PAPER • **OPEN ACCESS**

A novel adaptive weighting scheme for physics-informed neural networks with uncertainty prediction in presence of noisy data and outliers

To cite this article: Novella Rutigliano *et al* 2026 *Eng. Res. Express* **8** 045217

Engineering Research Express

OPEN ACCESS

RECEIVED
13 November 2025

REVISED
5 February 2026

ACCEPTED FOR PUBLICATION
12 February 2026

PUBLISHED
24 February 2026

PAPER

A novel adaptive weighting scheme for physics-informed neural networks with uncertainty prediction in presence of noisy data and outliers

Novella Rutigliano*, Riccardo Rossi and Pasquale Gaudio

Dipartimento di Ingegneria Industriale, Università degli Studi di Roma 'Tor Vergata', Via del Politecnico 1, Roma, 00133, Italy

* Author to whom any correspondence should be addressed.

E-mail: novella.rutigliano@alumni.uniroma2.eu

Keywords: physics-informed neural network, adaptive weighting, uncertainty, inverse problem

Abstract

Physics-Informed Neural Networks (PINNs) have recently emerged as a powerful approach, integrating data-driven deep learning with physical knowledge, and are increasingly utilised in fields such as physics, engineering, chemistry and mathematics. While PINNs offer advantages like resilience to noisy data, high-resolution predictions, and mesh-free simulation capabilities, they face challenges in balancing data-driven and physics-based losses. This work introduces a novel adaptive weighting scheme to enhance PINN convergence when dealing with noisy measurements combined with a PINN architecture capable of predicting uncertainty variables, using a model designed to estimate parameters (e.g., mean and standard deviation) for assumed distributions of predicted quantities. The methodology, based on Negative Log Likelihood (NLL) expectations, allows for different interesting features. At first, it weights the various diagnostics accordingly with their uncertainties, giving more importance to more accurate and reliable measurements. Second, the loss to physics is increased once a target NLL on the boundary is achieved, allowing for very accurate reconstructions. Last, the adaptive weighting scheme requires the selection of a new hyperparameter that it is easy to estimate. Moreover, if the hyperparameter is wrongly estimated, it is very easy to detect such an error and by a simple and general procedure it is possible to reach the best convergence in few iterations Additionally, the proposed methodology effectively detects problematic boundaries and diagnostics, offering robust performance across diverse applications. In this work applications of the presented method to synthetic cases simulating real experimental and engineering problems are proposedincluding heat transfer, fluid dynamics, and magnetohydrodynamics scenarios. This adaptive weighting framework demonstrates significant promise for inverse problem-solving, diagnostics assessment, and applications where data and physical models must be harmonised.

1. Introduction

One of the most groundbreaking developments in artificial intelligence is the integration of physical laws into the training of deep neural networks, known as Physics-Informed Neural Networks (PINNs), introduced by Raissi *et al* [1]. PINNs incorporate a term in the loss function to guide the network toward solutions that satisfy the underlying physical models. This approach has led to numerous advantages and applications that were previously unattainable. For instance, predictive models that embed physics can now be reliably used for extrapolation, as the physics-based loss functions act as a regularization mechanism, ensuring robust constraints even in regions with sparse or no data [2, 3]. Numerous studies have demonstrated that Physics-Informed Neural Networks are powerful tools for solving both forward and inverse problems [4], such as the Navier–Stokes numerical simulation [5, 6], tomography, and modelling. However, PINNs are not merely replacements for standard numerical solvers; they offer unique advantages that enable pioneering results. These

include being meshless, supporting domain decomposition, and facilitating the implementation of multi-physics [7–9]. Moreover, PINNs are not limited to numerical simulations; they also allow the incorporation of experimental measurements, which are typically affected by noise, into the training process. A particularly innovative feature of PINNs is their seamless integration of physics-driven knowledge (e.g., numerical simulations) with data-driven insights (e.g., experimental measurements). This capability enables direct modelling of partial differential equations (PDEs) and enhances diagnostic processing, leading to revolutionary advancements such as super-resolution [7, 10–12]. Five years after their introduction, PINNs have found applications across various fields of science and technology, including fluid dynamics [13], nuclear fusion and plasma physics [14], quantum mechanics [15], energy [16], and medicine [17].

Despite their potential and widespread use, Physics-Informed Neural Networks (PINNs) are still a relatively young technology that faces several limitations, particularly in their standard implementations. Setting up a PINN correctly is a complex task, and there does not yet appear to be a universal approach. Specific applications often demand tailored solutions, as reported in various studies, especially when dealing with complex physical equations that involve non-linearity, chaotic behaviours, discontinuities, or stiff dynamics.

One of the most challenging issues in Physics-Informed Neural Networks (PINNs) is determining the appropriate loss weighting schemes. A PINN is inherently governed by multiple loss functions, typically at least one for the physics ($L_{physics}$) and one for the data or boundary conditions (L_{data}). The total loss used to train the PINN is defined as the sum of these losses, each weighted by a parameter. In its simplest form, this can be expressed as $L_{total} = L_{data} + \alpha L_{physics}$, where α represents the weight assigned to the physics loss (with $\alpha = 1$ indicating that the physics loss is weighted equally to the data loss). The key challenge lies in determining the optimal value of α for a given problem. Moreover, data and physics losses are usually defined by other losses of single parts. Let us suppose the example of incompressible Navier–Stokes: the data loss would contain both velocity and pressure measurements, while the physics loss must satisfy both momentum and continuity equations. All these losses have different unit measures and scales, and therefore some scaling of the losses is needed. A common practice is to work with dimensionless equations and data, by normalising the various quantities with *ad hoc* procedures, which however do not fully solve the issues. Another important question to be considered is the presence of experimental data affected by noise. In numerical simulations, boundaries are expected to be not affected by any uncertainty and their values are imposed, while PINNs can be developed to tackle uncertainties. However, the loss weighting scheme should consider the fact that data are uncertain; if the data loss function is weighted too heavily compared to the physics loss ($\alpha \to 0$) the neural network will overfit also the noise, while if physics has a dominant role ($\alpha \to \infty$) the PINN will find one of the infinite solutions of the PDEs, without considering data (or ignoring most of them).

Several works analysed the importance of loss-weighting schemes and it is clear that the choice of α has an important role in the performances of PINNs. In Rohrhofer *et al*'s work [18], it has been systematically analysed the effect of the weight parameters, illustrating how same PINNs trained with different weights lead to a different distribution of the data and physics loss, and that they composed an apparent Pareto Front. They proposed to perform different PINN training, analyse the Pareto Front, and take the result that is more realistic. However, such an approach may require long computational times for some specific cases, requiring the previous method a parametric analysis of PINN training. To overcome this limitations, adaptive weighting schemes adaptive weighting schemes have been proposed. They propose to change the weight parameter by considering some expectations from the data and the model, reducing the problem to one PINN training [19]. Xia-Ting Jing *et al* in [20] adopt a weighted Huber loss function to avoid convergence, overfitting and loss imbalances during the training.

Another important consideration about Physics-Informed Neural Network working with experimental data is the estimation of uncertainties [21–23]. Let us suppose a simple physical problem described by one physical equation and closed by two measurements. Since the measurements are affected by noise, the PINN will not find the actual solution, but a new one where the boundary conditions are exactly the measurements, noise included. In such situations, it would be useful to estimate the uncertainty of the predictions, providing researchers with confidence intervals for their reconstructions/simulations. In the literature, different approaches have been proposed. The most efficient, but computationally demanding, is the Monte Carlo method [24]. The idea is to perform different trainings by perturbing the boundary condition with their estimated uncertainties and then evaluate the statistics of the predictions. Even if this method is truly effective and the unique at returning (for the number of trainings which goes to infinity) the exact prediction of the uncertainty, it is not practicable for many cases where one training may require days or weeks of computation. An alternative has been proposed based on the multi-output neural network [25]. This approach knocks down the huge computational times required by Monte Carlo approach, by requiring to the last layer to predict different values of the same variable, allowing for one training of one more complex neural network. However, even this approach may be computationally demanding for high dimensionality problems, and it is limited to specific cases (multi-output is not expected to work with chaotic systems, for example).

In this work, the authors introduce a new adaptive weighting scheme that can be easily combined with a PINN architecture able to predict both average values and their confidence intervals without a substantial

increase in neural network complexity or training time. This approach, tested with various synthetic cases, demonstrates its ability to accurately handle and predict uncertainties, and ensures that weights are updated to achieve optimal values in the apparent Pareto Front.

The paper is organised as follows: in the next section, titled 'Methods', we provide a brief overview of PINNs (section 2.1) followed by the new adaptive weighting scheme and the PINN architecture used in this work (section 2.2). Section 3 provides a detailed parametric analysis of the new methodology. At first, in section 3.1, the output of our PINN is compared with the standard MC-based approaches. Then, sections 3.2 and 3.3 show how the methodology performs in standard and pathologic (e.g. presence outliers) cases. Section 4 tests the new approach in more complex cases and different physics (heat transfer, fluid dynamics, and MHD) to prove that the methodology is performant, reliable, and easy to scale to different problems. Then, in section 5 the main conclusions are reported.

2. Methods

This work aims to provide a methodology based on Physics-Informed Neural Network to solve forward and backward problems in presence of noisy measurements, estimate the uncertainties of the predictions, and allow a fast convergence of the PINN through an advanced adaptive weighting scheme. In this section, a brief overview of how a PINN can be trained is provided. Then, our methodology PINN architecture and training methodology is explained.

2.1. A general introduction on PINN training process

A Physics-Informed Neural Network (PINN) is a deep neural network trained by enforcing physics-based constraints. A deep neural network (NN) is a universal approximator represented by the function $y = F(\theta, x)$, where y is the desired output, and F is a function that takes as input the input space x and the parameters θ. The goal of training is to tune the parameters θ by minimizing a loss function. In PINNs, the physics constraints play the most crucial role: the NN training process is similar to that of a standard supervised machine learning algorithm, but the loss includes not only measurements but also the physical behaviour of the system. PINNs can be considered a mesh-free method: the direct resolution of governing equations is transformed into a loss optimization process.

Since it can be challenging to describe a PINN algorithm in a general way, as the training is based on specific physics equations describing the phenomena, a canonical example is provided below. Consider the incompressible unsteady state 2D Navier–Stokes problem. The equations describing the problem are:

$$\begin{aligned}
\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} &= 0 \\
\rho\left(\frac{\partial u_x}{\partial t} + u_x\frac{\partial u_x}{\partial x} + u_y\frac{\partial u_x}{\partial y}\right) &= -\frac{\partial p}{\partial x} + \mu\left(\frac{\partial^2 u_x}{\partial x^2} + \frac{\partial^2 u_x}{\partial y^2}\right) \\
\rho\left(\frac{\partial u_y}{\partial t} + u_x\frac{\partial u_y}{\partial x} + u_y\frac{\partial u_y}{\partial y}\right) &= -\frac{\partial p}{\partial y} + \mu\left(\frac{\partial^2 u_y}{\partial x^2} + \frac{\partial^2 u_y}{\partial y^2}\right)
\end{aligned} \tag{1}$$

where the unknown quantities are u_x, u_y, p. So, the neural network takes in input the coordinates (x, y, t) and predict the fields u_x, u_y, p. Through the automatic differentiation techniques differential quantities are calculated and it is imposed that the predicted quantities to satisfy the Navier–Stokes equations by minimizing the physics loss function. The physics equation alone cannot be used to solve the problem since infinite solutions exist. The problem must be closed by boundary conditions that for PINNs can be measurements, boundary constrains, or both. So, starting from now, the superscript p indicates predicted quantities by the neural network, while the superscript t indicates the boundaries or the measurements.

A loss function is the quantification of the error (in this example the mean square error MSE) committed on the prediction respect to the reference value. We can so define the physics loss function as:

$$\begin{aligned}
L_{physics} = &\frac{1}{N}\sum_{i=1}^{N}\left[\left(\frac{\partial u_x^p}{\partial x} + \frac{\partial u_y^p}{\partial y}\right)\Bigg|_{x_i,y_i,t_i}\right]^2 \\
&+\frac{1}{N}\sum_{i=1}^{N}\left[\left(\rho\left(\frac{\partial u_x^p}{\partial t} + u_x^p\frac{\partial u_x^p}{\partial x} + u_y^p\frac{\partial u_x^p}{\partial y}\right) + \frac{\partial p^p}{\partial x} - \mu\left(\frac{\partial^2 u_x^p}{\partial x^2} + \frac{\partial^2 u_x^p}{\partial y^2}\right)\right)\Bigg|_{x_i,y_i,t_i}\right]^2 \\
&+\frac{1}{N}\sum_{i=1}^{N}\left[\left(\rho\left(\frac{\partial u_y^p}{\partial t} + u_x^p\frac{\partial u_y^p}{\partial x} + u_y^p\frac{\partial u_y^p}{\partial y}\right) + \frac{\partial p^p}{\partial y} - \mu\left(\frac{\partial^2 u_y^p}{\partial x^2} + \frac{\partial^2 u_y^p}{\partial y^2}\right)\right)\Bigg|_{x_i,y_i,t_i}\right]^2
\end{aligned} \tag{2}$$

The physics loss is calculated on a set of points that are known as collocation points (x_i, y_i, t_i). To improve resolution and accuracy of the prediction, the collocation points are generated with the quasi-random Sobol

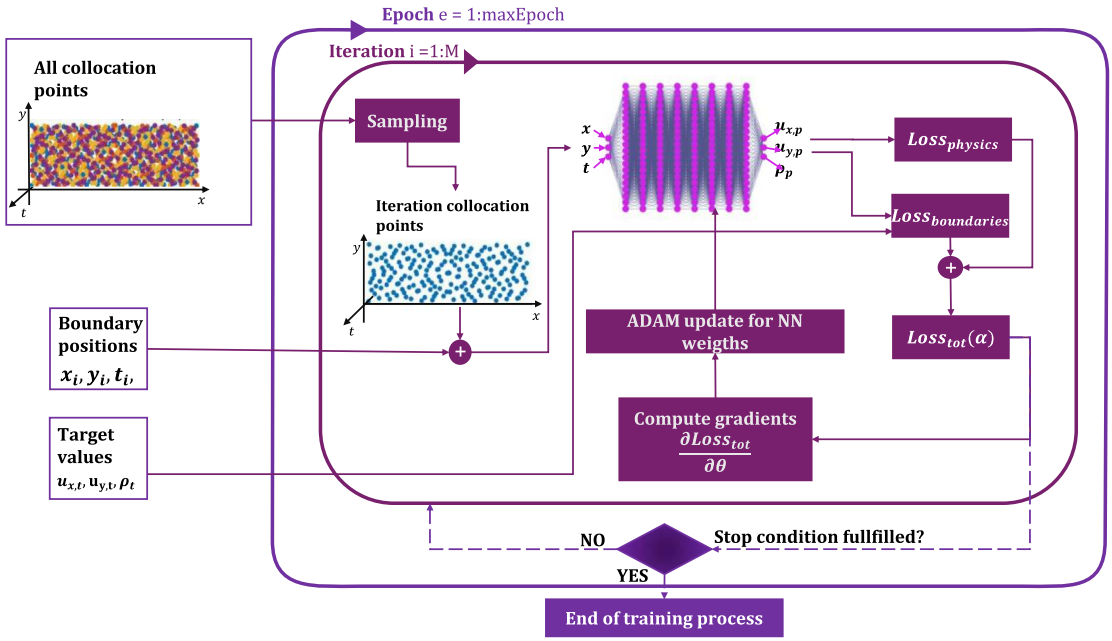

Figure 1. Schematic representation of PINN training.

