ECONSTOR
Make Your Publications Visible.

A Service of

ZBW Leibniz-Informationszentrum Wirtschaft
Leibniz Information Centre for Economics

Rosa, Mariapia De; Pompameo, Laura; Litvinenko, Alexander; Cuomo, Salvatore

Article — Published Version

HOMO-PINN: Hyperparameter Optimization of a Multi-output Physics-Informed Neural Network

Operations Research Forum

Provided in Cooperation with:
Springer Nature

Suggested Citation: Rosa, Mariapia De; Pompameo, Laura; Litvinenko, Alexander; Cuomo, Salvatore (2025) : HOMO-PINN: Hyperparameter Optimization of a Multi-output Physics-Informed Neural Network, Operations Research Forum, ISSN 2662-2556, Springer International Publishing, Cham, Vol. 6, Iss. 4,
https://doi.org/10.1007/s43069-025-00561-7

This Version is available at:
https://hdl.handle.net/10419/330385

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

https://creativecommons.org/licenses/by/4.0/

WWW.ECONSTOR.EU

Mitglied der Leibniz-Gemeinschaft

Operations Research Forum (2025) 6:153
https://doi.org/10.1007/s43069-025-00561-7

RESEARCH

HOMO-PINN: Hyperparameter Optimization of a Multi-output Physics-Informed Neural Network

Mariapia De Rosa[1] · Laura Pompameo[3] · Alexander Litvinenko[2] · Salvatore Cuomo[3]

Received: 12 June 2025 / Accepted: 25 September 2025
© The Author(s) 2025

Abstract
The good choice of hyperparameters is crucial for the successful application of deep learning (DL) networks in order to find accurate solutions or the best parameter in solving partial differential equations (PDEs), that are sensitive to errors in coefficient estimation. For this purpose, hyperparameter optimization of multi-output physics-informed neural networks (HOMO-PINNs) is based on the optimal search of PINN hyperparameters for solving PDEs with uncertain coefficients in the uncertainty quantification (UQ) field. By testing this novel methodology on different PDEs, the relationship between activation functions, the number of output neurons, and the degree of coefficient uncertainty can be observed. The experimental results show that adding output neurons to the neural network (NN), even if a theoretically incorrect activation function is chosen, keeps the predicted solution accurate.

Keywords Physics-informed neural network · Numerical methods · PDE · Uncertainty quantification · PINN · Hyperparameter optimization

1 Introduction

Partial differential equations (PDEs) are fundamental for modelling a wide range of physical phenomena in various scientific and engineering disciplines. In order to find an accurate solution to differential problems, physics-informed neural networks (PINNs) were developed by Raissi et al. [20], combining mathematical models and data. These networks are capable of using physical equations as a functional to minimize, thereby creating a NN that is suitable for solving problems in a manner that is consistent with the principles of physics. Consequently, PINNs have been employed to address

✉ Salvatore Cuomo
salvatore.cuomo@unina.it

1 Istitute of Biostructures and Bioimaging (IBB) of the National Research Council (CNR), Via Tommaso De Amicis, 95, 80145 Naples, Italy

2 RWTH Aachen University, Pontdriesch, 14-16, 52062 Aachen, Germany

3 University of Naples Federico II, Via Cintia, 26, 80126 Naples, Italy

Published online: 10 October 2025

real-world issues in a multitude of disciplines, such as fluid dynamics [3], flexible mechatronics and soft robotics [17], electronics [25], and railway tracking [6]. As is well documented in the literature, the coefficients of a PDE can have a significant impact on the solution [9] and its numerical approximation [10, 13]. Consequently, the extent of noise present in these coefficients necessitates the estimation of the error in predicting a solution by a PINN. For example, Qian et al. [19] provide an error analysis of PINN for the wave equation. While De Ryck et al. [7] prove bounds on the errors caused by the approximation of the incompressible Navier–Stokes equations with PINNs.

PINNs may help to solve PDEs with uncertain coefficients or random noise. In the uncertainty quantification (UQ) content, the chosen PDE is solved multiple times under random perturbations of coefficients to assess solution variability. This approach allows for evaluating the accuracy of the mean predicted solution against an analytical reference. In the context of elastic deformation in heterogeneous solids, Bharadwaja et al. [2] demonstrated that the mean and standard deviation of PINN solutions align well with Monte Carlo finite element results. Similarly, Li et al. [16] introduced a UQ-based method to identify collocation points where PINNs exhibit large prediction errors, providing a systematic way to enhance solution reliability.

Yang and Foster [24] proposed multi-output PINNs (MO-PINNs), which are capable of solving both forward and inverse problems governed by PDEs, even in the presence of noisy data. This approach not only predicts the solution, but also provides an associated uncertainty estimate, allowing quantification of the confidence in the results, particularly suited to the UQ task. Furthermore, Liu et al. [18] use MO-PINNs for one- and two-dimensional nonlinear time distributed-order models. In addition, Hao et al. [11] utilize MO-PINN to predict the evolution of time-varying coefficients and probability density function of quantity of interest (QoI). While Chang et al. [4] show how MO-PINNs reduce the relative error by an order of magnitude in the context of parameter identification. For the purpose of this paper, MO-PINNs are used due to their unique ability to train a single NN on each of the coefficient-based PDEs, and then visualize all outputs simultaneously.

Consequently, the selection of the number of output neurons, in conjunction with the other NN hyperparameters, has a significant impact on the efficiency of PINNs. For instance, Escapil-Inchauspé and Ruz [8] introduce a hyper-parameter optimization (HPO) algorithm and test it on the Helmholtz equation. While Sharma et al. [22] perform an optimization process for heat conduction problems with a discontinuous solution on many hyperparameters, including the number of hidden layers, learning rate, and activation functions. In particular, the output activation function, which is typically selected based on the range of values where the problem resides, plays a crucial role in the NN training phase. While the hyperbolic tangent (tanh) activation function is commonly used as the default option for PINNs, some studies have indicated that other activation functions can be more effective in specific scenarios (see [21]). For instance, research by Jagtap et al. [12] and Al Safwan et al. [1] suggests that the sigmoid and swish functions may offer advantages in certain cases. However, these studies do not provide a definitive conclusion on the optimal activation function for various PDE problems. Further investigation and comparison across different types of PDEs are needed to establish a clearer preference for activation functions in PINNs.

In order to determine the optimal choice of hyperparameters (e.g., the number of neurons and the activation function for the output layer), it is necessary to develop a robust methodology that can better predict the most accurate solution to the differential problem, even when the error on one or more coefficients that appear in the PDE increases (in the sense of being affected by a noise of larger variance). This study presents such methodology, introducing hyperparameter optimization of MO-PINNs (HOMO-PINNs), which are capable of combining an optimal search strategy for hyperparameters with the UQ for solutions to PDEs affected by uncertainties on coefficients.

The novelty of this paper is as follows:

- introduction of the HOMO-PINNs, obtained by combining multi-output PINNs with Hyperparameter Optimization, while presenting the quantification of uncertainties in PDE;
- testing of this methodology for groundwater flow and Poisson equations in order to showcase the links between hyperparameter choices and different percentages of error;
- discovering that a *wrong* activation function requires more output neurons in order to perform well when the coefficient error is large.

The paper is structured as follows: Section 2 provides the mathematical presentation of PINNs, focusing on multi-outputs, and outlines the hyperparameter optimization strategy; Section 3 is devoted to experimental results of the HOMO-PINN methodology on the two test PDEs; Section 4 includes the concluding remarks, summarizing the findings obtained through testing.

2 Methodology

This section focuses on the description of the HOMO-PINN approach (see Fig. 1), from the mathematical aspects of PINNs, through their multi-output variant, to the search for optimal hyperparameters.

