NOVEMBER 06 2025

Stable adaptive training for physics-informed neural networks in acoustic wave propagation

Márcio Marques ; Leonardo Mendonça ; Arthur Bizzi ; Leonardo Moreira ; Christian Oliveira ; Deborah Oliveira ; Lucas Fernandez ; Vitor Balestro ; João Pereira ; Daniel Yukimura ; Tiago Novello ; Pavel Petrov ; Lucas Nissenbaum

JASA Express Lett. 5, 112401 (2025)
https://doi.org/10.1121/10.0039767

JASA EXPRESS LETTERS ARTICLE asa.scitation.org/journal/jel

Stable adaptive training for physics-informed neural networks in acoustic wave propagation

Márcio Marques,[1] Leonardo Mendonça,[1] Arthur Bizzi,[2] Leonardo Moreira,[3] Christian Oliveira,[1] Deborah Oliveira,[4] Lucas Fernandez,[5] Vitor Balestro,[1,6] João Pereira,[7] Daniel Yukimura,[1] Tiago Novello,[1] Pavel Petrov,[1,a)] and Lucas Nissenbaum[1]

[1]*Instituto de Matemática Pura e Aplicada, Rio de Janeiro, Brazil*
[2]*École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland*
[3]*Universidade do Estado do Rio de Janeiro, Rio de Janeiro, Brazil*
[4]*McGill University, Montreal, Canada*
[5]*Laboratório Nacional de Computação Científica, Petrópolis, Brazil*
[6]*Universidade Federal Fluminense, Niterói, Brazil*
[7]*University of Georgia, Athens, Georgia 30602, USA*

Abstract: Physics-informed neural networks (PINNs) have emerged as a promising tool for simulating various phenomena. However, their application in underwater acoustics remains challenging, primarily due to the need to sample large computational domains and to convergence to trivial solutions. This study presents a strategy to address these issues by combining adaptive domain sampling with absorbing boundary conditions. The adaptive sampler dynamically focuses computational effort on regions where the acoustic energy is localized, while the absorbing boundaries perform training stabilization. Numerical experiments show that our method improves the stability and convergence of PINN training, leading to more accurate and reliable wave propagation simulations.

[Editor: Nickolas Vlahopoulos] https://doi.org/10.1121/10.0039767

Received: 26 August 2025 **Accepted:** 17 October 2025 **Published Online:** 6 November 2025

1. Introduction

Modeling sound propagation is a fundamental challenge in underwater acoustics. Over the past decades, several important classes of methods have been developed and extensively used for various practical purposes (Jensen *et al.*, 2011; see also numerous references therein). Despite the great progress in accuracy and efficiency, there is a persistent interest in the development of new approaches to acoustic field modeling aiming at reaching new levels of performance and precision that could meet new demands emerging in applications.

Physics-informed neural networks (PINNs) are currently attracting increasing interest due to their flexibility in handling partial differential equations (PDEs), as well as in incorporating possibly sparse or noisy data both in direct and inverse problems (Raissi *et al.*, 2019; Rasht-Behesht *et al.*, 2022; Ito *et al.*, 2025; Liu and Gerstoft, 2024). PINNs have already been successfully applied to the solution of wave propagation problems in heterogeneous media, both in the time (Moseley *et al.*, 2023; Sethi *et al.*, 2023; Zhang *et al.*, 2023) and frequency domains (Song *et al.*, 2021; Alkhalifah and Huang, 2024; Li *et al.*, 2025), including applications to underwater acoustics (Yoon *et al.*, 2024; Xi *et al.*, 2024). There also exist neural network-based solutions for sound propagation problems where loss function is not related to a PDE residual (Varon *et al.*, 2023; Mallik *et al.*, 2022).

For the acoustic wave equation, many challenges arise in time-domain formulations, particularly for complex media. The spectral bias problem makes it hard to learn high-frequency components when using simple architectures (Wang *et al.*, 2021b; Rahaman *et al.*, 2019) [note that the spectral bias is a common phenomenon for neural networks that even manifests in frequency-domain approaches (Yoon *et al.*, 2024)]. Moreover, the multilayer perceptron (MLP)-based architecture's lack of causality combined with the strong spatiotemporal localization of wavefields makes the training unbalanced and often causes it to converge to a trivial solution or fail to propagate the initial conditions (Penwarden *et al.*, 2023). The combination of techniques such as Fourier features (Tancik *et al.*, 2020) and absorbing boundary conditions (BCs) (Higdon, 1986) is now the state-of-the-art for propagating wavefields in complex media, such as the Marmousi model. However, this approach, proposed by Ding *et al.* (2025), only works with time-domain decomposition strategies and requires a very large number of collocation points and optimization steps to achieve reasonable accuracy, demanding more memory and training time (TT).

a)Corresponding author: pavel.petrov@impa.br

This work presents a streamlined framework that combines Fourier features and absorbing BCs with an adaptive sampling strategy and adaptive weight balancing guided by the neural tangent kernel (NTK) method (Jacot *et al.*, 2018). Numerical experiments show that our approach yields highly accurate solutions for low-frequency sound wave propagation in a sea with complex bottom structure, while achieving comparable or better accuracy with substantially fewer collocation points and optimization steps than previous state-of-the-art methods.