[26, 27] sequence. Since the NN training is an iterative algorithm, for each iteration a 'minibatch' of N collocation points is selected. Note that this approach makes the PINNs a meshless method: in fact, the result is not a grid of values in specific region of the space, but a general function that given the coordinates return the variable. This approach offers advantages such as high-resolution reconstructions., as described in [1].

The boundary loss is formulated in the same way and can be written as:

$$\begin{aligned} L_{boundary} = & \frac{1}{M_{u_x}} \sum_{i=1}^{M_{u_x}} (u_x^p(x_i, y_i, t_i) - u_x^t(x_i, y_i, t_i))^2 \\ & + \frac{1}{M_{u_y}} \sum_{i=1}^{M_{u_y}} (u_y^p(x_i, y_i, t_i) - u_y^t(x_i, y_i, t_i))^2 \\ & + \frac{1}{M_p} \sum_{i=1}^{M_p} (p_x^p(x_i, y_i, t_i) - p_x^t(x_i, y_i, t_i))^2 \end{aligned} \tag{3}$$

Then, the loss to minimise is formulated as:

$$L_{tot} = L_{boundary} + \alpha L_{physics} \tag{4}$$

Where α is a hyperparameter that weights the importance of physics with respect to boundaries.

A schematic representation of a PINN algorithm is shown in figure 1 and its functioning can be resumed as follows: at the beginning of each epoch, the collocations points are divided into M minibatches and each minibatch is used in one iteration. For each iteration, the physical quantities are predicted at the minibatch and at the boundary positions, respectively used to calculate the loss boundary and the loss physics. So, the total loss is calculated and its gradient with respect to the neural network parameters θ is then calculated. The ADAM algorithm then uses this gradient to update the neural network's weights. At the end of each epoch, the loss is used to evaluate the stop condition (different stop conditions can be used) and, if the stop condition is satisfied, the training stops, and the neural network is considered trained.

2.2. A new PINN for uncertainty prediction with adaptive weighting

In order to introduce both the uncertainty prediction and the adaptive weighting scheme for our PINN model, it is worth considering a simple but representative case study, i.e. the 1D Poisson equation:

$$\lambda \frac{\partial^2 u(x)}{\partial x^2} = f(x) \tag{1a}$$

We consider a specific case where the analytical solution is known, $u(x) = \sin^3(6x)$, such that the various comparisons in the following will be done with the actual solution where no numerical errors are present. The diffusion coefficient λ has been set equal to 0.01 and then $f(x)$ is 108 $\lambda \sin(6x)(2\cos^2(6x) - \sin^2(6x))$.

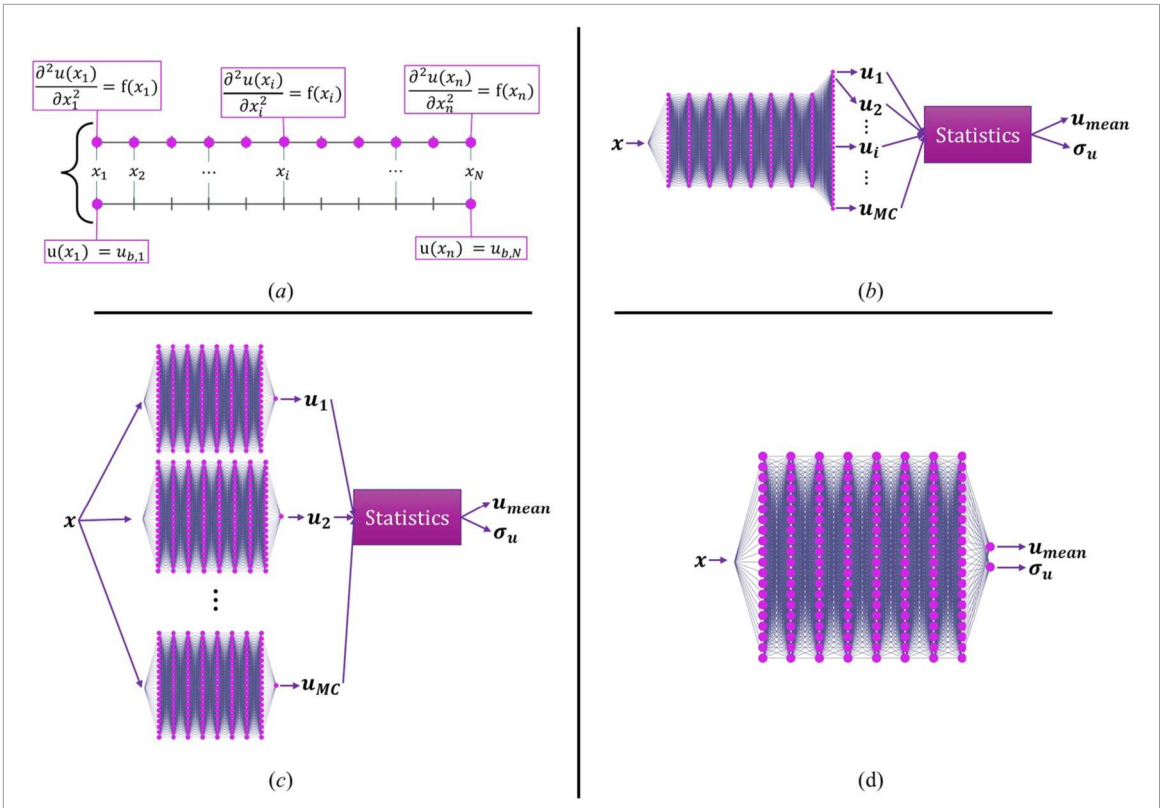

Figure 2. Schematic comparison of FEM solver (a), MO-PINN (b), MC-PINN (c) and ML-PINN (d).

Let us now consider the case where we know $f(x)$ in any point and we measure the quantity u in two points: $u_{b,1} = u(x_1)$ and $u_{bnN} = u(x_N)$. Let us suppose that we want to reconstruct this quantity in the domain ranging from $x_1 = -0.7$ to $x_N = 0.7$. Given the simplicity of the problem, we can use the standard Finite Element Method (see figure 2(a)), or FEM, to reconstruct the quantity u. The two extreme values of u are supposed to be measured and then affected by an uncertainty σ_m. Therefore, we may also be interested in predicting the uncertainty of such predictions. A standard approach is the one based on MonteCarlo. We can solve the same problem M times, perturbing both boundaries with their respective uncertainties each time, resulting in M predictions slightly different. These M predictions are then used to calculate the mean and standard deviation of the prediction (or, eventually, the pdf of the quantity as a function of x).

Such a procedure is at the base of the MonteCarlo PINN (MC-PINN) and the Multi-Output PINN (MO-PINN), whose architectures shown in figures 2(c) and (b) respectively. In the first case (MC-PINN), the procedure is exactly the same of the one described before: different PINNs are trained by perturbing the boundaries and then used for a statistical evaluation. The MC-PINN is for sure the best method from a theoretical point of view, but requires huge computational times, limiting its application to simple problems or to researchers with uncommon computational resources. The MO-PINN provides a trade-off between accuracy and computational load, since there is only one PINN with M output layers. Then, the neural network is trained taking into account the M outputs which are requested to fit M perturbed boundaries.

The method proposed in this work applies a different procedure: instead of performing M predictions and then the statistics, it directly predicts the statistics, i.e. the mean and the standard deviation (figure 2(d)). This enormous simplification in terms of computational load (we go from M PINNs or M neurons in the output layer to 1 PINN with 2 outputs per physical quantity) is paid in terms of modelling of the uncertainty. In fact, a new physics or regularisation loss must be introduced for the prediction of the uncertainty. Unfortunately, the uncertainty propagation in PDEs is not known in most cases. Therefore, being the model not known, a regularisation equation can be used. Typical regularisation equations are the ones based on the minimisation of the sum of squares of the first (g_1) or second derivatives (g_2) of the uncertainty, that for our problem are:

$$g_1(x) = \frac{\partial \sigma_u}{\partial x}$$
$$g_2(x) = \frac{\partial^2 \sigma_u}{\partial x^2} \tag{2a}$$

This type of regularisation will involve that, as discussed in the next sections, the uncertainty will be just an interpolation between the uncertainties of various measurements. Despite such an approach does not give the

actual uncertainty of the prediction, in several cases it is a good approximation. Moreover, this methodology combined with the adaptive weighing scheme shown in the next paragraph allow for innovative methodologies to threat noisy data and outliers. Therefore, defined our regularisation equation for the uncertainty, our minimisation problem could be written like:

$$Loss_{total} = Loss_{boundaries} + \alpha Loss_{physics} + \beta Loss_{\sigma} \tag{3a}$$

Where α is the weight for the physics loss and β the weight for the uncertainty loss. As it has been already analysed by other researchers [28], the weighting plays an important role at defining the accuracy of the PINN. Let us now consider a case where we have not just two boundary measurements, but we have ten equally spaced measurements (over constrained problem), all with an uncertainty equal to σ_m. Using the MC FEM approach, it is possible to reconstruct the quantity u varying α (while β is not needed for MC FEM and for the moment not considered).

It is essential to note that while the equations of physical variables are expected to be exact, the equation for the uncertainty is merely a regularization equation, which requires the uncertainty to be smooth within the domain. This has some drawbacks and limitations, especially for equations where it is expected that the uncertainty propagation is deeply non-linear. In such cases, only MonteCarlo-based approaches can return reliable predictions, inturn, with high computational costs. However, this limitation is overcome in cases like inverse problems, where many diagnostics constrain the uncertainty in different regions of the domain.

Figure 3 shows the results of the parametric analysis varying α. The first column shows the results for $\sigma_u^M = 0.01$, the second column for $\sigma_m = 0.1$ and the third column for $\sigma_m = 0.2$. The colormap represents the MSE (in logarithm scale) between the true solution and the reconstructed. The top row shows the results for the entire range of $Loss_{boundaries}$ and $Loss_{physics}$ explored, while the bottom row shows the results for the region where the optimum is reached. The dashed lines represent the value of the uncertainty imposed to the boundaries (σ_m). At first, it can be clearly noted that different values of α lead the method to converge to completely different combinations of $Loss_{boundaries}$ and $Loss_{physics}$, and this was expected from the analysis by [28]. But a second important consideration can be done: all the best reconstructions (i.e. the blue points, where the MSE is the lowest) are close to the dashed lines. Even this result is not surprising, since having a $Loss_{boundaries}$ lower than measurement uncertainty means that we are overfitting the measurements while a too large $Loss_{boundaries}$ means that we are accepting large error on measurements and therefore we are accepting one of the infinite solutions. This is true if not outliers are present and such a consideration will be deeply analysed in the next sections of the paper. This analysis suggests that we have a loss boundary target and that we can adapt the physics weight on this target (if we know that the uncertainty of my measurement is 0.1, we cannot imagine that the error between the prediction and the measurements is order of magnitude different). This is the principle on what our adaptive weighting scheme is based, as it will be shown in the next.

The loss for the boundaries (and measurements) can be written using the normalised negative log-likelihood:

$$Loss_{boundaries} = \frac{1}{MC\,N}\sum_{i=1}^{N}\sum_{j=1}^{MC}\left[\log\left(\frac{\sigma_{p,i}}{\sigma_{m,i}}\right) + \frac{(u_{p,i} - u_{m,i,j})^2}{2\quad\sigma_{p,i}}\right] \tag{3b}$$

Where $u_{m,i,j}$ is the j-th measurement of the i-th diagnostics. In many cases, it is impossible to have large MC values and sometimes only MC equal one is possible (e.g. in unsteady phenomena). In such cases, the same approach of MonteCarlo methodology is used: the measurement $u_{m,i}$ is the average value (this is the best estimation we have in the case) and MC perturbed values are generated with a gaussian random noise equal to $\sigma_{m,i}$. This is another key aspect of the authors' method: the target is perturbed at the beginning of each iteration through a systematic error, ε_{pert}, to be used as input where ε_{pert} is a Gaussian zero-mean error with standard deviation equal to $\sigma_{m,i}$. Therefore, the target for boundary/measurements is not a single value but a distribution of values.

The loss physics and the loss for the uncertainty are written with a standard MSE (equations (4) and (5)). They can be normalised by specific constants (C_p and C_σ) to make the loss comparable. The normalisation between physics/uncertainty equations (in case of systems) is of fully importance (see [19, 29, 30]). On the contrary, it is not important to normalise the physics/uncertainty losses with respect to the boundary loss since the weight of the various losses automatically change with the new adaptive scheme.

For our specific case, the two losses will be:

$$\begin{aligned} Loss_{physics} &= \frac{1}{C_p}\frac{1}{N}\sum\left(\lambda\frac{\partial^2 u_p(x)}{\partial x^2} - f(x)\right)^2 \\ Loss_{\sigma} &= \frac{1}{C_\sigma}\frac{1}{N}\sum\left(\frac{\partial \sigma_p}{\partial x}\right)^2 \end{aligned} \tag{4a}$$

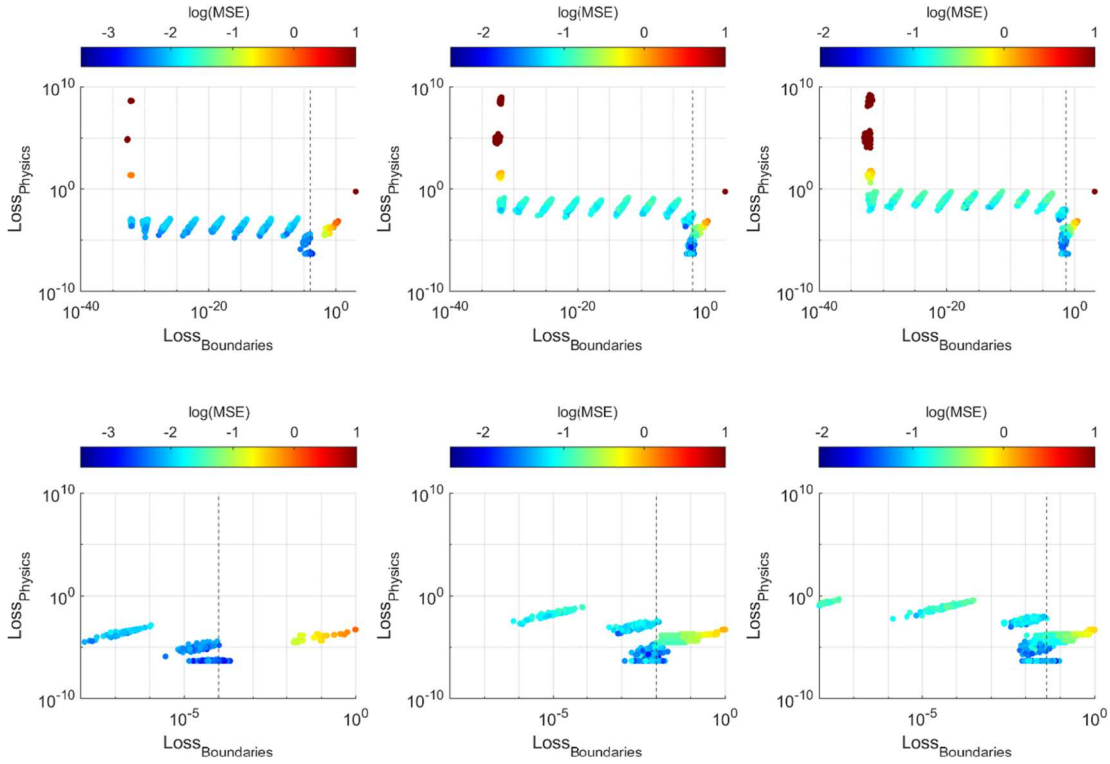

Figure 3. Parametric analysis of FEM reconstruction varying the weight α. The first row shows the results for the entire explored domain of the two losses, while the second row zooms in the plots where the minimum MSE are achieved. Each column refers to a case with a different uncertainty value (the dashed lines).

