

Article

Physics-Informed Neural Networks for Modal Wave Field Predictions in 3D Room Acoustics

Stefan Schoder 

Institute of Fundamentals and Theory in Electrical Engineering, Graz University of Technology, Inffeldgasse 18/I, 8010 Graz, Austria; stefan.schoder@tugraz.at

Abstract: The generalization of Physics-Informed Neural Networks (PINNs) used to solve the inhomogeneous Helmholtz equation in a simplified three-dimensional room is investigated. PINNs are appealing since they can efficiently integrate a partial differential equation and experimental data by minimizing a loss function. However, a previous study experienced limitations in acoustics regarding the source term. A challenging but realistic excitation case is a confined (e.g., single-point) excitation area, yielding a smooth spatial wave field periodically with the wavelength. Compared to studies using smooth (unrealistic) sound excitation, the network's generalization capabilities regarding a realistic sound excitation are addressed. Different methods like hyperparameter optimization, adaptive refinement, Fourier feature engineering, and locally adaptive activation functions with slope recovery are tested to tailor the PINN's accuracy to an experimentally validated finite element analysis reference solution computed with openCFS. The hyperparameter study and optimization are conducted regarding the network depth and width, the learning rate, the used activation functions, and the deep learning backends (PyTorch 2.5.1, TensorFlow 2.18.0 1, TensorFlow 2.18.0 2, JAX 0.4.39). A modified (feature-engineered) PINN architecture was designed using input feature engineering to include the dispersion relation of the wave in the neural network. For smoothly (unrealistic) distributed sources, it was shown that the standard PINNs and the feature-engineered PINN converge to the analytic solution, with a relative error of 0.28% and $2 \times 10^{-4}\%$, respectively. The locally adaptive activation functions with the slope lead to a relative error of 0.086% with a source sharpness of $s = 1$ m. Similar relative errors were obtained for the case $s = 0.2$ m using adaptive refinement. The feature-engineered PINN significantly outperformed the results of previous studies regarding accuracy. Furthermore, the trainable parameters were reduced to a fraction by Bayesian hyperparameter optimization (around 5%), and likewise, the training time (around 3%) was reduced compared to the standard PINN formulation. By narrowing this excitation towards a single point, the convergence rate and minimum errors obtained of all presented network architectures increased. The feature-engineered architecture yielded a one order of magnitude lower accuracy of 0.20% compared to 0.019% of the standard PINN formulation with a source sharpness of $s = 1$ m. It outperformed the finite element analysis and the standard PINN in terms time needed to obtain the solution, needing 15 min and 30 s on an AMD Ryzen 7 Pro 8840HS CPU (AMD, Santa Clara, CA, USA) for the FEM, compared to about 20 min (standard PINN) and just under a minute of the feature-engineered PINN, both trained on a Tesla T4 GPU (NVIDIA, Santa Clara, CA, USA).

Keywords: PINNs; FEM; Helmholtz equation; DeepXDE; openCFS; acoustics; waves; 3D; PDE; room acoustics; time-harmonic wave field; absorber



Academic Editors: Edoardo Piana, Marco Caniato and Paolo Bonfiglio

Received: 24 November 2024

Revised: 14 January 2025

Accepted: 16 January 2025

Published: 18 January 2025

Citation: Schoder, S. Physics-Informed Neural Networks for Modal Wave Field Predictions in 3D Room Acoustics. *Appl. Sci.* **2025**, *15*, 939. <https://doi.org/10.3390/app15020939>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Today, there are persisting challenges in room acoustics simulations, with errors mainly arising from uncertain input parameters of used building material, resulting in a high uncertainty in the used boundary conditions [1,2]. In particular, at low-frequency waves, the material parameters are off [3,4], hindering more accurate simulation results. These uncertain simulation input data underline the importance of using algorithms that allow for the on-the-fly minimization of the model discrepancy with reality. As a selected algorithm, physics-informed neural networks (PINNs) can predict the wave field and infer the material coefficients or boundary condition parameters with the use of data [3]. However, the computation of the time-harmonic acoustic wave field in large 3D geometries represents a challenge for current wave-based numerical and machine-learned methods [4,5]. The solutions to the Helmholtz equation in 3D are essential for understanding low-frequency room acoustic scenarios. The Finite Element Methods (FEM) analysis provides accurate results [4], but the extension to inverse parameters estimation is tedious compared to PINNs. PINNs show promising capabilities for that purpose, such that the interest in tailoring these algorithms to the accuracy of FEM has become an interesting topic.