2.1 Physics-Informed Neural Networks

PINNs represent a significant advancement in the field of computational modeling, bridging the gap between traditional data-driven approaches and physics-based simulations. By embedding physical laws (described by PDEs) directly into the neural network's learning process, PINNs enhance the model's predictive capabilities, especially in scenarios with limited or noisy data.

The PINN architecture is based on a fully-connected feed-forward neural network (FC-FFNN), where each neuron in each layer is connected to every neuron in the previous layer. The mathematical operations, within a single layer, can be described in terms of matrix multiplication followed by the application of a (linear or non-linear) activation function. Let $\mathbf{x} \in \mathbb{R}^d$, for the l-th layer in the network, the output is

$$\mathbf{h}^{(l)} = \phi^{(l)} \left(\mathbf{W}^{(l)} \mathbf{h}^{(l-1)} + \mathbf{b}^{(l)} \right),$$

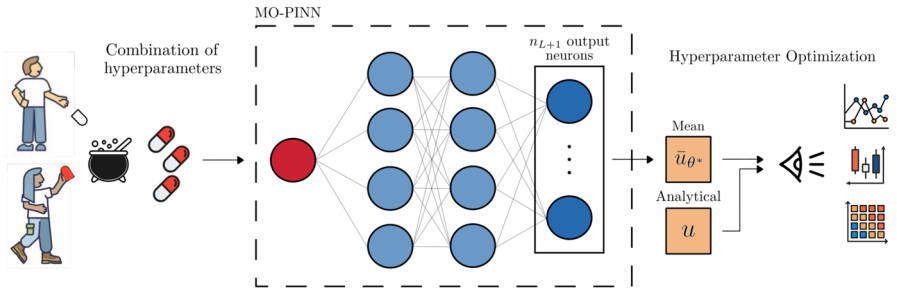


Fig. 1 Pipeline for the HOMO-PINN. Once the hyperparameters to be tested have been selected, all the possible combinations of them are generated as vectors. For each of these, a MO-PINN is trained, which returns the predicted solutions associated with different samples of the coefficient. The mean of these solutions is then compared with the analytical solution to identify the optimal combination of hyperparameters through graphical analysis

where $\phi^{(l)}$ is the activation function, $\mathbf{W}^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$ is the weight matrix, $\mathbf{b}^{(l)} \in \mathbb{R}^{n_l}$ is the bias vector and n_l is the number neurons in the l-th layer. Note that $\mathbf{h}^{(l-1)}$ is the output of the previous layer and $\mathbf{h}^{(0)} = \mathbf{x}$. Then, the FC-FFNN output is given by

$$u_\theta(\mathbf{x}) = \left(\mathbf{h}^{(L+1)} \circ \mathbf{h}^{(L)} \circ \cdots \circ \mathbf{h}^{(0)}\right)(\mathbf{x}), \tag{2.1.1}$$

where L is the number of hidden layers and θ indicates the set of hyperparameters of the NN.

A distinctive feature of PINNs is the *loss function* expressed by the discrepancy between the NN output Eq. 2.1.1 and the physical law. Let us consider a general PDE of the form

$$\begin{aligned} \mathcal{F}(u(\mathbf{x}), \nabla u(\mathbf{x}), \nabla^2 u(\mathbf{x}), \dots) = 0, \\ \mathcal{I}(u(\mathbf{x})) = 0, \qquad \mathcal{B}(u(\mathbf{x})) = 0, \end{aligned} \tag{2.1.2}$$

where u is the solution to the PDE, $\mathbf{x}$ is the vector of space-time coordinates, while $\mathcal{F}, \mathcal{I}$, and $\mathcal{B}$ are differential operators referring to the PDE, its initial conditions (IC), and boundary conditions (BC), respectively. The residuals of Eq. 2.1.2 are defined as

$$\begin{aligned} \mathcal{R}^\theta_{\text{PDE}}(\mathbf{x}) = \mathcal{F}(u_\theta(\mathbf{x}), \nabla u_\theta(\mathbf{x}), \nabla^2 u_\theta(\mathbf{x}), \dots) \\ \mathcal{R}^\theta_{\text{IC}}(\mathbf{x}) = \mathcal{I}(u_\theta(\mathbf{x})), \quad \mathcal{R}^\theta_{\text{BC}}(\mathbf{x}) = \mathcal{B}(u_\theta(\mathbf{x})) \end{aligned}$$

Then, the loss function can be defined as follows

$$\mathcal{L}(\theta) := \frac{\lambda_{\text{PDE}}}{N_{\text{PDE}}} \sum_{i=1}^{N_{\text{PDE}}} |\mathcal{R}^\theta_{\text{PDE}}(\mathbf{x}_i)|^2 + \frac{\lambda_{\text{IC}}}{N_{\text{IC}}} \sum_{j=1}^{N_{\text{IC}}} |\mathcal{R}^\theta_{\text{IC}}(\mathbf{x}_j)|^2 + \frac{\lambda_{\text{BC}}}{N_{\text{BC}}} \sum_{k=1}^{N_{\text{BC}}} |\mathcal{R}^\theta_{\text{BC}}(\mathbf{x}_k)|^2, \tag{2.1.3}$$

where N_{PDE}, N_{IC} and N_{BC} are the number of collocation points used to evaluate the PDE residual, initial condition residual, and boundary condition residual, respectively

and λ_{PDE}, λ_{IC} and λ_{BC} their weights. Therefore, the PDE is approximated by finding an optimal set θ^* of NN hyperparameters that minimizes Eq. 2.1.3 as follows

$$\theta^* := \underset{\theta}{\operatorname{argmin}}\, \mathcal{L}(\theta), \tag{2.1.4}$$

using an optimization algorithm (e.g., ADAM [14]). After identifying the optimal hyperparameter set Eq. 2.1.4, the solution $u_{\theta^*}(\mathbf{x})$ is derived, which provides the best approximation to the actual solution of the problem.

2.2 Multi-output PINNs (MO-PINNs)

To address the inherent uncertainties in PDE coefficients arising from factors such as measurement error, material heterogeneity, and model approximations, MO-PINNs extend traditional PINNs and enable the simultaneous prediction of multiple possible solutions under various uncertainty scenarios. MO-PINNs are also useful for approximating statistical properties such as the mean and variance of the solution. The key idea is that instead of training multiple PINNs for different realizations of uncertain parameters, a single neural network is designed with multiple outputs, each representing a different realization of the solution.

In this context, UQ is naturally integrated by considering a stochastic representation of the PDE coefficients. Specifically, in the framework introduced in Subsection 2.1, let us suppose that the operator $\mathcal{F}$ in Eq. 2.1.2 also depends on an *uncertain* coefficient $\tilde{K}(\mathbf{x}, \omega)$, i.e.,

$$\mathcal{F}(\tilde{K}(\mathbf{x}, \omega), u(\mathbf{x}), \nabla u(\mathbf{x}), \nabla^2 u(\mathbf{x}), \dots) = 0. \tag{2.2.1}$$

The uncertainty of $\tilde{K}(\mathbf{x}, \omega)$ is introduced by a multiplicative noise as follows

$$\tilde{K}(\mathbf{x}, \omega) = K(\mathbf{x})\varepsilon(\mathbf{x}, \omega), \tag{2.2.2}$$

where $\varepsilon(\mathbf{x}, \omega) \in \mathbb{R} \times \mathcal{S}$ is a random noise field[1], $\mathcal{S}$ a random space, ω a random event (or a random variable).