Finally, our total loss can be written as:

$$\begin{aligned} Loss_{tot} &= \frac{Loss_{boundaries} + \alpha \cdot Loss^{tot}_{physics}}{1+\alpha} \\ Loss^{tot}_{physics} &= \frac{Loss_{physics} + \beta \cdot Loss_{\sigma}}{1+\beta} \end{aligned} \tag{5}$$

As anticipated, α and β are not constant values but there are updated to ensure that the best result can be achieved.

The parameter α is updated to achieve the solution with the lowest physics loss when the boundary loss is (almost) equal to the target. The definition of the boundary loss as normalised negative log-likelihood together with the replication method led to an ideal target for the loss boundary ($Loss_{target}$) equal to 0.5 for most cases (see appendix B for derivation and section 3 for parametric analyses).

The parameter β has the role to balance the relative importance given to the uncertainty with respect to the physics equation such that, within an epoch, the PINN will try to minimise both losses. In fact, too high β will cause the PINN to disregard the physics while too low β will cause the PINN to predict unreliable uncertainty. However, β has a minor role with respect to α and, if the researchers can derive a good value for its specific problem, it may be set equal to this value without requiring automatic update. Generally speaking, if one does not want to implement an automatic update of β, we should expect that β is lower than one, since the uncertainty equation is based on a regularisation equation (and therefore the $Loss_{\sigma}$ target is larger than zero), while the physics is based on a true physical equation whose error is expected to be (ideally) zero.

From a computational point of view, the two hyperparameters are updated each epoch: α is updated by considering the difference between $Loss_{target}$ and $Loss_{boundary}$, while β changes as a function of the difference between $Loss_{physics}$ and $\beta Loss_{\sigma}$.

$$\begin{aligned} \Delta\alpha &= 0.1 \cdot \tanh\left[-2(\overline{Loss_{boundary}} - Loss_{target})\right] \\ \alpha_{epoch} &= \max\left(\alpha_{epoch-1}(1+\Delta\alpha),\ \alpha_{\min}\right) \end{aligned} \tag{6}$$

$$\begin{aligned} \Delta\beta &= 0.1 \cdot \tanh\left[-2(\overline{Loss_{physics}} - \beta\overline{Loss_{\sigma}})\right] \\ \beta_{epoch} &= \max\left(\beta_{epoch-1}(1+\Delta\beta),\ \beta_{\min}\right) \end{aligned} \tag{7}$$

Where the bar represents the mean value of the quantity over the entire epoch and $\alpha_{\min}$ and $\beta_{\min}$ are two constant minimal values for α and β (being also their initialising values) respectively set equal to 0.1 and 10^{-3}.

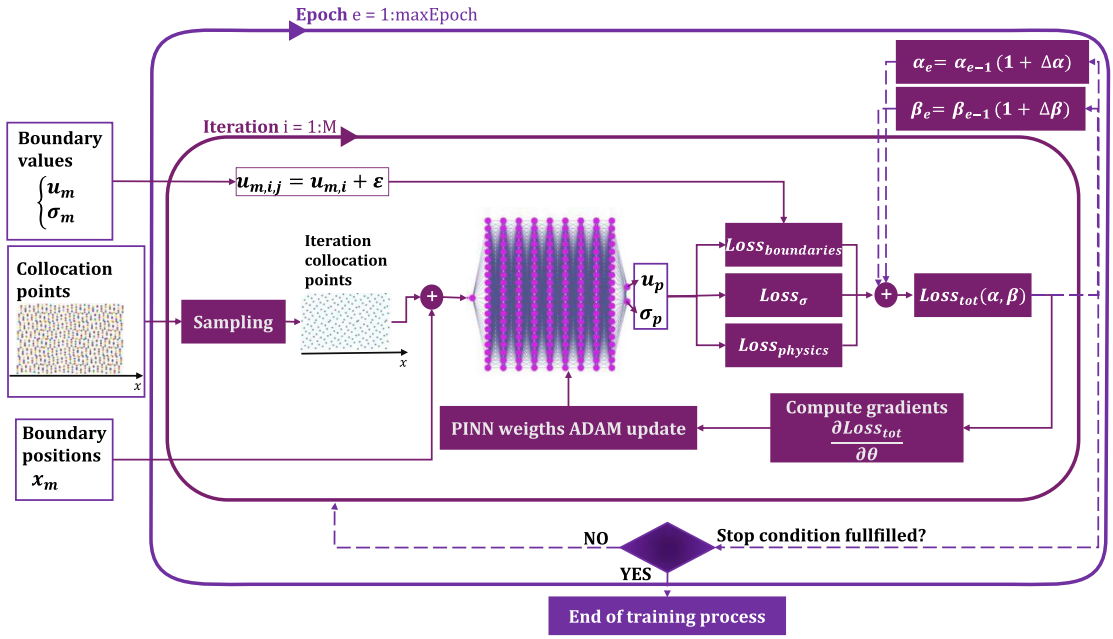

Figure 4. Schematic representation of the PINN algorithm proposed in this work.

It is worth highlighting that the adaptive weighting scheme can be easily scaled to other PINN architecture. This will be shown in section 4.2, where a standard PINN architecture is used.

The schematic representation of our PINN algorithm is shown in figure 4 and its functioning can be resumed as follows:

1. The collocations points are divided into M minibatches at the beginning of each epoch.
2. Each iteration sees one different minibatch that it is used to calculate the losses for the physics and the uncertainty. At the same time, MC perturbations of the N measurements are generated and the loss for the boundary is calculated. So, the total loss is evaluated and its gradients with resect to the loss is calculated. So, the net parameter update algorithm (in our case ADAM) computes the new net parameter and the iteration ends.
3. Once M iterations are made, one epoch is concluded, the average losses are evaluated, α and β are updated with the adaptive scheme and the epoch ends. If one of the convergence criteria are met, the training concludes, otherwise a new epoch starts.

3. Systematic analysis of the new method

This section aims at showing the potentialities of this new method in different conditions.

At first, the method is validated by comparing the results from the ML-PINN with the MC-FEM, the MC-PINN and the MO-PINN. Then, the adaptive weighting scheme is tested in a case where it is supposed that the knowledge of the measurement uncertainty is perfectly known (no faulty diagnostics). Then, an operational procedure for the loss target is proposed for real-world applications where faulty diagnostics are present.

3.1. ML-PINN method validation

The comparison between the methods is made for the 1D Poisson shown in equation (1). The domain for x ranges from -0.7 to 0.7, the $f(x)$ is assumed to be $108\ \lambda \sin(6x)(2\cos^2(6x) - \sin^2(6x))$, λ equals 0.01 and the two boundaries are assumed to be measured with uncertainty σ_m. The analytical solution, along with the boundaries, is shown in figure 5 (top-left). Both cases of uncertainty, σ_m equal to 0.01 and 0.1, are tested, and the same reconstruction points are sampled for comparison.

The four methodologies have been compared following this setup:

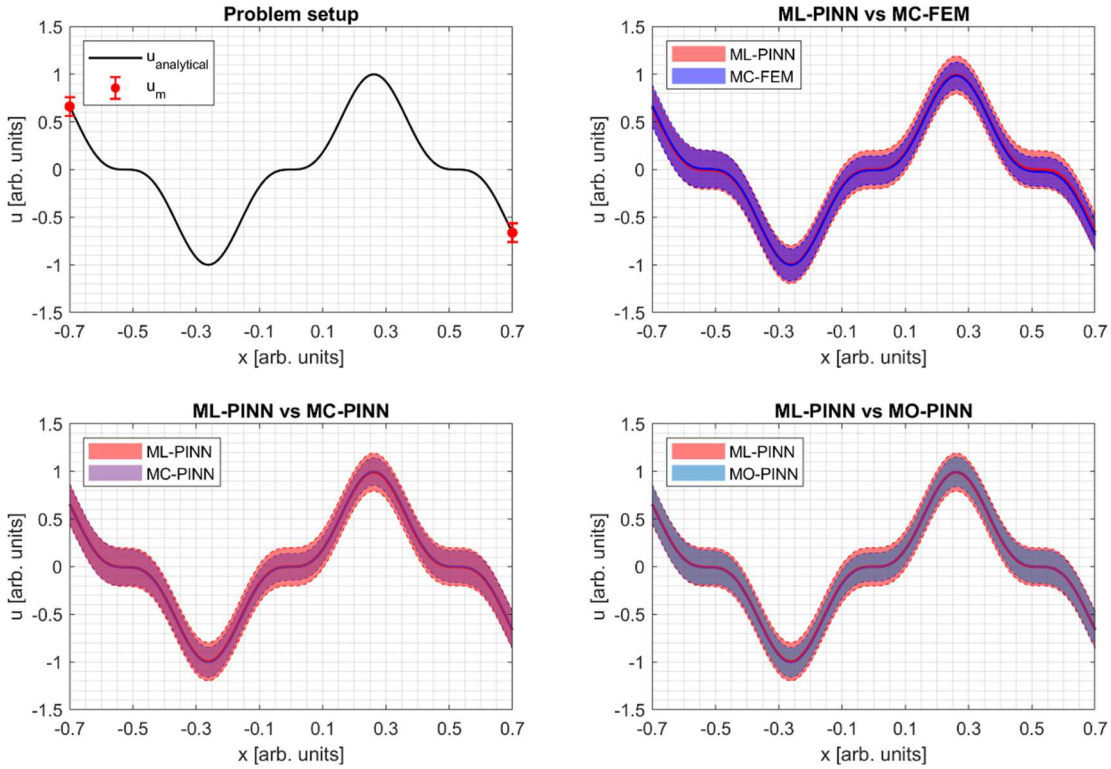

Figure 5. Analytical solution with boundary conditions for the method validation case (top-left). Comparison between the mean and standard deviation prediction of MC-FEM versus ML-PINN (top-right), MC-PINN versus ML-PINN (bottom-left) and MO-PINN versus ML-PINN (bottom-right). The central lines represent the average values, while the shaded areas are the confidence intervals of the predictions evaluated with two standard deviations.

1. The MC-FEM was obtained using the standard finite element method by discretizing the domain into 200 elements, and the statistical results were obtained by replicating the inversion 30 times, perturbing the boundaries by their uncertainties.
2. he MC-PINN is based on training 30 fully connected deep neural networks and performing the statistics similarly to the MC-FEM.
3. The MO-PINN has been set up with 30 output layers so that 30 different predictions of u are obtained with one neural network, by using different perturbed boundaries like the standard MC-FEM or MC-PINN. Again, the statistics of the 30 predictions are performed.
4. The ML-PINN, that has been trained as described in the previous section.

In these four cases, no adaptive weighting schemes have been used and all the weights have been set up equal to 1. Therefore, in this case the scope is to compare the uncertainty predicted by our model with other methodologies.

Figure 5 (top-right) compares the reconstructions from the MC-FEM with the ML-PINN, while figure 5 (bottom-left) and (bottom-right) compares the ML-PINN with MC-PINN and MO-PINN, when the uncertainty of the boundary conditions (σ_m) is equal to 0.1. All the figures suggest that the four methodologies return, qualitatively speaking, fully comparable reconstruction.

However, it is interesting to note that the ML-PINN has the lowest RMSE, indicating that the mean predicted quantity is closer to the analytical solution. Such a comparison is done in table 1, where both the normalised negative log-likelihood (NNLL) and the root mean square error (RMSE) calculated by comparing the analytical solution with the reconstructed quantities. In the table, the analyses for both measurement uncertainties equal to 0.01 and 0.1 are shown. At first, one can be note that all four methods return a NNLL below one, indicating that all the reconstructions statistically contain the analytical solution in one standard deviation. The ML-PINN has the highest NNLL, and therefore it is the least accurate. However, it is interesting to note that the ML-PINN has the lowest RMSE, indicating that the mean predicted quantity is closer to the analytical solution. Such an apparent contradiction can be easily explained by remembering that our reconstruction requires an equation for the uncertainty, which in our case was based on a regularization equation. This can be

Table 1. Comparison between MC-FEM, MC-PINN, MO-PINN and ML-PINN for.

		MC-FEM	MC-PINN	MO-PINN	ML-PINN
$\sigma_m = 0.01$	NNLL	0.61	0.91	0.87	0.92
	RMSE	7.36×10^{-3}	9.97×10^{-3}	9.39×10^{-3}	4.75×10^{-4}
$\sigma_m = 0.1$	NNLL	0.73	0.71	0.73	0.92
	RMSE	8.31×10^{-2}	8.12×10^{-2}	8.19×10^{-2}	4.56×10^{-3}

Table 2. Parametric analysis of the loss target for the ideal case.

Loss target	NNLL	RMSE	Loss boundary	Loss physics	Loss sigma	α
0.50	1.52	$1.79\,10^{-2}$	0.47	$4.20\,10^{-2}$	$4.20\,10^{-2}$	$1.5\,10^{-3}$
0.55	0.72	$9.50\,10^{-4}$	0.54	$2.60\,10^{-5}$	$2.50\,10^{-5}$	$2.7\,10^{6}$
0.75	0.50	$2.74\,10^{-3}$	0.71	$4.50\,10^{-5}$	$5.60\,10^{-5}$	$2.4\,10^{6}$
1.00	0.35	$1.31\,10^{-3}$	0.98	$2.40\,10^{-5}$	$2.47\,10^{-5}$	$2.4\,10^{7}$
10.00	15.18	$9.90\,10^{-2}$	1.01	$2.45\,10^{-5}$	$1.40\,10^{-5}$	$1.0\,10^{9}$

observed in the comparison plots in figure 5, where the uncertainty of the ML-PINN is always slightly larger than that of the other methods in the central region. Therefore, our approach better reconstructs the average quantity but has a worse estimation of the uncertainty. Of course, the RMSE of MC-based models would decrease by increasing the number of replications (which, in the present case, was set to 30), at the cost of computational time.

3.2. Adaptive weighting analysis

Given that our PINN can replicate with sufficient accuracy both the average value and the uncertainty of the predicted quantities, adaptive weighting is henceforth used (AW-ML-PINN), and various cases are investigated. The adaptive weighting method proposed by the authors requires the definition of the $Loss_{target}$. In section 2, it was anticipated that the lower limit for $Loss_{target}$ is 0.5. However, this is an ideal limit (see appendix A), and it is worth understanding how this loss target should be selected in real applications.

In this section, the AW-ML-PINN is analysed with the 1D Poisson model described before. Two types of analyses have been performed. In the first part, identified as 'ideal cases', the boundaries have the exact value of the analytical solution, but however our model assumes that they are affected by uncertainty. The second part, identified as 'real cases', investigates scenarios where the boundaries are affected by an error. In both parts, unless stated otherwise, it is assumed that the statistical uncertainties of the measurements are perfectly known. For both the 'ideal' and 'real' cases, a parametric analysis of AW-ML-PINN has been performed varying the hyperparameter $Loss_{target}$.

First, the ideal case is considered. The weight for the uncertainty relative to the physics loss (β) has been set constant and equal to one, so the method adapts only α. Five values of loss target have been analysed: 0.5, 0.55, 0.75, 1 and 10. The convergence criteria is the maximum number of epochs, set equal to 5000.