So far, PINNs have been used to solve the Helmholtz equation [6–8] only in 1D and 2D domains for acoustic simulations [3,9–15], ultra-sound application [16], electromagnetic waveguides [17,18], and aeroacoustics [19]. It was already observed in [11] that the training deteriorates with increasing wave number due to the frequency-principle [20,21] and deteriorates when changing from the Dirichlet boundary condition to a Neumann boundary condition. A weak point regarding the acoustics predictions of previous studies using PINNs is that the solution of the inhomogeneous Helmholtz equation is similar to the widespread exciting source of the partial differential equation. A cosine function in space describes the wave field, and a scaled and unrealistic cosine function models the source covering the whole room. However, typical sources in acoustics are confined [22,23], resulting in a substantial increase in the prediction discrepancies from a 2.5% to 92% relative error [5]. These confined sources excite a spatial periodic wave field. This characteristic is known as the dispersion relation of the acoustic wave. Compared to previous studies focusing on unrealistic smooth excitation [9,11], we will analyze the network's generalization capabilities towards realistic excitation scenarios caused by a (loudspeaker-like) confined source. A structured hyperparameter study provides intuition about the network key parameter changes (concerning the results of [11]), the activation functions, and various backends, including PyTorch [24], TensorFlow 1 and 2 [25], and JAX [26]. All results are compared to results from a validated FEM analysis using openCFS [4]. Rerunning the training of the PINN reported in [5] on a Tesla T4 GPU with Tensorflow 1, the training time of the 3D PINN took about 84 min compared to a solver time of the FEM analysis of 16 min on a standard notebook CPU. In [11], the randomly distributed point-wise l^2 error measure was used as a loss metric. It is highly questionable whether a random distribution of evaluation points computes a reliable error measure for assessing numerical convergence. The accuracy is evaluated in a best-practice manner by comparing the results on a structured evaluation grid with the FEM results and computing an L^2 error norm. As a follow-up to the hyperparameter study, additional data points are added by residual-based adaptive refinement. In addition to the standard neural network architectures, the locally adaptive activation functions with slope recovery [27] and the multi-scale Fourier feature network [28] are investigated for potentially improved neural network performance. In addition to these structures, a modified architecture was proposed using feature engineering to include the wave's dispersion relation in the neural network. It shows how the input features transform the FEM reference solution into a monotonic line that can be more easily predicted by the neural network. For this feature-engineered network structure,

a hyperparameter optimization using Gaussian processes-based Bayesian optimization yielded a substantial reduction in the trainable parameters (around 5% compared to the standard PINN formulation) and the training time (around 60 s compared to 20 min of the standard PINN formulation) while it slightly improved the prediction accuracy.

This article starts with Section 2, introducing the numerical examples of 3D room-like geometry, the differential equation with sound-hard wall boundary conditions, and a forcing source including the narrowing of the source by a sharpness parameter s . Section 3 presents the theoretical background of a standard PINN, the parameter space of the experimental hyperparameter analysis, the multi-scale Fourier feature network with randomly sampled Fourier features [28], and the locally adaptive activation functions with slope recovery [27]. Section 4 presents the investigated hypotheses. This is followed by the results section that presents the details of the investigations and the comparison of the FEM analysis by numerical error analysis. Based on the results, the conclusions of the multi-scale Fourier feature network led to the feature-engineered network architecture using the acoustic dispersion relation for feature engineering. Section 6 summarizes the significance and concludes the findings regarding the hypotheses.

2. Room Acoustic Application

A room's linear acoustic wave field can be described by solving the inhomogeneous Helmholtz equation with suitable boundary conditions [29]. We present the application example as it will be essential to understand the details of the used PINN variants (see Section 3).

