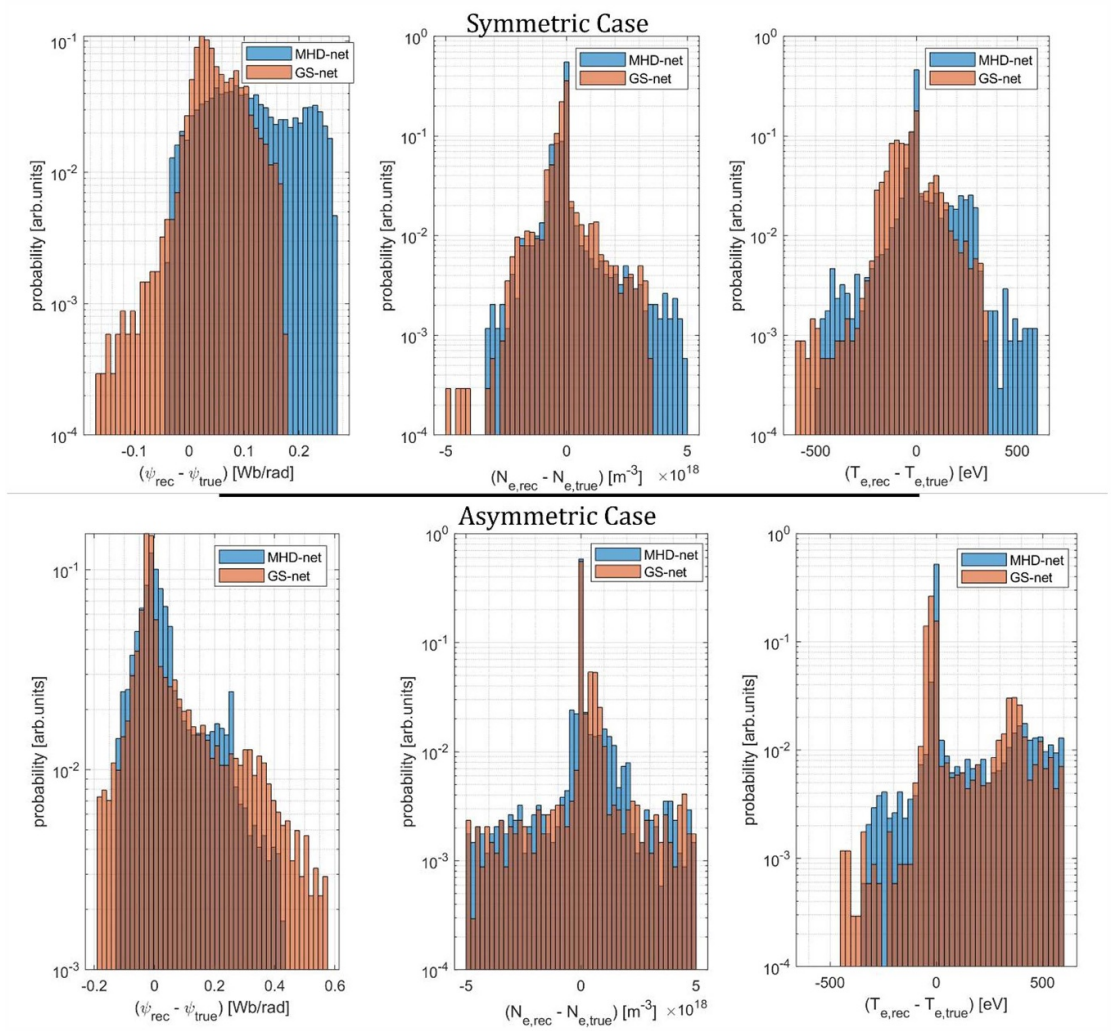

Figure 8. Error histogram of poloidal flux, electron density and temperature for the symmetric and asymmetric case using the MHD-net and the GS-net.

Conversely, the MHD-Net remains stable in both cases. The H-Net results are not shown, since its performance is worse in both situations and, at least for the present implementation, it cannot yet be considered a promising solution.

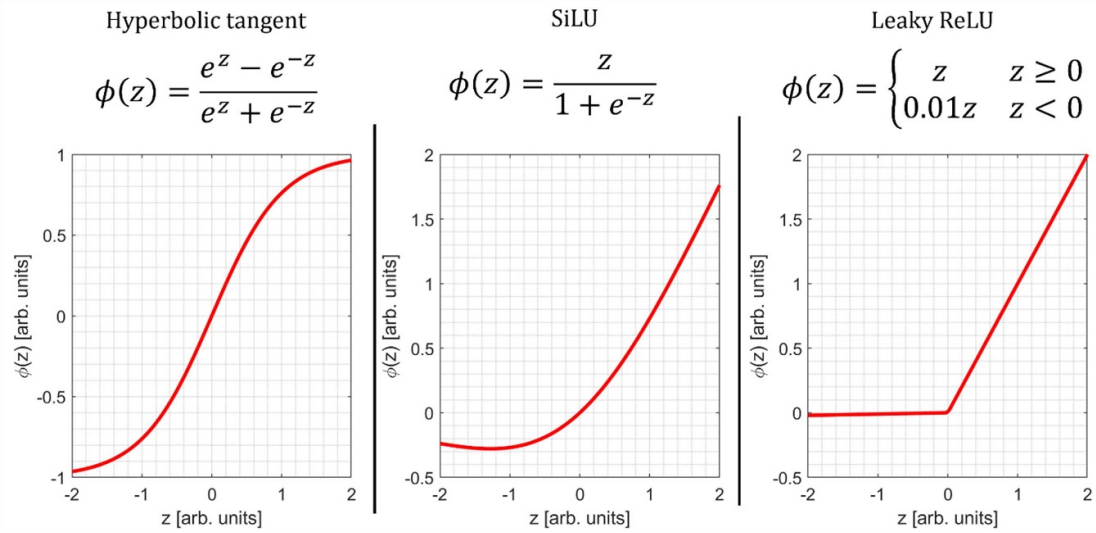

Figure 9. Activation functions trends. From the left: hyperbolic tangent, SiLU and Leaky ReLU.

3.3. Activation function

Independent of the chosen architecture, another key aspect influencing the performance of deep neural networks is the selection of the activation function (the function that each neuron applies to the input to obtain the output) in the hidden layers (by definition, the network layers that are neither in the input layer nor in the output layer). Activation functions introduce the nonlinearity required for learning complex mappings between inputs and outputs; without them, a network composed solely of linear operations would effectively reduce to a simple linear regression model.

One of the earliest and most widely adopted activation functions is the hyperbolic tangent (tanh), which produces outputs bounded between -1 and 1 and ensures zero-centred activations. However, tanh is known to suffer from the vanishing gradient problem, which can considerably slow convergence and lead to inactive neurons that no longer update their weights during training.