The paper is organized as follows. In Sec. 2, we define the main equations and BCs for underwater sound propagation. In Sec. 3, the neural network architecture is discussed, while sampling strategies for its training and for weight balancing are introduced in Sec. 4. The performance of the proposed method is illustrated in Sec. 5 through a numerical example.

2. Wave equation and absorbing BCs

A non-stationary acoustic field $p(t,x,z)$ generated by a point source located at (x_0,z_0) is usually described by the wave equation of the form

$$\frac{1}{c^2}\frac{\partial^2 p}{\partial t^2}-\frac{\partial^2 p}{\partial x^2}-\frac{\partial^2 p}{\partial z^2}=S(t,x,z), \tag{1}$$

where $c(x,z)$ is the sound velocity distribution, and $S(t,x,z)$ is the source term. A common choice is $S=s(t)\,\delta(x-x_0)\,\delta(z-z_0)$, where $s(t)$ represents the waveform of a signal transmitted by a point source in plane geometry (Jensen *et al.*, 2011). Equation (1) is typically complemented by suitable BCs, such as the pressure-release condition $p=0$ at the sea surface.

In this study, we intentionally avoid specifying the exact BCs considered. Instead, we solve the problem in a rectangular domain $\Omega=\{(x,z)|\,x\in[-L,L],z\in[0,H]\}$, assuming a propagation environment beyond the boundaries of this domain. In practice, this is usually the case for both vertical boundaries $x=\pm L$ and at least one horizontal boundary $z=H$. Under the assumptions that $c(x,z)=c(L,z)$ for $x>L$, $c(x,z)=c(-L,z)$ for $x<-L$, and $c(x,z)$ is constant for $z\geq H$ and for $z\leq 0$, we can impose so-called absorbing BCs (also called non-reflecting/artificial/radiation BCs) at these boundaries. In particular, we employ high-order Higdon's absorbing BCs (Higdon, 1986), which can be written as

$$\prod_{j=1}^{N}\left(\frac{\partial}{\partial t}+C_j\frac{\partial}{\partial \mathbf{n}}\right)p=0, \tag{2}$$

where N is the order of the condition, and $\mathbf{n}$ is the outward-normal vector at the given point of the boundary. Hereafter, we consider an initial-boundary value problem for Eq. (1) with absorbing BCs [Eq. (2)] and zero Cauchy data

$$p|_{t=0}=0 \quad \text{and} \quad \partial p/\partial t|_{t=0}=0. \tag{3}$$

3. PINNs

The PINNs framework, proposed by Raissi *et al.* (2019), can integrate real data and physical laws by training a neural network to fit the data, while minimizing a physics-informed loss that includes PDE residuals, initial conditions, and BCs. This flexibility allows PINNs to be employed either as data-driven models or as physics-based solvers without data assimilation. We will focus on the latter in this paper, as it serves as a strong starting point when data is available.

The architecture of these neural networks and the loss function has great influence in the training process (Wang *et al.*, 2021b). In this section, we will discuss some architectures and possible loss functions allowing us to efficiently solve wave propagation problems in complex media.

3.1 Fourier feature networks

In this context, the solution of a PDE is approximated by a neural network p_θ, whose parameters are optimized so that the neural network closely matches the solution. In this work, acoustic pressure $p(t,x,z)$ is represented by a MLP enhanced by Fourier feature mapping (Tancik *et al.*, 2020). Specifically, a Fourier feature network of depth $L\in\mathbb{N}$ is defined by

$$p_\theta(t,x,z)=t^2\big(f_{L+1}\circ f_L\circ\cdots\circ f_1\circ\gamma(t,x,z)\big), \tag{4}$$

where $f_\ell(y)=\phi(W^{(\ell)}y+b^{(\ell)})$, for $\ell=1,\ldots,L$, denote the hidden layers, and $f_{L+1}=W^{(L+1)}y+b^{(L+1)}$ is the output layer. Each layer is parametrized by trainable weights $W^{(\ell)}\in\mathbb{R}^{d_\ell\times d_{\ell-1}}$ and biases $b^{(\ell)}\in\mathbb{R}^{d_\ell}$, with ϕ denoting a non-linear activation function. The Fourier feature encoder is defined by a vector of frequencies $\gamma\in\mathbb{R}^{2k}$ where $\mathbf{v}=(t,x,z)^T$,

$$\gamma(\mathbf{v})=\begin{pmatrix}\cos(2\pi F\mathbf{v})\\ \sin(2\pi F\mathbf{v})\end{pmatrix}, \tag{5}$$

where Fourier basis frequencies $F\in\mathbb{R}^{k\times 3}$ are sampled from a Gaussian distribution, and sine and cosine are computed coordinate-wise. Theses architectures are based on the classic MLPs, which are universal function approximators

(Hornik *et al.*, 1989) and are commonly used to represent PDE solutions. These are unfortunately also known to suffer from *spectral bias*, where the network often fails to capture high-frequency components (Rahaman *et al.*, 2019). The use of Fourier feature mapping helps mitigate this issue (Tancik *et al.*, 2020; Wang *et al.*, 2022; Wang *et al.*, 2021b).