2.1. Case Study Setup

In particular, a simplified room-like 3D domain $\Omega = [0, 1]^3 \text{ m}^3$ is considered. This room setup presents the lowest geometrical complexity while retaining the wall reflections interacting with the distributed and confined source regions. The inhomogeneous Helmholtz equation as the governing equation with homogeneous Neumann boundary conditions on $\partial\Omega = \Gamma_N$ reads as

$$(\Delta + k^2)p(\omega, \mathbf{x}) = g(\omega, \mathbf{x}) \quad \mathbf{x} \in \Omega \quad (1)$$

$$\nabla p(\mathbf{x}) \cdot \mathbf{n} = 0, \quad \mathbf{x} \in \Gamma_N. \quad (2)$$

In this equation, $p \in \mathbb{C}$ is the time-harmonic acoustic pressure, $\mathbf{x} = (x, y, z)$ is the space coordinate, $k = \omega/c = 2\pi/\lambda$ is the wave number, λ the wavelength, ω the angular frequency, $c = 1$ the speed of sound, g the forcing term, $\Delta = \nabla \cdot \nabla$ is the Laplacian operator, and ∇ the gradient. Homogeneous Neumann boundary conditions model the room walls. The forcing consists of

$$g(x, y, z) = -2k^2 \cos(kx) \cos(ky) \cos(kz) e^{-\frac{\|\mathbf{x} - \mathbf{x}_0\|_2^2}{2s^2}}, \quad (3)$$

with a source sharpness parameter s , the Euclidean distance $\|\mathbf{x} - \mathbf{x}_0\|_2$, and a source location $\mathbf{x}_0 = (0.5, 0.5, 0.5) \text{ m}$. In the case of $s \rightarrow 0$, the source leads to a point source confined in the source location $\mathbf{x}_0$. As a follow-up to the previous 3D room acoustic examples [9–15], the source is modeled more realistically by a point source in the center of the room modeled for sufficiently small s .

2.2. Analytic Solution

For $s \rightarrow \infty$, the forcing can be simplified to

$$g(x, y, z) = -2k^2 \cos(kx) \cos(ky) \cos(kz). \quad (4)$$

This source leads to an analytic solution of

$$p(x, y, z) = \cos(kx) \cos(ky) \cos(kz) \quad (5)$$

and it satisfies the homogeneous Neumann boundary condition $\nabla p(x, y, z) \cdot \mathbf{n} = 0$. In the current case, the solution is only real-valued $p \in \mathbb{R}$; however, in the general case, the solution is complex $p = \Re(p) + i\Im(p)$ and may be treated by one network with two output neurons or two independent networks with one output each, providing the independent solutions parts $\Re(p)$ and $\Im(p)$. In the case of a finite s , the physical system is in resonance, and slight damping is included to make the system of equations well-posed $c^2 = c_0^2 + ic_1^2$, with $c_1^2 = 0.04$. In this case, the pressure $p \in \mathbb{C}$ is a complex function.

2.3. Spatial Frequency Content

The spatial frequency $f_x = 1/\lambda_x$ of the solution is directly connected to the excited source by the linearity of the Helmholtz equation and is provided by the value of $\lambda_x = \lambda$. For cases with $s > \lambda$, the solution wavelength is primarily determined by λ . However, in the case of $s < \lambda$, the wavelength content of the Gaussian envelope becomes important, which is determined by the Fourier transform of it

$$\mathcal{F}[e^{-(x/s)^2}](f_x) = \sqrt{2\pi} s e^{-2(\pi s f_x)^2}. \quad (6)$$

The frequency content governed by the Gaussian f_x is sufficiently low for values of f_x being smaller than two standard deviations. In our case, this condition reads $f_x < 1/(\pi s)$ or more specifically $\lambda > \pi s$.