A subsequent and popular alternative is the rectified linear unit (ReLU), which alleviates the vanishing gradient issue by maintaining a constant gradient for positive input values. ReLU is computationally efficient and typically accelerates convergence, although it can still suffer from the so-called dying ReLU problem when neurons receive persistently negative inputs. To mitigate this issue, the Leaky ReLU introduces a small, nonzero slope for negative inputs, preserving the ability of the network to learn even when activations are below zero. This modification improves robustness while maintaining the simplicity of the ReLU function.

More recently, the sigmoid linear unit (SiLU), also known as the Swish activation, has attracted attention due to its smooth, nonmonotonic behaviour. SiLU maintains small gradients for all input values, effectively avoiding inactive neurons and often improving generalisation performance. Although it is slightly more computationally demanding than ReLU based functions, its advantages are particularly relevant in complex or large-scale architectures. The graphic representation of the three best performing activation functions is reported in figure 9.

This section presents a parametric analysis of equilibrium reconstructions using three activation functions (tanh, Leaky ReLU, and SiLU) with the objective of identifying which function provides the best trade-off among convergence speed, computational time, and reconstruction accuracy.

The first key result is that, within the considered cases, no significant differences have been observed in computational time. The average time per epoch (100 iterations per epoch) was 2.75 ± 0.12 s for tanh, 2.41 ± 0.07 s for Leaky ReLU, and 3.35 ± 0.05 s for SiLU.

The second analysis focused on the evolution of the loss functions versus the training epochs. Figure 10 (left) shows the evolution of the two main loss components (physics and data), while figure 10 (right) reports the behaviour of the total loss. The results indicate that:

- The Leaky ReLU activation fails to correctly reconstruct the plasma quantities. This is likely due to its limited nonlinearity, which arises solely from the discontinuity at zero and is therefore inadequate for modelling smooth, continuous physical fields. Consequently, for the present PINN architecture, Leaky ReLU should be avoided.

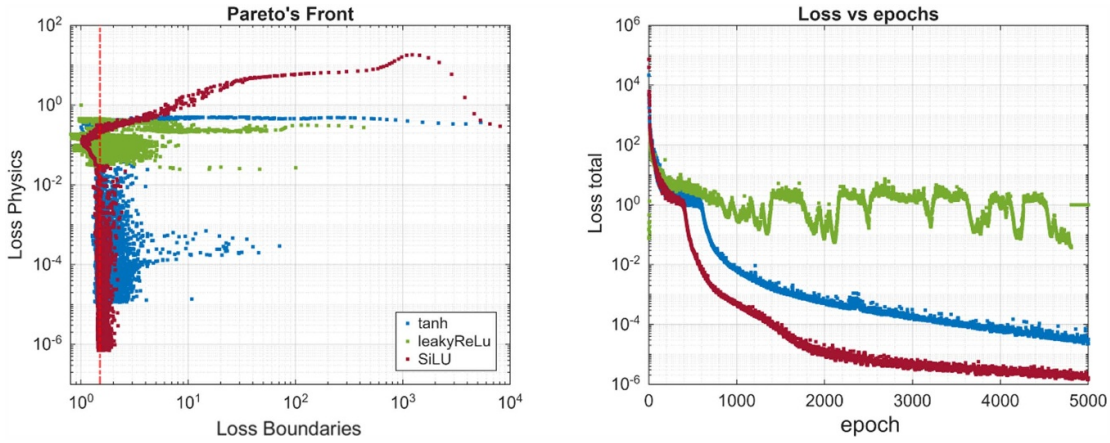

Figure 10. Pareto's front (left) and total loss during the training (right) of the PINN using three different activation functions.

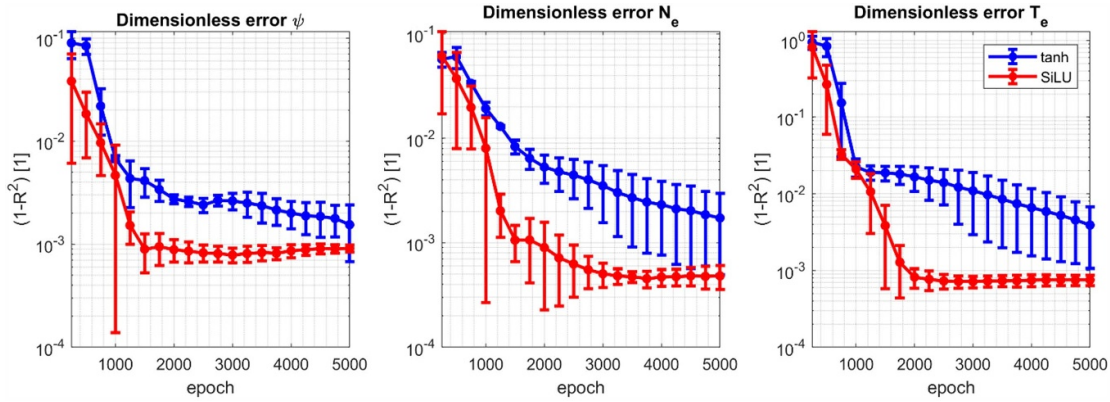

Figure 11. Mean squared error normalised by the variance ($1-R^2$) of poloidal flux, electron density, and electron temperature fields as a function of the epoch using the activation function tanh (blue) and SiLU (red).

- The SiLU activation yields the best performance in terms of both convergence speed and final physics loss, reaching values below 10^{-6} after 5000 epochs, compared with tanh, which converges to a minimum slightly above 10^{-5}.

In terms of reconstruction capability, both the tanh and SiLU activation functions yield very good results, with mean squared errors normalised by the variance ($1-R^2$) converging to values of the order of 10^{-3} (see figure 11). In contrast, the Leaky ReLU exhibits significant oscillations during training and converges to poor reconstructions, with relative mean squared errors exceeding 15%. The resulting profiles are clearly inconsistent with a physically valid plasma equilibrium, confirming that this activation function is not suitable for the present PINN based reconstruction framework.

3.4. Transfer learning

The final study has focused on assessing the impact of using a pretrained neural network within the PINN framework. One of the main limitations of PINNs lies in their case specific nature: each configuration typically requires a new training process, which entails high computational costs. It is therefore essential to evaluate whether, and to what extent, the use of a pretrained neural network can accelerate convergence while maintaining comparable performance.

To this end, PINNs previously trained with a specific scenario has been deployed to reconstruct a different equilibrium. The results obtained with these pretrained PINNs have been compared with those from randomly initialised PINNs.

Figure 12 illustrates the evolution of the physics and boundary losses for both configurations: the randomly initialised PINN (top left) and the pretrained PINN (bottom left). In this case, the pretrained model is based on a PINN trained on a DN scenario and subsequently tested on the SN case (see section 2.2).