This approach is particularly beneficial in scenarios where material properties such as conductivity, diffusivity, or elasticity coefficients are not well-defined and may exhibit spatial variability due to inherent randomness in the medium. By modeling $K(\mathbf{x})$ as a random variable, one can capture these variations, thereby ensuring that the PINN solutions are capable of accounting for real-world uncertainties.

Let us denote with n_{L+1} the number of samples $(\varepsilon_1(\mathbf{x}), \dots, \varepsilon_{n_{L+1}}(\mathbf{x}))$, from which we obtain the corresponding coefficients $\tilde{K}_r(\mathbf{x})$. Then, according to the given definition

[1] A **random field** is a mathematical model used to describe spatially or temporally varying random phenomena. It generalizes the concept of a stochastic process to multiple dimensions, allowing randomness to be defined over space, time, or both. A random field $K(\mathbf{x}, \omega)$ is a family of random variables indexed by a point $\mathbf{x}$ in a domain $D \subseteq \mathbb{R}^d$ and defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, where $\mathbf{x}$ represents spatial (or spatiotemporal) coordinates in D, $\omega \in \Omega$ is a sample from the probability space, and $K(\mathbf{x}, \omega)$ is a random variable for each fixed $\mathbf{x}$. For a fixed location $\mathbf{x}$, the function $K(\mathbf{x}, \omega)$ describes a random variable, while for a fixed realization ω, the function $K(\mathbf{x}, \omega)$ describes a deterministic function over space.

of a MO-PINN, the number of output neurons has to be fixed to n_{L+1}, and, recalling the first term of Eq. 2.1.3, the PDE residual becomes

$$\mathcal{R}^{\theta}_{\text{PDE}}(\mathbf{x}_i) = \frac{1}{n_{L+1}} \sum_{r=1}^{n_{L+1}} |f_r^{\theta}(\mathbf{x}_i)|^2, \tag{2.2.3}$$

where

$$f_r^{\theta} = \mathcal{F}\left(\tilde{K}_r(\mathbf{x}), u_\theta(\mathbf{x}), \nabla u_\theta(\mathbf{x}), \nabla^2 u_\theta(\mathbf{x}), \dots\right), \qquad r = 1, \dots, n_{L+1}. \tag{2.2.4}$$

Therefore, the loss function Eq. 2.1.3 can be replaced by

$$\mathcal{L}(\theta) := \frac{\lambda_{\text{PDE}}}{N_{\text{PDE}}} \sum_{i=1}^{N_{\text{PDE}}} \left(\frac{1}{n_{L+1}} \sum_{r=1}^{n_{L+1}} |f_r^{\theta}(\mathbf{x}_i)|^2 \right) + \frac{\lambda_{\text{IC}}}{N_{\text{IC}}} \sum_{j=1}^{N_{\text{IC}}} |\mathcal{R}^{\theta}_{\text{IC}}(\mathbf{x}_j)|^2 + \frac{\lambda_{\text{BC}}}{N_{\text{BC}}} \sum_{k=1}^{N_{\text{BC}}} |\mathcal{R}^{\theta}_{\text{BC}}(\mathbf{x}_k)|^2. \tag{2.2.5}$$

By minimizing the loss function in Eq. 2.1.4, a set of solutions $u^{(1)}_{\theta^*}(\mathbf{x}), \dots, u^{(n_{L+1})}_{\theta^*}(\mathbf{x})$ is obtained, each corresponding to a PDE realization with residual Eq. 2.2.4. The final prediction is then given by their average, $\bar{u}_{\theta^*}(\mathbf{x})$.

A key advantage of this approach is that it enables a direct evaluation of the mean and variance of the predicted solutions, which is imperative for comprehending the impact of uncertainty on the system's behavior. By training the network on multiple realizations of K, an ensemble of solutions is generated, thereby providing a probabilistic interpretation of the PDE solution rather than a single deterministic outcome. This guarantees the reliability of the solutions, even when PDE parameters are subject to variability.

2.3 Hyperparameter Optimization

The performance of MO-PINNs, as for any NN, is highly dependent on the choice of hyperparameters, including the number of hidden layers, neurons per layer, learning rate, activation functions, and the number of output neurons. To systematically identify the optimal configuration, a hyperparameter optimization strategy is essential (see [15, 23]).

In this study, the hyperparameter tuning process was executed via a grid search, a method involving the exhaustive exploration of a manually defined subset of the hyperparameter space. This process is repeated for each element in the subset, with the objective of identifying the element that minimizes the prediction error. The hyperparameters considered include the number n_{L+1} of output neurons and the activation function $\phi^{(L+1)}$ for the output layer. For each combination of these, and for different percentages of errors in the distribution of ε_r, the MO-PINN was trained on the fixed dataset $\{\mathbf{x}_j\}_{j=1}^{M_{train}} \subseteq \mathcal{D} \subseteq \mathbb{R}^d$ minimizing the loss function Eq. 2.2.5. Then, the same

combinations were tested on $\{\mathbf{x}_s\}_{s=1}^{M_{test}}$, evaluating the performance of the network based on the mean squared error (MSE) between the predicted and true solutions of the PDE, i.e.,

$$\text{MSE} := \frac{1}{M_{test}} \sum_{s=1}^{M_{test}} [u(\mathbf{x}_s) - \bar{u}_{\theta^*}(\mathbf{x}_s)]^2 .$$

For each percentage of error in the coefficient K, the optimal set of hyperparameters was identified as the combination that resulted in the lowest MSE, ensuring that the model achieves high accuracy while maintaining robustness to uncertainties.

A critical aspect of hyperparameter optimization in MO-PINNs is balancing accuracy and computational efficiency. While the addition of more output neurons has been shown to enhance uncertainty quantification by capturing a more extensive range of potential solutions, it concomitantly increases the computational burden. The selection of the activation function influences both numerical stability and the expressiveness of the model. For instance, the employment of smooth activation functions, such as Swish, can assist in addressing issues related to gradient vanishing. Conversely, traditional activation functions, including Sigmoid, may yield superior performance when the output range is constrained. These considerations highlight the complexity of hyperparameter selection in MO-PINNs, emphasizing the need for a systematic approach to optimize performance while maintaining computational feasibility. For simplicity, from now on let us denote $\bar{u}_{\theta^*}(\mathbf{x})$ with $\bar{u}(\mathbf{x})$.

3 Numerical Results

This section focuses on evaluating the HOMO-PINN methodology by addressing what happens if one considers more than one noisy PDE coefficient and uses them to obtain multiple predicted solutions, and if the mean of these is somehow close to the analytical solution to the problem. For this purpose, let us suppose the coefficient K in Eq. 2.2.1 to be affected by a non-linear noise defined as in Eq. 2.2.2, where

$$\varepsilon_r(x, \omega) = \exp\left\{ {\xi_r}^{(1)}(\omega) \sin(2\pi x) + {\xi_r}^{(2)}(\omega) \cos(8\pi x) + {\xi_r}^{(3)}(\omega) \cos\left(\frac{\pi}{8} x\right) \right\} \tag{3.0.1}$$

with ${\xi_r}^{(i)}$ having a Gaussian distribution with fixed mean $\mu = 1$ and variance $\sigma^2 \in \{0.05, 0.1, 0.2\}$.

For all cases, different numbers of output neurons and activation functions for the output layer are considered, while the weights in the loss function Eq. 2.2.5 are always set as $\lambda_{\text{PDE}} = 1/5$ and $\lambda_{\text{IC}} = \lambda_{\text{BC}} = 2/5 \cdot n_{L+1}$. The rationale behind this setup is that, as the number n_{L+1} increases, the PDE residual tends to dominate the optimization, which may lead to weak enforcement of initial and boundary conditions. By scaling the IC and BC contributions proportionally to n_{L+1}, the different terms of the loss are kept balanced, mitigating potential convergence issues. The specific ratios (1/5 and 2/5) were determined empirically to provide a robust trade-off between accuracy and stability in the test cases. The hidden layers of the HOMO-PINN have the same architecture for all tests, consisting of four layers of 40 neurons each, with hyperbolic

tangent (tanh) as activation function. For the training, 30,000 epochs were conducted using a learning rate of 3×10^{-3} and employed the ADAM optimizer.