Table 2 presents the results of the parametric analysis. It can be observed that a loss target equal to the ideal target (0.5) does not allow for optimal convergence (larger NNLL and RMSE) and this was expected because of statistical considerations: setting the target equal to the ideal limit involves that the adaptive weighting scheme is not able to increase α since the adaptive method increases the weight only when the loss of the boundary is below the target (occurrence that it is statistically impossible). This is demonstrated by the fact that α is $\sim 10^{-3}$ at the end of the training, with consequent high loss physics and uncertainty. By slightly increasing the loss target (0.55), the method is now able to increase the weight, that reaches values around 10^6 after 5000 epochs. Such a huge increase of the weight while keeping the loss boundaries close to the ideal target allowed for very good reconstructions in terms of both RMSE and NLL. Such a good reconstruction is observed also for larger loss targets, like 0.75 and 1.00. However, it is important to remember that using larger loss targets means accepting one of the infinite solutions of the PDE with boundaries different from the measurement one. This is evident when a very large loss target is chosen: very bad reconstruction can be achieved even if the physics and uncertainty losses are very small (see results for loss target equal to 10.0 in the table).

A last but fundamental consideration by observing table 2 is mandatory. Using the adaptive scheme allows for reconstruction errors in terms of both NNLL and RMSE much smaller than the case with fixed weights (table 1). This demonstrates the first main advantage of our adaptive scheme, which clearly allows for improved

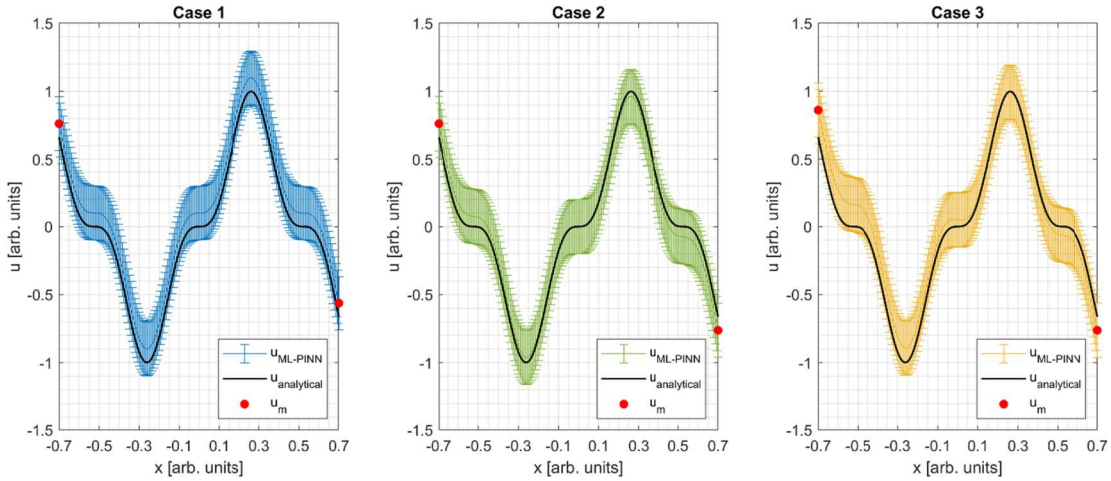

Figure 6. $u(x) \pm 2\sigma_u$ reconstruction for Case 1, Case 2 and Case 3 (from left to right).

reconstructions and PDE solving. Note that the reconstruction is better even if compared with the MC-based methodologies.

Three 'real' cases were investigated where the values of the boundaries are affected by an error. Let us consider the following notation: the analytical (true) values of the boundaries are denoted as $u_0 = u(x = -0.7)$ and $u_1 = u(x = 0.7)$ while the measured one as $u_{m,1} = u_m\ (x = -0.7)$ and $u_{m,2} = u_m\ (x = 0.7)$. In the first case, both boundaries overestimate the analytical solution by one standard deviation and therefore $u_{m,1} = u_{analytical,1} + \sigma_m$ and $u_{m,2} = u_{analytical,2} + \sigma_m$. The second case has errors with opposite signs: $u_{m,1} = u_{analytical,1} + \sigma_m$ and $u_{m,2} = u_{analytical,2} - \sigma_m$. The third case has opposite signs and the left boundary overestimate the value of two standard deviations $u_{m,1} = u_{analytical,1} + 2\ \sigma_m$ and $u_{m,2} = u_{analytical,2} - \sigma_m$.

Performing the parametric analysis of the $Loss_{target}$, yields a pattern more similar to the ideal case. Too large values will lead to a random solution of the infinite ones, while too low targets do not allow for physics loss minimisation. Figure 6 shows the predicted confidence intervals (2σ) for the three cases and it can be observed that in all three cases the analytical solution is contained inside. Both the NNLL and RMSE are obviously larger than the ideal case, since our model will find the solution with the perturbed boundaries. However, if the errors of the boundaries are in line with their known uncertainty, the models will predict confidence intervals containing the actual solution.

3.3. AW-ML-PINN applied to pathologic cases

In the previous two sections, we demonstrated that the AW-ML-PINN allows for reliable reconstructions, which means that the true solution is contained in the reconstructed confidence intervals. Naturally, the better the diagnostics are, the lower the error (both RMSE and NLL). However, in many real-world cases, perfect knowledge of the diagnostic characteristics is often unavailable, and anomalies (e.g., outliers) may be present, requiring complex analyses and ad hoc post-processing of the diagnostics.

In this section, we present three cases in which we measure the quantity u at ten different positions, with one exhibiting off-normal behaviour. Initially, it will be shown that a limited and guided parametric analysis allows for good convergence. Moreover, the predicted quantity and uncertainty enable easy detection of faulty diagnostics. Case 1 is characterised by the presence of an accurate (average error equal to zero) but non-precise measurement (large uncertainty), case 2 by the presence of a non-accurate (average error different from zero) but precise (correct uncertainty) measurement, and case 3 by the presence of a non-accurate non-precise measurement. The three cases are shown in the first row of figure 7.

The parametric analysis varying the loss target from 0.55 to 3.00 for the three cases is summarised in figure 7 and table 3. The results in the table are average values of three independent runs.

Let us begin with case 1. If there is a fault in the measurement and it is unknown, our expected ideal $Loss_{target}$ will be 0.5 (and thus a good real target would be 0.55). However, running the AW-ML-PINN with this target will not yield a good result, statistically speaking, as the measurement will show an error significantly larger than the expected standard deviation, as it is shown in figure 7 (second row, first column). In this case, the AW scheme will set α to zero, suggesting that the boundaries with the selected uncertainties do not match the physics. By increasing the target value to 0.75, the AW-ML-PINN will find a solution with $Loss_{boundary}$ equal to $Loss_{target}$, and it will increase α to very high values, resulting in very low physics and sigma losses, suggesting that this $Loss_{target}$ accepts solutions coherent with the physics. Moreover, it is possible to analytically demonstrate

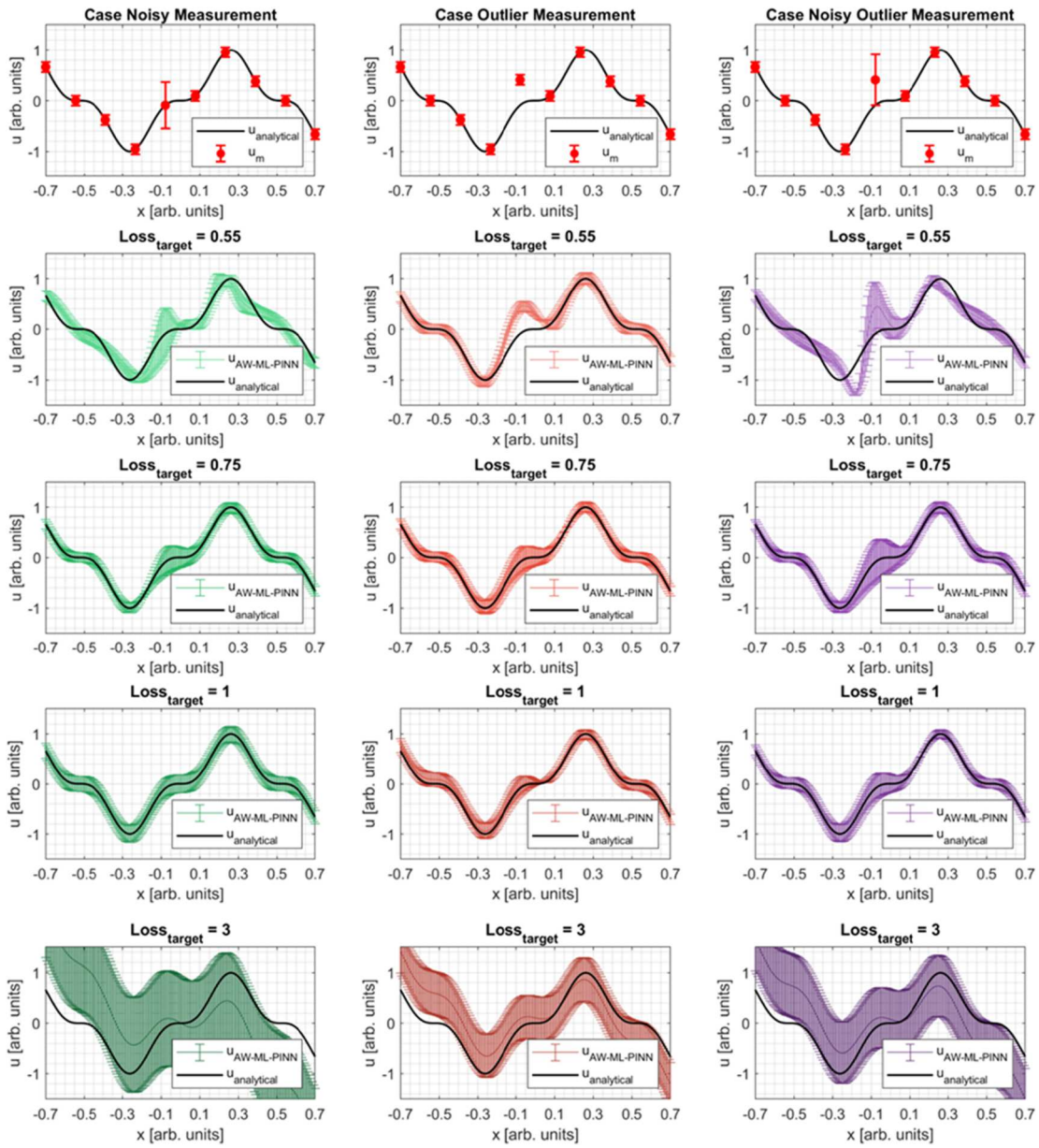

Figure 7. AW-ML-PINN applied to pathologic cases. The top row shows the three configurations analysed with the analytical solution and the measurements used. From the second row, the reconstructions using the AW-ML-PINN is reported for case 1 (first column), case 2 (second column) and case 3 (third column), using a loss target equal to 0.55 (second row), 0.75 (third row), 1.00 (fourth row) and 3 (fifth row).

that the ideal loss target in this case is larger than 0.5:

$$\begin{aligned} Loss_{boundary} &= \frac{1}{10}\sum\left[\log\left(\frac{\sigma^p_{u,i}}{\sigma^M_u}\right) + \frac{(u_{p,i}-u_{m,i})^2}{2\sigma^{p^2}_{u,i}}\right] \\ &= \frac{1}{10}\sum_{i=1}^{9}\left[\log\left(\frac{\sigma^p_{u,i}}{\sigma^M_u}\right) + \frac{(u_{p,i}-u_{m,i})^2}{2\sigma^{p^2}_{u,i}}\right] + \frac{1}{10}\left[\log\left(\frac{\sigma^p_{u,j}}{\sigma^M_u}\right) + \frac{(u_{p,j}-u_{m,j})^2}{2\sigma^{p^2}_{u,j}}\right] \\ &= \frac{9}{10}\left(\frac{1}{2}\right) + \frac{1}{10}\left(\log\left(\frac{\sigma^p_{u,j}}{\sigma^M_u}\right) + \frac{(\sigma^{M,fault}_u)^2}{2\sigma^{p^2}_{u,j}}\right) \end{aligned} \tag{8}$$

Where $\sigma^{M,fault}_u$ is the actual standard deviation of the fault diagnostic. The predicted standard deviation which minimises second term is exactly $\sigma^{M,fault}_u$ and therefore:

Table 3. AW-ML-PINN results of the parametric analysis for the pathologic cases.

Target	NLL	RMSE	Loss boundary	Loss physics	Loss sigma	α	β
Case 1 - Noisy measurement							
0.55	34.75	0.13	0.66	$2.89\,10^{1}$	$2.55\,10^{-2}$	$1.00\,10^{-6}$	1.00
0.75	0.97	$6.67\,10^{-4}$	0.75	$2.88\,10^{-5}$	$3.25\,10^{-3}$	$2.37\,10^{2}$	0.26
1	1.61	$6.38\,10^{-3}$	1.00	$2.56\,10^{-5}$	$2.00\,10^{-3}$	$3.25\,10^{7}$	$1.01\,10^{-5}$
3	2.36	0.38	1.89	$2.93\,10^{-5}$	$6.70\,10^{-4}$	$1.00\,10^{9}$	$9.44\,10^{-7}$
Case 2 - Outlier measurement							
0.55	47.91	0.13	0.55	1.62	$2.00\,10^{-3}$	0.11	1.00
0.75	1.20	0.01	0.75	$5.24\,10^{-5}$	$2.40\,10^{-3}$	334.86	0.39
1	3.81	0.05	1.00	$3.74\,10^{-5}$	$1.75\,10^{-3}$	$8.46\,10^{6}$	$5.98\,10^{-5}$
3	7.96	0.46	2.27	$3.96\,10^{-5}$	$2.03\,10^{-3}$	$8.77\,10^{8}$	$1.22\,10^{-2}$
Case 3 - Noisy and outlier measurement							
0.55	19.63	0.19	0.67	116.06	$4.30\,10^{-3}$	$1.00\,10^{-6}$	1.00
0.75	1.08	0.01	0.75	$3.66\,10^{-5}$	$5.95\,10^{-3}$	107.44	$1.71\,10^{-2}$
1	1.09	0.02	0.99	$2.62\,10^{-5}$	$2.26\,10^{-3}$	$1.79\,10^{7}$	$1.42\,10^{-3}$
3	2.51	0.32	2.57	$3.32\,10^{-5}$	$3.75\,10^{-4}$	$1.00\,10^{9}$	$2.66\,10^{-6}$

$$\begin{aligned} Loss_{target} &= \frac{9}{10}\left(\frac{1}{2}\right) + \frac{1}{10}\left(\log\left(\frac{\sigma_u^{M,fault}}{\sigma_u^M}\right) + \frac{1}{2}\right) \\ &= \frac{1}{2}\left(0.5 + \frac{1}{10}\log\left(\frac{\sigma_u^{M,fault}}{\sigma_u^M}\right)\right) \sim 0.73 \end{aligned} \tag{9}$$

In this case, since the theoretical standard deviation was 0.1 while the actual standard deviation of the fault diagnostic is 1, the ideal $Loss_{target}$ is 0.73, explaining why 0.55 was not a good target, while 0.75 allows for good convergence. The reconstruction obtained with the AW-ML-PINN for case 1 and $Loss_{target}$ is 0.75 is shown in figure 7 (first column, third row). It is worth highlighting that the analysis of the predicted standard deviation allows for the direct identification of the faulty diagnostic.