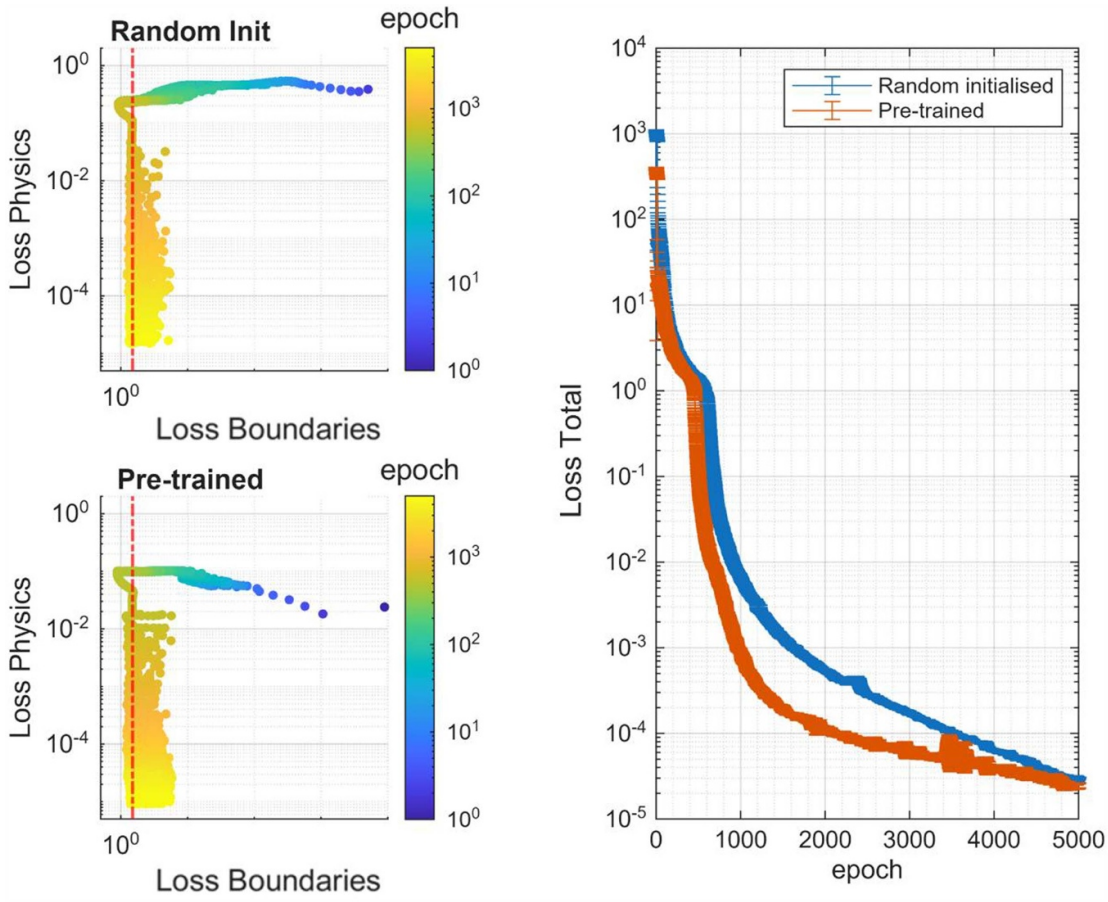

Figure 12. Loss boundaries vs loss physics evolution for randomly initialised (top-left) and pre-trained (bottom-left) PINNs. The colour map refers to the epochs. On the right, statistical evolution of total losses for randomly initialised and pre-trained PINNs.

The results reveal two main findings:

1. Both PINNs achieve good and comparable convergence within the Pareto front after 5000 epochs. This convergence is facilitated by the adaptive weighting scheme (see section 3.1), which enables self-tuning of the physics related loss terms, ensuring stable and effective optimisation.
2. The total loss of the pretrained PINN decreases faster than that of the randomly initialised model particularly at the beginning of the training. For instance, after 1000 epochs, the total loss for the pretrained PINN is approximately 7×10^{-4}, compared with 5×10^{-3} for the randomly initialised one. Moreover, the loss of the pretrained PINN is always lower than that of the randomly initialised one. This behaviour demonstrates the potential of pretrained PINNs to accelerate significantly the reconstruction process for experimental pulse analyses at any level of the desired loss.

The difference between the plasma configuration used for pre-training (DN configuration) and the single-null scenario being reconstructed enhances the advantages of using a pre-trained network.

4. Conclusion

This work presents a series of studies and parametric analyses aimed at optimising PINNs for the multi diagnostic reconstruction approach described in [29]. While the methodology and its application are clearly relevant to the specific field described in the paper, certain aspects (such as adaptive weighting) may also find broader applicability in other PINN problems across different physical and engineering domains.

The first parametric analysis focused on the implementation of an adaptive weighting scheme specifically developed for multi diagnostic experimental data. The results clearly demonstrate that moving from a static to an adaptive weighting scheme enables the attainment of an optimal Pareto front without

requiring extensive parametric analyses. Furthermore, the adaptive weighting scheme allows exploring regions of the Pareto front that are inaccessible under fixed weighting conditions, leading to more accurate and reliable reconstructions.

The second analysis has investigated the effect of the PINN architecture. From a general perspective, physical constraints can be introduced through either hard or soft formulations. Accordingly, three architectures have been developed and tested: the MHD-Net, the GS-Net, and the H-Net. The MHD-Net predicts plasma quantities without enforcing explicit constraints between them. Instead, it takes spatial coordinates as input and directly outputs the plasma quantities of interest across the entire domain. In contrast, the GS-Net strictly enforces the assumption that electron density, electron temperature, and the coefficient F are functions of the magnetic flux, ψ. This is achieved through a two-step process: a first network predicts the magnetic flux from the geometric coordinates, while a second network uses ψ as input to output N_e, T_e, and F. Finally, the H-Net refines this approach by modelling N_e, T_e, and F as the sum of two components: one that depends on ψ and another that does not, thereby accounting for poloidal asymmetries. These architectures have been evaluated under two main conditions. In the first case, an ideal scenario with perfect symmetry and full consistency with the underlying physics, the results indicate that the PINN with the hard constraint (the GS-Net) achieve the best performance in terms of computational speed. However, in the second case, when deviations from ideal assumptions are introduced (for example, poloidal asymmetries in the electron density and temperature profiles), the GS-Net, due to its rigid constraints, tends to produce artefacts with larger errors compared with the MHD-Net. The H-Net has shown limited performance in most cases and is therefore not considered a viable option at this stage. The main conclusion of this analysis is that, for ideal conditions such as numerical simulations, the GS-Net is likely the best choice, whereas for experimental cases involving asymmetries and other non-ideal effects, the MHD-Net provides superior robustness and accuracy.

The third study examines the role of the activation function, that is, the nonlinear transformation applied by each neuron in the hidden layers. Three commonly used activation functions (tanh, Leaky ReLU, and SiLU) have been compared. The results show that Leaky ReLU generally performs poorly, while tanh achieves good accuracy but with significantly higher computational time compared with SiLU, likely due to the vanishing gradient problem typical of this function. Conversely, SiLU offers comparable accuracy with substantially faster convergence, rendering it the most efficient option.