We note that we have included a product by a t^2 term to the output layer of the neural network in Eq. (4). This term enforces a null wavefield and derivative at $t = 0$ (Ding *et al.*, 2025) and leads to a network that by construction satisfies the initial conditions in Eq. (3). An alternate way this is frequently done is to impose these initial conditions as a soft constraint, by adding them as boundary constraints at $t = 0$ (Raissi *et al.*, 2019; Hu *et al.*, 2024). This approach, however, may lead to unstable training due to unbalanced gradients and is not guaranteed to fit the initial conditions exactly (Wang *et al.*, 2021a).

3.2 Loss function

The loss function plays a central role in the training process. It is commonly defined as a weighted sum of constraints accounting for the PDE, the initial and BCs, and, when available, observational data. The constraints for the wave equation and the absorbing BC at each collocation point $\mathbf{v}_i = (t_i, x_i, z_i)$ can be expressed in terms of residuals

$$\mathcal{R}_{\text{PDE}}(\theta, \mathbf{v}_i) = \left(\frac{1}{c^2}\frac{\partial^2}{\partial t^2} - \frac{\partial^2}{\partial x^2} - \frac{\partial^2}{\partial z^2}\right)p(\mathbf{v}_i) - S(\mathbf{v}_i), \quad \mathcal{R}_{\text{ABC}}(\theta, \mathbf{v}_i) = \prod_{j=1}^{N}\left(\frac{\partial}{\partial t} + C_j\frac{\partial}{\partial \mathbf{n}}\right)p(\mathbf{v}_i), \tag{6}$$

where $C_j = c(\mathbf{v}_i)$ for all j. The corresponding loss terms to these residuals, evaluated over the sets of collocation points $\mathcal{X}_{\text{PDE}}$ and $\mathcal{X}_{\text{ABC}}$ with N_{PDE} and N_{ABC} points, respectively, are defined as the empirical means

$$\mathcal{L}_{\text{PDE}}(\theta, \mathcal{X}_{\text{PDE}}) = \frac{1}{N_{\text{PDE}}}\sum_{\mathcal{X}_{\text{PDE}}} |\mathcal{R}_{\text{PDE}}(\theta, \mathbf{v}_i)|^2, \quad \mathcal{L}_{\text{ABC}}(\theta, \mathcal{X}_{\text{ABC}}) = \frac{1}{N_{\text{ABC}}}\sum_{\mathcal{X}_{\text{ABC}}} |\mathcal{R}_{\text{ABC}}(\theta, \mathbf{v}_i)|^2. \tag{7}$$

Additional loss terms for the initial condition are not required because it is automatically enforced [see Eq. (4)].

4. Sampling strategies and balancing

In Sec. 3, we discussed how a PINN is trained to minimize a loss function over a set of sample points distributed over its domain. There are a few ways this set can be constructed. A relatively straightforward and canonical approach is to randomly sample N_U points at each training step according to a uniform distribution over the domain (Raissi *et al.*, 2019). Wave propagation problems in time domain, however, often have localized solutions, meaning the majority of these points will be in regions where the solution is simple, or even zero, as can be seen in Fig. 1. In an attempt to improve training, we implement residue-based adaptive distribution sampling similar to Wu *et al.* (2023), in order to dynamically update the sampling distribution during training to prioritize the domain regions where the loss function has larger values.

4.1 Adaptive sampling

Our sampling strategy works as follows: at the end of each training epoch (corresponding to 100 steps of the gradient descent algorithm), we split the domain into a regular grid of n^3_{grid} rectangular boxes and evaluate the wave equation residue $\mathcal{R}_{\text{PDE}}$ at the points $\mathbf{v}_{ijk} = (t_i, x_j, z_k)$ given by the center of each box B_{ijk}. The probability to sample from a box B_{ijk} is defined to be proportional to the squared-loss at the center or

$$P_{\text{adp}}(\mathbf{v} \in B_{ijk}) \propto \mathcal{R}_{\text{PDE}}(\theta, \mathbf{v}_{ijk})^2. \tag{8}$$

A point is then selected uniformly from the randomly sampled box.

We then uniformly sample a set $\mathcal{X}_U$ of N_U points and a set $\mathcal{X}_A$ of N_A adaptive points according to P_{adp} at each training step, calculating the loss function as a combination of both, plus the BCs

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{PDE}}(\theta, \mathcal{X}_U) + \beta\mathcal{L}_{\text{PDE}}(\theta, \mathcal{X}_A) + \mathcal{L}_{ABC}(\theta, \mathcal{X}_{\text{ABC}}), \tag{9}$$

where the weight parameter β can be set as a fixed hyperparameter or adjusted dynamically as described in Sec. 4.2.