The experiments were executed on an NVIDIA GeForce RTX 3090 GPU with an Intel(R) Core(TM) i9-9900K CPU @ 3.60GHz, 8-Cores Processor and 128 GB of RAM, in TensorFlow 2.0 using a Python interpreter.

3.1 Groundwater Flow Equation

The behavior of groundwater flow is influenced by various factors, including the hydraulic conductivity of the subsurface materials, the porosity of the geological formations, the presence of natural and artificial recharge and discharge sources, and the temporal variations in these factors. The general form of the groundwater flow equation is

$$\nabla \cdot (\mathbf{K} \nabla u) + S \frac{\partial u}{\partial t} = Q \qquad t > 0, \tag{3.1.1}$$

where u is the hydraulic head, $\mathbf{K}$ is the hydraulic conductivity tensor, S is the storage coefficient, and Q represents sources or sinks of water (see [5] for details). The equation that describes the water extraction from a point sink in an infinite aquifer is obtained by Eq. 3.1.1 specifying the source term using V_a as the instantaneous source or sink water volume located in δ, which indicates the Dirac delta function. Then,

$$\nabla \cdot (\mathbf{K}(t, \mathbf{x}) \nabla u(t, \mathbf{x})) + S \frac{\partial u(t, \mathbf{x})}{\partial t} = V_a \delta(t) \delta(\mathbf{x}), \tag{3.1.2}$$

for all $\mathbf{x} \in \mathbb{R}^n$, $t > 0$, with initial conditions and boundary conditions

$$u(t, \mathbf{x})|_{t=0} = 0 \quad \forall \mathbf{x} \in \mathbb{R}^n \qquad u(t, \mathbf{x})|_{\mathbf{x} \to \pm\infty} = 0 \quad \forall t > 0$$

To numerically solve the problem Eq. 3.1.2, it is necessary to estimate the Dirac delta function δ, as it is not differentiable everywhere and can therefore only be described in terms of distributions. A Dirac approximation can be built as

$$\delta_\eta(x) = \frac{1}{\eta} \varphi\left(\frac{x}{\eta}\right), \tag{3.1.3}$$

where for $y = x/\eta$

$$\varphi(y) = \begin{cases} \frac{1}{2}(1 + \cos(\pi y)) & \text{if } |y| \leq 1 \\ 0 & \text{otherwise.} \end{cases}$$

From now on, let us consider the one-dimensional spatial case, with the Delta approximation Eq. 3.1.3, then Eq. 3.1.2 becomes

$$-K \frac{\partial^2 u}{\partial x^2} + S \frac{\partial u}{\partial t} = V_a \delta_\eta(t) \delta_\eta(x), \tag{3.1.4}$$

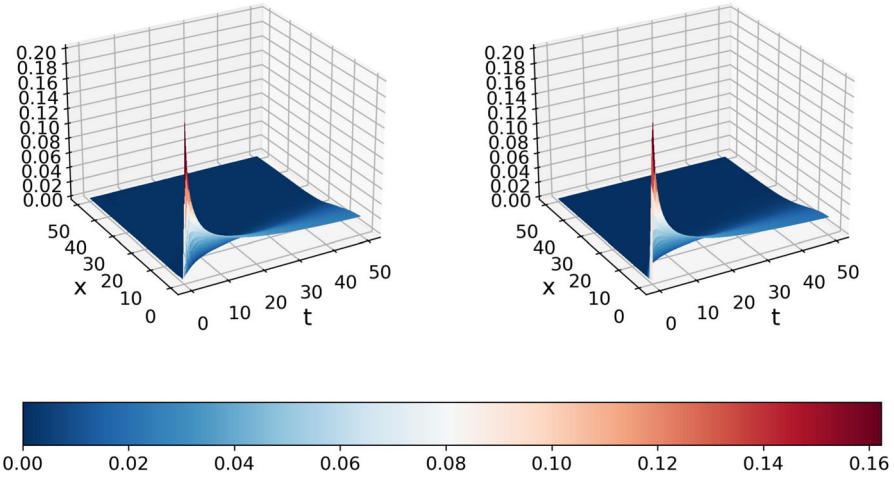


Fig. 2 Predicted (left) and analytical (right) solution for Eq. 3.1.4 through PINNs using sigmoid as activation function for the one neuron output layer

for all $x \in \mathbb{R}, t > 0$, whose analytical solution is

$$u(t,x) = \frac{V_a}{2S}\sqrt{\frac{S}{K\pi t}}\, e^{-\frac{Sx^2}{4Kt}}. \tag{3.1.5}$$

Let us suppose the hydraulic conductivity K in Eq. 3.1.4 to be noisy, i.e., $\tilde{K}_r = K\,\varepsilon_r(x)$, where $K = 1$ and ε_r is defined in Eq. 3.0.1. In addition, let us assume $(t,x) \in \mathcal{D} = [0,50] \times [0,50]$, while setting $S = 1$ and $V_a = \frac{1}{2}$. Then, the residual in Eq. 2.2.4 for this case study is

$$f_r^\theta = -\tilde{K}_r \frac{\partial^2 u}{\partial x^2} + \frac{\partial u}{\partial t} - \frac{1}{2}\delta_\eta(t)\delta_\eta(x), \qquad r = 1, \ldots, n_{L+1}.$$

The hyperparameters considered in the optimization phase include $n_{L+1} \in \{3, 5, 10\}$, the number of output neurons, and $\phi^{(L+1)} \in \{\text{sigmoid}, \tanh, \text{swish}\}$, the activation function for the output layer.

In general, for problems like the groundwater flow equation with this set of IC, BC, and parameters, where the analytical solution lies within the interval [0, 1], the sigmoid activation function is recommended due to its inherent properties. In fact, Fig. 2 shows the accuracy of the predicted solution with respect to the analytical one. This is confirmed by the absolute value of the point-wise difference between the two, as displayed in Fig. 3, where the error is concentrated in the points interested by the Dirac delta approximation Eq. 3.1.3. Nevertheless, even if an unsuited activation function is selected, leading to sub-optimal individual results, the mean of these results remains comparably accurate to that obtained using the optimal activation function. Consequently, increasing n_{L+1} enhances the accuracy of the mean, bringing it closer to the analytical solution. For instance, as can be noticed in Fig. 4, when using swish

