Towards Optimally Weighted Physics-Informed Neural Networks in Ocean Modelling

Taco de Wolff, Hugo Carrillo, Luis Martí, Nayat Sanchez-Pi

Towards Optimally Weighted Physics-Informed Neural Networks in Ocean Modelling

Taco de Wolff, Hugo Carrillo, Luis Martí & Nayat Sánchez-Pi
Inria Chile Research Center
Av. Apoquindo 2827, piso 12, Las Condes, Santiago, Chile
{taco.dewolff,hugo.carrillo,luis.marti,nayat.sanchez-pi}@inria.cl

Abstract

The carbon pump of the world's ocean plays a vital role in the biosphere and climate of the earth, urging improved understanding of the functions and influences of the ocean for climate change analyses. State-of-the-art techniques are required to develop models that can capture the complexity of ocean currents and temperature flows. This work explores the benefits of using physics-informed neural networks (PINNs) for solving partial differential equations related to ocean modeling; such as the Burgers, wave, and advection-diffusion equations. We explore the trade-offs of using data vs. physical models in PINNs for solving partial differential equations. PINNs account for the deviation from physical laws in order to improve learning and generalization. We observed how the relative weight between the data and physical model in the loss function influence training results, where small data sets benefit more from the added physics information.

1 Introduction

The ocean plays a key role in the biosphere [18]. It regulates the carbon cycle by absorbing emitted CO_2 through the biological pump and dissipates a large part of the heat that is retained in the atmosphere by the remaining CO_2 and other greenhouse gases. The biological pump is driven by photosynthetic microalgae, herbivores, and decomposing bacteria. Understanding the drivers of micro- and macroorganisms in the ocean is of paramount importance to understand the functioning of ecosystems and the efficiency of the biological pump in sequestering carbon and thus abating climate change.

Similarly, there is also a clear scientific consensus about the effects of climate change on the ocean. Changes like the shift in temperature, an increase of acidification, deoxygenation of water masses, perturbations in nutrient availability, and biomass productivity to mention a few, have a dramatic impact on almost all forms of life in the ocean with further consequences on food security, ecosystem services, and the well-being of coastal communities and humankind in general.

Consequently, we end up with a cyclic dependency: to understand and create strategies to mitigate climate change it is of primordial importance to understand the ocean and to understand the ocean it is necessary to understand how climate change is impacting it. Therefore, this is not only an urgent but also a scientifically demanding task that must be addressed with a cohort approach involving different scientific areas: state-of-the-art artificial intelligence, machine learning, applied math, modelling, and simulation, and of course marine biology and oceanography.

That is why it can be hypothesized that it is necessary to develop new state-of-the-art artificial intelligence, machine learning, and mathematical modelling tools that will enable us to move forward with the understanding of our ocean and to understand, predict, and hopefully mitigate the consequences of climate change.

Preprint. Under review.

The ocean is mostly a fluid in almost permanent movement – subjected to currents, tides, waves and other interactions – where different where organisms, atmosphere, and continents interact. Modelling oceanic fluid dynamics is essential but also very expensive in the computational sense. Mathematical models like the Navier-Stokes equations [21] can express most of the processes of interest [14]. However, the high computational cost involved in applying them with the correct precision level makes their application intractable. Machine learning (ML) and, in particular, neural networks are a competitive alternative. However, these approaches in general require a large amount of data to serve as examples for the learning process. In oceanographic research, obtaining data is challenging and expensive. The approach of hybrid models seems to be a good alternative to tackle this problem.

Physics-informed neural networks (PINNs) [16] are a hybrid approach that take into account a data-based neural network model and a physics-informed mechanistic model, which are two different paradigms. They offer a framework where existing knowledge about a physical phenomenon and empirical data gathered about it. This general concept has been previously explored and is known as data assimilation [25], but PINNs bring a novel and sound approach to consolidate the existing models and sampled data. This feature makes PINNs particularly appealing for the above-described problems and has lead to some preliminary studies [5, 13].

Furthermore, it could be argued that by incorporating a rule that promotes the consistency of the neural with *a priori* existing knowledge, the internal representations created during the training phase could be easier to understand and interpret by domain experts and, therefore, become not only a modelling but a research tool with a more broad impact.

This paper deals with the problem of predicting values of physical laws at given points in space and time relying on PINNs with a focus on ocean modelling. Our main goal is provide a first attempt -to the best of our knowledge- at determining what is the importance and relevance of the physical component in PINNs in improving the predictive power of a neural network concerning the amount of data available. By answering this question we aim to find a procedure for efficient configuration of parameters with numerical experiments inferred from the characteristics of the problem and the data available.