Figure 1 shows collocation points sampled uniformly and adaptively during training of the PINN for the problem described in Sec. 5. Uniformly sampled points are spread throughout the domain; therefore, a significant fraction of them are in regions where the wavefield is null, as seen in Fig. 1(B). Adaptive points, however, concentrate near the region with the largest velocity gradients, where the network must learn to reflect the wave, thus allowing the network to focus on the more complex parts of the solution.

A similar form of adaptive sampling was introduced by Wu *et al.* (2023); however, our approach can be regarded as a stratified and more efficient version. Another important difference is that we have a weight parameter β that allows the weight of the adaptive points to be set independently of the number of points, allowing a dynamic adjustment during training that leads to more stable results. A similar approach (albeit without the NTK) called retain-resample-release was proposed in Daw *et al.* (2022).

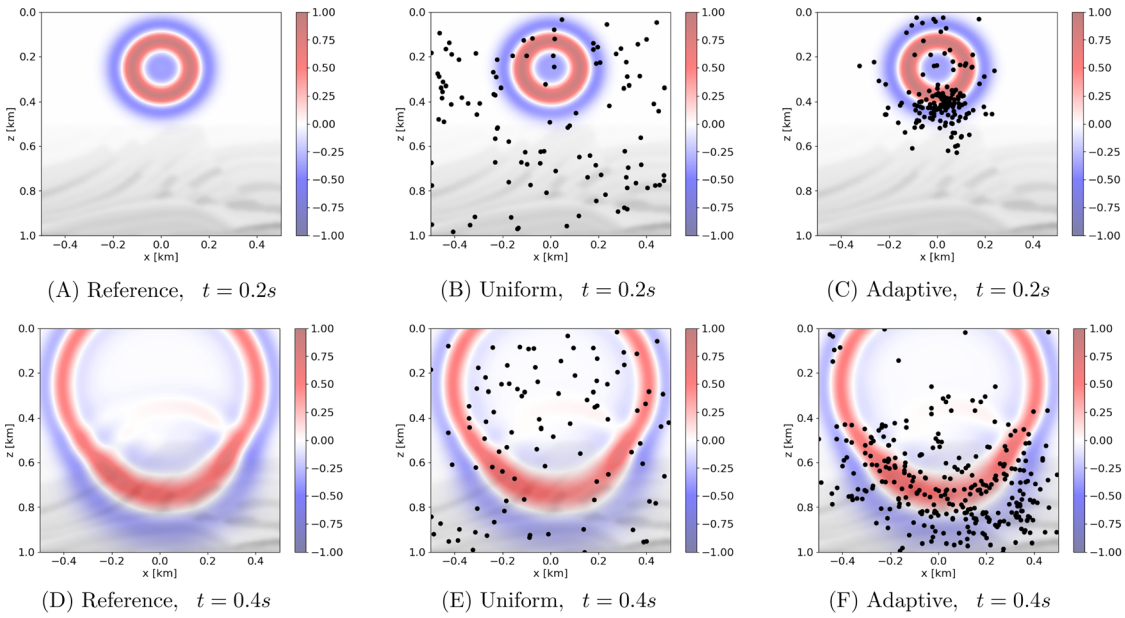


Fig. 1. Representation of the points picked by the adaptive and uniform samplers at $t = 0.2$ s and $t = 0.4$ s, superposed with the PINN solution in Pa, for the model trained on the Marmousi domain after 100 training epochs. (A) and (D) show the reference solution obtained through the finite difference method, which mostly occupies a fraction of the domain. In the background, the Marmousi velocity field is shown in gray-scale. As can be seen in (C) and (F), adaptive sampling uses more points where the function is non-zero.

4.2 NTK and weight balancing

Minimizing all loss terms in Eq. (9) is a multi-objective optimization problem, in which each and every loss term should be simultaneously minimized. PINNs can run into difficulties balancing these terms, which may lead to a training dynamic where only one component of the loss actually reaches an optimum (Wang *et al.*, 2022; Wang *et al.*, 2021b). This is exacerbated in problems where the different losses have different convergence speeds, expressed in terms of the eigenvalues of the NTK matrix (Jacot *et al.*, 2018). The analysis of the training dynamics using the NTK method suggests a weight balancing technique as proposed by (Wang *et al.*, 2022), which is now widely used in modern PINNs frameworks (Wu *et al.*, 2024; Zhongkai *et al.*, 2024; Wang *et al.*, 2024).