The final analysis explored the use of pretrained neural networks within the PINN framework. The objective is to assess whether a network pretrained on a different scenario could achieve good results with reduced computational time. The results demonstrated that the pretrained approach is statistically faster than randomly initialised PINNs, achieving comparable accuracy in shorter time.

The devised PINNs are now mature for application to experimental data. A systematic comparison of their results with established equilibrium codes such as EFIT [54] is also under way. From a technical point of view, the most important development would probably consist of identifying an efficient and reliable procedure to associate confidence intervals to the PINNs' outputs. Another promising direction for future work is the creation of a database of pretrained PINNs that could serve as initial models, allowing selection based on similar diagnostic characteristics to further accelerate convergence.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://github.com/QEP-Repository/PINN_MHD_Equilibria_NetOptimisation [61].

Conflict of interest

The authors declare no competing interests.

Code and Data Availability

Synthetic data have been generated using TokaLab, an open-source toolbox: https://github.com/TokaLab.

PINNs codes: https://github.com/QEP-Repository/PINN_MHD_Equilibria_NetOptimisation.

Author contributions

Novella Rutigliano 0009-0005-4350-4019
Data curation (equal), Formal analysis (equal), Investigation (equal), Methodology (equal), Software (equal), Validation (equal), Visualization (equal), Writing – original draft (equal)

Andrea Murari 0000-0002-3932-3865
Formal analysis (equal), Methodology (equal), Validation (equal), Writing – review & editing (equal)

Pasquale Gaudio 0000-0003-0861-558X
Funding acquisition (equal), Methodology (equal), Project administration (equal), Supervision (equal), Writing – review & editing (equal)

Michela Gelfusa 0000-0001-5158-7292
Funding acquisition (equal), Methodology (equal), Project administration (equal), Supervision (equal), Writing – review & editing (equal)

Riccardo Rossi 0000-0003-4414-6119
Conceptualization (lead), Formal analysis (equal), Investigation (lead), Methodology (equal), Supervision (equal), Writing – review & editing (equal)

Appendix A: Diagnostics modelling

TokaLab diagnostic set used in this work is shown in figure 1(b). Each diagnostic provides multiple measurements (which are the '# of measurements' choice in table 2). In the following, the subscript 'i' denotes one of these measurements.

Pick-up coils [42, 55, 56] are induction sensors capable of measuring the local magnetic field along the coil axis direction (n) characterised by an angle γ respect to the poloidal surface axis. Each measurement is calculated as follows:

$$B_{\mathrm{PU}_i} = B\left(R_i, z_i\right) \cdot n\left(R_i, z_i\right) = B_R\left(R_i, z_i\right) \cos\left(\gamma_i\right) + B_z\left(R_i, z_i\right) \sin\left(\gamma_i\right) \tag{A.1}$$

where R_i and z_i are the coordinates of the coil centre.

Flux loops (FL) and saddle loops (SL) [55] return magnetic flux measurements by evaluating the magnetic field linked to the loop:

$$\psi_{\mathrm{FL}_i} = \psi\left(R_i, z_i\right) \tag{A.2}$$

$$\Delta\psi_{\mathrm{SL}_i} = \psi\left(R_2, z_2\right) - \psi\left(R_1, z_1\right) \tag{A.3}$$

where, for the SL, 1 and 2 are the extremities of the saddle, considering a counterclockwise orientation.

Thomson scattering [55] is a laser-based diagnostics evaluating the electron density and electron temperature locally along the laser propagation direction. The principle of the measurement is related to a radiation beam scattered by the plasma,

$$\begin{aligned} N_{e_{TS_i}} &= N_{\mathrm{e}}\left(R_i, z_i\right) \\ T_{e_{TS_i}} &= T_{\mathrm{e}}\left(R_i, z_i\right) \end{aligned} \quad . \tag{A.4}$$

Here, the subscript i denotes the ith channel of the Thomson scattering measurement at the coordinate (R_i,z_i).

The interferometer–polarimeter [55] is also a laser-based diagnostic. The interferometer measures the line-integrated density (LID) along the radiation propagation path:

$$\mathrm{LID}_i = \int N_{\mathrm{e}} \mathrm{d}Z \tag{A.5}$$

where dZ is the infinitesimal distance along the line of sight.

For polarimetric measurements (Faraday rotation and Cotton–Mouton phase shift), the complete Stokes formulation [57] is used in combination with the hot plasma assumption [58]. A detailed analysis of the impact of such a formulation and a systematic summary of the Faraday rotation and Cotton–Mouton phase shift derivation is reported in [29, 59],

$$\begin{aligned} \mathrm{Far} &= \frac{1}{2}\int \Omega_3^{(T)} \mathrm{d}Z = \frac{1}{2} C_3 \lambda^2 \int N_\mathrm{e}(Z)\, B_z(Z)\,(1-2\tau)\,\mathrm{d}Z \\ \mathrm{CM} &= \int \Omega_1^{(T)} \mathrm{d}Z = C_1 \lambda^3 \int N_\mathrm{e}(Z) \left(B_x^2(Z) - B_y^2(Z)\right) \left(1+\frac{9}{2}\tau\right) \mathrm{d}Z \end{aligned} \tag{A.6}$$

where Z is the laser beam propagation direction and $\mathbf{\Omega}^{(T)}$ a vector modelling the plasma under the hot plasma assumption (see [29] for a complete discussion of the reference system linked to the propagation of the laser beam):

$$\mathbf{\Omega}(Z) = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} = \begin{bmatrix} C_1 \lambda_l^3 N_\mathrm{e}(Z) \left(B_x^2(Z) - B_y^2(Z)\right) \\ 2 C_1 \lambda_l^3 N_\mathrm{e}(Z)\, B_x(Z)\, B_y(Z) \\ C_3 \lambda_l^2 N_\mathrm{e}(Z)\, B_z(Z) \end{bmatrix} \tag{A.7}$$

where $\tau = T_\mathrm{e}/\left(m_\mathrm{e} c^2\right)$ is the relativistic factor [58], C_1 and C_3 are two characteristic constants ($C_1 = 2.45 \times 10^{-11}$, $C_3 = 5.26 \times 10^{-13}$) [57], and λ_l is the wavelength of the probing beam in m.

Under the hot plasma assumption, also the LID must be corrected as follows to take into account relativistic effects:

$$\mathrm{LID} = \int N_\mathrm{e} \left(1 - \frac{3}{2}\tau\right) \mathrm{d}z. \tag{A.8}$$

The motional Stark effect (MSE) diagnostic is based on the measurement of the light polarisation emitted by a beam of hydrogen neutral atoms injected into the plasma [55]. Due to the motion of the ions inside the tokamak magnetic field, an apparent electric field is generated that causes the Stark effect. By measuring the polarisation of light, it is possible to determine the orientation of the resultant magnetic field at a given point in the plasma. This orientation, defined by the angle θ with respect to the poloidal plane, characterises the field's direction.

$$\mathrm{MSE}_i = \theta(R_i, z_i). \tag{A.9}$$

It is important to underline that the TokaLab's modularity allows for an easy implementation of new diagnostics.