The contributions of this papers are:

1. to study the influence of the physics information and the scenarios where using it is more effective,
2. a first step towards the automatic parameterization of PINNs using oceanographic modelling as test case, and
3. we study the effects of other hyperparameters as the length and width of the neural network and activation functions on the optimal weight.

The rest of this paper is organized as follows. In Section 2 we present a brief rationale regarding the context of the paper, PINNs, modelling ocean fluids and the particular equations that we want to address (see 2.1). After that, in Section 3, we deal with the theoretical aspects of our work. Then, in Section 4, we describe and analyze the experimental studies involving PINNs for the estimation of the solution and parameters of the PDEs. Finally, in Section 6 we put forward our conclusions of outline out future work.

2 Foundations

High-dimensional partial differential equations (PDEs) [17] are a common fixture in areas as diverse as physics, chemistry, engineering, finance, etc. Their enormous success has been hampered by its limited computational viability. Numerical methods for solving PDEs such as finite difference or finite element methods become infeasible in higher dimensions due to the explosion in the number of grid points and the demand for reduced time step sizes.

Addressing PDE scalability by approximating them by neural networks has been considered in various forms either by creating a training data set by directly sampling from an *a priori* fixed mesh or by directly sampling the solution in a random manner (see [19] for an overview). Consequently, a database of synthetic data coming from the numerical resolution of PDE-based models is generated on a broad range of scenarios. These data sets are then used to train and validate deep neural networks as in [4, 6].

These approaches have the limitation of high computational costs for creating such data sets. To subvert the high computational costs, PINNs were recently proposed by [16] as a hybrid approach that considers a process where a source of physical knowledge in the form of PDEs is available. They specify the problem of training the neural networks as a multi-objective learning task, where we want to minimize the error with the data as well as the error with a physical law. Now we have three problems instead of one: minimizing the error with the data, minimizing the error with the physical law, and the combination of training for both objectives. [16] solves this by incorporating a physics loss term to the loss function, including a relative weight between both loss terms.

As it is a recent topic, there is not much literature about weighted PINNs. Having optimal weights for the loss functions in PINNs could help us to improve the performance of deep learning solvers for PDEs and so make the computations cheaper than regular PINNs. In [23], the authors propose optimal weights for linear PDEs under the knowledge of boundary values and the differential equation. They use the structure and properties of the PDEs studied.

In [26], the authors propose a self-adaptive loss balanced PINNs (lbPINNs) to solve the incompressible Navier-Stokes equation by an empirical search of the weights. In [24], the authors also chose empirically the weights to the loss functions in PINNs for study myocardial perfusion in MRI. On the other hand, in [8] the authors evaluate the different results given by a few amount of weights in the loss functions for the advection-diffusion equation.

In this work, we study the influence of the weight on a validation quality metric, in particular the relative error of the neural network solution. In addition, we study the effects of other hyperparameters as the length and width of the neural network and activation functions on the optimal weight.

2.1 Model equations related to the Ocean

The ocean is characterized by its fluid nature. It is constituted mainly by water in different forms that is subjected to the effects of waves created by wind, currents, tides, among others. Therefore, in order to understand the ocean it is necessary to take into account this fundamental characteristic and to model it and the intervening phenomena. This gains particular relevance with the emergence of the climate change phenomenon and, therefore, we need trying to model, forecast and device policies to attempt to mitigate its effects.

Modelling oceanic fluid dynamics is essential but also very expensive in the computational sense. Mathematical models like the Navier-Stokes equations [21] can express most of the processes of interest. However, the high computational cost involved in applying them with the correct precision level makes their application intractable.

There are a number of equations that are of particular interest when modelling the ocean that are the focus of this work. In particular,

- *Burgers equation* [3]: appears often as a simplification to understand the main properties of the Navier-Stokes equations. It is a one-dimensional equation where the pressure is neglected but the effects of the nonlinear and viscous terms remain, hence as in the Navier-Stokes equations a Reynolds number can be defined [15]. Hence, it is frequently used as a tool that is used to understand some of the inside behavior of the general problem.
- *Wave equation* [20]: in this work we include the wave equation since, under certain suitable conditions, it can be seen as a reduced model of the shallow-water equation which models the water height behavior in coasts and channels.
- *Advection-diffusion equations* [22]: models the behavior of temperature in a fluid, considering its velocity in the advective term and a diffusion phenomenon of the temperature.

For the sake of briefness, a full description of these equations along with the details about how they are used in our experiments is given on Section 4.

In this work we study the behavior of the relative error (validation loss) with respect to the weight in the physical part, for Burgers, wave, and advection-diffusion equations. In [8] the authors study PINNs for advection-difussion equations, also considering different weights. On the other hand, it is very usual in the literature to find only the use of boundary and initial conditions as data, while a large amount of interior data is considered for data driven discovery of PDEs.