Case 2 represents a more complex scenario in real applications. In fact, while case 1 is quite easy to identify since the uncertainty of one diagnostic can be easily measured by performing repetitive measurements in steady conditions, offsets are much more difficult to identify. However, by using the AW-ML-PINN and by changing the $Loss_{target}$ until the convergence is reached allows rapid convergence. Even in this case, the ideal loss target in the presence of one outlier can be evaluated (see appendix B for all steps):

$$Loss_{target} = \frac{1}{2}\left(0.5 + \log\left(\frac{\sqrt{(\sigma_u^M)^2 \quad + (\Delta u_{offset})^2}}{\sigma_u^M}\right)\right) \sim 0.66 \tag{10}$$

The results for this case are resumed in figure 7 (central column) and table 2. As for the previous case, a smaller target than the ideal target (that for this case with the outlier is around 0.66) requires a non-physical solution, meaning that α saturates at low values (0.11) and the physical loss remains high (1.62). This leads to un unphysical solution with high NNLL and RMSE. By applying a target equal to 0.75 (which is larger than the ideal target), a good reconstruction is achieved (the best one in terms of NNLL, equal to 1.20). The parameter α increases to high values, resulting in a very small physics loss ($\sim 10^{-5}$). Increasing the targets to 1 and 3 causes the reconstruction accuracy to decrease in terms of both NNLL and RMSE. Again, this is due to the fact that the PINN is allowed to have boundary conditions significantly different from the measurement values. It is worth noting that simply comparing the reconstructed diagnostics (i.e., the diagnostic values given by the reconstructed field) with the measurements allows for easy identification of faulty diagnostics, indicating that this method can also be used for outlier or fault detection. Such a feature will be demonstrated also in section 4.1 for surface averaged measurements.

The last pathologic case (case 3) is just a combination of the two previous cases. Different behaviours are not theoretically expected or experimentally observed, as shown in figure 7 (third column) and table 3. Again, too low loss targets do not allow for physical solutions, while too large targets make solving equivalent to solving a poorly constrained problem. This is also demonstrated by the standard deviation of the parametric analysis here reported. Running case 3 three times with a good loss target (0.75), the NNLL standard deviation is around 0.06 (~5% of the average value), while using a loss target equal to 3 leads to an NNLL variability of 0.84 (~33% of the average value).

Further analyses to stress test the methodology in presence of different levels of noise are discussed in appendix D.

4. Real application-inspired cases

From the theoretical consideration in section 2 and the parametric analysis in section 3, the following considerations can be summarised:

1. The ML-PINN can allow for accurate reconstruction of both average and standard deviation of the quantities, mostly comparable with MC-based methods, without the need for high computational cost.
2. The proposed adaptive weighting scheme allows for improved reconstructions. Moreover, the unique hyper-parameter, the loss target, can be analytically determined if all relevant information is known. If it is not the case, a simple iterative procedure can be used simply until the AW scheme increases the weight α.

Despite the results of section 3 are very positive, the method has been applied to a quite simple problem: only one physical quantity to predict, local measurements, and one partial differential equation. Therefore, we decided to test our methodology in more complex situations, through synthetic cases that mirror real-world applications. The three applications have been thought to show the applicability of the method with different physics: a typical problem in heat transfer, one in fluid dynamics, and one in magneto-hydrodynamic (MHD) have been considered. Moreover, each problem deal with different typical issue in physics and engineering.

4.1. Heat transfer: surface-averaged data and outliers

The first case is based on the heat transfer physics in the presence of missing information (some boundaries will be assumed to be unknown). The problem will be closed by measuring some variables which will present some limitations: they will be affected by noise, there will be some outliers, and the diagnostic will provide surface-integrated measurements.

We consider a metallic plate subjected to a heat flux (figure 8(a)) on the left side (x = 0), while both the upper side (y = 0.5) and the lower side (y = 0) are under adiabatic conditions, where the temperature gradient is zero across the boundaries ($\partial T/\partial y = 0$). The right side (x = 1) is supposed to be isothermal with a temperature T equal to 300 K. The plate is made of a homogeneous material with a thermal conductivity of $k = 100\ \mathrm{Wm^{-1}K^{-1}}$. The analytical solution is reported in figure 8(c). Moreover, we assume the presence of a thermal camera [31] with a finite resolution (20×10 pixels) with a gaussian proportional error of 1%. The measurement of each pixel of the camera is simulated as the surface-integrated measurement (figure 8(d)) of the analytical temperature solution according with the following equation:

$$T_{\mathrm{int}} = \int T_{analytical} \cdot W_{pixel}\, dS_{pixel} \tag{11}$$

Where W_{pixel} is an opportune matrix that represents view factors of the surface integrated measurement, $T_{\mathrm{int},k}$ is the thermal-camera measurement of the pixel k and $\sigma_{\mathrm{int},k}$ is its uncertainty. However, we supposed that some pixels have some faults and therefore they are affected by an offset equal to 5 $\sigma_{\mathrm{int},k}$.

Aim of this first case is to demonstrate that given the thermal-camera measurements, the physics equation describing the phenomenon (i.e. heat Fourier equation in equation (11)) and the knowledge that the top and bottom contours are adiabatic, we can easily identify the broken pixels and accurately reconstruct the true solution.

Figure 8 (b) shows the architecture of the neural network used for the AW-ML-PINN. Inputs are the spatial coordinates x and y while the outputs are the two variables, i.e. the temperature (T_p) and its uncertainty ($\sigma_{T,p}$).

The measurements from the thermo-camera are used to calculate the $Loss_{boundaries}$:

$$Loss_{boundaries} = \frac{1}{N_{pixels}} \sum \left[\log\left(\frac{\sigma_{\mathrm{int},p,i}}{\sigma_{\mathrm{int},m,i}} \right) + \frac{1}{2} \frac{(T_{\mathrm{int},m,i} - T_{\mathrm{int},p,i})^2}{\sigma^2_{\mathrm{int},p,i}} \right] \tag{12}$$

Where $T_{\mathrm{int},p,i}$ is the expected i-th thermo-camera pixel given the predicted temperature field (calculated with equation (11)) and $\sigma_{\mathrm{int},p,i}$ is the predicted uncertainty of the pixel given temperature uncertainty field, calculated using the uncertainty propagation theory. The condition of adiabatic bottom and top walls has been put inside the loss physics together with the diffusion equation. It is worth noting that the adiabatic condition can also be included in the boundary loss function, simply following the considerations in appendix C. The loss of the uncertainty is based on the minimisation of the sum of the second derivatives (implying a diffusive uncertainty regularisation).

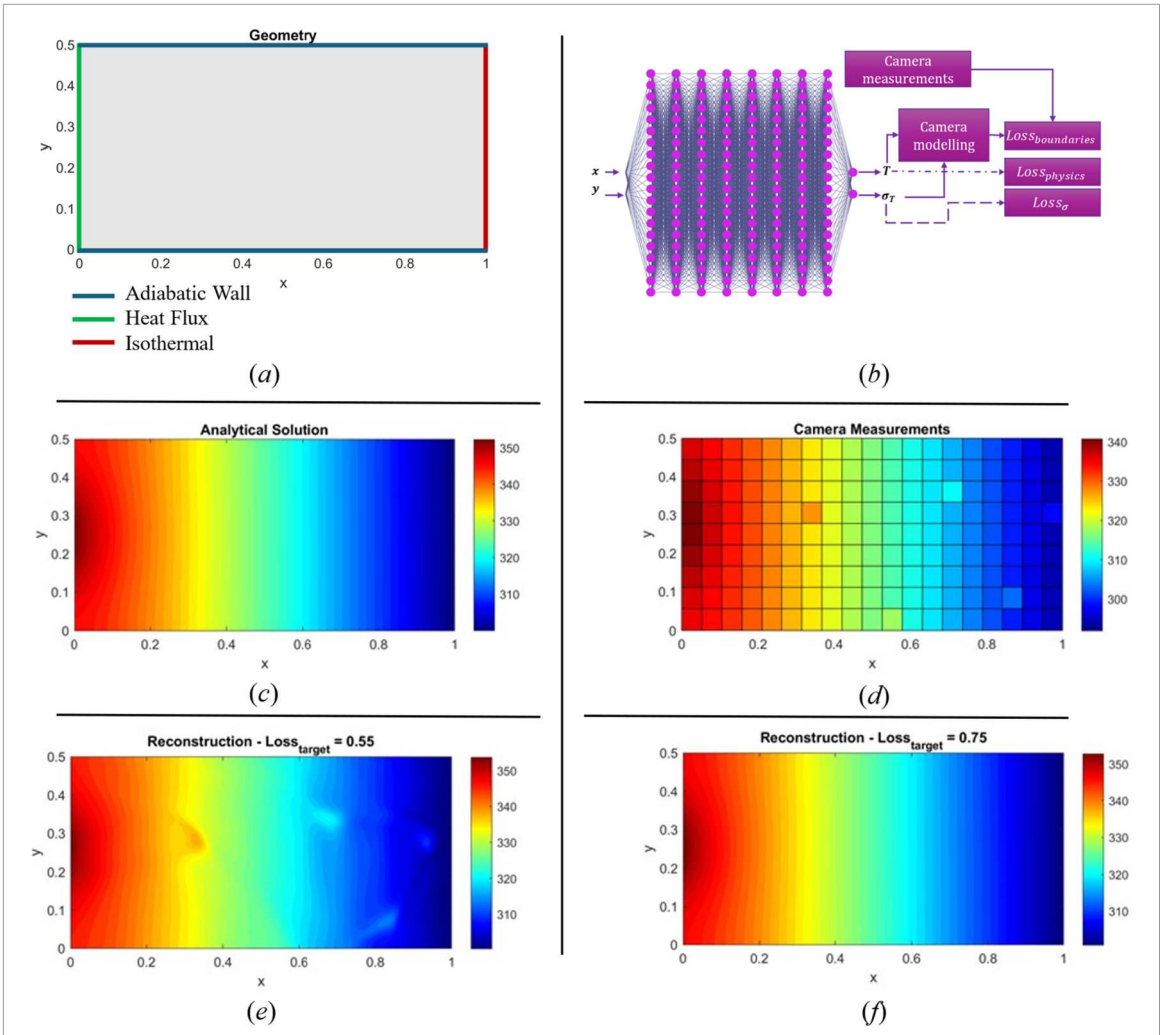

Figure 8. Heat transfer case: geometry and boundary condition of the problem (a), PINN architecture (b), target solution of the problem (c), simulated thermal camera measurements with outliers (d), and temperature field reconstruction using a target equal to 0.55 (e) and 0.75 (f).

$$Loss_{physics} = \frac{1}{N_{edge}\left(\frac{\Delta T}{\Delta L_y}\right)^2}\sum\left[\left(\frac{\partial T_p}{\partial y}\bigg|_{y=0}\right)^2 + \left(\frac{\partial T_p}{\partial y}\bigg|_{y=0.5}\right)^2\right] + \frac{1}{N_{physics}\left(\frac{\sigma_T^{\min}}{\Delta L_x^2}\right)^2}\sum\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right)^2 \tag{13}$$

$$Loss_{\sigma} = \frac{1}{N_{physics}\left(\frac{\sigma_T^{\min}}{\Delta L_x^2}\right)^2}\sum\left(\frac{\partial^2 \sigma_T}{\partial x^2} + \frac{\partial^2 \sigma_T}{\partial y^2}\right)^2 \tag{14}$$

$$Losses = \frac{Loss_{boundary} + \alpha \cdot \frac{Loss_{physics} + \beta \cdot Loss_{\sigma}}{1+\beta}}{1+\alpha} \tag{15}$$

Both the heat transfer loss (last term of equation (13)) and the uncertainty loss have been calculated in generated collocation points (using Sobol distribution [26, 27]). The adiabatic conditions are calculated in a fixed grid of points on both the bottom and top edges. The boundary and the physics losses have been normalised by some available quantities in the real application. The adiabatic condition has been normalised by the average vertical derivative of the temperature measured by the thermal camera, while the heat transfer and the uncertainty losses have been normalised by the minimum uncertainty value of the thermal camera $\sigma_T^{\min}$ (while the term ΔL_x^2 is used to make formally dimensionless the equation but it is not quantitatively relevant in this example being equal to one).

From an operative point of view, at the beginning we supposed to not know that there are outliers and therefore we set the ideal loss target ($Loss_{target} = 0.55$). The reconstructed field with this target is reported in

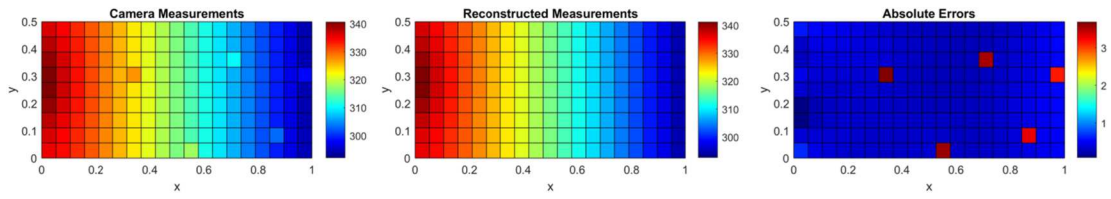

Figure 9. Simulated (left) and reconstructed (middle) thermal camera measurements. The absolute difference (right) allows for the detection of the five faulty pixels. Reconstructed measurements are based on the results obtained with loss target equal to 0.75.

figure 8(e). It can be shown that it substantially differs from the analytical solution of figure 8(c), showing some unexpected hot spots. This is due to the outliers of the thermal camera, whose measurements are reported in figure 8(d). In fact, asking an ideal loss target in presence of outliers involves that outliers are overfitted. But the question would be: in a real case, where I have not the analytical solution, how I can identify this problem? The answer is in the analysis of the other losses. In fact, our adaptive scheme with a too strict loss target was forced to put α to very small values (0.013), allowing the PINN to disregard the physics equations. In fact, the loss physics at the end of the training is 5.13, a too large value considering that its ideal target should be zero. The RMSE and NNLL for this case are 0.73 ($\pm$0.60) K and 1.28 ($\pm$0.07) respectively. The maximum error between analytical and reconstruction is 6.5 K.

A consequent operative approach would be to increase the loss target. For the second reconstruction attempt, the loss target was set to 0.75. The result is shown in figure 8(f). What can be noted is that the hot spots of the previous reconstruction disappeared and that the reconstructed field is much closer to the true solution. Moreover, in this case α reached 1130.3 at the end of the training (stop condition was reached for epochs equal to 5000), while the physics loss is 4.36 10^{-5}. This suggest that the PINN was able to significantly reduce the loss on the physics with respect to the previous case, preventing outliers' overfitting and resulting in a more reliable solution. In this case, the RMSE is 0.40 ($\pm$0.11) K with a maximum error of 1.21 K. The NLL is 1.45 ($\pm$0.002). It is worth highlighting that with this approach we were able to reconstruct the temperature field with a resolution much larger than the one offered by the thermal camera, typical advantage of the PINNs [10].

Moreover, by using the temperature field reconstructed with $Loss_{target} = 0.75$, we are able to detect faulty pixels. In fact, one may calculate the absolute error between the predicted and the measured thermal camera pixels, clearly observing that there are five pixels that exhibits very large errors and therefore identifiable as outliers or faulty pixels (figure 9).