2.4. Validated FEM Reference Simulation

An experimentally validated FEM setup [22] is used to compute the reference solution with openCFS [30]. The mesh approximates a wavelength λ per 50 degrees of freedom N_{dof}/λ , satisfying the conditions of [31]. The mesh consists of 1.06 million elements. For quadratic polynomial basis functions, the mesh will provide accurate results for $\lambda > 0.06125$ m and $s > 0.06125/\pi$ m [32]. A simulation run on a standard notebook CPU (AMD Ryzen 7 Pro 8840HS with 3.3 GHz (AMD, Santa Clara, CA, USA)) takes 15 min and 30 s. The mesh study conducted led to the relative errors

$$e_{\text{rel}} = \sqrt{\frac{\sum_{i=1}^{|\mathcal{T}_{\text{test}}|} (p_{i,\text{FEM}} - p_{i,\text{FEMx}})^2}{\sum_{i=1}^{|\mathcal{T}_{\text{test}}|} (p_{i,\text{FEM}})^2}}, \quad (7)$$

where $|\mathcal{T}_{\text{test}}|$ is the number of evaluation points, $p_{i,\text{FEM}}$ is the FEM reference pressure of the $N_{dof}/\lambda = 50$ mesh, and $p_{i,\text{FEMx}}$ is the pressure result from the coarser mesh reported in Table 1.

Table 1. Convergence of the finite element results for varying source sharpness parameters s . For $s \rightarrow \infty$, the errors are compared to the analytic case. In the other two cases, the error is compared to the $N_{dof}/\lambda = 50$ case.

N_{dof}/λ	e_{rel} in % $s \rightarrow \infty$	$s = 1$	$s = 0.2$	CPU Time
20	1.3×10^{-2}	1.35	1.69	5 s
25	4.5×10^{-3}	0.73	1.18	15 s
30	1.8×10^{-3}	0.34	0.66	33 s
35	8.0×10^{-4}	0.35	0.46	1 min 11 s
40	3.5×10^{-4}	0.43	0.33	3 min 43 s
45	1.5×10^{-4}	0.22	0.20	5 min 12 s
50	5.0×10^{-5}	-	-	15 min 30 s

3. Physics-Informed Neural Networks for Wave-Based Room Acoustics

PINNs combine neural networks and partial differential equations (PDEs) using automatic differentiation and a modified loss function [33]. Figure 1 shows an L -layer-feed-forward neural network [34,35] PINN that can solve the Helmholtz Equation (2) with Neumann boundary conditions.

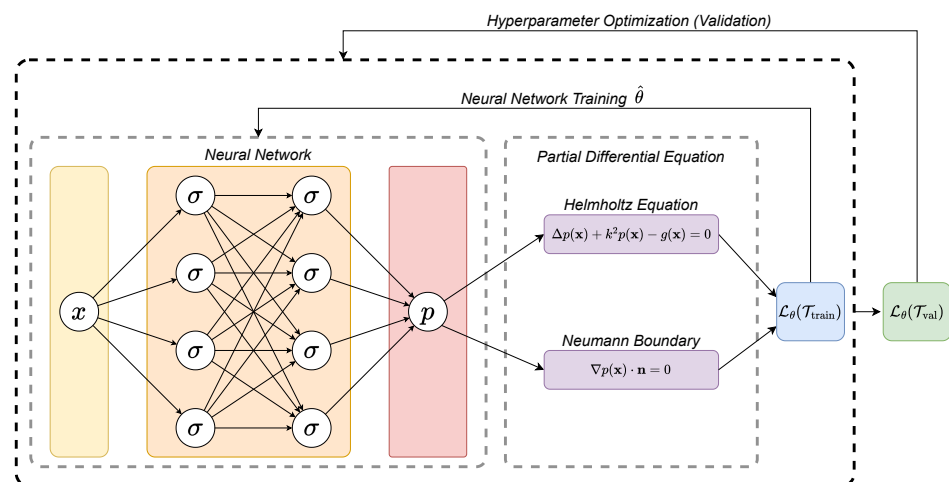


Figure 1. Schematic representation of a standard PINN. A neural network of $L = 3$ and $N = 4$ learns the mapping $x \rightarrow p(x)$. The residual includes the partial differential equation. The neural network trainable parameters are optimized towards the optimal parameters $\hat{\theta}$. In an outer loop, the network's hyperparameters are optimized regarding a validation dataset.