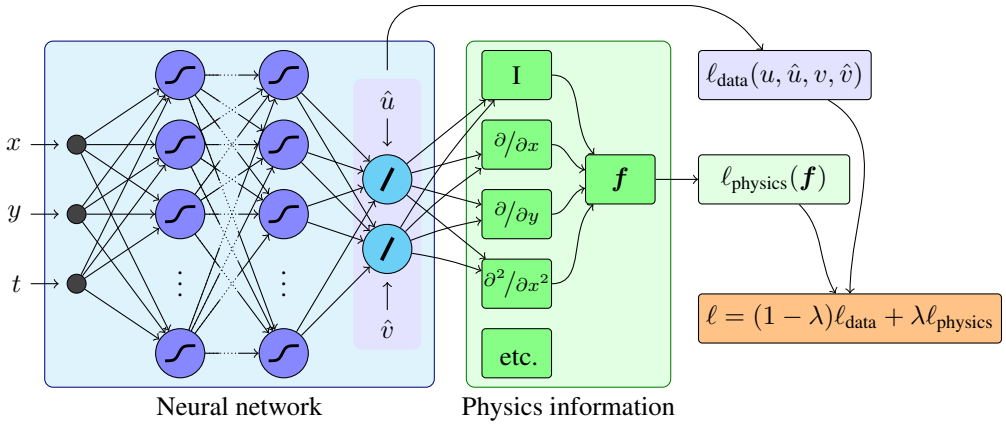


Figure 1: Schematic representation of a physics-informed neural network with inputs x, y, and t; outputs $\hat{u}$ and $\hat{v}$. In this work we applied metaheuristics for determining the optimal hyperparameters for each case. Using automatic gradient calculation we can differentiate the neural network by its input variables and construct a physics error function f. The loss function involves a loss term for the data and a loss term for the physics function.

In this work we consider a small amount of scattered data from the whole closed cylinder where the spatiotemporal PDEs are defined, which can simulate more likely the acquisition of sparse data in a huge domain.

3 Method

As stated, our main goal is to understand the viability and particularities of the use of PINNs for oceanographic modelling. One particularity of this type of problem is the scarcity of data, specially if contrasted for the data-hungry needs of deep learning models that are currently the state of the art. That is why it is so important to understand the value that the physics information term can bring in order to "fill the gaps" where data is scarce or not available.

We also hypothesize that, even in cases where there is a sufficient amount of data, adding a physics-informed loss function can potentially lead to a better interpretability of the trained model. This term can be seen as regularization term that would prompt the network to create representations that are consistent with the current knowledge of the phenomena. This, then should allow in the future to have an easier interpretation, assimilation, validation and acceptance by domain experts.

PINNs are neural network models that are trained to obey laws in physics described by partial differential equations (PDEs). They can be used used to solve supervised tasks in which we both minimize the error with respect to the data and to the physics law. We define the loss function as follows

$$\ell = (1-\lambda)\ell_{\text{data}} + \lambda\ell_{\text{physics}} \tag{1}$$

where ℓ_{data} is the loss with respect to the data, ℓ_{physics} is the loss with respect to the PDEs of a physics law, and $\lambda \in [0, 1]$ is the relative weight of the physics loss. If f^j is the error of each physical condition, then using the mean-squared error for both losses we obtain

$$\ell = (1-\lambda) \sum_{i=0}^{N_u-1} (y_i - \hat{y}_i)^2 + \lambda \sum_j \sum_{i=0}^{N_f-1} (f_i^j)^2 \,. \tag{2}$$

Any neural network such as fully connected, recurrent, and convolutional networks can be used, where we make use of automatic gradient calculations as used by back-propagation to compute derivatives of the output variables with respect to the input variables. An example of such a network is shown in Fig. 1, with f representing the PDE rewritten as $f = 0$.

The network is designed with input variables x, y, and t and output variables $\hat{u}$ and $\hat{v}$. The output variables are used directly to calculate the data loss term, while for the physics loss term we

differentiate the variables with respect to the input variables as needed for the physics loss function. Observe that the neural network must have input variables that correspond to the physics law's derivative terms. That is, if $\partial u/\partial x$ is part of our physics loss function, then x must be an input variable and u an output variable of our neural network.

The objective is to train both in a supervisory manner with measured or simulated data (i. e. ℓ_{data}), as well as training to minimize the departure from the physics law. The ℓ_{physics} term ensures that the neural network generalizes better for unseen data by preventing overfitting. The physics loss encourages that the output variables are not just trained to a local region around the input values of the given data, but that also their first and/or second-degree derivations (depending on the PDEs) match with our physical understanding of the model thus spurring better predictive power outside the region of the training data.