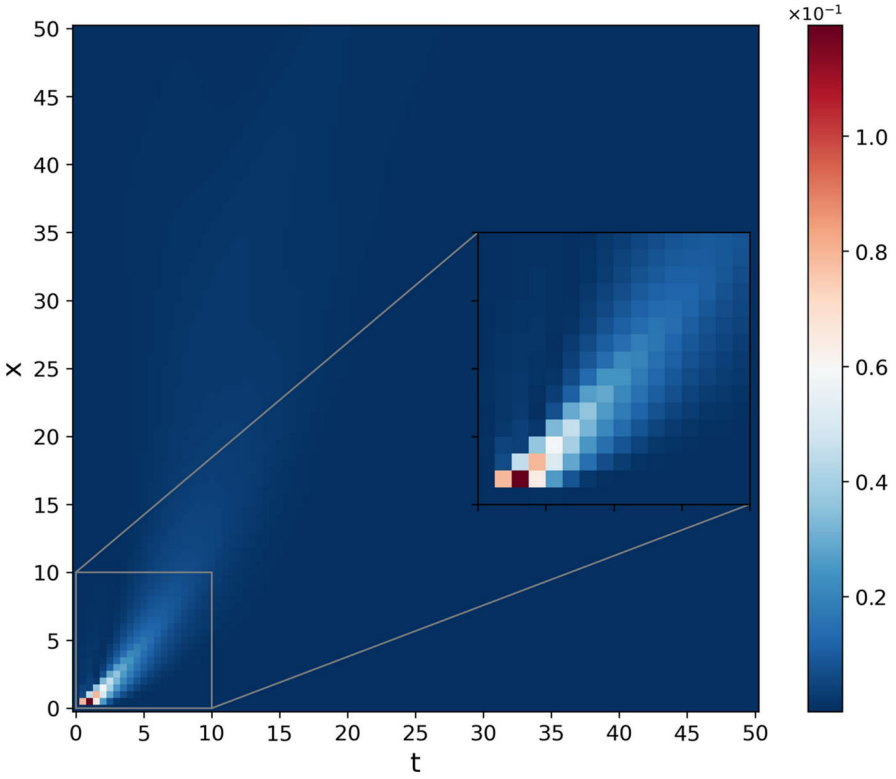


Fig. 3 Absolute value of the point-wise difference between the analytical solution and its prediction through a PINN with one output neuron and sigmoid as the activation function for the output layer. The zoomed-in part of the plot shows the area where this error is concentrated. A solution takes values in [0, 1]

as activation function and setting the number of output neurons to 10, for a 0.2 noise variance on the coefficient K, the mean $\bar{u}_{\theta^*}$ of the predicted solutions $u^{(1)}_{\theta^*}, \ldots, u^{(10)}_{\theta^*}$, associated with different samples $\tilde{K}_1, \ldots, \tilde{K}_{10}$ of the coefficient, stays adjacent to the analytical solution u (see Eq. 3.1.5). As far as the ten single predicted solutions $u^{(r)}_{\theta^*}$ that are plotted in grey in Fig. 4, a far more significant error with respect to the analytical solution can be noticed. This is due not only to the theoretically wrong choice of activation function, but also to the fact that the diffusion coefficient $\tilde{K}_r$ used has a noise percentage of 20%, leading the PINN to solve a different type of the differential problem from the original one.

Figure 5 summarizes the test results with every combination of the considered hyperparameters. More specifically, x, y-axes correspond to the number of neurons and the activation function used for the output layer, respectively, while each colored square represents the maximum among the values, for each of the noise variances on K, of either

$$\text{MaxAbsErr} := \max_{t\in[0,50]} \|u - \bar{u}\|_{L^2([0,50])} \quad \text{or} \quad \text{RelErr} := \frac{\|u(10,x) - \bar{u}(10,x)\|_{L^2([0,25])}}{\|u(10,x)\|_{L^2([0,25])}} \tag{3.1.6}$$

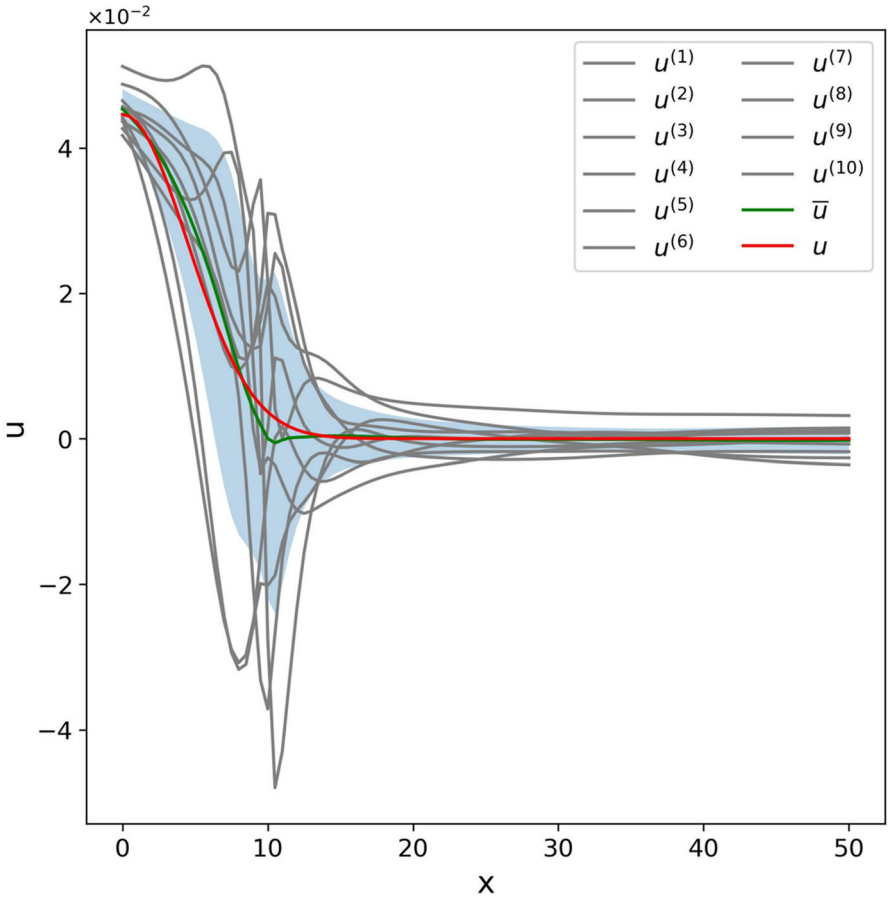
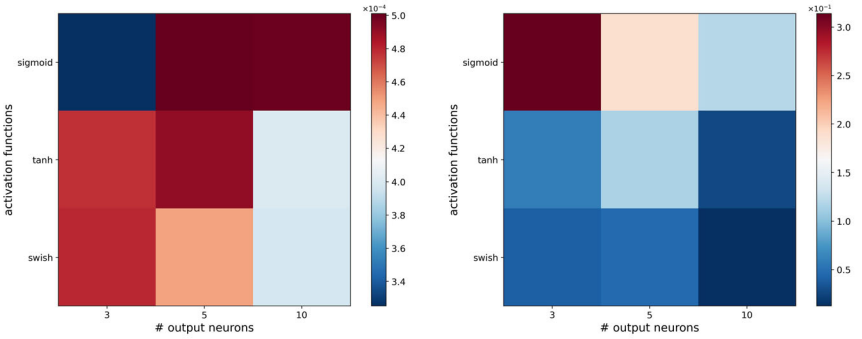


Fig. 4 2D projection for $t = 10$ of single predicted MO-PINN solutions $u^{(r)}$ (in grey), their mean $\bar{u}$ (in green) and analytical solution u (in red) to groundwater flow equation. The predictions are obtained with 10 output neurons using swish activation function on the output layer, while the samples of $\tilde{K}_r$ are generated using Eqs. 2.2.2 and 3.0.1 with variance 0.2

for Fig. 5a and b. The choices for t and x in the definition of RelErr are dictated by the shape of the solution (see Fig. 2). For this kind of problem, considering a restricted relative-kind error leads to more precise evaluations. In support of this, Fig. 5b better shows that the swish activation function is the most suitable when considering a number of output neurons greater than 3, despite it being theoretically not appropriate for this problem. Using the two metrics introduced in Eq. 3.1.6 and focusing on the UQ perspective, Fig. 6 allows to validate the previous results.

Indeed, it shows, despite expecting sigmoid as the preferred activation function, its too variable errors when increasing the noise percentage and therefore the number of output neurons to compensate for it. On the other side, the errors for the other activation functions remain relatively constant for the different noise values.