The following description gives the mathematical definitions of the individual blocks of the schematic in Figure 1. The prediction behavior of an L -layer-feed-forward neural network is defined by an operator from the input x (spatial locations) to the estimated output (pressure of the acoustic wave field) $\hat{p}(x)$. Let $\mathcal{N}(\mathbf{x}) : \mathbb{R}^{d_{in}} \rightarrow \mathbb{R}^{d_{out}}$ be an L -layer-feed-forward neural network which predicts $\hat{p}(x)$, with N_ℓ neurons in the ℓ -th layer. The argument of the nonlinear activation function σ is a tensor product of a weight matrix and inputs, and a bias vector is added

$$\sigma\left(\mathbf{W}^\ell \sigma\left(\mathcal{N}^{\ell-1}(\mathbf{x})\right) + \mathbf{b}^\ell\right). \quad (8)$$

The weight matrix in the ℓ -th layer is denoted by $\mathbf{W}^\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ and the bias vector by $\mathbf{b}^\ell \in \mathbb{R}^{N_\ell}$, respectively. The feed-forward neural network is recursively defined by

$$\begin{aligned}
 &\text{Input layer: } \mathcal{N}^0(\mathbf{x}) = \mathbf{x} \in \mathbb{R}^{d_{\text{in}}} \\
 &\text{Hidden layers: } \sigma(\mathcal{N}^\ell(\mathbf{x})) = \sigma(\mathbf{W}^\ell \sigma(\mathcal{N}^{\ell-1}(\mathbf{x})) + \mathbf{b}^\ell) \in \mathbb{R}^{N_\ell}, \quad \text{for } 1 \leq \ell \leq L-1, \\
 &\text{Output layer: } \mathcal{N}^L(\mathbf{x}) = \mathbf{W}^L \sigma(\mathcal{N}^{L-1}(\mathbf{x})) + \mathbf{b}^L \in \mathbb{R}^{d_{\text{out}}}.
 \end{aligned} \tag{9}$$

In the 3D case, the coordinates of one point $\mathbf{x} = (x, y, z)$ are provided as the input at the input layer. Defined by the neural network, the output prediction at the output layer L is calculated by

$$\hat{p}(\mathbf{x}) = \mathbf{W}^L \sigma(\mathcal{N}^{L-1}(\mathbf{x})) + \mathbf{b}^L = \mathbf{W}^L \sigma(\mathbf{W}^{L-1} \sigma(\mathcal{N}^{L-2}(\mathbf{x})) + \mathbf{b}^{L-1}) + \mathbf{b}^L = \dots \tag{10}$$

$$= \mathcal{N}^L \circ \sigma \circ \mathcal{N}^{L-1} \circ \sigma \circ \dots \circ \sigma \circ \mathcal{N}^1 \circ \mathbf{x} = \mathcal{N}(\boldsymbol{\theta}, \mathbf{x}), \tag{11}$$

being a series of consecutive transformation functions of $\mathcal{N}^\ell$ and sigma. The transformation is determined by the nonlinear operation $\mathcal{N}(\boldsymbol{\theta}, \mathbf{x})$. Its arguments are the trainable parameters $\boldsymbol{\theta}$ of all weights and biases $\boldsymbol{\theta} = \{\mathbf{W}^\ell, \mathbf{b}^\ell\}_{1 \leq \ell \leq L}$ and the network input $\mathbf{x}$.

During training, $\boldsymbol{\theta}$ will be learned by unlabeled training data $\mathcal{T}_{\text{train}} = \{\mathbf{x}_i\}_{i=1}^{|\mathcal{T}_{\text{train}}|}$. The neural network operator $\hat{p} = \mathcal{N}(\boldsymbol{\theta}, \mathbf{x})$ approximates a continuous function mapping $h(\mathbf{x}) = p$, regarding the minimal error described by the loss function $\mathcal{L}(\boldsymbol{\theta}, \mathcal{T}_{\text{train}})$. The loss will be minimized by

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^{|\boldsymbol{\theta}|}} (\mathcal{L}(\boldsymbol{\theta}, \mathcal{T}_{\text{train}})), \tag{12}$$

such that the weights and biases $\boldsymbol{\theta}$ are optimized to the converged ones $\hat{\boldsymbol{\theta}}$. Within the study, we used a mini-batch variant of the gradient-based optimization algorithm Adam [36] and the quasi-Newton method LBFGS [37]. The weights and biases are adapted by the gradient of the loss function $\nabla_{\boldsymbol{\theta}} \mathcal{L}$ to decrease it iteratively, using backpropagation [38] and automatic differentiation [39].