Appendix B: Losses formulation

In this appendix section, the formulation of the losses for the PINNs presented in this paper is described in detail. The total loss function of the PINN can be expressed as a weighted sum of the data loss and physics loss:

$$\mathrm{Loss}_\mathrm{total} = \frac{\mathrm{Loss}_\mathrm{boundaries} + \alpha\, \mathrm{Loss}_\mathrm{physics}}{1+\alpha} \tag{B.1}$$

where the parameter α can be fixed or variable along the training. A discussion on the importance of automatic update of the hyperparameter α is presented in section 3.1.

The boundary loss evaluates the error between measurements (y_b) and the reconstructed diagnostics (y_P) from the predicted plasma quantities (see appendix A to have more detail on how every diagnostic is modelled).

The general formulation for data losses can be considered as a weighted mean square error (where the weight w is the number of measurements of each diagnostic) normalised by the diagnostic uncertainty (σ_b):

$$\mathrm{Loss}_\mathrm{boundaries} = \frac{1}{W_\mathrm{tot}} \sum \frac{w\left(y_p - y_b\right)^2}{\sigma_b^2} \tag{B.2}$$

where $W_{\rm tot} = \sum^{w}$.

We now define contributions for all the losses that concur to the formulation of $\mathrm{Loss}_{\rm boundaries}$:

$$\mathrm{Loss}_{\rm boundaries} = \mathrm{Loss}_{\rm PU} + \mathrm{Loss}_{\rm FL} + \mathrm{Loss}_{\rm SL} + \mathrm{Loss}_{\rm TS} + \mathrm{Loss}_{\rm MSE} + \mathrm{Loss}_{\rm LID} + \mathrm{Loss}_{\rm Far} + \mathrm{Loss}_{\rm CM} \tag{B.3}$$

where acronyms stand for pick-up coils (PU), FL, SL, MSE, interferometer (LID), Faraday rotation (Far), Cotton–Mouton phase shift (CM).

For each of these losses the $y_{\rm p}$ is calculated as already described in appendix A. Data losses can so be written as:

$$\begin{aligned}
\mathrm{Loss}_{\rm PU} &= \frac{1}{W_{\rm tot}} \sum w_{\mathrm{PU}_i} \frac{\left(B_{\mathrm{PU}_i} - B_{\mathrm{PU}_{p,i}}\right)^2}{\sigma^2_{\rm PU}} \\
\mathrm{Loss}_{\rm FL} &= \frac{1}{W_{\rm tot}} \sum w_{\mathrm{FL}_i} \frac{\left(\psi_{\mathrm{FL}_i} - \psi_{\mathrm{FL}_{p,i}}\right)^2}{\sigma^2_{\rm FL}} \\
\mathrm{Loss}_{\rm SL} &= \frac{1}{W_{\rm tot}} \sum w_{\mathrm{SL}_i} \frac{\left(\Delta\psi_{\mathrm{SL}_i} - \Delta\psi_{\mathrm{SL}_{p,i}}\right)^2}{\sigma^2_{\rm SL}} \\
\mathrm{Loss}_{\rm MSE} &= \frac{1}{W_{\rm tot}} \sum w_{\mathrm{MSE}_i} \frac{\left(\theta_{\mathrm{MSE}_i} - \theta\psi_{\mathrm{MSE}_{p,i}}\right)^2}{\sigma^2_{\rm MSE}} \\
\mathrm{Loss}_{\rm TS} &= \frac{1}{W_{\rm tot}} \sum \left(w_{\mathrm{TS}_{Ne}} \frac{\left(N_{e\mathrm{TS}_i} - N_{e\mathrm{TS}_{p,i}}\right)^2}{\sigma^2_{\mathrm{TS}_{Ne}}} + w_{TS_{Te}} \frac{\left(T_{e\mathrm{TS}_i} - T_{e\mathrm{TS}_{p,i}}\right)^2}{\sigma^2_{\mathrm{TS}_{Te}}} \right) \\
\mathrm{Loss}_{\rm LID} &= \frac{1}{W_{\rm tot}} \sum w_{\mathrm{LID}_i} \frac{\left(\mathrm{LID}_i - \mathrm{LID}_{p,i}\right)^2}{\sigma^2_{\rm LID}} \\
\mathrm{Loss}_{\rm Far} &= \frac{1}{W_{\rm tot}} \sum w_{\mathrm{Far}_i} \frac{\left(\mathrm{Far}_i - \mathrm{Far}_{p,i}\right)^2}{\sigma^2_{\rm Far}} \\
\mathrm{Loss}_{\rm CM} &= \frac{1}{W_{\rm tot}} \sum w_{\mathrm{CM}_i} \frac{\left(\mathrm{CM}_i - \mathrm{CM}_{p,i}\right)^2}{\sigma^2_{\rm CM}}
\end{aligned} \tag{B.4}$$

where the subscript p refers to 'predicted' and so to the reconstructed measurements from the predicted plasma fields.

Only for what concerns the GS-Net PINN (see section 3.2) two more contributions must be considered for the training of the learnable parameter λ. These boundaries are formulated as a cross-entropy [60]: in such a way the network learns that λ is a distribution between 0 and 1 (being set to 0 when no plasma is present). The cross entropy is a commonly used loss function in machine learning, meant to measure the difference between a model's predicted probability distribution and the true distribution. It is aimed at penalising confident wrong predictions, lower values indicating a better model. The two main contributions define λ's cross-entropy on walls and on Thomson scattering measurements (ensuring that measurements that could be almost 0 because of noise are not considered 'outside' the plasma).

These two contributions can so be written as:

$$\mathrm{CE} = \mathrm{CE}_{\rm wall} + \mathrm{CE}_{\rm TS} = \sum -\log\left(1 - \lambda|_{\rm PhysicalPoints} + \varepsilon\right)\left(1 - i_{\rm p}\right) + \sum -\log\left(1 - \lambda|_{\rm TSpoints} + \varepsilon\right) i_{\rm p} \tag{B.5}$$

where $i_{\rm p}$ is an index equal to 1 if inside the wall and 0 outside, Physicalponits the mini-batch on which the physics prediction is made and TSpoints the points, on which the Thomson Scattering diagnostic is reconstructed and $\varepsilon$is a small offset, set to 2.26×10^{-16}, introduced to prevent the argument of the logarithm from being exactly zero.