(a) Max absolute error between the analytical solution and the mean of the predicted solutions, as defined in Eq. (??).

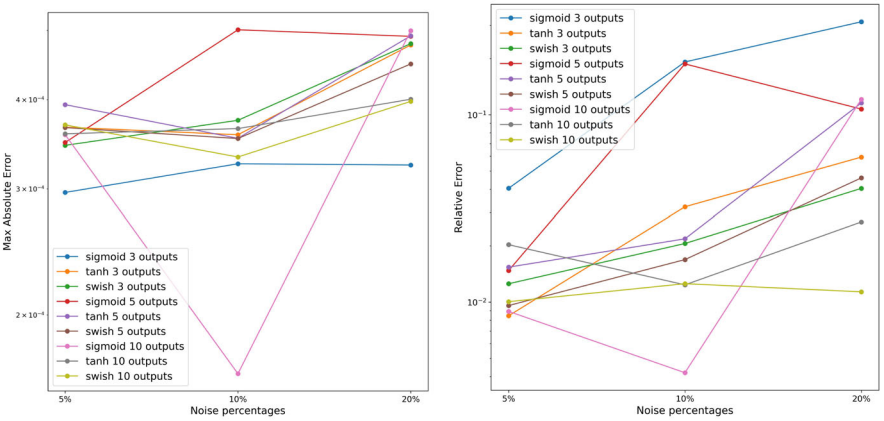
(b) Relative error between the analytical solution and the mean of the predicted solutions, as defined in Eq. (??).

Fig. 5 Heatmap of errors for various combinations of the considered hyperparameters, i.e., number of output neurons on x-axis and activation function for the output layer on y-axis

Lastly, Fig. 7 shows the values of

$$\text{AbsErr} = \|u - \bar{u}\|_{L^2(\Omega)} \tag{3.1.7}$$

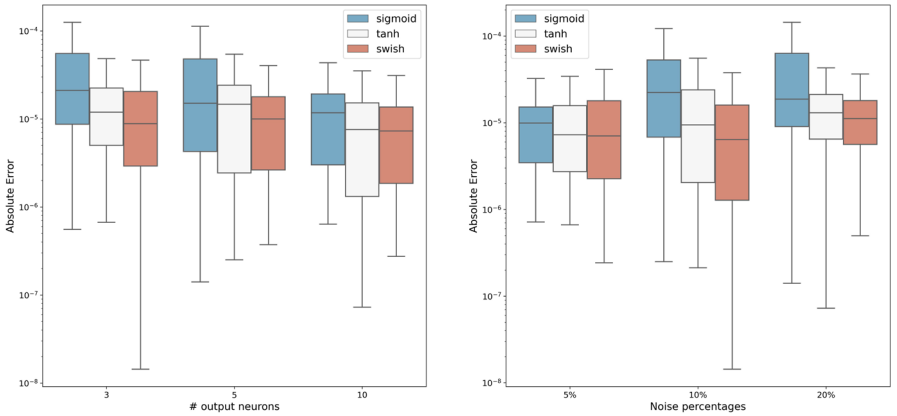
between the predicted and the analytical solution, for all $t \in [0, 50]$, where $\Omega = [0, 50]$. Both the boxplots, comparing the number of output neurons (Fig. 7a) and the noise percentages (Fig. 7b), provide empirical support for the assertion that swish is the optimal choice for this type of problem. Indeed, an examination of both figures reveals that the median of all AbsErr values (represented by the line within the rectangle) is lowest in each subgroup for the swish activation function.



(a) Max absolute error between the analytical solution and the mean of the predicted solutions, as defined in Eq. (3.1.6).

(b) Relative error between the analytical solution the mean of the predicted solutions, as defined in Eq. (3.1.6).

Fig. 6 Errors for various combinations of the considered hyperparameters, i.e., number of output neurons and activation function for the output layer (in different colors). The x-axis indicates the noise percentages on the coefficient K (see Eqs. 2.2.2 and 3.0.1) of the considered PDE, while the y-axis the respective errors



(a) Number of output neurons versus absolute error.

(b) Noise percentage versus absolute error.

Fig. 7 Comparison boxplots for number of output neurons and noise percentages on the coefficient K in Eq. 3.1.4. The y-axis refers to the absolute error between the analytical solution and the median of the predicted solutions, as defined in Eq. 3.1.7. The different colors refer to the choice of activation function

3.2 Poisson Equation

The Poisson equation represents a fundamental second-order PDE that is applicable across a range of mathematical physics and engineering contexts. It generalizes Laplace's equation and can be expressed in the following general form

$$K\,\Delta u = g, \tag{3.2.1}$$

highlighting the diffusion coefficient K, where Δ indicates the Laplacian operator, u is the unknown scalar potential function, and g is a known source term representing the distribution of sources within the domain. The Poisson equation arises naturally in contexts where the potential field is influenced by a spatially varying source term. This means that the equation effectively models scenarios where the distribution of some quantity, such as charge density, mass density, or heat sources, varies throughout a region. The variation in the source term directly impacts the behavior of the potential field, thus necessitating the development of a mathematical framework that can accommodate these changes.

For the purpose of this paper, let us consider the one-dimensional case of Eq. 3.2.1 with $x \in \mathbb{R}$. Choosing the coefficient $K = 1/(3\pi^2)$ and by taking the analytical solution

$$u(x) = \sin^3(\pi x), \tag{3.2.2}$$

the source term can be obtained as

$$g(x) = 2\sin(\pi x)\cos^2(\pi x) - \sin^3(\pi x). \tag{3.2.3}$$

By replacing Eq. 3.2.3 in Eq. 3.2.1, then

$$\frac{1}{3\pi^2}u''(x) = 2\sin(\pi x)\cos^2(\pi x) - \sin^3(\pi x). \tag{3.2.4}$$

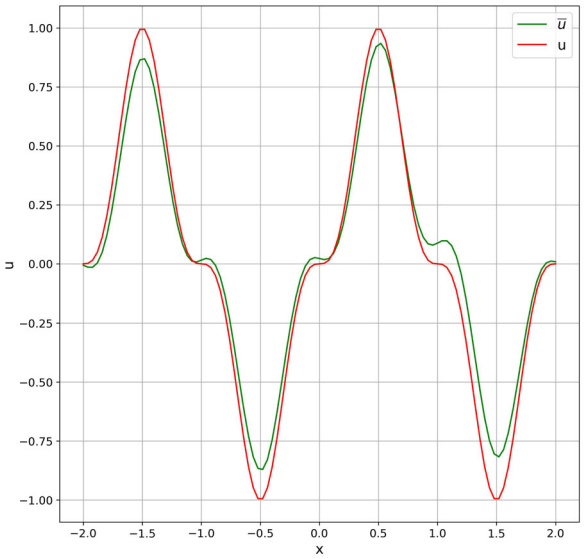


Fig. 8 Predicted (in green) and analytical (in red) solution for Eq. 3.2.4 through PINNs using tanh as activation function for the one neuron output layer

with domain $\mathcal{D} = [-2, 2]$ and initial and boundary conditions $u(-2) = u'(-2) = 0$. As for groundwater flow equation (see Section 3.1), assume the diffusion coefficient K to be affected by a non-linear multiplicative noise (see Eqs. 2.2.2 and 3.0.1), con-