The residual of the Helmholtz equation r_{PDE} and the respective mean-squared error loss function term is defined by

$$\begin{aligned}
 r_{\text{PDE}}(\mathbf{x}) &= \Delta \hat{p}(\mathbf{x}) + k^2 \hat{p}(\mathbf{x}) - g(\mathbf{x}) \quad \forall \mathbf{x} \in D_{\text{PDE}} \\
 \mathcal{L}_{\text{PDE}}(\boldsymbol{\theta}, \mathcal{T}_f) &= \frac{1}{|\mathcal{T}_{\text{train}, \Omega}|} \sum_{i=1}^{|\mathcal{T}_{\text{train}, \Omega}|} \| r_{\text{PDE}}(\mathbf{x}) \|^2
 \end{aligned} \tag{13}$$

with $\|\cdot\|$ being the l_2 -norm. The loss function incorporates the partial differential equation into the training process. It is evaluated at randomly sampled space points $\mathbf{x}$ within the computational domain Ω , forming the dataset $\mathcal{T}_{\text{train}}$. It consists of two subsets $\mathcal{T}_{\text{train}} = \mathcal{T}_{\text{train}, \Omega} \cup \mathcal{T}_{\text{train}, \Gamma}$. The points in the domain Ω form the subset $\mathcal{T}_{\text{train}, \Omega} = \{\mathbf{x}_i\}_{i=1}^{|\mathcal{T}_{\text{train}, \Omega}|}$ with a total number of $|\mathcal{T}_{\text{train}, \Omega}|$ samples. This set is used to evaluate the PDE residual. The second subset collects the points on the boundary Γ , $\mathcal{T}_{\text{train}, \Gamma} = \{\mathbf{x}_i\}_{i=1}^{|\mathcal{T}_{\text{train}, \Gamma}|}$ with a total number of $|\mathcal{T}_{\text{train}, \Gamma}|$ samples. This set is used to evaluate the Neuman boundary condition residual. The Neumann boundary condition residual r_{NBC} and the respective mean-squared error loss function term is given by

$$\begin{aligned}
 r_{\text{NBC}} &= \nabla p(\mathbf{x}) \cdot \mathbf{n} \quad \forall \mathbf{x} \in D_{\text{NBC}} \\
 \mathcal{L}_{\text{NBC}}(\boldsymbol{\theta}, \mathcal{T}_{b2}) &= \frac{1}{|\mathcal{T}_{\text{train}, \Gamma}|} \sum_{i=1}^{|\mathcal{T}_{\text{train}, \Gamma}|} \| r_{\text{NBC}}(\mathbf{x}) \|^2.
 \end{aligned} \tag{14}$$

The total loss function can be described by a weighted sum of individual error terms

$$\mathcal{L} = w_{\text{PDE}} \mathcal{L}_{\text{PDE}} + w_{\text{NBC}} \mathcal{L}_{\text{NBC}}, \tag{15}$$

where w_{PDE} and w_{NBC} are weights.