Wang *et al.* (2022) derives the NTK in the context of PINNs containing multiple loss terms, given by differential forms $\mathcal{R}_1$, $\mathcal{R}_2$, sampled on a set of points $\mathcal{X}_1$, $\mathcal{X}_2$. In our application, $\mathcal{R}_1$ and $\mathcal{R}_2$ are the PDE residual $\mathcal{R}_{\text{PDE}}$ on the uniform and adaptive points, respectively. The residuals evolve with the training step τ as a dynamical system as follows:

$$\frac{d}{d\tau}\begin{bmatrix}\mathcal{R}_1(\theta(\tau),\mathcal{X}_1)\\ \mathcal{R}_2(\theta(\tau),\mathcal{X}_2)\end{bmatrix} = -\mathbf{K}(\tau)\cdot\begin{bmatrix}\mathcal{R}_1(\theta(\tau),\mathcal{X}_1)\\ \mathcal{R}_2(\theta(\tau),\mathcal{X}_2)\end{bmatrix}, \tag{10a}$$

$$\mathbf{K}(\tau) := \begin{bmatrix}\mathbf{K}_{11}(\tau) & \mathbf{K}_{12}(\tau)\\ \mathbf{K}_{21}(\tau) & \mathbf{K}_{22}(\tau)\end{bmatrix}, \text{ where } \quad (\mathbf{K}_{kl})_{ij} := \frac{2}{N_l}\left\langle \frac{d\mathcal{R}_k(\theta,\mathcal{X}_k^i)}{d\theta}, \frac{d\mathcal{R}_l(\theta,\mathcal{X}_l^j)}{d\theta}\right\rangle\Bigg|_{\theta=\theta(\tau)}, \tag{10b}$$

where $\mathcal{X}_k^i$ is the ith element of the set $\mathcal{X}_k$, and $\langle\cdot,\cdot\rangle$ is the inner product. In practical terms, we expect training to occur along $\mathbf{K}$'s eigenvectors proportionally to the magnitude of its eigenvalues, i.e., the residuals will decay primarily in the direction of the largest eigenvalues.

In the acoustic problems studied, however, the NTK eigenvalues for the adaptive sampler's loss are more than 3 orders of magnitude larger than those of the uniformly sampled points, as observed in preliminary studies. This means that learning on these points tends to be much more aggressive, thus effectively dominating the training and preventing the network from properly learning the wave equation on the uniform points and the BCs, which can make learning unstable.

With this in mind, one can employ a dynamic weighting algorithm based on Wang *et al.* (2022) to ensure that both uniform and adaptive sampled points are being learned with similar rates at each step of the training process. During training, one updates the value of β according to the trace of the components of $\mathbf{K}$ related to the uniform and adaptive loss $\mathbf{K_{uni}} := \mathbf{K}_{11}$ and $\mathbf{K_{adp}} := \mathbf{K}_{22}$, respectively,

$$\beta^{(0)} = 10^{-3}, \quad \beta^{(i+1)} = \frac{tr(\mathbf{K_{uni}}^{(i)})}{tr(\mathbf{K_{adp}}^{(i)})}. \tag{11}$$

By setting the weights as a function of the trace, which equals the sum of the eigenvalues, we ensure that, even if the eigenvalues of $\mathbf{K_{adp}}$ are significantly larger than those of $\mathbf{K_{uni}}$, which would imply a greater effective learning rate for this loss, the weight balancing strategy described here will scale down β so as to keep the evolution of different losses comparable and prevent over-prioritizing one loss term over the other. The weight definition described here is made more computationally efficient by the fact that only the diagonal of $\mathbf{K}$ needs to be computed for updating the weights in Eq. (11). We also note that our experiments have indicated that including the weight of absorbing boundary conditions (ABCs) into NTK does not help to further stabilize training.

5. Experiments and results

5.1 Experiment setup

We consider the case of sound propagation in the ocean with water depth $h = 500\,\text{m}$ and constant sound velocity 1500 m/s within the water column and fluid bottom with strongly inhomogeneous sound velocity distribution according to the Marmousi model (Martin *et al.*, 2006) [as shown in Fig. 2(A)]. We apply a Gaussian filter with $\sigma = 10\,\text{m}$ to the lower half of the domain, which has little effect on the solution, given that the wavelengths are on the order of magnitude of 200 m, but helps with training stabilization. The physical domain is considered to extend infinitely beyond the four borders of the area in which the PINN is to be trained as defined in Sec. 2.

Due to the modus operandi of PINNs, they naturally tend to produce a solution assuming that propagation medium extends infinitely beyond the boundaries of the computational domain. In some recent research (Rasht-Behesht *et al.*, 2022), this property is used to train PINNs without explicitly specifying BCs. As claimed by Ding *et al.* (2025) and shown in the following experiments, the use of BCs [Eq. (2)] offers a much better alternative both in the accuracy and the stability of training, especially in the case of adaptive sampling. In our experiments we used second-order Higdon's conditions.