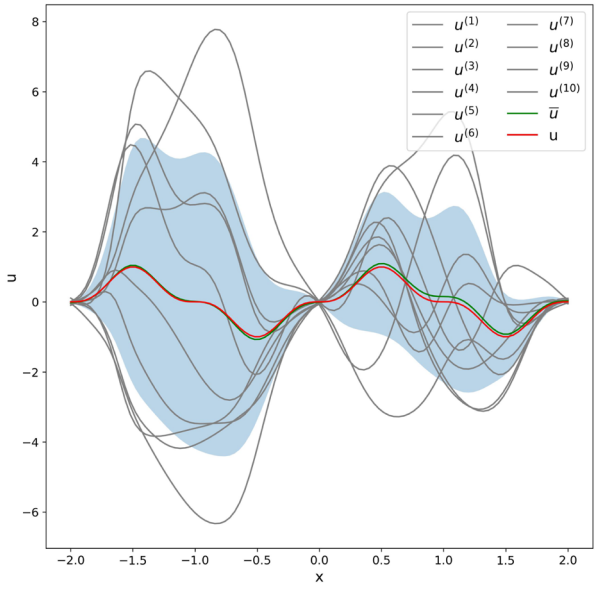


Fig. 9 MO-PINN solutions $u^{(r)}$ (in grey), their mean $\bar{u}$ (in green) and analytical solution u (in red) to Poisson equation. The predictions are obtained with 10 output neurons using ELU activation function, while $\tilde{K}_r$ are sampled with variance 0.2

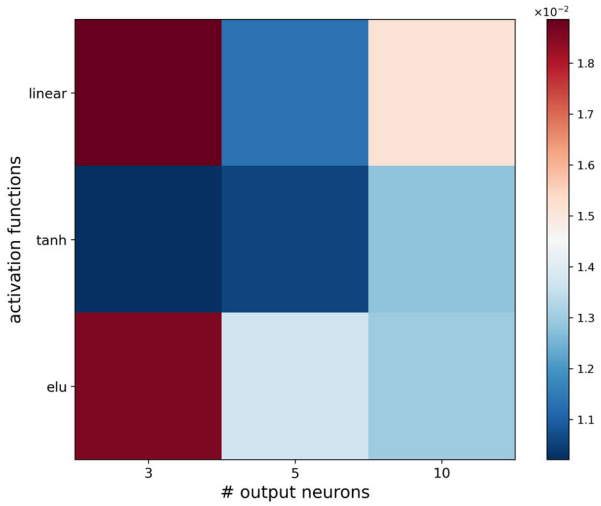


Fig. 10 Heatmap display for various combinations of the considered hyperparameters (i.e., number of output neurons on x-axis and activation function for the output layer on y-axis) of the absolute error between the analytical solution and the mean of the predicted solutions, as defined in Eq. 3.1.7

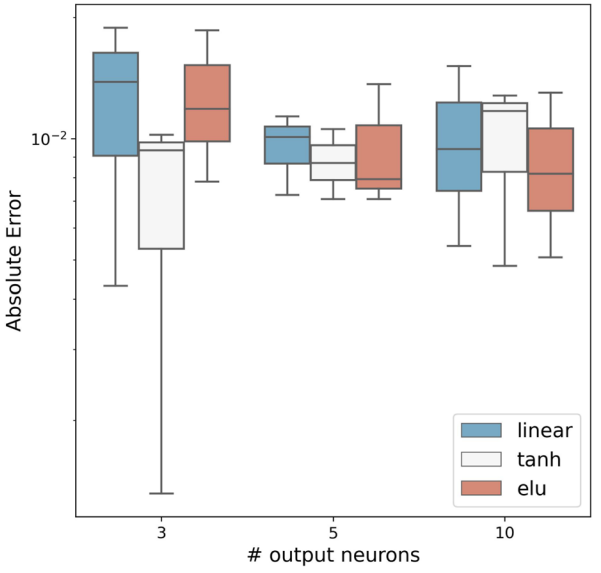


Fig. 11 Comparison boxplot for number of output neurons (on the x-axis), while the y-axis refers to the absolute error defined in Eq. 3.1.7. The line in each box represents the median, whereas the different colors refer to the choice of activation function

sidering the same three different percentages of noise based on the variance of the ε_r sampling, i.e., $\sigma^2 \in \{0.05, 0.1, 0.2\}$. Then, the PDE residual in Eq. 2.2.3 becomes

$$\mathcal{R}^{\theta}_{\text{PDE}}(x_i) = \frac{1}{n_{L+1}} \sum_{r=1}^{n_{L+1}} |\tilde{K}_r u''(x_i) - 2\sin(\pi x_i)\cos^2(\pi x_i) + \sin^3(\pi x_i)|^2.$$

Figure 8 illustrates that the analytical solution, as defined in Eq. 3.2.2, is constrained to the interval $[-1, 1]$, thereby suggesting that tanh represents the optimal choice of activation function for a PINN with one output neuron, i.e., $n_{L+1} = 1$. As explained in Section 2.3, a grid search has been conducted on all possible combinations of the hyperparameters, specifically the number of output neurons $n_{L+1} \in \{3, 5, 10\}$ and the activation function for the output layer $\phi^{(L+1)} \in \{\text{linear, tanh, ELU}\}$, in order to evaluate the performance of each combination. Figure 9 shows the results obtained with ELU as activation function for the output layer and 10 as the number of output neurons. As can be noticed, despite the single predicted solutions corresponding to the coefficient $\tilde{K}_r$ with variance 0.2 being inaccurate, their mean is close to the analytical solution. Indeed, although the most appropriate activation function for this problem, i.e., tanh, has optimal results, Fig. 10 displays that the other two chosen activation functions improve their accuracy when the number of output neurons increases. In support of this, Fig. 11 highlights the irrelevance of the choice of the activation function when n_{L+1} grows. While, as can be seen in Fig. 12, the choice of the activation function is crucial for the stability of the problem when the coefficient is affected by a 20%

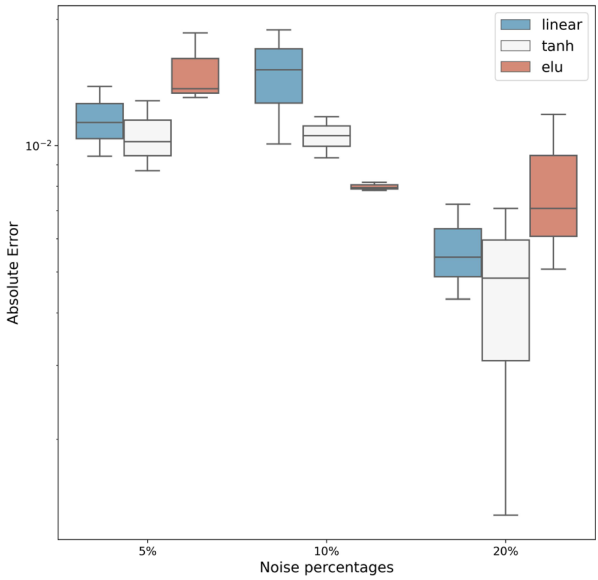


Fig. 12 Comparison boxplot for noise percentages on the coefficient K in Eq. 3.2.1 (on the x-axis), while the y-axis refers to the absolute error defined in Eq. 3.1.7. The line in each box represents the median, whereas the different colors refer to the choice of activation function

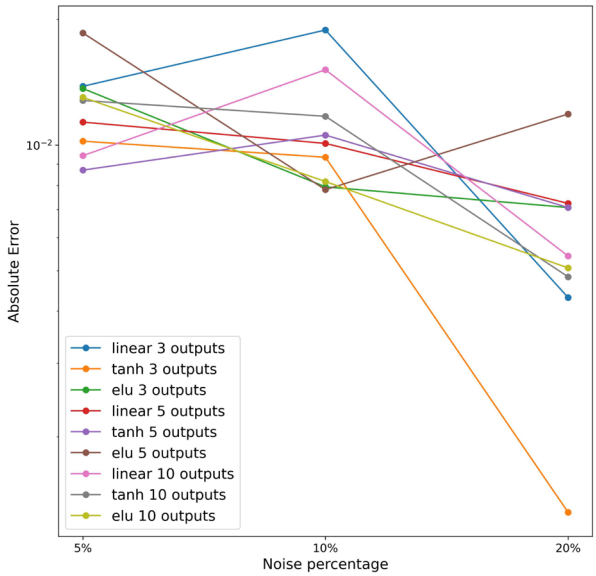


Fig. 13 Errors for various combinations of the considered hyperparameters, i.e., number of output neurons and activation function for the output layer (in different colors). The x-axis indicates the noise percentages on the coefficient K of the considered PDE, while the y-axis the respective absolute errors

noise. In this case, using tanh (in orange in Fig. 13) is necessary to significantly lower the approximation error on the solution.