4.2. Navier–stokes: direct modelling from experimental data

In this case, we examine a two-dimensional time-dependent fluid dynamics problem. We consider the flow of an incompressible fluid through an orifice in a controlled environment [32], as shown in figure 10(a). Initially, the fluid is at rest, and the flow begins at $t_0 = 0s$. The system is observed for a total of 10 s, and at each second, a set of 20 $\times$ 10 velocity measurements is recorded (simulating, for instance, a low-resolution Particle Image Velocimetry, or PIV, diagnostics) along with two pressure measurements, one above and one below the orifice. The measurement points for both velocity and pressure are shown in figure 10(a).

The true solution, obtained with a Computational Fluid Dynamics (CFD) solver, was used to generate the measurements of both PIV [33] and pressure probes. Random Gaussian noises of 0.05 m s^{-1} and 0.01 Pa have been added to the velocity and pressure measurements. The CFD solves the incompressible Navier–Stokes equation. The fluid density was set equal to 1 kg m^{-3}, while the viscosity to 0.001 Pa s. No slip conditions have been imposed on the walls while constant pressure of 0 Pa has been imposed for x equal to 1 m. The inlet velocity has been imposed with the following equation:

$$u_{inlet} = \frac{r_{orifice}^2 - y^2}{r_{orifice}^2} \min(t,\ 1) \tag{16}$$

In this case, we assume that we know the boundary and initial conditions of our problem, the governing equations, and the measurements from the PIV and pressure probes, but we do not know the viscosity of the fluid. This represents a case of 'incomplete physics' where a fundamental physical parameter is missing. Our goal is to develop a PINN-based model to reconstruct the velocity and pressure fields and estimate the viscosity of the fluid.

Instead of directly predicting the velocity, we predict a variable φ such that the velocity is the rotor of φ. This will automatically imply that the conservation of mass is satisfied, reducing our physics losses to the conservation of momentum. Moreover, the neural network is asked to predict the pressure as a function of spatial

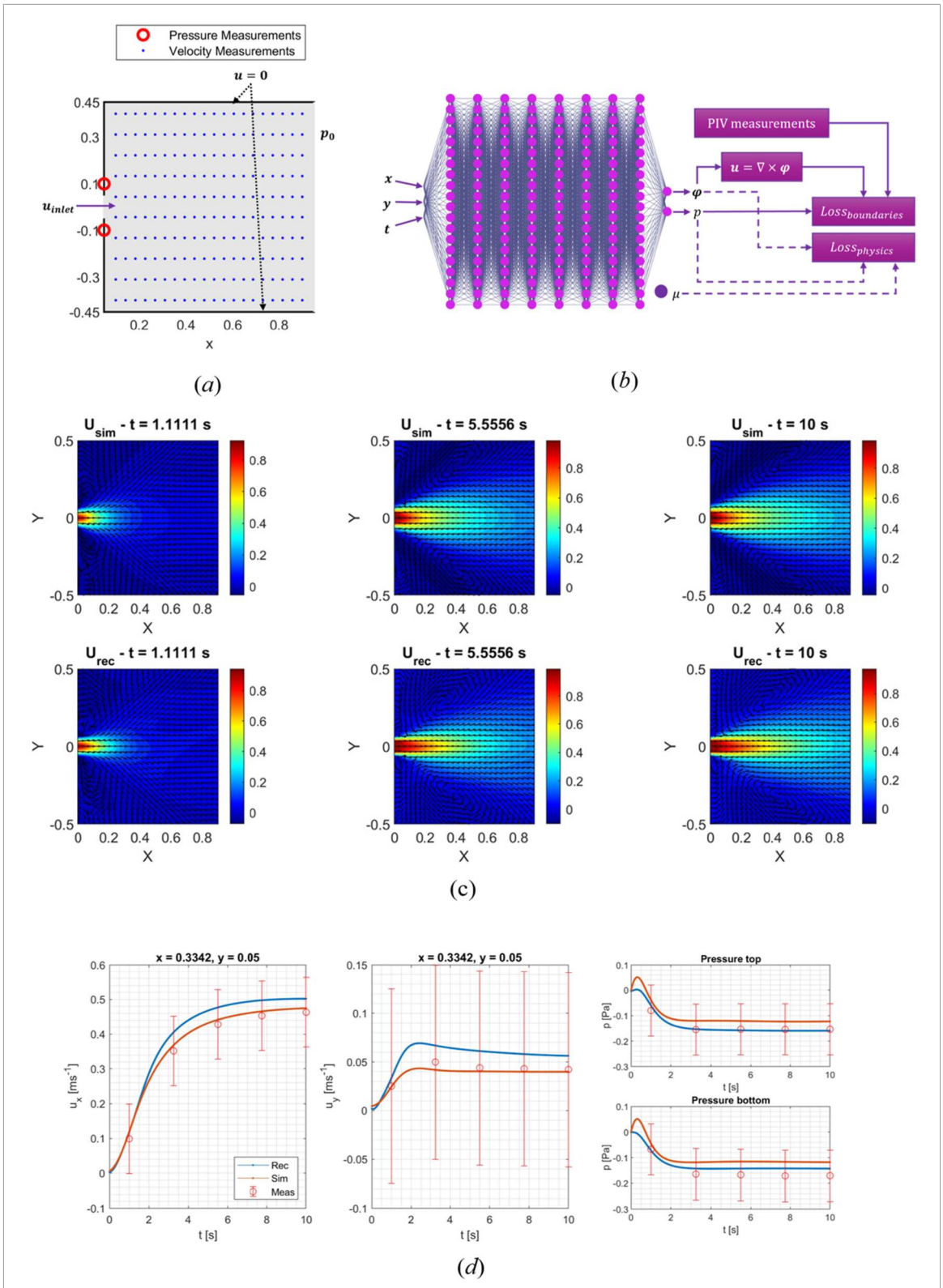

Figure 10. Fluid dynamics case: geometry and boundary conditions of the problems (a), PINN architecture (b), simulated versus reconstructed velocity field at different time instants (c), and simulated versus reconstructed velocity and pressure as a function of time (d).

and time coordinates and a constant parameter that represents the dynamic viscosity of the fluid μ_p. The neural network architecture is shown in figure 10(b).

In order to show the applicability of the adaptive weighting scheme to other types of PINN, in this example we used a standard methodology where the uncertainty is not predicted. In such a case, one can modify the normalised negative log-likelihood by substituting the predicted uncertainty with the measured one:

$$Loss_{boundary} = \frac{1}{N_{PIV}}\sum\left[\frac{1}{2}\frac{(u_{x,m}-u_{x,p})^2}{\sigma_u^2} + \frac{1}{2}\frac{(u_{y,m}-u_{y,p})^2}{\sigma_u^2}\right] + \frac{1}{N_{probe}}\sum\frac{1}{2}\frac{(p_m-p_p)^2}{\sigma_p^2} \tag{17}$$

Unsurprisingly, we obtain the classical normalised MSE formulation. his modified loss retains the same properties as the methodology described in previous sections of the paper. The physics loss contains both the ideal boundary conditions, assumed to be certain, and the two momentum equations:

$$Loss_{physics} = Loss_{NS,x} + Loss_{NS,y} + Loss_{inlet} + Loss_{outlet} + Loss_{wall} \tag{18}$$

Where:

$$Loss_{NS,x} = \frac{1}{N_p\left(\frac{p_0}{\Delta L}\right)^2}\sum\left[\rho\left(\frac{\partial u_{x,p}}{\partial t} + u_{x,p}\frac{\partial u_{x,p}}{\partial x} + u_{y,p}\frac{\partial u_{x,p}}{\partial y}\right) + \frac{\partial p_p}{\partial x} - \mu_p\left(\frac{\partial^2 u_{x,p}}{\partial x^2} + \frac{\partial^2 u_{x,p}}{\partial y^2}\right)\right]^2 \tag{19}$$

$$Loss_{NS,y} = \frac{1}{N_p\left(\frac{p_0}{\Delta L}\right)^2}\sum\left[\rho\left(\frac{\partial u_{y,p}}{\partial t} + u_{x,p}\frac{\partial u_{y,p}}{\partial x} + u_{y,p}\frac{\partial u_{y,p}}{\partial y}\right) + \frac{\partial p_p}{\partial y} - \mu_p\left(\frac{\partial^2 u_{y,p}}{\partial x^2} + \frac{\partial^2 u_{y,p}}{\partial y^2}\right)\right]^2 \tag{20}$$

$$Loss_{inlet} = \frac{1}{N_{inlet}\,\sigma_u^2}\sum\left[(u_{x,inlet} - u_{x,p})^2 + (u_{y,p})^2\right] \tag{21}$$

$$Loss_{outlet} = \frac{1}{N_{outlet}\,\sigma_p^2}\sum(p_p)^2 \tag{22}$$

$$Loss_{wall} = \frac{1}{N_{wall}\,\sigma_u^2}\sum\left[(u_{x,p})^2 + (u_{y,p})^2\right] \tag{23}$$

And therefore, the total loss to minimise is just:

$$Loss = \frac{Loss_{boundary} + \alpha \cdot Loss_{physics}}{1+\alpha} \tag{24}$$

The training is done by imposing a loss target equal to 0.55 and, since there were not outliers or faulty diagnostics, this training converged directly to a very good reconstruction.

The RMSE between the numerical simulation and the PINN reconstruction is 0.034 ($\pm$0.013) m s^{-1} for u_x, 0.016 ($\pm$0.004) m s^{-1} for u_y, and 0.033 ($\pm$ 0.011) Pa for the pressure. Moreover, the PINN estimated the fluid's dynamic viscosity as 0.0012 Pa s, representing a 20% error relative to the true value (0.001 Pa s), a good result considering that the uncertainty on the velocity measurements is equal to 0.05 m s^{-1}, while the natural variability of the velocity field is 0.13 m s^{-1} (~38% measurement uncertainty).

Figure 10(c) shows the comparison between the simulated and reconstructed velocity field (the colourmap represents the magnitude of the velocity, while the black arrows the direction), while figure 10(d) shows the numerical versus reconstructed time traces of the velocity and pressure together with the available measurements in that locations.

4.3. MHD: reconstruction of hidden quantities

The last example is based on the ideal steady-state static magneto-hydrodynamic (MHD) physics:

$$\begin{aligned}\nabla\cdot\mathbf{B} &= 0\\ \mu_0\mathbf{J} &= \nabla\times\mathbf{B}\\ \nabla p &= \boldsymbol{J}\times\boldsymbol{B}\end{aligned} \tag{25}$$

We consider a simplified example of magnetically confined plasma in nuclear fusion [14, 34, 35]. Specifically, we examine a 1D solution in cylindrical coordinates, assuming that all derivatives along θ and z are equal to zero. We suppose a case of a plasma subjected to an axial magnetic field B_z and in which circulates a plasma current J_z that originates a poloidal magnetic field $B_\theta(r)$. The magnetic field is therefore defined as $\boldsymbol{B} = (0,\ B_\theta,\ B_z)$.

In typical tokamak plasmas, we can obtain internal measurements of the pressure (for example, electron pressure measured by Thomson scattering) but not internal measurements of the magnetic field. Therefore, we apply PINN to a scenario where internal quantities are reconstructed from external (or boundary) measurements. The solution adopted is based on a typical magnetic field profile commonly used in tokamaks research [36] and the target or true quantities are reported in figure 11(a).

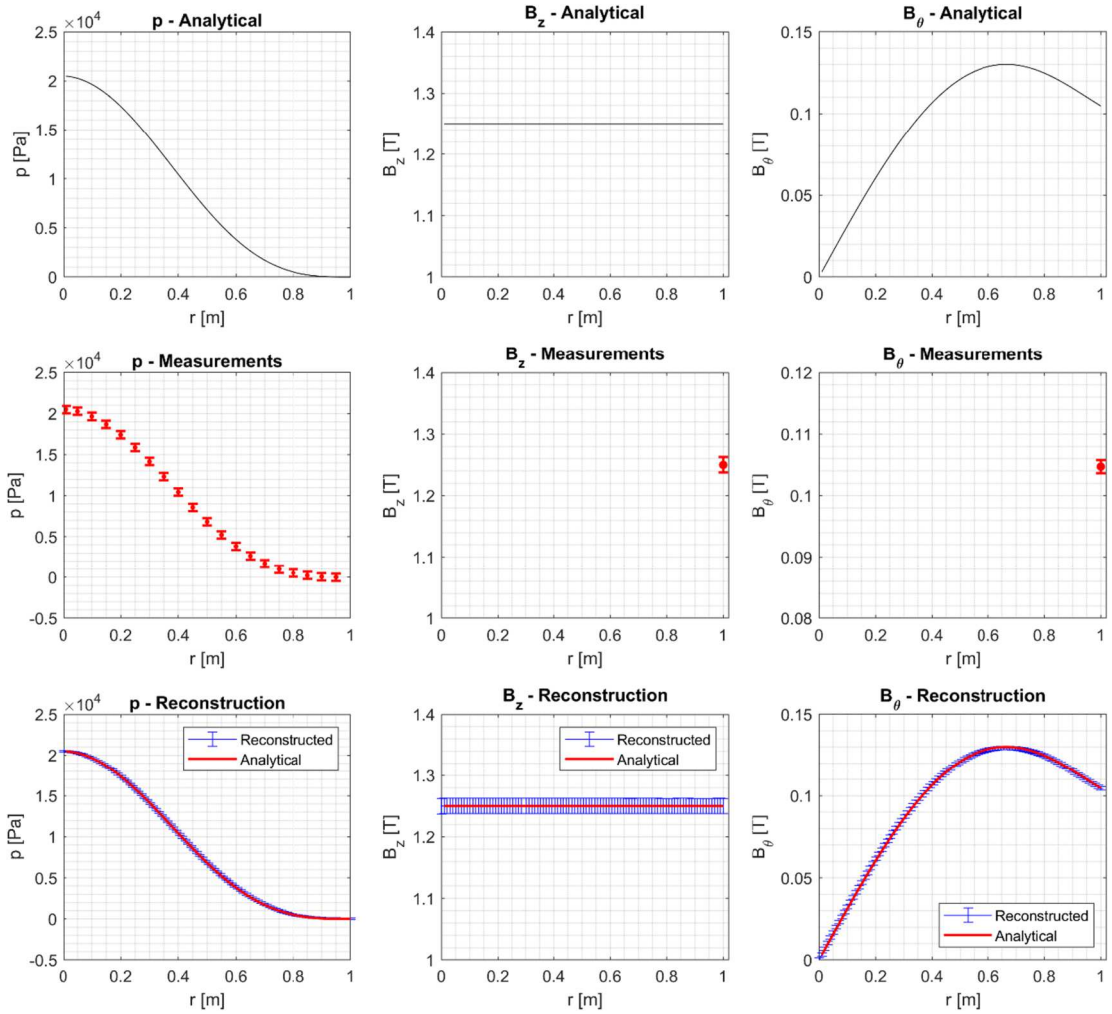

Figure 11. MHD case: analytical (or target) pressure and magnetic profile (top row), measurements used for the reconstruction (middle row), and reconstructed quantities (bottom row).

Assu have a diagnostic to measure the pressure at various internal points (similar to a lidar or Thomson scattering diagnostic in plasma [37]) and only two measurements of the magnetic field components at the edge of the plasma (typically achievable by pick-up magnetic coils [34, 37]). All the diagnostics are supposed to have an uncertainty ($\sigma_{p,m} = 450\ Pa$, $\sigma_{B_z,m} = 13\ mT$ and $\sigma_{B_\theta,m} = 1\ mT$). Figure 11(b) shows the measurements used and their error bars.

The architecture of the neural network is shown in figure 12. The neural network is asked to predict the average and the standard deviation of the pressure and the component of the magnetic field along z and θ. The magnetic field component along r is assumed to be equal to zero because of physical considerations ($\nabla \cdot \boldsymbol{B} = 0$).