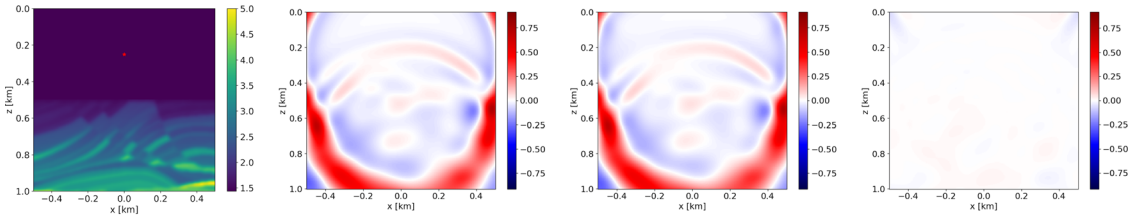
In each experiment, we use the input term in Eq. (1) defined as the product $S(t,x,z) = G(x,z)s(t)$ of a Gaussian $G(x,z) = \mathrm{e}^{-((x-x_0)^2+(z-z_0)^2)/(2\alpha^2)}$, approximating the point source, and a Ricker wavelet representing its waveform

$$s(t) = M_0\left(1 - 2\pi^2 f_0^2\left(t - \frac{1}{f_0}\right)^2\right)\mathrm{e}^{\pi^2 f_0^2 (t-1/f_0)^2}. \tag{12}$$

The source position is $(x_0, z_0) = (0\,\text{m}, 250\,\text{m})$, and its width, magnitude, and central frequency are $\alpha = 25\,\text{m}$, $M_0 = 10^{-2}\,\text{Pa}\cdot\text{m}^{-2}$, $f_0 = 10\,\text{Hz}$, respectively. We train the PINN and evaluate the solution on the time interval from $t = 0$ to $t = 1\,\text{s}$.

Using the PyTorch open-source library, we trained a neural network with five hidden layers and 150 neurons per layer and activation function given by the *Swish* function (Ramachandran *et al.*, 2017) with the input Fourier features layer having 180 frequencies randomly distributed with $\sigma = 1.0$. The starting learning rate was set to $lr = 3 \times 10^{-3}$, and an exponential decay schedule of 1% every 100 steps was added. The PDE loss was sampled on 10 000 collocation points. When using the adaptive sampler, they were split into 9000 points uniformly sampled and 1000 points sampled adaptively, while, for the absorbing BCs, 1000 points were sampled in each boundary segment. Each network was trained for 300 epochs of 100 Adam steps each. Experiments were repeated ten times with different random seeds, and the mean and standard deviation of the metrics are reported here in order to investigate the effect of stochasticity in initialization and sampling.

In the adaptive experiments, the NTK dynamic weighting strategy was used to balance the weight β of the adaptively sampled points relative to the uniformly sampled ones. For computational performance, the NTK was calculated on a subsample of $N_{NTK} = 2000$ collocation points, and at every ten training epochs, after a warm-up period of 30 epochs



(A) Velocity field (B) Reference (C) PINN solution (D) Residual

Fig. 2. (A) Sound velocity distribution (in m/s). Source position is marked by red star. (B) Reference solution (acoustic pressure in Pa) at $t = 0.5\,\text{s}$. (C) Wavefield predicted by our model at $t = 0.5\,\text{s}$. (D) Error between the predicted pressure field and the reference.

Table 1. Relative L1 and L2 loss and TT for the wave propagation in the Marmousi domain. The best value obtained for each metric is marked in bold.

	No BC			Higdon order-2 BC		
Sampler	$L1RE \downarrow$	$L2RE \downarrow$	TT (s)	$L1RE \downarrow$	$L2RE \downarrow$	TT (s)
Uniform	1.032 ± 1.532	0.589 ± 0.787	$\mathbf{578.3 \pm 7.0}$	0.516 ± 0.405	0.39 ± 0.367	1271.6 ± 21.8
Adaptive	0.2 ± 0.092	0.119 ± 0.039	1189.3 ± 6.0	$\mathbf{0.086 \pm 0.024}$	$\mathbf{0.05 \pm 0.012}$	1954.8 ± 19.3

during which the weight β was kept fixed. As the absorbing BCs are being used as a regularization, rather than a hard constraint, we keep their weight fixed and adjust only the value of β.

Experiments were run on a computer with a GeForce RTX 4090 GPU, i9-13900K CPU, and 128 GB RAM.

5.2 Results and analysis

The solution obtained is evaluated on a regular grid of points in the solution domain by means of the $L1$ and $L2$ relative errors (Zhongkai *et al.*, 2024), denoted as $L1RE$ and $L2RE$, respectively, and defined as

$$L1RE = \frac{\sum_i |p_\theta(\mathbf{v}_i) - p_{\text{ref}}(\mathbf{v}_i)|}{\sum_i |p_{\text{ref}}(\mathbf{v}_i)|} \quad \text{and} \quad L2RE = \sqrt{\frac{\sum_i (p_\theta(\mathbf{v}_i) - p_{\text{ref}}(\mathbf{v}_i))^2}{\sum_i p_{\text{ref}}(\mathbf{v}_i)^2}}. \tag{13}$$

The reference solution p_{ref} is obtained using an explicit finite difference solver with 8th-order central difference schemes in space and a 2nd-order central difference scheme in time. The spatial and temporal step sizes are 5×10^{-3} km and 5×10^{-4} s, respectively. To simulate an infinite domain, the computational domain is extended sufficiently so that any reflections do not return to the original region of interest.