The data loss for the GS-Net can be so written as:

$$\mathrm{Loss}_{\rm boundaries,\,GS-Net} = \mathrm{Loss}_{\rm boundaries} + \mathrm{CE}. \tag{B.6}$$

The loss physics can be generally described as the sum of five terms: the fitting of the MHD equation, the condition of the null plasma current in the wall, the conditions of null density and temperature on the wall and the contributions of density and temperature gradients with respect to the magnetic flux lines. The formulation is conducted in parallel for the 'standard' losses formulation (applicable to all the cases except for the GS-Net) and the GS-Net formulation.

For the MHD equations only the pressure boundary loss is needed since both the null divergence of the magnetic field and the Ampere's law are automatically fulfilled by the PINN's architectures,

$$\begin{aligned}
\mathrm{Loss}_{\mathrm{MHD}} &= \frac{1}{N_{\mathrm{PhysicalPoints}}} \sum_{i=1}^{N_{\mathrm{PhysicalPoints}}} \frac{f^2_{\mathrm{MHD}}}{C^2_{\mathrm{MHD}}} \\
f^2_{\mathrm{MHD}} &= \left[\left(J_\phi B_z - J_z B_\phi - \frac{\mathrm{d}p}{\mathrm{d}R} \right)^2 + \left(J_R B_z - J_z B_R \right)^2 + \left(J_R B_\phi - J_\phi B_R - \frac{\mathrm{d}p}{\mathrm{d}z} \right)^2 \right] \cdot i_{\mathrm{p}} \\
C_{\mathrm{MHD}} &= J_0 B_{p,0} = \frac{B^2_{p,0}}{a_0 \mu_0} = \frac{B^2_{t,0}}{a (2\pi R_0 \mu_0)^2}
\end{aligned} \tag{B.7}$$

where $N_{\mathrm{physicalpoints}}$ is the mini-batch size, $R_0 = 6$ m and $a_0 = 2$ m.

To ensure that current and pressure outside the wall are zero, we introduced the $\mathrm{Loss}_{\mathrm{Vacuum}}$:

$$\begin{aligned}
\mathrm{Loss}_{\mathrm{vacuum}} &= \frac{1}{N_{\mathrm{PhysicalPoints}}} \sum_{i=1}^{N_{\mathrm{PhysicalPoints}}} \frac{f^2_{\mathrm{vacuum}}}{C^2_{\mathrm{vacuum}}} \\
\frac{f^2_{\mathrm{vacuum}}}{C^2_{\mathrm{vacuum}}} &= \left[\frac{p_e^2}{p_0^2} + \frac{J_R^2}{J_0^2} + \frac{{J_\phi}^2}{J_0^2} + \frac{J_z^2}{J_0^2} \right] \left(1 - i_p \right).
\end{aligned} \tag{B.8}$$

With vacuum it is intended all the regions where the vacuum Grad–Shafranov equation is imposed (outside the wall for all networks and outside LCFS for the GS-Net). The normalisation factor $p_0 = N_{\mathrm{e},0} T_{\mathrm{e},0} = 10^{20} \mathrm{m}^{-3} 2 \times 10^4 \mathrm{eV} = 321\,800$ Pa and $J_0 = B_p / (a_0 \mu_0)$ (and $B_{p,0}$ formulated as already shown in the third equation of (B.7)).

For the GS-Net:

$$\mathrm{Loss}_{\mathrm{vacuum,\,GS-Net}} = \frac{1}{N_{\mathrm{PhysicalPoints}}} \sum_{i=1}^{N_{\mathrm{PhysicalPoints}}} \frac{f^2_{\mathrm{vacuum}}}{C^2_{\mathrm{vacuum}}} (1 - \lambda). \tag{B.9}$$

The fourth term of the physics loss function is related to the null density and temperature on the walls:

$$\mathrm{Loss}_{\mathrm{wall}} = \frac{1}{N_{\mathrm{wall}}} \sum_{w=1}^{N_{\mathrm{wall}}} \left(\frac{N_{\mathrm{e}}(R_w, z_w)^2}{N^2_{\mathrm{e},0}} + \frac{T_{\mathrm{e}}(R_w, z_w)^2}{T^2_{\mathrm{e},0}} \right) \tag{B.10}$$

where N_{wall} are the 100 points, in which the wall has been discretised, $(R_w,\, z_w)$ the coordinates of a point belonging to the wall and the normalisation factors the same already introduced.

For the GS-Net the contribution of is included resulting in the following formulation of the $\mathrm{Loss}_{\mathrm{wall}}$:

$$\mathrm{Loss}_{\mathrm{wall,GS-Net}} = \frac{1}{N_{\mathrm{wall}}} \sum_{w=1}^{N_{\mathrm{wall}}} \left(\frac{N_{\mathrm{e}}(R_w, z_w)^2}{N^2_{\mathrm{e},0}} \lambda + \frac{T_{\mathrm{e}}(R_w, z_w)^2}{T^2_{\mathrm{e},0}} \lambda \right). \tag{B.11}$$

Only for the MHD-Net and the H-Net PINNs, a soft constraint on the density and temperature gradients has been implemented (which is not necessary for the GS-Net, since, in that case, a hard constraint is present on the dependency of both density and temperature on the magnetic flux).

Indeed, the parallel gradient is expected to be small (a common assumption made on magnetic flux lines is that they are iso-density and iso-temperature). The orthogonal gradient is also softly imposed to be small to avoid strong non-physical fluctuations.

Density and temperature gradients can be written as:

$$\begin{aligned}
\left(\nabla \cdot N_{\mathrm{e}} \right)_\perp &= \left(\nabla \cdot N_{\mathrm{e}} \right) \cdot n \\
\left(\nabla \cdot N_{\mathrm{e}} \right)_\parallel &= \left(\nabla \cdot N_{\mathrm{e}} \right) \cdot t \\
\left(\nabla \cdot T_{\mathrm{e}} \right)_\perp &= \left(\nabla \cdot T_{\mathrm{e}} \right) \cdot n \\
\left(\nabla \cdot T_{\mathrm{e}} \right)_\parallel &= \left(\nabla \cdot T_{\mathrm{e}} \right) \cdot t.
\end{aligned} \tag{B.12}$$

With the parallel (t) and orthogonal (n) unit vectors evaluated from the magnetic flux gradient direction:

$$n = \begin{pmatrix} n_R \\ n_z \end{pmatrix} = \begin{pmatrix} \frac{\partial \Psi}{\partial R} \frac{1}{|\Psi|} \\ \frac{\partial \Psi}{\partial z} \frac{1}{|\Psi|} \end{pmatrix}; t = \begin{pmatrix} t_R \\ t_z \end{pmatrix} = \begin{pmatrix} -n_z \\ n_R \end{pmatrix}. \tag{B.13}$$