3.1. Initial Setup [5]

The wavelength is selected to be $\lambda = 1/2$ m. The PINN is setup using DeepXDE 1.12.2 (Python 3.10, Cuda 12.2) and trained on a Nvidia Tesla T4 GPU (NVIDIA, Santa Clara, CA, USA). There are 10 random collocation points per wavelength along each direction for training and 30 used for validation are defined. This amounts to $n_{\text{train}} = 20$ data points per direction and a total $|\mathcal{T}_{\text{train},\Omega}| = n_{\text{train}}^3 = 8000$ points inside the domain. On the surface, a total number of $|\mathcal{T}_{\text{train},\Gamma}| 6n_{\text{train}}^2 = 2400$ is given. A fully connected neural network consists of $L = 4$ layers and a layer width of $W = 180$ neurons. A sinus activation function is used with Glorot uniform bias and weights initialization [40]. The loss term is defined in Equation (15), with $w_{\text{NBC}} = 5$, and $w_{\text{PDE}} = 1$. The wave field pressure inside the room on a $z = 0.5$ m plane is illustrated in Figure 2. It shows that the real part of the pressure spatially varies between -1 Pa and $+1$ Pa, and one can see the wavelength. Figure 2a shows the exact solution of the acoustic pressure ($s \rightarrow \infty$). Figure 2b shows a visualization of the predicted acoustic pressure by the PINN. The network is trained over 30,000 iterations by the ADAM optimizer with a learning rate of $\eta = 0.001$, resulting in a validation loss of 0.1248. After 200,000 iterations by the ADAM optimizer, the validation loss decreased further to 0.0952 and the relative error e_{rel} to 0.36% compared to the analytic solution. Figure 2c shows the result after finetuning the results from Figure 2b with the L-BFGS optimizer for 2000 iterations. It brings down the test loss to 0.0609 (see Figure 2c) and e_{rel} to 0.35%. Visually, there are no differences between these plots, and the method seems to be promising.

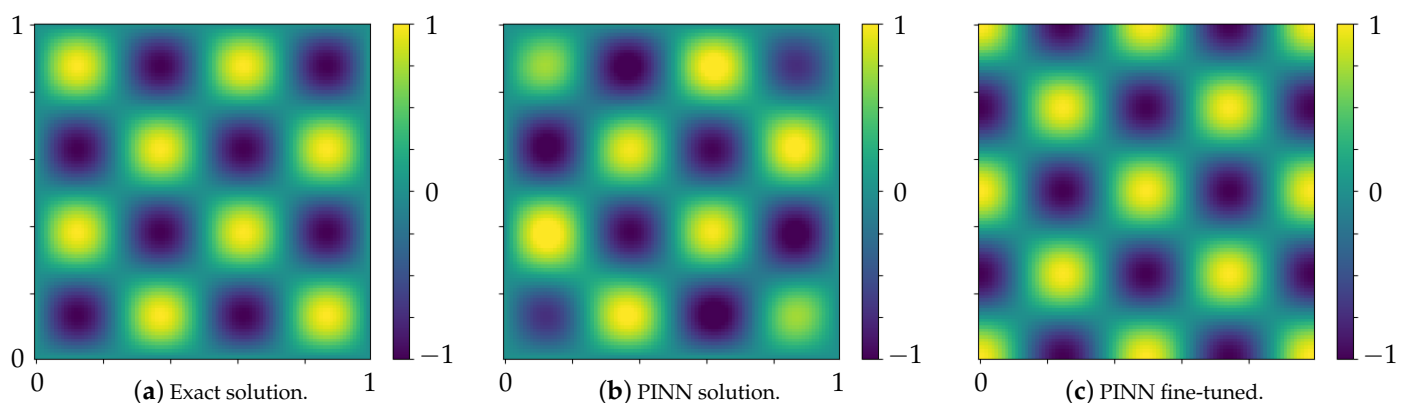


Figure 2. The real part of the acoustic pressure as a result of (a) the analytic expression, (b) standard PINN trained with the ADAM optimizer, and (c) the standard PINN fine-tuned by the L-BFGS optimizer from the solution of (b) at a plane with $z = 0.5$ m. Adapted with permission from Ref. [5].

3.2. Hyperparameter Study

To further tune the network's prediction accuracy, an explorative hyperparameter study was performed using the following parameter space θ_{NN}

$$\theta_{\text{NN}} = \{L - 1, W, w_{\text{NBC}}, \sigma, \mathcal{B}\}, \quad (16)$$

with

$$\begin{aligned} L - 1 &= \{1, 2, 3, 4, 5\} \\ W &= \{20, 40, 60, 120, 180, 300\} \\ w_{\text{NBC}} &= \{500, 50, 5, 1, 1/5, 1/50\} \\ \sigma &= \{\sin, \text{ELU}, \text{GELU}, \text{ReLU}, \text{SELU}, \text{sigmoid}, \text{SiLU}, \text{swish}, \text{tanh}\} \\ \mathcal{B} &= \{\text{PyTorch 2.5.1 (P)}, \text{TensorFlow 2.18.0 1 (T1)}, \\ &\quad \text{TensorFlow 2.18.0 2 (T2)}, \text{JAX 0.4.39 (J)}\}. \end{aligned}$$