We denote $\mathbb{N}_l \times \mathbb{M}_l$ as a multi-layer perceptron (MLP) of $\mathbb{N}_l$ hidden layers and $\mathbb{M}_l$ neurons per layer. As part of the experimentation we assess different activation functions. As our optimization problem is non-convex, the stochastic gradient-descent Adam optimizer was used [10] with different learning rates depending on the problem. The weights of the neural network are initialized randomly using Xavier's initialization method [7]. As we intend to analyze how activation functions play a role in learning the derivations of PDEs the $\tanh$, GELU, Softplus, LogSigmoid, Sigmoid, TanhShrink, CELU, Softsign, and ReLU activation functions.

We select a sample subset $(X_i, Y_i) \forall i \in [0, N_u - 1]$ at random from the entire solution space for training the data loss, with $X_i \in \mathbb{R}^D$ the D features and $Y_i \in \mathbb{R}^P$ the labels of dimension P. For the physics loss we select a subset $X_i \forall i \in [0, N_f - 1]$ at random for which we evaluate the PDE error. Note that for the physics loss the solution is not needed, allowing applications to be able to train even when few data are available of the solution. In general we pick $N_u \ll N_f$.

4 Experiments

As stated in Section 2, we consider three models: the first ones, the Burgers equation, is a nonlinear one-dimensional and the other two, wave, and advection-diffusion equations, are linear and two dimensional PDEs.

4.1 Datasets preparation

In order to generate data for training, validation, and testing, we simulate the PDEs using a Fourier spectral method for the one dimensional model, finite element methods (FEM) for the wave equation and the explicit analytical solution for the advection-diffusion equation. Simulations for the two-dimensional problems were performed using FEniCS [1].

Burgers equation We consider the following Burgers equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}\,, \tag{3}$$

where ν is a diffusion coefficient. This equation usually appears in the context of fluid mechanics, and more specifically models one-dimensional internal waves in deep ocean. It represents a hyperbolic conservation law as $\nu \to 0$ and it is the simplest model for analyzing the effect of nonlinear advection and diffusion in a combined way. Notice that the Burgers equation can be written as a physics loss term of (2) as $f_{\text{Burgers}}[u] = 0$.

We simulated the Burgers equation in the spatiotemporal domain $[0, L] \times [0, T]$, using a Fourier spectral method as described in [2].

The diffusion coefficient is taken as $\nu = \left(\frac{0.01}{\pi}\right)$.

We consider the initial condition

$$u(x, 0) = -\sin(\pi x), \text{ with } x \in (-1, 1)\,, \tag{4}$$

and boundary conditions

$$u(1, t) = u(-1, t) = 0, \text{ with } t \in (0, 1)\,, \tag{5}$$

with $N = 512$ spatial points in a uniform grid in $[-1, 1]$ and 100 points in time defined in a uniform grid in $[0, 1]$.

Wave equation In this case, we consider the wave equation

$$\frac{\partial^2 \eta}{\partial t^2} - \nabla \cdot (H \nabla \eta) = 0\,, \tag{6}$$

where $\eta : \overline{\Omega} \times \mathbb{R}_0^+ \to \mathbb{R}$ represents the superficial fluctuations of a water container or channel and $H : \overline{\Omega} \to \mathbb{R}_0^+$ is the depth of the water from a reference level. This equation can be written as $f_{\text{waves}}[\eta, H] = 0$. This is where the physics loss is taken from for this PDE.

We simulate the wave equation in a rectangular domain in the time interval $]0, T_f[$, with Dirichlet boundary conditions for η as

$$\eta = 0, \text{ on } \partial\Omega \times]0, T_f[\,, \tag{7}$$

and initial conditions on Ω

$$\begin{array}{ll} \eta(x, y, 0) & = \exp\left(-10 \cdot \left((x - 0.5)^2 + (y - 0.75)^2\right)\right)\,, \\ \frac{\partial \eta}{\partial t}(x, y, 0) & = 0\,. \end{array} \tag{8}$$

On the other hand, the depth H is given by

$$H(x, y) = (1 - x)(2 - \sin(3\pi y)) \quad \text{in } \overline{\Omega}\,. \tag{9}$$

We implement FEM in this equation considering Lagrange finite elements of degree 1. The time scheme is explicit and $T_f = 1.0$, $n = 100$.