The regularisation term to be minimised can so be defined as:

$$\begin{aligned}
\text{Loss}_{\nabla\cdot N_e} &= \frac{1}{N_{\text{PhysicalPoints}}} \sum_{i=1}^{N_{\text{PhysicalPoints}}} \left(f^2_{(\nabla\cdot N_e)_\perp} + f^2_{(\nabla\cdot N_e)_\parallel} \right) \\
\text{Loss}_{\nabla\cdot T_e} &= \frac{1}{N_{\text{PhysicalPoints}}} \sum_{i=1}^{N_{\text{PhysicalPoints}}} \left(f^2_{(\nabla\cdot T_e)_\perp} + f^2_{(\nabla\cdot T_e)_\parallel} \right) \\
f^2_{(\nabla\cdot N_e)_\perp} &= L^2_\perp \left(\nabla\cdot N_e\right)^2_\perp \\
f^2_{(\nabla\cdot N_e)_\parallel} &= L^2_\parallel \left(\nabla\cdot N_e\right)^2_\parallel \\
f^2_{(\nabla\cdot T_e)_\perp} &= L^2_\perp \left(\nabla\cdot T_e\right)^2_\perp \\
f^2_{(\nabla\cdot T_e)_\parallel} &= L^2_\parallel \left(\nabla\cdot T_e\right)^2_\parallel
\end{aligned} \tag{B.14}$$

where $L_\perp$ and $L_\parallel$ are two normalisation lengths equal to $2\pi R_0\,[\text{m}]$ and $10^{-3}[\text{m}]$ respectively.

At the end, the overall formulation of the $\text{Loss}_{\text{Physics}}$ for the MHD-Net and the H-Net PINNs can be considered as follows:

$$\text{Loss}_{\text{physics}} = \text{Loss}_{\text{MHD}} + \text{Loss}_{\text{vacuum}} + \text{Loss}_{\text{wall}} + \beta\left(\text{Loss}_{\nabla\cdot Ne} + \text{Loss}_{\nabla\cdot Te}\right) \tag{B.15}$$

where $\beta = 10^{-2}$ imposes the soft constraint on the density and temperature gradients.

For the GS-Net the $\text{Loss}_{\text{Physics}}$ can be written as follows:

$$\text{Loss}_{\text{physics,GS}-\text{Net}} = \text{Loss}_{\text{MHD}} + \text{Loss}_{\text{vacuum,GS}-\text{Net}} + \text{Loss}_{\text{wall,GS}-\text{Net}} \tag{B.16}$$

References

[1] Wang T, Mazon D, Svensson J, Liu A, Zhou C, Xu L, Hu L, Duan Y and Verdoolaege G 2019 Bayesian data analysis for Gaussian process tomography *J. Fusion Energy* **38** 445–57

[2] Rodionov R, Nemtcev G and Krasilnikov A 2022 Fusion neutron emissivity tomography for ITER vertical neutron camera *Nucl. Instrum. Methods Phys. Res. A* **1040** 167127

[3] Jardin A, Bielecki J, Mazon D, Peysson Y, Król K, Dworak D and Scholz M 2021 Implementing an x-ray tomography method for fusion devices *Eur. Phys. J. Plus* **136** 706

[4] Odstrčil T, Pütterich T, Odstrčil M, Gude A, Igochine V and Stroth U 2016 Optimized tomography methods for plasma emissivity reconstruction at the ASDEX Upgrade tokamak *Rev. Sci. Instrum.* 123505 **87**

[5] Wyss I, Murari A, Peluso E, Gelfusa M, Gaudio P and Rossi R 2024 On the accuracy of a fast time resolution inversion method for the detection of different radiation patterns in fusion reactors *Fusion Eng. Des.* **205** 114527

[6] Svoboda J, Cavalier J, Ficker O, Imríšek M, Mlynář J and Hron M 2021 Tomotok: python package for tomography of tokamak plasma radiation *J. Instrum.* **16** C12015

[7] Blum J and Buvat H 1997 *An Inverse Problem in Plasma Physics: The Identification of the Current Density Profile in a Tokamak* (Springer) pp 17–36

[8] Lamérand L, Auroux D, Ghendrih P, Rapetti F and Serre E 2024 Inverse problem for determining free parameters of a reduced turbulent transport model for tokamak plasma *Adv. Comput. Math.* **50** 39

[9] Angioni C 2021 Impurity transport in tokamak plasmas, theory, modelling and comparison with experiments *Plasma Phys. Control. Fusion* **63** 073001

[10] Xie X, Lan T, Liu H, Zhu X, Mao W, Lan T and Ding W 2025 Neural-network based electron density profile inversion for interferometer on EAST tokamak *Plasma Phys. Control. Fusion* **67** 045001

[11] Faugeras B, Blum J and Boulbe C 2022 First equilibrium reconstruction for ITER with the code NICE *J. Instrum.* **17** C02024

[12] Fitzpatrick R 2024 Inverse aspect-ratio expanded tokamak equilibria *Phys. Plasmas* **31** 082505

[13] Di Grazia L E, Mattei M and Pironti A 2025 An extended Kalman filter approach for tokamak plasma equilibrium reconstruction *Fusion Eng. Des.* **220** 115363

[14] Kaltsas D A and Throumoulopoulos G N 2022 Neural network tokamak equilibria with incompressible flows *Phys. Plasmas* **29** 022506

[15] Raissi M, Perdikaris P and Karniadakis G E 2019 Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations *J. Comput. Phys.* **378** 686–707

[16] Hornik K, Stinchcombe M and White H 1989 Multilayer feedforward networks are universal approximators *Neural Netw.* **2** 359–66

[17] Chen Z, Liu Y and Sun H 2021 Physics-informed learning of governing equations from scarce data *Nat. Commun.* **12** 6136

[18] Zhu M, Zhang H, Jiao A, Karniadakis G E and Lu L 2023 Reliable extrapolation of deep neural operators informed by physics or sparse observations *Comput. Methods Appl. Mech. Eng.* **412** 116064

[19] Lawal Z K, Yassin H, Lai D T and Che Idris A 2022 Physics-informed neural network (PINN) evolution and beyond: a systematic literature review and bibliometric analysis *Big Data Cogn. Comput.* **6** 140

[20] Botarelli T, Fanfani M, Nesi P and Pinelli L 2025 Using physics-informed neural networks for solving Navier-Stokes equations in fluid dynamic complex scenarios *Eng. Appl. Artif. Intell.* **148** 110347

[21] Rossi R, Murari A, Craciunescu T, Wyss I, Mazon D, Pau A, Costantini A and Gelfusa M 2025 Time-resolved, physics-informed neural networks for tokamak total emission reconstruction and modelling *Nucl. Fusion* **65** 036030