Boundaries loss is calculated as follows:

$$\begin{aligned} Loss_{boundaries} = \frac{1}{N_p}\sum \Bigg(& \log \frac{\sigma_{p,p}}{\sigma_{p,m}} + \frac{(p_p - p_m)^2}{2\ \sigma_{p,p}^2} \Bigg) \\ & + \left(\log \frac{\sigma_{B_z,p}}{\sigma_{B_z,m}} + \frac{(B_{z,p} - B_{z,m})^2}{2\ \sigma_{B_z,p}^2} \right) \\ & + \left(\log \frac{\sigma_{B_\theta,p}}{\sigma_{B_\theta,m}} + \frac{(B_{\theta,p} - B_{\theta,m})^2}{2\ \sigma_{B_\theta,p}^2} \right) \end{aligned} \tag{26}$$

Concerning the losses for the physics, we have various terms. The Ampere's law and the force balance can be combined [] so that the following physics loss must be minimised:

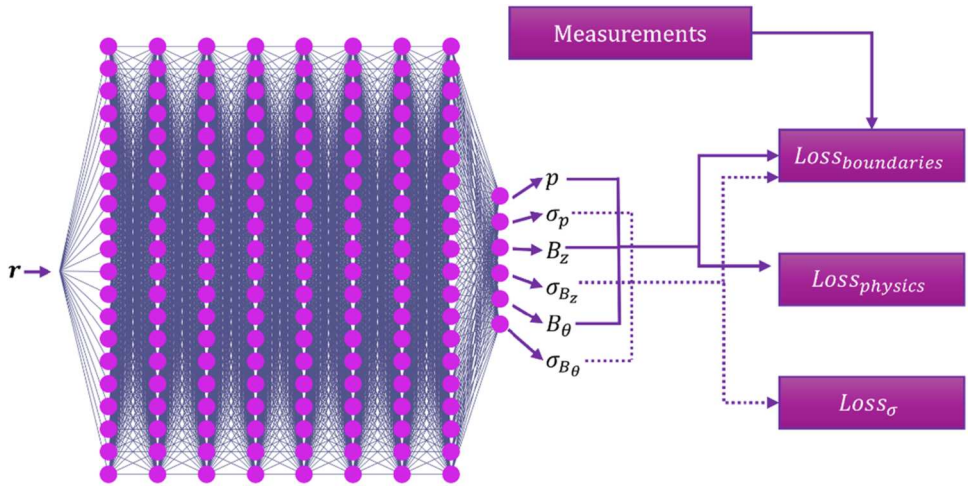

Figure 12. PINN architecture for the case MHD case.

$$Loss_{physics} = \frac{1}{N}\left[\left(\frac{r_0}{B_0}\right)^2\sum\left(\frac{\partial B_z}{\partial r}\right)^2 + \left(\frac{r_0}{p_0}\right)^2\sum\left(\frac{\partial p}{\partial r} + J_z B_\theta\right)^2\right] + \frac{p(r=1)^2}{\sigma_{p_0}^2} + \frac{B_\theta(r=0)^2}{\sigma_{B_\theta}^2} + \left(\frac{r_0}{p_0}\right)^2 \frac{\partial p}{\partial r}\Big|_{r=0}^2 \quad (27)$$

Where the first two terms are calculated in a grid generated with a 1D Sobol distribution, the third term is an ideal wall condition (pressure equal to zero), the fourth term comes from physical considerations, and the last term impose that the pressure first derivative is equal to zero to ensure the cylindrical symmetry condition. All the terms are normalised by appropriate scaling factors based on the reference values r_0, B_0, p_0, σ_{p_0} and $\sigma_{B\theta}$. The uncertainty regularisation equation is based on minimising the first derivative of the uncertainty, normalised with the opportune factors:

$$Loss_\sigma = \frac{1}{N}\left[\left(\frac{r_0}{B_0}\right)^2\sum\left(\frac{\partial \sigma_{B_z}}{\partial r}\right)^2 + \left(\frac{r_0}{B_0}\right)^2\sum\left(\frac{\partial \sigma_{B_\theta}}{\partial r}\right)^2 + \left(\frac{r_0}{p_0}\right)^2\sum\left(\frac{\partial \sigma_p}{\partial r}\right)^2\right] \quad (28)$$

Then, the total loss is evaluated with the equation (5).

The reconstruction has been achieved setting the target equal to 0.55. The comparison between the analytical solution and the reconstructed quantities are shown in figure 11 (bottom). The RMSE for the pressure is 173 Pa with a NNLL of 0.99, while the RMSE of the two components of the magnetic fields are $4.16\ 10^{-2}$ mT (NNLL = 0.92) and 1.10 mT (NNLL = 1.51) for the z and θ components, respectively. The loss physics at the end of the training is $1.35\ 10^{-4}$ ($\alpha = 3.4\ 10^3$), suggesting that the case has not faults in the diagnostics (as imposed by us).

5. Conclusions

Physics-Informed Neural Networks are a relatively new methodology that integrates traditional data-driven deep learning with prior knowledge of physics. Despite being introduced just few years ago, PINNs have rapidly gained widespread use across a broad range of scientific disciplines, including physics, engineering, chemistry, and mathematics. Their success can be attributed to several key features. Not only PINNs can incorporate imperfect or incomplete knowledge of physical systems, but they are also robust to noisy data. Additionally, they enable high-resolution predictions, support mesh-free simulations, facilitate domain decomposition, and offer many other versatile capabilities, making them highly attractive for a wide variety of applications.

However, PINNs still require further analysis, study, and improvement in certain areas. One key challenge is balancing the different loss terms, particularly between boundary/data losses and physics-based losses. Several studies have explored this issue, demonstrating that the choice of weighting significantly affects the quality of the PINN results. Researchers have proposed various methodologies in search of a more general weighting scheme. However, current research suggests that PINNs often require tailored solutions for specific problems, rather than a one-size-fits-all approach.

In this work, we proposed a new adaptive weighting scheme to improve convergence of PINNs when applied to measurements (data with noise) together with a PINN architecture that aims at predicting the

uncertainty of the predicted quantities. Such a new architecture aims at predicting directly the estimator of the distribution of the various quantity. In our work, the quantities have been supposed to be normally distributed, and therefore the neural network has been asked to predict the mean and the standard deviation, but it is worth highlighting that any distribution may be used, with the necessary modification on the loss distribution and in the network outputs. The adaptive weighting is based on Negative Log Likelihood expectation from diagnostic data. In fact, in most cases, it is expected to know the statistical behaviour of the diagnostics, allowing for a good weighting of the various diagnostics according with their uncertainty. Such a weighting of the various diagnostic is automatically achieved thanks the use of the NLL metrics for the boundary loss. In standard situations, the expected boundary loss (what we called 'target') is known and equal to 0.5 (ideal value). We based our new adaptive weighting scheme on the concept of the boundary target: once the PINN reached the target, the weight to physics loss is iteratively increased.

Several parametric analyses have been performed, and results can be resumed as in the following:

1. The predicted uncertainty using the new PINNs does not reflect the actual uncertainty, but it is only an approximation. However, in many cases, especially in over constrained cases where more diagnostics are present, this approach is sufficiently accurate. This is because when diagnostics lead the uncertainty of the reconstruction, our model is close to actual uncertainty. On the contrary, in regions where there are not diagnostics, the predicted uncertainty is guided by the regularisation loss of the uncertainty, that is an approximation and not the 'true' equation. However, if the 'true' equation for uncertainty is known, it can be easily changed with the regularisation one.
2. The adaptive weighting scheme offers a guided and simple methodology to reach the best convergence of PINNs. What is interesting is that the combination of the target with the uncertainty prediction allows for a methodology that does not require specific adjustments. In fact, the AW scheme is easy to understand too low targets involves saturation of α to very low values. Therefore, from an operative perspective, the user is asked just to increase the loss target until the adaptive weighing schemes increases the physics weight (α) to large values, with consequent strong decrease of the physics loss. This means that, in the absence of outliers in the measurements, the adaptive weighting scheme operates fully automatically. However, if an unknown outlier is present (whether in the measurements or the boundary conditions) a brief user-in-the-loop intervention becomes necessary. Even in such cases, only a few iterative steps are required, and the algorithm remains capable of simultaneously solving the problem and identifying the outliers.
3. The use of our adaptive weighing scheme proved to allow very precise reconstruction, with RMSE below MC-based methodology, especially for limited MC replications.
4. The proposed methodology automatically filters out problematic boundaries, like fault diagnostics and outliers. Therefore, the adaptive weighting scheme may be used to develop PINN-based methodology able to 'diagnose the diagnostics' without the need of iterative and manual procedures.

Section 4 was used to show how to apply our adaptive weighting scheme for PINN to more complex cases. Therefore, three cases based on heat transfer, fluid dynamics and MHD have been developed by referring to typical application in physics and engineering. The section proved that the methodology is robust even when applied to complex cases like the incomplete physics fluid dynamics case, without the need of specific adjustments. Moreover, in the fluid dynamic case we used a standard neural network and not a ML-PINN, showing that the adaptive weighting scheme is robust even with other PINN architecture.

In conclusions, in this paper we presented a simple, robust and efficient adaptive weighting scheme for PINN that can be easily applied to all problems where measurements and physics knowledge can be combined: tt can be used to solve inverse problems with higher accuracy by taking into account diagnostic uncertainty, to detect and identify faults in the diagnostic, perform modelling of incomplete equations by combining data and physics.

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/AW-ML-PINN.

Appendix A. Normalised NLL

In this work, a modified version of the negative log-likelihood, named normalised negative log-likelihood, has been used for both training our models and analyse the results. In physics and engineering, it is common to solve inverse problem or to fit function by referring to the maximisation of the likelihood (L). In our case, we predict both the mean quantity u_p and its uncertainty σ_p, and therefore for a gaussian distribution L is:

$$L = \prod \frac{1}{\sqrt{2\pi}\,\sigma_p} e^{-\frac{(u_p-u_t)^2}{2\sigma_p^2}} \tag{A1}$$

However, it is common to address the maximisation of the likelihood by minimising the negative logarithm of the likelihood, known as negative log-likelihood (NLL):

$$NLL = -\ln\left(\prod \frac{1}{\sqrt{2\pi}\,\sigma_p} e^{-\frac{(u_p-u_t)^2}{2\sigma_p^2}}\right) = \sum\left[\ln\left(\sqrt{2\pi}\,\sigma_p\right) + \frac{(u_p-u_t)^2}{2\sigma_p}\right] \tag{A2}$$

In the case where the sigma is not predicted but imposed, the minimisation of the NLL is equivalent to the minimisation of the uncertainty weighted MSE. Moreover, if all the measurements have the same uncertainty, the problem can be reduced to the minimisation of the standard MSE.

The NLL returns values that depends on the boundary conditions. For example, if one has a perfect predictor $u_p = u_t$ and $\sigma_p = \sigma_t$, the NLL will be $\sum\ln\left(\sqrt{2\pi}\,\sigma_t\right)$, involving that the ideal NLL value depends on uncertainty. Of course, this is correct and in line with the actual meaning of the NLL. However, in some cases, one may be interested in using a normalised value that it is independent of the problem. For this reason, we decided to use an alternative NLL that is independent of the uncertainty of the measurements. Starting from A.1, one can easily define a normalised likelihood as follows:

$$Normalised\ Likelihood = \prod\left[\frac{\frac{1}{\sqrt{2\pi}\,\sigma_p} e^{-\frac{(u_p-u_t)^2}{2\sigma_p^2}}}{\frac{1}{\sqrt{2\pi}\,\sigma_t} e^{-\frac{(u_t-u_t)^2}{2\sigma_t^2}}}\right] = \prod\left[\frac{\sigma_t}{\sigma_p} e^{-\frac{(u_p-u_t)^2}{2\sigma_p^2}}\right] \tag{A3}$$

And therefore, one can define the Normalised Negative Log-Likelihood as:

$$NNLL = \sum\left[\ln\left(\frac{\sigma_p}{\sigma_t}\right) + \frac{(u_p-u_t)^2}{2\sigma_p}\right] \tag{A4}$$

The NNLL has the property to tends to zero when the reconstruction converges to the exact solution.

Despite the NNLL is a good metric to evaluate the quality of the reconstructions, alone it does not to decouple the error on the prediction of the average value and of the uncertainty. This is clearly shown in figure A1, where there are infinite values of σ_p/σ_t and $(u_p - u_t)^2/(2\sigma_p^2)$. For this reason, in the paper the model is analysed by calculating both the RMSE and the NNLL.

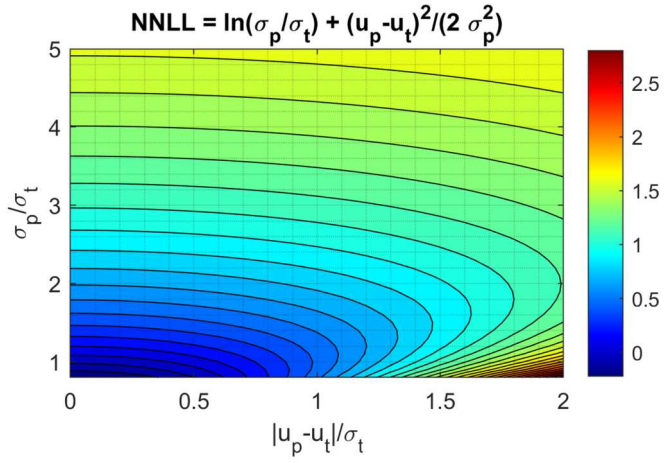

Figure A1. NNLL as a function of the error of the two statistics descriptors of the prediction normalised by the measurement uncertainty.

Appendix B. Loss target calculation

The loss target is the value of the loss boundary that must be reached before the adaptive weighting scheme starts to increase α, i.e. the weight given to the loss of the physics and uncertainty. The correct selection of the loss target plays a fundamental role in the quality of the results that can be obtained by the AW-ML-PINN, and therefore in this appendix we discussed how it can be defined in different situations. We considered the following cases:

1. Well-determined systems
2. Overdetermined systems
3. Overdetermined systems in presence of diagnostic faults

At first, we consider the well-determined case, where the number N of measurements and boundary conditions are the minimum to close our problem (sections 3.1 and 3.2). So, we have N measurements at different locations of our domain, and we supposed to have M repeated measurement for each i-th location. Therefore, the loss boundary for the i-th measurement will be:

$$\begin{aligned} Loss_{boundary,i} = & \frac{1}{M}\sum_{j=1}^{M}\left[\ln\left(\frac{\sigma_{p,i}}{\sigma_{m,i}}\right)+\frac{1}{2}\frac{(u_{p,i}-u_{m,i,j})^2}{\sigma_{p,i}^2}\right] \\ = & \ln\left(\frac{\sigma_{p,i}}{\sigma_{m,i}}\right)+\frac{1}{2\sigma_{p,i}^2}\frac{1}{M}\sum_{j=1}^{M}[(u_{p,i}-u_{m,i,j})^2] \end{aligned} \tag{B1}$$

The loss target, from ad ideal point of view, is the minimum value of the achievable boundary and therefore:

$$\begin{aligned} \frac{\partial(Loss_{boundary,i})}{\partial\sigma_{p,i}} = 0 \rightarrow & \frac{1}{\sigma_{p,i}}-\frac{1}{\sigma_{p,i}^3}\frac{1}{M}\sum_{j=1}^{M}[(u_{p,i}-u_{m,i,j})^2] \\ = 0 \rightarrow & \sigma_{p,i}=\sqrt{\frac{1}{M}\sum_{j=1}^{M}[(u_{p,i}-u_{m,i,j})^2]} \end{aligned} \tag{B2}$$

$$Loss_{target,i}=\ln\left(\frac{\sqrt{\frac{1}{M}\sum_{j=1}^{M}[(u_{p,i}-u_{m,i,j})^2]}}{\sigma_{M,i}}\right)+\frac{1}{2} \tag{B3}$$

If the diagnostic has no faults, the uncertainty of the diagnostic is exactly equal to $\sigma_{m,i}$, the predicted quantity $u_{p,i}$ will be equal to the average value of the measurements ($u_{p,i}=\frac{1}{M}\sum_{j=1}^{M}u_{m,i,j}$), therefore the RMSE will converges to the diagnostic uncertainty and the target of the diagnostic will be equal to 0.5. The average loss target is just the average of loss target of each i-th diagnostics, and therefore if all diagnostics satisfy the previous requirements the average ideal loss target will be equal to 0.5.