Advection-diffusion equation We consider the advection-diffusion equation for heat transfer, assuming that the diffusion coefficient is constant, there are no sources nor sinks of heat, and the temperature depends only on x, y, and t, that is,

$$\frac{\partial T}{\partial t} = D \nabla^2 T - \boldsymbol{u} \cdot \nabla T\,, \tag{10}$$

where $T : \overline{\Omega} \times \mathbb{R}_0^+ \to \mathbb{R}$ is the temperature in K, $D = 0.1 \frac{\text{m}^2}{\text{s}}$ is the thermal diffusivity considered in this work, and $\boldsymbol{u} : \overline{\Omega} \times \mathbb{R}_0^+ \to \mathbb{R}^2$ the velocity field in $\frac{m}{s}$. The equation can be written as a physics loss term as

$$f_{AD}[u, v, T] = 0\,. \tag{11}$$

We simulate the advection-diffusion equation in $\Omega \times (0, T_f)$ where $\Omega = (0, L_x) \times (0, L_y)$, where $L_x = L_y = 1.0$ and $T_f = 1.0$. In addition, $D = 0.02$, $u = (\cos(\phi)$ and $v = \sin(\phi))$, where $\phi = 22.5°$. We consider the following initial conditions

$$T(x, y, 0) = \exp\left(-\frac{x^2 + y^2}{D}\right), \text{ in}\Omega\,, \tag{12}$$

and boundary condition

$$T(x, y, t) = \frac{1}{4t + 1} \exp\left[-\frac{x^2 + y^2 + t^2(u^2 + v^2) - 2t(xu + yv)}{D(4t + 1)}\right] \qquad \text{on } \partial\Omega\,. \tag{13}$$

As the coefficients are constant and the domain is a square, we are able to solve this linear equation by known analytic methods. The explicit solution of this equation is

$$T(x, y, t) = \frac{1}{4t + 1} \exp\left[-\frac{x^2 + y^2 + t^2(u^2 + v^2) - 2t(xu + yv)}{D(4t + 1)}\right]\,, \tag{14}$$

so we generate this solution in a uniform unit square mesh with 40 points in each spatial dimension.