The $L1RE$ and $L2RE$ error metrics, shown in Table 1, show a significantly greater accuracy of the model trained with Higdon BCs and adaptive sampling when compared to all other configurations. Adaptive sampling also makes the training outcome more stable, as evidenced by the smaller standard deviation in $L1$ and $L2$ error. Although the use of adaptive sampling and Higdon BCs requires a longer TT than the baseline, it offers strong results in terms of accuracy and training stability.

A state-of-the-art approach by Ding *et al.* (2025) that was used to handle problems of similar complexity employs techniques such as Fourier features (Tancik *et al.*, 2020) and absorbing BCs (Higdon, 1986) to propagate through complex media. However, it demands time-domain decomposition strategies and requires 80 000 collocation points and 45 000 optimization steps to achieve reasonable accuracy (on the order of 10%). To emphasize the advantages of our stable adaptive framework, we show that it achieves similar results with only 10 000 collocation points and 30 000 optimization steps, demanding much less memory and TT.

Figure 2 shows that the model trained with adaptive sampling and ABCs captures the qualitative behavior of the solution with stunning fidelity, closely resembling the reference numerical solution (see also animation in Mm. 1). As can be seen in Fig. 2(D), the most pronounced errors are associated with the treatment of corners in Higdon's condition. Similar results have been obtained in experiments with other environments.

Mm. 1. Animation of the PINN solution for the smooth Marmousi domain. File of type "gif"

6. Conclusion and future work

In this study, we propose a new training strategy for PINNs that approximates the solution of non-stationary wave equations, specifically designed to handle the problems of acoustic wave propagation in complex media. The proposed PINN structure is based on existing state-of-the-art concepts such as Fourier features, adaptive sampling, and ABCs, tailoring the network to the typical structure of a non-stationary acoustic field due to an impulse point source, and on the NTK technique for balancing the weights of the multiple loss functions in the training process. We emphasize that although it is widely accepted that PINNs produce solutions that implicitly satisfy ABCs, imposing them explicitly helps to stabilize the training by removing ambiguity associated with the waves coming from outside the domain that still satisfy wave equation.

It is shown by the numerical experiments discussed that the accuracy of the solution of the wave equation obtained by the proposed PINN reaches the limits imposed by the mathematical tools involved (in our case, the L^2-error is comparable to the error caused by approximate absorbing BCs). On the other hand, training our PINN requires substantially less computational effort than for the existing state-of-the-art architectures.

It is important to stress that, however promising our results are, many issues still need to be addressed. In particular, it is important to include higher-order Higdon's conditions to overcome accuracy limitations associated with the latter (e.g., using the auxiliary variable approach). Ideally, it would be advantageous to train the neural network to automatically adjust the coefficients C_j in Eq. (2). The current implementation of our PINN does not take density jumps at media

interfaces (or, more generally, density variations) into account. This could be resolved, e.g., by introducing a weak formulation of the wave equation. Finally, attenuation and elasticity effects also need to be taken into account. Although this would require a substantial complication of the presented model, there is also no simple way to handle these issues within the traditional finite-difference or finite-elements models.

Acknowledgments

This research was supported by Petrobras. We also acknowledge financial support from Google.

Author Declarations

Conflict of Interest

The authors have no conflicts to disclose.