4 Conclusions

In this study, a novel approach, HOMO-PINNs, has been developed and presented in order to enhance the accuracy of solutions to PDEs under uncertainty. This methodology integrates hyperparameter optimization with UQ to identify the most effective hyperparameters for PINNs. The experimental findings on both groundwater flow and Poisson equations demonstrate the robustness of HOMO-PINNs in handling various degrees of uncertainty on the stochastic coefficient $K(x)$. The numerical results observed that higher variance in $K(x)$ results in increased solution dispersion, necessitating more output neurons to accurately capture uncertainty. Conversely, the selection of appropriate activation functions has been shown to significantly mitigate the instability introduced by random fluctuations in $K(x)$. It was therefore observed that the selection of the number of output neurons and the choice of activation functions impact the performance of the NN. It is worth noting that our findings imply that, in instances where an erroneous activation function is employed, increasing the number of output neurons can facilitate the preservation of the accuracy of the predicted solution. The mean prediction of MO-PINNs demonstrates resilience to high uncertainty, contingent on the implementation of an optimized hyperparameter strategy. This adaptability underscores the necessity of optimizing hyperparameters tailored to the distinctive

characteristics of each problem, thereby illustrating the interplay between hyperparameter selections and the accuracy of solutions across varying levels of coefficient noise.

Acknowledgements We would like to thank the reviewers for their valuable comments, which helped to improve our paper. S. Cuomo also acknowledges *GNCS-INdAM* and the *UMI-TAA, UMI-AI* research groups. This work has been supported by the project: *PNRR Centro Nazionale HPC, Big Data e Quantum Computing*, ($CN\_00000013$)(CUP: $E63C22000980007$), under the NRRP MUR program funded by the NextGenerationEU. S. Cuomo has been partially supported by the Italian PRIN project *Numerical Optimization with Adaptive accuracy and applications to machine learning* (CUP: $E53D23007690006$), Progetti di Ricerca di Interesse Nazionale 2022.

Author Contributions M.D. and L.P. wrote the original draft, curated the conceptualization of the methodology, software, and validation of the results. A.L. and S.C. supervised the methodology and conceptualization, by also participating in the validation and visualization.

Funding Open access funding provided by Università degli Studi di Napoli Federico II within the CRUI-CARE Agreement.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Competing Interests The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Al Safwan A, Song C, Waheed UB (2021) Is it time to swish? Comparing activation functions in solving the Helmholtz equation using PINNs, In: 82nd EAGE Annual Conference & Exhibition, European Association of Geoscientists & Engineers. pp 1–5
2. Bharadwaja B, Nabian MA, Sharma B, Choudhry S, Alankar A (2022) Physics-informed machine learning and uncertainty quantification for mechanics of heterogeneous materials. Integ Mater Manufact Innov 11:607–627
3. Cai S, Mao Z, Wang Z, Yin M, Karniadakis GE (2021) Physics-informed neural networks (PINNs) for fluid mechanics: a review. Acta Mech Sin 37:1727–1738
4. Chang W, Yuchen F, Yongqing Z, Xin L, Chaoqun Z, Heyang W (2023) Multi-output physics-informed neural networks model based on the Runge-Kutta method. Chinese J Theoret Appl Mech 55:2405–2416
5. Cuomo S, Rosa M, Giampaolo F, Izzo S, Cola VS (2023) Solving groundwater flow equation using physics-informed neural networks. Comput Math Appl 145:106–123
6. Cuomo S, Rosa M, Piccialli F, Pompameo L (2024) Railway safety through predictive vertical displacement analysis using the PINN-EKF synergy. Math Comput Simul 223:368–379
7. Ryck T, Jagtap AD, Mishra S (2024) Error estimates for physics-informed neural networks approximating the Navier-Stokes equations. IMA J Numer Anal 44:83–119

8. Escapil-Inchauspé P, Ruz GA (2023) Hyper-parameter tuning of physics-informed neural networks: application to Helmholtz problems. Neurocomputing 561:126826
9. Fokas A (2004) Boundary-value problems for linear PDEs with variable coefficients. Proceedings of the Royal Society of London. Series A: Math, Phys Eng Sci 460:1131–1151
10. Graham IG, Kuo FY, Nuyens D, Scheichl R, Sloan IH (2011) Quasi-Monte Carlo methods for elliptic PDEs with random coefficients and applications. J Comput Phys 230:3668–3694
11. Hao TT, Yan WJ, Chen JB, Sun TT, Yuen KV (2024) Multi-output multi-physics-informed neural network for learning dimension-reduced probability density evolution equation with unknown spatio-temporal-dependent coefficients. Mech Syst Signal Process 220:111683
12. Jagtap AD, Kawaguchi K, Karniadakis GE (2020) Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. J Comput Phys 404:109136
13. Khoo Y, Lu J, Ying L (2021) Solving parametric PDE problems with artificial neural networks. Eur J Appl Math 32:421–435
14. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980
15. Le DK, Guo M, Yoon JY (2023) Hyperparameter optimization for physics-informed neural networks utilizing genetic algorithm. Available at SSRN 4590874
16. Li J, Long X, Deng X, Jiang W, Zhou K, Jiang C, Zhang X (2024) A principled distance-aware uncertainty quantification approach for enhancing the reliability of physics-informed neural network. Reliab Eng Syst Safety 245:109963
17. Li W, Lee KM (2021) Physics informed neural network for parameter identification and boundary force estimation of compliant and biomechanical systems. Int J Intell Robot Appl 5:313–325
18. Liu W, Liu Y, Li H, Yang Y (2023) Multi-output physics-informed neural network for one-and two-dimensional nonlinear time distributed-order models. Netw Heterogen Media 18
19. Qian Y, Zhang Y, Huang Y, Dong S (2023) Error analysis of physics-informed neural networks for approximating dynamic PDEs of second order in time. arXiv:2303.12245
20. Raissi M, Perdikaris P, Karniadakis GE (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
21. Ramachandran P, Zoph B, Le QV (2017) Searching for activation functions. arXiv:1710.05941
22. Sharma P, Evans L, Tindall M, Nithiarasu P (2023) Hyperparameter selection for physics-informed neural networks (PINNs)-application to discontinuous heat conduction problems. In: Part B (ed) Numerical Heat Transfer. Fundamentals, pp 1–15
23. Wang Y, Han X, Chang CY, Zha D, Braga-Neto U, Hu X (2023) Auto-PINN: understanding and optimizing physics-informed neural architecture. arXiv:2205.13748v2
24. Yang M, Foster JT (2022) Multi-output physics-informed neural networks for forward and inverse PDE problems with uncertainties. Comput Methods Appl Mech Eng 402:115041
25. Zhao S, Peng Y, Zhang Y, Wang H (2022) Parameter estimation of power electronic converters with physics-informed machine learning. IEEE Trans Power Electron 37:11567–11578

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.