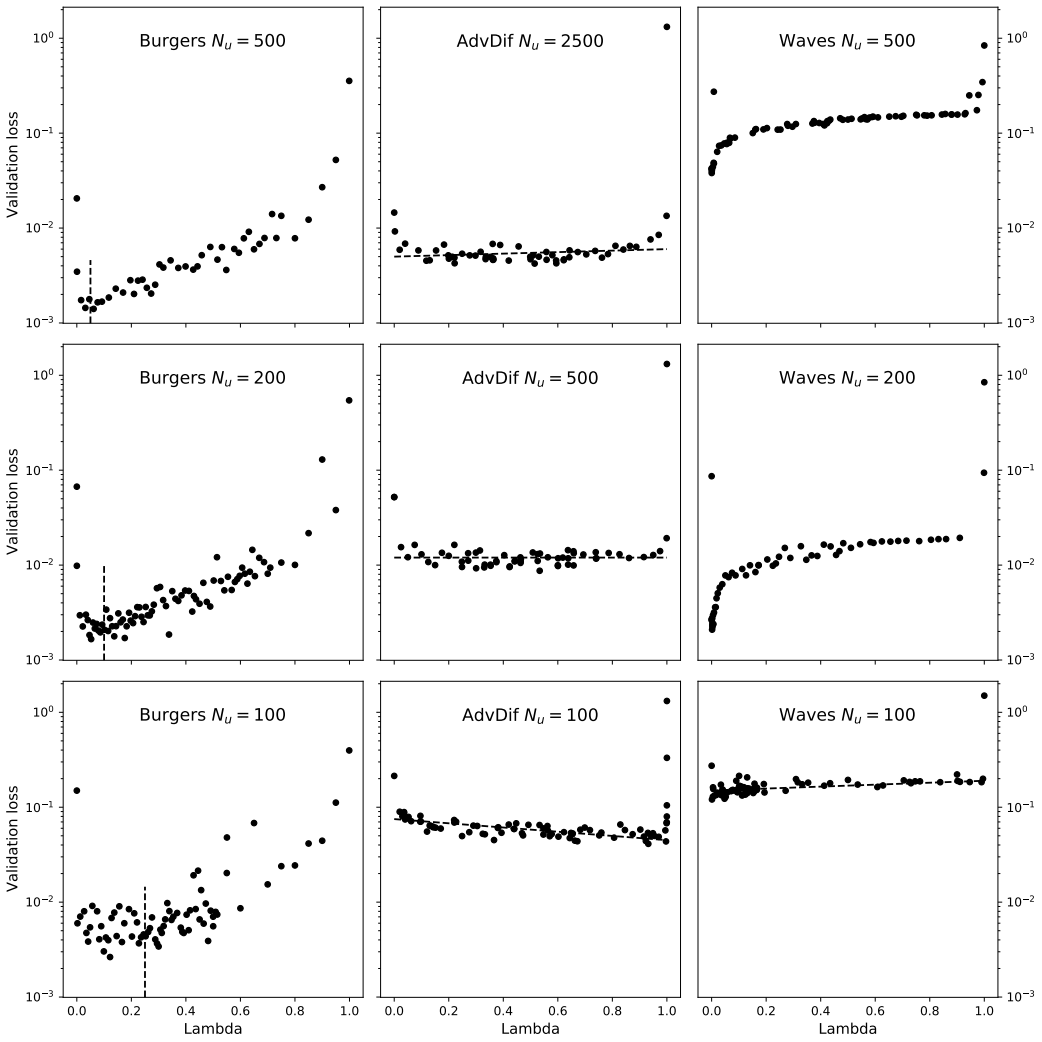


Figure 2: The relative error for different values of N_u and for different problems. Each dot represents the final value of the relative error after training. We observe how the choice of λ affects training results, and we observe a slight shift to higher λ values as N_u decreases. The dashed lines were added manually to aid in observing trends. For Burgers we note that the minimum shifts towards 1.0 as N_u decreases, and for advection-diffusion the slope becomes more negative as N_u decreases. The wave equation is more less evident, but it can be seen that for $N_u = 500$ and $N_u = 200$ the minimum lies around zero, while for $N_u = 100$ the trend has flattened and the results would be similar for $\lambda \in [0.1, 0.9]$ roughly.

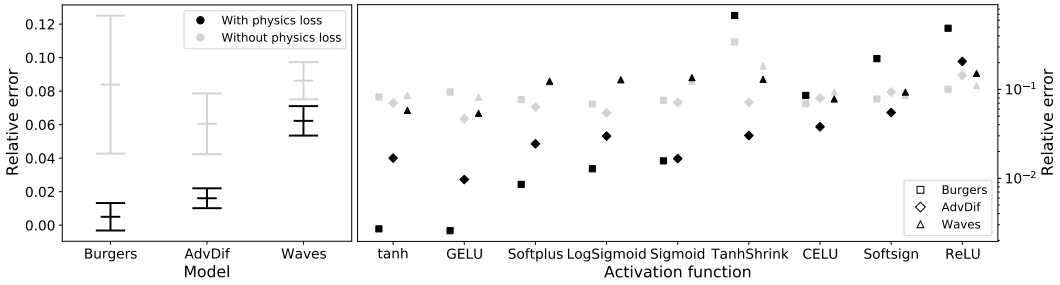
5 Results and discussion

See Table 1 for an overview of the used parameters for each of the models. The hyperparameters, such as the number and size of the hidden layers; the optimizer; and the number of epochs, were chosen either manually or by a grid search. All experiments were executed using Google Colab, a platform to run notebooks on fast graphics processing units (GPUs). The ability of GPUs to run calculations in parallel greatly enhances the speed of evaluating the neural network and calculating the gradients.[1]

[1]The data sets, source code, and results are available online at `https://github.com/Inria-Chile/assessing-pinns-ocean-modelling` and are released under the CeCILL license.

Table 1: Parameters for each of the models. Here epochs are the numbers of iterations over the training set, and N_u and N_f are the number of points used for the data loss and the physics loss respectively. N_u^{val} is the number of data points used to for the validation data loss.

Model	**NN setup**	**Optimizer**	**Epochs**	N_u	N_f	N_u^{val}
Burgers	8×20	Adam (lr=0.001)	25000	100, 200, 500	12800	25600
Advec.-Diffusion	3×40	Adam (lr=0.002)	20000	100, 500, 2500	5000	168100
Wave	5×50	Adam (lr=0.001)	10000	100, 200, 500	10000	21800



(a) Errors with/without physics loss.

(b) Relative error using different activation functions.

Figure 3: Summary of the experimental results. Fig. 3a shows results for 25 runs with and without physics loss for each of the models. For Fig. 3b, N_u is 200, 500, and 200 and λ is 0.1, 0.5, and 0.002 for the Burgers, advection-diffusion, and wave equations respectively.

To be able to compare the performance between models, the validation data loss is calculated over the entire data set and then normalized as

$$\text{Relative Error} = \frac{\sqrt{\text{MSE}(Y_{val}, \hat{Y}_{val})}}{\|Y_{val}\|_2}, \tag{15}$$

where Y_{val} are the labels of the validation data set, $\hat{Y}_{val}$ the output of the neural network, and MSE the mean squared error. In the case of advection-diffusion the labels Y_i have been normalized to have zero mean and unit standard deviation to bring down the scale of the temperature, which is in Kelvin, to a range around zero. This obviates the need for the division when calculating the relative error. Additionally, to reduce the jitter in the loss while training, we use the lowest validation data loss of the last 250 epochs to calculate the relative error.