However, in real cases there are different reasons which involve that a loss target equal to 0.5 may lead to some troubles in terms of convergence and stability. At first, the MSE converges exactly to the diagnostic uncertainty only for large number of M. Second, the uncertainty of the diagnostics is usually estimated and therefore affected by uncertainty itself. Last, our adaptive scheme increases α only for loss boundary lower than the loss target. For these reasons, it is convenient to use slightly larger loss targets (e.g. 0.55).

When there is not the possibility to have repeated measurements (e.g. during transient phenomena), the uncertainty estimation can be guided by the perturbation approach (see section 2). In such a case, the ideal loss target is 0.5 only if the measurements are exact but we suppose that they are uncertainty. In such a case, 0.55 is a good operational value (see sections 3.1 and 3.2).

The second case consists of analysing overdetermined problems. This class of problems is very interesting for a huge number of applications, such as data-integrated analysis and modelling of PDE. The general formulation is the same of equation (B3). For very large values of M and diagnostics without faults (no offsets), the MSE of equation (B3) converges again to the standard deviation of the diagnostics and therefore the ideal target is 0.5.

However, there are more complex cases to be considered that may apply to real applications. Therefore, let us now consider the case where no repetitions are allowed (M = 1). In this case, the perturbation method will be used. In such a case, each i-th measurement will have an error respect the true solution ε_i. Therefore, the perturbed measurement will be $u_{M,i,j}=u_{T,i}+\varepsilon_i+\varepsilon_{pert,i,j}$ where $u_{T,i}$ is the true value to be reconstructed, ε_i is the error committed by the diagnostics, and $\varepsilon_{pert,i,j}$ is the perturbation. The aim of our PINN is to reconstruct the

true solution and so we suppose that $u_{p,i} = u_{T.i}$ and therefore the loss target will be:

$$Loss_{target,i} = \ln\left(\frac{\sqrt{\frac{1}{2}\frac{1}{M}\sum_{j=1}^{M}[(\varepsilon_i + \varepsilon_{pert,i,j})^2]}}{\sigma_{M,i}}\right) + \frac{1}{2} = \ln\left(\frac{\sqrt{\varepsilon_i^2 + \sigma_{M,i}^2}}{\sigma_{M,i}}\right) + \frac{1}{2} \tag{B4}$$

By averaging the target of the various diagnostic, supposing that our diagnostics are not affected by systematic errors and that the estimated (or expected) uncertainties are correct, the loss target can be expected to be:

$$Loss_{target} = \frac{1}{2} + \frac{1}{N}\sum_{i=1}^{N}\ln\left(\frac{\sqrt{\varepsilon_i^2 + \sigma_{M,i}^2}}{\sigma_{M,i}}\right) = \frac{1}{2} + \frac{1}{N}\sum_{i=1}^{N}\ln\left(\sqrt{1 + \frac{\varepsilon_i^2}{\sigma_{M,i}^2}}\right) \sim 0.76 \tag{B5}$$

So, for over-determined cases with only one repetition for each i-th measurement the ideal loss target is around 0.76 (numerically evaluated). From an operational point of view, a value between 0.9 and 1 should be a good solution.

At end, it is worth it to analyse what happen in case of fault diagnostics. We consider three cases (the same reported in section 3.3):

1. One too noisy unexpected diagnostic (the true uncertainty is unknown and equal to $\sigma_{M,k,fault}$).
2. One diagnostic with high offset values (there is a systematic error equal to Δ_{offset}).
3. One diagnostic with both high offsets and noise (combination of the two previous cases).

Moreover, we will consider the case where M repetitions (and not perturbations) are possible. For all diagnostics, expect the fault one, the loss target will be equal to 0.5, while the fault one (i = k) will have a different target as a function of the case:

$$Case\ \ 1{:}\ Loss_{target,k} = \ln\left(\frac{\sigma_{M,k,fault}}{\sigma_{M,k}}\right) + \frac{1}{2} \tag{B6}$$

$$Case\ \ 2{:}\ Loss_{target,k} = \ln\left(\frac{\sqrt{\sigma_{M,k}^2 + \Delta_{offset}^2}}{\sigma_{M,k}}\right) + \frac{1}{2} \tag{B7}$$

$$Case\ \ 3{:}\ Loss_{target,k} = \ln\left(\frac{\sqrt{\sigma_{M,k,fault}^2 + \Delta_{offset}^2}}{\sigma_{M,k}}\right) + \frac{1}{2} \tag{B8}$$

The equations above clearly explain the reason why a target equal to the ideal on does not allows the AW-ML-PINN to converge to good solutions, while by increasing the loss target step-by-step allows for the model to automatically exclude unphysical measurements (e.g. outliers) from the reconstruction (See sections 3.3 and 4.1).

Appendix C. Treatments 0f ideal boundaries

In PINN applications it is quite common to have two different types of boundaries, i.e. ideal boundary conditions and measured boundary conditions. Let us consider as example the fluid dynamics case reported in section 4.3. We have some boundaries that must be true because of physical considerations and are not uncertain, like the no-slip conditions on the wall, and others that are affected by uncertainties (like the velocity from the Particle-Image Velocimetry). Therefore, one may ask if the ideal boundaries should be put in the physics losses or in the boundary losses.

Unfortunately, we have not a general answer and probably the better solution is problem dependent. So, we want just to show how to change the losses and the target as a function of the solution. To do that, let us use the following nomenclature:

- $Loss_{real}$ is the boundary loss of uncertain quantities defined with the NNLL as shown in the paper.
- $Loss_{ideal}$ is the boundary loss of ideal boundaries (like no-slip conditions in fluid dynamics).
- $Loss_{PDE}$ is the loss that aims to minimise the error on the PDE.
- $Loss_{\sigma}$ is the loss of the uncertainty.

Therefore, we have two possibilities. The first on is putting the ideal loss in the physics loss:

$$Loss_{physics} = \frac{1}{N_{PDE}}\sum Loss_{PDE,i} + \frac{1}{N_{ideal}}\sum Loss_{ideal,i} \tag{C1}$$

And therefore all the considerations in our paper apply. Consider that this is the solution adopted by the authors in section 4.1 for adiabatic conditions, 4.2 for the various boundary conditions and 4.3 for the wall and symmetry conditions.

The alternative would be including the ideal loss in the boundary loss:

$$Loss_{boundary} = \frac{1}{N_{real}}\sum Loss_{real,i} + \frac{1}{N_{ideal}}\sum Loss_{ideal,i} \tag{C2}$$

In such a case, it is important just update the loss target by considering that the target of the loss ideal is zero.

Appendix D. Stress-test with increasing levels of noise

A stress test of the methodology is conducted by increasing the level of noise. Let us refer to the hyperstatic case introduced in section 3.3, but without outliers. The top row of figure D1 shows the test cases: three scenarios with noisy measurements for which noise is equal to σ, 2σ, 3σ (where $\sigma = 0.1$), respectively. For all the three cases, adaptive weighting scheme is used with $Loss_{target} = 0.75$.

The reconstruction performances, in terms of NLL and dimensionless RMSE, are quantified in table D1. The results show that in the worst case (3 σ), the reconstruction returns an average error below the uncertainty (1.5 σ), validating the reliability of the method. The results are also shown graphically in figure D1, where it can be observed that the ideal signal is always inside the predicted error bars, confirming the capability to obtain high accuracy and reliability.

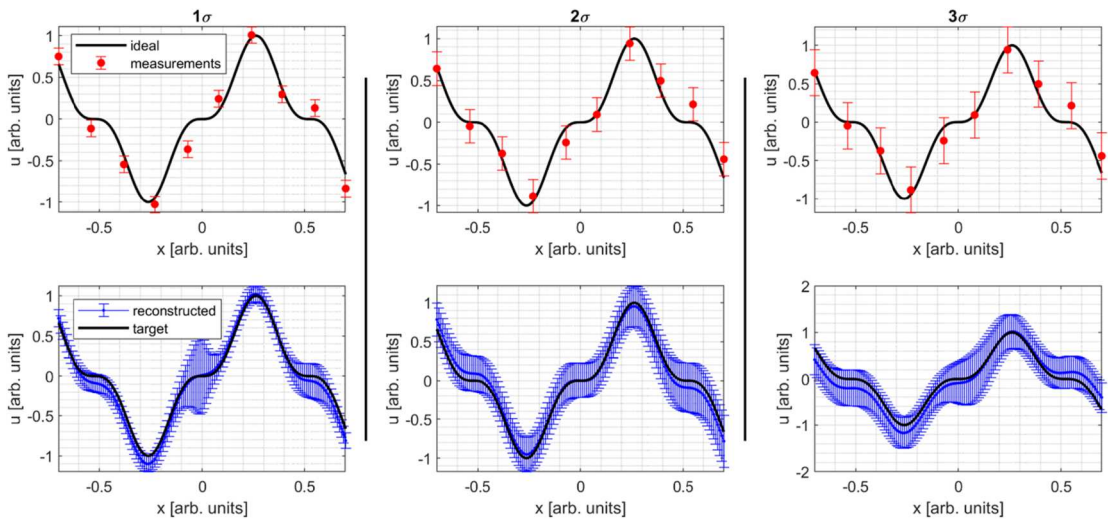

Figure D1. Top row: measurements in the \sigma, 2\sigma, 3\sigma cases versus the target (or ideal) function u. Bottom row: reconstruction for the respective case.

Table D1. NLL and normalised RMSE for the three investigated cases. The normalised RMSE is calculated by dividing the RMSE for 0.1, which is the value of σ.

	Case σ	Case 2σ	Case 3σ
NLL	−125.35	−61.64	0.43
$RMSE/\sigma$	0.70	0.70	1.5

References

[1] 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
[2] Chen Z, Liu Y and Sun H 2021 Physics-informed learning of governing equations from scarce data *Nat. Commun.* **12** 6136
[3] 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
[4] Lawal Z K, Yassin H, Lai D T C and Che Idris A 2022 Physics-informed neural network (PINN) evolution and beyond: a systematic literature review and bibliometric analysis *MDPI* (https://doi.org/10.3390/bdcc6040140)
[5] 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
[6] Jin X, Cai S, Li H and Karniadakis G E 2021 NSFnets (Navier–Stokes flow nets): physics-informed neural networks for the incompressible Navier–Stokes equations *J. Comput. Phys.* **426** 109951
[7] Wang X, Wen H and Wang B 2024 Super-resolution flow-field reconstruction in rotating detonation combustors *Aerosp. Sci. Technol.* **144** 108740
[8] Kim D and Lee J 2024 A Review of physics informed neural networks for multiscale analysis and inverse problems *Multiscale Science and Engineering* **6** 1–11
[9] Karniadakis G E, Kevrekidis I G, Lu L, Perdikaris P, Wang S and Yang L 2021 Physics-informed machine learning *Nature Reviews Physics* **3** 422–40
[10] Sofos F and Drikakis D 2025 A review of deep learning for super-resolution in fluid flows *Phys. Fluids* **37** 041303
[11] 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
[12] 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
[13] Cai S, Mao Z, Wang Z, Yin M and Karniadakis G E 2021 Physics-informed neural networks (PINNs) for fluid mechanics: a review *Acta Mech. Sin.* **37** 1727–38
[14] 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
[15] Zhang S, Lan P and Su J-J 2021 Wave-packet behaviors of the defocusing nonlinear Schrödinger equation based on the modified physics-informed neural networks *Chaos* **31** 113107
[16] Huang B and Wang J 2023 Applications of physics-informed neural networks in power systems - a review *IEEE Trans. Power Syst.* **38** 572–88
[17] Sahli Costabal F, Yang Y, Perdikaris P, Hurtado D E and Kuhl E 2020 Physics-informed neural networks for cardiac activation mapping *Front. Phys.* **8** 42
[18] Rohrhofer F M, Posch S, Gößnitzer C and Geiger B C 2023 Data versus physics: the apparent pareto front of physics-informed neural networks *IEEE Access* **11** 86252–61
[19] Xiang Z, Peng W, Liu X and Yao W 2022 Self-adaptive loss balanced physics-informed neural networks *Neurocomputing* **496** 11–34
[20] 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
[21] Yang L, Meng X and Karniadakis G E 2021 B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data *J. Comput. Phys.* **425** 109913
[22] Zhang D, Lu L, Guo L and Karniadakis G E 2019 Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems *J. Comput. Phys.* **397** 108850
[23] Feng R 2023 Physics-informed deep learning for rock physical inversion and its uncertainty analysis *Geoenergy Science and Engineering* **230** 212229
[24] Kroese D P and Rubinstein R Y 2012 Monte Carlo methods *WIREs Computational Statistics* **4** 48–58
[25] Yang M and Foster J T 2022 Multi-output physics-informed neural networks for forward and inverse PDE problems with uncertainties *Comput. Methods Appl. Mech. Eng.* **402**
[26] Sobol' I M, Asotsky D, Kreinin A and Kucherenko S 2011 Construction and comparison of high-dimensional sobol' generators *Wilmott* 64–79 **2011**
[27] Sobol I M 1967 'On the distribution of points in a cube and the approximate evaluation of integrals *USSR Computational Mathematics and Mathematical Physics* **7** 86–112
[28] Rohrhofer F M, Posch S, Gobnitzer C and Geiger B C 2023 Data versus physics: the apparent pareto front of physics-informed neural networks *IEEE Access* **11** 86252–61
[29] Elhamod M *et al* 2022 CoPhy -PGNN: learning physics-guided neural networks with competing loss functions for solving eigenvalue problems *ACM Trans. Intell. Syst. Technol.* **13** 1–23
[30] 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
[31] Minkina W and Dudzik S 2009 *Infrared Thermography* (Wiley) 10.1002/9780470682234
[32] Hoidn O, Mishra A A and Mehta A 2023 Physics constrained unsupervised deep learning for rapid, high resolution scanning coherent diffraction reconstruction *Sci. Rep.* **13** 22789
[33] Gori F, Petracci I and Angelino M 2013 Flow evolution of a turbulent submerged two-dimensional rectangular free jet of air. average particle image velocimetry (PIV) visualizations and measurements *Int. J. Heat Fluid Flow* **44** 764–75
[34] Wesson J 2011 *Tokamaks* (Oxford University Press)
[35] Shafranov V D 1966 Plasma equilibrium in a magnetic field *Reviews of Plasma Physics* **2** 103
[36] Zohm H 2014 *Magnetohydrodynamic Stability of Tokamaks* (Wiley) 10.1002/9783527677375
[37] Hutchinson I H 2002 *Principles of Plasma Diagnostics* (Cambridge University Press) 10.1017/CBO9780511613630