Data Availability

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

References

Alkhalifah, T., and Huang, X. (**2024**). "Physics-informed neural wavefields with Gabor basis functions," Neural Netw. **175**, 106286.
Daw, A., Bu, J., Wang, S., Perdikaris, P., and Karpatne, A. (**2022**). "Mitigating propagation failures in physics-informed neural networks using retain-resample-release (R3) sampling," arXiv:2207.02338.
Ding, Y., Chen, S., Miyake, H., and Li, X. (**2025**). "Physics-informed neural networks with Fourier features for seismic wavefield simulation in time-domain nonsmooth complex media," IEEE Trans. Geosci. Remote Sens. **63**, 5916913.
Higdon, R. L. (**1986**). "Absorbing boundary conditions for difference approximations to the multi-dimensional wave equation," Math. Comput. **47**(176), 437–459.
Hornik, K., Stinchcombe, M., and White, H. (**1989**). "Multilayer feedforward networks are universal approximators," Neural Netw. **2**(5), 359–366.
Hu, Z., Shukla, K., Karniadakis, G. E., and Kawaguchi, K. (**2024**). "Tackling the curse of dimensionality with physics-informed neural networks," Neural Netw. **176**, 106369.
Ito, R., Ishikawa, K., Tanigawa, R., and Oikawa, Y. (**2025**). "Three-dimensional sound field reconstruction from optical projections using physics-informed neural networks," JASA Express Lett. **5**(6), 064801.
Jacot, A., Gabriel, F., and Hongler, C. (**2018**). "Neural tangent kernel: Convergence and generalization in neural networks," in *Advances in Neural Information Processing Systems*, edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N.Cesa-Bianchi, and R.Garnett (Curran Associates, Inc., Montreal, Canada), Vol. 31.
Jensen, F. B., Kuperman, W. A., Porter, M. B., Schmidt, H., and Tolstoy, A. (**2011**). *Computational Ocean Acoustics* (Springer, New York).
Li, Z., Zhang, T., and Cheng, L. (**2025**). "Adaptive physics-informed neural networks for underwater acoustic field prediction," JASA Express Lett. **5**(6), 068301.
Liu, R., and Gerstoft, P. (**2024**). "Spatial acoustic properties recovery with deep learning," J. Acoust. Soc. Am. **155**(6), 3690–3701.
Mallik, W., Jaiman, R. K., and Jelovica, J. (**2022**). "Predicting transmission loss in underwater acoustics using convolutional recurrent autoencoder network," J. Acoust. Soc. Am. **152**(3), 1627–1638.
Martin, G. S., Wiley, R., and Marfurt, K. J. (**2006**). "Marmousi2: An elastic upgrade for Marmousi," Leading Edge **25**(2), 156–166.
Moseley, B., Markham, A., and Nissen-Meyer, T. (**2023**). "Finite basis physics-informed neural networks (FBPINNs): A scalable domain decomposition approach for solving differential equations," Adv. Comput. Math. **49**(4), 62.
Penwarden, M., Jagtap, A. D., Zhe, S., Karniadakis, G. E., and Kirby, R. M. (**2023**). "A unified scalable framework for causal sweeping strategies for physics-informed neural networks (PINNs) and their temporal decompositions," J. Comput. Phys. **493**, 112464.
Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., and Courville, A. (**2019**). "On the spectral bias of neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, edited by K. Chaudhuri and R. Salakhutdinov, Vol. 97, pp. 5301–5310.
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.
Ramachandran, P., Zoph, B., and Le, Q. V. (**2017**). "Searching for activation functions," arXiv:1710.05941.
Rasht-Behesht, M., Huber, C., Shukla, K., and Karniadakis, G. E. (**2022**). "Physics-informed neural networks (pinns) for wave propagation and full waveform inversions," J. Geophys. Res. Solid Earth **127**(5), e2021JB023120, https://doi.org/10.1029/2021JB023120.
Sethi, H., Pan, D., Dimitrov, P., Shragge, J., Roth, G., and Hester, K. (**2023**). "Hard enforcement of physics-informed neural network solutions of acoustic wave propagation," Comput. Geosci. **27**(5), 737–751.
Song, C., Alkhalifah, T., and Waheed, U. B. (**2021**). "Solving the frequency-domain acoustic VTI wave equation using physics-informed neural networks," Geophys. J. Int. **225**(2), 846–859.
Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., and Ng, R. (**2020**). "Fourier features let networks learn high frequency functions in low dimensional domains," Adv. Neural Info. Process. Syst. **33**, 7537–7547.
Varon, A., Mars, J., and Bonnel, J. (**2023**). "Approximation of modal wavenumbers and group speeds in an oceanic waveguide using a neural network," JASA Express Lett. **3**(6), 066003.
Wang, S., Li, B., Chen, Y., and Perdikaris, P. (**2024**). "Piratenets: Physics-informed deep learning with residual adaptive networks," J. Mach. Learn. Res. **25**(402), 1–51.
Wang, S., Teng, Y., and Perdikaris, P. (**2021a**). "Understanding and mitigating gradient flow pathologies in physics-informed neural networks," SIAM J. Sci. Comput. **43**(5), A3055–A3081.

Wang, S., Wang, H., and Perdikaris, P. (**2021b**). "On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks," Comput. Methods Appl. Mech. Eng. **384**, 113938.
Wang, S., Yu, X., and Perdikaris, P. (**2022**). "When and why PINNs fail to train: A neural tangent kernel perspective," J. Comput. Phys. **449**, 110768.
Wu, C., Zhu, M., Tan, Q., Kartha, Y., and Lu, L. (**2023**). "A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks," Comput. Methods Appl. Mech. Eng. **403**, 115671.
Wu, H., Luo, H., Ma, Y., Wang, J., and Long, M. (**2024**). "Ropinn: Region optimized physics-informed neural networks," Adv. Neural Inf. Process. Syst. **37**, 110494–110532.
Xi, Q., Fu, Z., Xu, W., Xue, M.-A., Rashed, Y. F., and Zheng, J. (**2024**). "FEM-PIKFNN for underwater acoustic propagation induced by structural vibrations in different ocean environments," Comput. Math. Appl. **176**, 46–54.
Yoon, S., Park, Y., Gerstoft, P., and Seong, W. (**2024**). "Predicting ocean pressure field with a physics-informed neural network," J. Acoust. Soc. Am. **155**(3), 2037–2049
Zhang, Y., Zhu, X., and Gao, J. (**2023**). "Seismic inversion based on acoustic wave equations using physics-informed neural network," IEEE Trans. Geosci. Remote Sens. **61**, 4500511.
Zhongkai, H., Yao, J., Su, C., Su, H., Wang, Z., Lu, F., Xia, Z., Zhang, Y., Liu, S., Lu, L., and Zhu, J. (**2024**). "Pinnacle: A comprehensive benchmark of physics-informed neural networks for solving pdes," Adv. Neural Inf. Process. Syst. **37**, 76721–76774.