Learning the optimal value of λ, the relative weighting of the physics loss with respect to the data loss is a metalearning objective in order to improve subsequent training of the same PINN. In Fig. 2 we show the results of evaluating different problems using different values for N_u to find an optimal value of λ. A sharp rise in the relative error can be observed at $\lambda = 0.0$ and $\lambda = 1.0$, which are equivalent to training using solely the data loss or physics loss terms respectively. We note that the optimal value for λ depends on the problem at hand and specifically on the scale of the data loss versus the physics loss. Normalizing the data set labels would in part solve the problem, but the scale of the physics loss remains hard to normalize. In our experiments however, we note that for typical values of λ the comparison of the data and physics loss remains roughly $0.1 < \ell_{\text{data}}/\ell_{\text{physics}} < 10$.

Per problem we observe that at decreasing values for N_u, the optimal value of λ shifts slightly towards 1.0. This trend suggests that as data become more sparse, the physics loss takes higher importance. As less data are available, the information input from the physics loss helps training significantly, while the reverse holds as more data are available. At the limit where N_u is sufficiently large to be able to train the network autonomously, the optimal value of λ tends to zero.

In cases where few data points are known (i.e. N_u is small), the choice of the sample subset can have a large influence on training results. In Fig. 3a the relative error is shown for 25 random data subset selections with and without the physics loss. The added physics loss term reduces the relative

error, averaged over all three models, by 65% and reduces its standard deviation by 56%. PINNs thus improve training results and reduce variability when the number of data points is low.

An analysis of how activation functions play a role in learning the derivations of PDEs, a comparison is made with nine activation functions in Fig. 3b. Here we compare the $\tanh$, GELU, Softplus, LogSigmoid, Sigmoid, TanhShrink, CELU, Softsign, ReLU activation function for their performance. It can be observed that especially the $\tanh$ and GELU activation functions and to a lesser extend Softplus work well. The ReLU function produces poor results due to its inability to learn second or higher derivatives. Due to the fact that a DNN using the ReLU activation function produces a piece-wise linear function [12], its second derivative is zero except at the points where the linear segments meet, where it will be a Dirac function at $x = 0$, inhibiting learning second derivatives or higher. The Softsign and CELU functions have similar problems given their discontinuity of the second derivative at $x = 0$.

6 Final remarks

Information from the PDE serves in cases where data availability is limited to the extent that it would either be unable to train the network satisfactorily using solely the data, or where results would vary widely depending on the selection of the data points. The addition of the physics loss term both improves training results and improves the stability of the training, even more so when the weight in the loss function is optimal.

Finding the optimal weight of the loss function is a complex problem and this is, to the best of our knowledge, the first study to understand the behavior of it with respect to different hyperparameters and PDEs. In the future, we expect to find the optimal weights using only the data available. In addition, it is also the first attempt at the study of PINNs for ocean modelling equations with very simplified geometries. We expect to extend our results to more complex and realistic geometries in order to use our results in real applications.

As λ has a dual role: the prior confidence in the training data vs. the PDE and the scaling of these two terms, future work could understand each role of this parameter, which could be a possible explanation of why the optimal weight in some of our experiments are so different between different models. In addition, as we observed for very small data sets, it would be interesting to find what "good" sample subsets of the problem for training are more stable under optimization. This would be the first step to estimate the minimal amount of data required to train the neural networks in a robust way.

As the equations of interest are inherently wave-like, we propose to explore the benefits of Fourier transformations for learning the equations more effectively, as was done by [11]. This would enable the network to learn patterns of the data in the frequency space. Additionally, to generalize the model better for unseen data, we propose the use of dropout by [9] to randomly turn off a percentage of the neurons. This forces the network to learn redundancies in case certain neurons are turned off, increasing the stability of the network, and the likelihood to generalize better for unseen data.

Future work on this topic is needed as it is required to develop better models for understanding the ocean and integrate those models with others that explain the interaction between organisms and the environment. This is essential because of the constantly changing nature of the ocean.

We believe that PINNs have the potential of providing such tools, and therefore, serve as a building block of a more comprehensive solution that addresses the ultimate issue of understanding and mitigating the climate change crisis. However, this potential also comes with important risks as using an inadequate, biased, or non-transparent tool could lead instead to a deterioration. In this direction, as we already mentioned in the paper, we are interested in combining PINNs with explainable AI and powerful visualization tools. Similarly, extended validation is necessary by domain experts.

Acknowledgments and Disclosure of Funding

This work is funded by project CORFO 10CEII-9157 Inria Chile and Inria Challenge project OcéanIA (desc. num 14500).

References

[1] Alnæs, M. S., Blechta, J., Hake, J., Johansson, A., Kehlet, B., Logg, A., Richardson, C., Ring, J., Rognes, M. E., and Wells, G. N. (2015). The FEniCS project version 1.5. *Archive of Numerical Software*, 3(100).