[22] Sun R, Jeong H, Zhao J, Gou Y, Sauret E, Li Z and Gu Y 2024 A physics-informed neural network framework for multi-physics coupling microfluidic problems *Comput. Fluids* **284** 106421
[23] Wang X, Wen H and Wang B 2024 Super-resolution flow-field reconstruction in rotating detonation combustors *Aerosp. Sci. Technol.* **144** 108740
[24] Kim D and Lee J 2024 A review of physics informed neural networks for multiscale analysis and inverse problems *Multiscal. Sci. Eng.* **6** 1–11
[25] Karniadakis G E, Kevrekidis I G, Lu L, Perdikaris P, Wang S and Yang L 2021 Physics-informed machine learning *Nat. Rev. Phys.* **3** 422–40
[26] Sofos F and Drikakis D 2025 A review of deep learning for super-resolution in fluid flows *Phys. Fluids* **37** 041303
[27] Gao H, Sun L and Wang J-X 2021 Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels *Phys. Fluids* **33** 073603
[28] Sautory T and Shadden S C 2024 Unsupervised denoising and super-resolution of vascular flow data by physics-informed machine learning *J. Biomech. Eng.* **146** 091006
[29] Rutigliano N, Rossi R, Murari A, Gelfusa M, Craciunescu T, Mazon D and Gaudio P 2025 Physics-informed neural networks for the modelling of interferometer-polarimetry in tokamak multi-diagnostic equilibrium reconstructions *Plasma Phys. Control. Fusion* **67** 065029
[30] Rossi R, Gelfusa M and Murari A 2023 On the potential of physics-informed neural networks to solve inverse problems in tokamaks *Nucl. Fusion* **63** 126059
[31] Jang B, Kaptanoglu A A, Gaur R, Pan S, Landreman M and Dorland W 2024 Grad–Shafranov equilibria via data-free physics informed neural networks *Phys. Plasmas* **31** 032510
[32] Kumar J, Zarzoso D, Grandgirard V, Ebert J and Kesselheim S 2023 Physics informed neural networks applied to the description of wave-particle resonance in kinetic simulations of fusion plasmas (arXiv:2308.12312)
[33] Seo J, Kim I H and Nam H 2024 Leveraging physics-informed neural computing for transport simulations of nuclear fusion plasmas *Nucl. Eng. Technol.* **56** 5396–404
[34] Rahman H U, Hussain A, Ilyas M, Ahmed M and Ur Rehman H 2025 A multi-domain physics-informed neural network for transient thermal analysis of a tokamak divertor *Fusion Eng. Des.* **216** 115036
[35] McDevitt C J 2023 A physics-informed deep learning model of the hot tail runaway electron seed *Phys. Plasmas* **30** 092501
[36] Rutigliano N, Murari A, Gaudio P, Gelfusa M and Rossi R 2026 Multi-diagnostics reconstruction of magnetic equilibrium and kinetic profiles using physics-informed neural networks with applications to JET *Nucl. Fusion* **66** 046006
[37] Shafranov V D 1966 Plasma equilibrium in a magnetic field, reviews of plasma physics *Rev. Mod. Plasma Phys.* **2** 103
[38] Grad H and Rubin H 1958 Hydromagnetic equilibria and force-free fields *J. Nucl. Energy* **7** 284–5
[39] TokaLab Team TokaLab (available at: https://tokalab.github.io/) (Accessed 7 August 2025)
[40] Coleman M and McIntosh S 2020 The design and optimisation of tokamak poloidal field systems in the BLUEPRINT framework *Fusion Eng. Des.* **154** 111544
[41] Frassinetti L, Beurskens M N A, Scannell R, Osborne T H, Flanagan J, Kempenaars M, Maslov M, Pasqualotto R and Walsh M 2012 Spatial resolution of the JET Thomson scattering system *Rev. Sci. Instrum.* **83** 013506
[42] Wesson J 2011 *Tokamaks* (Oxford University Press)
[43] Boboc A, Macdonald J, Felton R, Brown M J, Studholme W and Cramp S 2024 JET far-infrared interferometer/polarimeter diagnostic system—40 years of lessons learned *Plasma Phys. Control. Fusion* **66** 085011
[44] Sobol' I M, Asotsky D, Kreinin A and Kucherenko S 2011 Construction and comparison of high-dimensional Sobol' generators *Wilmott* **2011** 64–79
[45] Sobol' I M 1967 On the distribution of points in a cube and the approximate evaluation of integrals *USSR Comput. Math. Math. Phys.* **7** 86–112
[46] Kingma P D and Ba J 2015 Adam: a method for stochastic optimization *3rd Int. Conf. for Learning Representations (San Diego)* pp 7–9
[47] van der Meer R, Oosterlee C W and Borovykh A 2022 Optimally weighted loss functions for solving PDEs with neural networks *J. Comput. Appl. Math.* **405** 113887
[48] Gao B, Yao R and Li Y 2025 Physics-informed neural networks with adaptive loss weighting algorithm for solving partial differential equations *Comput. Math. Appl.* **181** 216–27
[49] Jing X-T, Bai Y-L, Hou B-Y and Huang C 2025 Physics-informed neural networks coupled with a residual-driven dynamic weighted Huber loss function *New J. Phys.* **27** 094602
[50] Rutigliano N, Rossi R and Gaudio P 2026 A novel adaptive weighting scheme for physics-informed neural networks in presence of noisy data and outliers *Eng. Res. Express* **8** 045217
[51] Rohrhofer F M, Posch S, Gößnitzer C and Geiger B C 2023 Data vs. physics: the apparent Pareto front of physics-informed neural networks *IEEE Access* **11** 86252–61
[52] Bischof R and Kraus M A 2025 Multi-objective loss balancing for physics-informed deep learning *Comput. Methods Appl. Mech. Eng.* **439** 117914
[53] Xiang Z, Peng W, Liu X and Yao W 2022 Self-adaptive loss balanced physics-informed neural networks *Neurocomputing* **496** 11–34
[54] Lao L L, John H S, Stambaugh R D, Kellman A G and Pfeiffer W 1985 Reconstruction of current profile parameters and plasma shapes in tokamaks *Nucl. Fusion* **25** 1611–22
[55] Hutchinson I H 2002 *Principles of Plasma Diagnostics* (Cambridge University Press) (https://doi.org/10.1017/CBO9780511613630)
[56] Donné A J H *et al* 2007 Chapter 7: diagnostics *Nucl. Fusion* **47** S337–84
[57] Segre S E 1999 A review of plasma polarimetry—theory and methods *Plasma Phys. Control. Fusion* **41** R57–R100
[58] Mirnov V V, Ding W X, Brower D L, Van Zeeland M A and Carlstrom T N 2007 Finite electron temperature effects on interferometric and polarimetric measurements in fusion plasmas *Phys. Plasmas* **14** 102105
[59] Rossi R, Orsitto F P, Spolladore L, Wyss I and Gaudio P 2020 On the interpretability and uncertainty propagation of polarimetric measurements in thermonuclear plasmas as a function of the input polarisation and laser wavelength *Plasma Phys. Control. Fusion* **62** 105019
[60] de Boer P-T, Kroese D P, Mannor S and Rubinstein R Y 2005 A tutorial on the cross-entropy method *Ann. Oper. Res.* **134** 19–67
[61] Rutigliano N and Rossi R (available at: https://github.com/QEP-Repository/PINN_MHD_Equilibria_NetOptimisation)