[2] Basdevant, C., Deville, M., Haldenwang, P., Lacroix, J., Ouazzani, J., Peyret, R., Orlandi, P., and Patera, A. (1986). Spectral and finite difference solutions of the burgers equation. *Computers & fluids*, 14(1):23–41.

[3] Burgers, J. (1948). A mathematical model illustrating the theory of turbulence. In Von Mises, R. and Von Kármán, T., editors, *Advances in Applied Mechanics*, volume 1, pages 171–199. Elsevier.

[4] Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. (2018). Neural ordinary differential equations. In *Advances in neural information processing systems*, pages 6571–6583.

[5] de Wolff, T., Carrillo Lincopi, H., Martí, L., and Sanchez-Pi, N. (2021). Assessing physics informed neural networks in ocean modelling and climate change applications. In Sanchez-Pi, N. and Martí, L., editors, *AI: Modeling Oceans and Climate Change Workshop at ICLR 2021*.

[6] Dupont, E., Doucet, A., and Teh, Y. W. (2019). Augmented neural ODEs. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 3140–3150. Curran Associates, Inc.

[7] Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W. and Titterington, M., editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

[8] He, Q. and Tartakovsky, A. M. (2020). Physics-informed neural network method for forward and backward advection-dispersion equations. *arXiv preprint arXiv:2012.11658*.

[9] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580.

[10] Kingma, D. P. and Ba, J. (2017). ADAM: A method for stochastic optimization.

[11] Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. (2020). Fourier neural operator for parametric partial differential equations.

[12] Liu, B. and Liang, Y. (2021). Optimal function approximation with ReLU neural networks. *Neurocomputing*, 435:216–227.

[13] Lütjens, B., Crawford, C. H., Veillette, M., and Newman, D. (2021). PCE-PINNs: Physics-informed neural networks for uncertainty propagation in ocean modeling. In Sanchez-Pi, N. and Martí, L., editors, *AI: Modeling Oceans and Climate Change Workshop at ICLR 2021*.

[14] McLean, D. (2012). *Understanding aerodynamics: arguing from the real physics*. John Wiley & Sons.

[15] Orlandi, P. (2000). *The Burgers equation*, pages 40–50. Springer Netherlands, Dordrecht.

[16] Raissi, M., Perdikaris, P., and Karniadakis, G. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707.

[17] Rao, K. S. (2010). *Introduction to partial differential equations*. PHI Learning Pvt. Ltd.

[18] Sánchez-Pi, N., Martí, L., Abreu, A., Bernard, O., de Vargas, C., Eveillard, D., Maass, A., Marquet, P. A., Sainte-Marie, J., Salomon, J., Schoenauer, M., and Sebag, M. (2020). Artificial intelligence, machine learning and modeling for understanding the oceans and climate change. In Dao, D., Sherwin, E., Donti, P., Kaack, L., Kuntz, L., Yusuf, Y., Rolnick, D., Nakalembe, C., Monteleoni, C., and Bengio, Y., editors, *Tackling Climate Change with Machine Learning workshop at NeurIPS 2020*.

[19] Sirignano, J. and Spiliopoulos, K. (2018). DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:339–1364.

[20] Stoker, J. J. (2011). *Water waves: The mathematical theory with applications*, volume 36. John Wiley & Sons.

[21] Temam, R. (2001). *Navier-Stokes equations: Theory and numerical analysis*, volume 343. American Mathematical Soc.

[22] Union, J. I. G. (2013). Advection diffusion equation models in near-surface geophysical and environmental sciences. *J. Ind. Geophys. Union (April 2013)*, 17(2):117–127.

[23] van der Meer, R., Oosterlee, C., and Borovykh, A. (2020). Optimally weighted loss functions for solving PDEs with neural networks. *arXiv preprint arXiv:2002.06269*.

[24] van Herten, R. L., Chiribiri, A., Breeuwer, M., Veta, M., and Scannell, C. M. (2020). Physics-informed neural networks for myocardial perfusion mri quantification. *arXiv preprint arXiv:2011.12844*.

[25] Vetra-Carvalho, S., van Leeuwen, P. J., Nerger, L., Barth, A., Altaf, M. U., Brasseur, P., Kirchgessner, P., and Beckers, J.-M. (2018). State-of-the-art stochastic data assimilation methods for high-dimensional non-Gaussian problems. *Tellus A: Dynamic Meteorology and Oceanography*, 70(1):1–43.

[26] Xiang, Z., Peng, W., Zheng, X., Zhao, X., and Yao, W. (2021). Self-adaptive loss balanced physics-informed neural networks for the incompressible navier-stokes equations. *arXiv preprint arXiv:2104.06217*.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See in particular Section 6
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Section 5.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 5.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section 5.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 5.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [Yes] See Section 5.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See Section 5.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]