Regarding the long simulation time reported in [5,11] and the difficulty in interpreting the outcome of [11], we first decided to run several hyperparameters by a manual line search approach. As the initial value of the unvaried hyperparameters, the reference setup from Figure 2b was used. The investigation procedure started with varying the network depth, followed by the network width, the residual weight, the activation function, and finally, the backends. In each step, the best model characteristics were used for the follow-up line search in the next hyperparameter dimension. The best model was selected based on the relative error compared to the validated FEM solution presented in Section 2.4.

3.3. Locally Adaptive Activation Functions

The locally adaptive activation function (LAAF) with slope recovery [27] is a method designed to improve the training speed and performance of physics-informed neural networks. It introduces a scalable parameter, a so-called activation slope na_i^ℓ (for layer ℓ and neuron i), to the activation function optimized during training (see Figure 3). The locally adaptive activation function neural network (at the neuron level) is defined as

$$\begin{aligned} \text{Input layer: } \mathcal{N}^0(\mathbf{x}) &= \mathbf{x} \in \mathbb{R}^{d_{\text{in}}} \\ \text{Hidden layers: } 1 \leq \ell \leq L - 1, \\ \sigma\left(na^\ell \mathcal{N}^\ell(\mathbf{x})\right) &= \sigma\left(na^\ell (\mathbf{W}^\ell \sigma(na^{\ell-1} \mathcal{N}^{\ell-1}(\mathbf{x})) + \mathbf{b}^\ell)\right) \in \mathbb{R}^{N_\ell}, \\ \text{Output layer: } \mathcal{N}^L(\mathbf{x}) &= \mathbf{W}^L \sigma(na^{L-1} \mathcal{N}^{L-1}(\mathbf{x})) + \mathbf{b}^L \in \mathbb{R}^{d_{\text{out}}}; \end{aligned} \quad (17)$$

The output prediction at the output layer L is calculated by

$$\hat{p} = \mathcal{N}^L \circ \sigma \circ na^{L-1} \mathcal{N}^{L-1} \circ \sigma \circ \dots \circ \sigma \circ na^1 \mathcal{N}^1 \circ \mathbf{x}, \quad (18)$$

a series of transformation functions. The slope recovery term

$$S(a) = \frac{L - 1}{\sum_{\ell=1}^{L-1} \exp\left(\frac{\sum_{i=1}^{N_\ell} a_i^\ell}{N_\ell}\right)} \quad (19)$$

encourages the network to increase the activation slope, leading to faster training. The loss function is extended to include the slope recovery term by

$$\mathcal{L} = w_{\text{PDE}} \mathcal{L}_{\text{PDE}} + w_{\text{NBC}} \mathcal{L}_{\text{NBC}} + w_a S(a). \quad (20)$$

The analysis in [27] shows that LAAFs effectively multiply the gradients by conditioning matrices, accelerating convergence without explicitly calculating these matrices. Experiments demonstrated that LAAF works for deep neural networks in the application of standardized datasets MNIST [41], Fashion-MNIST [42], KMNIST [43], Semeion [44], SVHN [45], and CIFAR [46]. Compared to fixed and global adaptive activations, neuron-

wise LAAF exhibited faster training speeds when approximating a discontinuous function and, in combination with PINNs, the inverse problem of identifying the material coefficient of Poisson's equation and Burgers' equation in 2D. LAAF contributes to the overall challenge of reducing the number of iterations needed for convergence compared to the fixed activation function. This feature was tested with the backend TensorFlow 1 in DeepXDE and $n = 10$ for the reliable hyperparameters with a network depth of $L - 1 = 2$, each layer consisting of 180 neurons; $w_{NBC} = 5$, the n of the DeepXDE implementation of layer-wise (a uniform $a_i^\ell = a^\ell$ for all neurons in the layer) LAAF is selected to be 2. The parameters selected were a good compromise for the standard PINN.

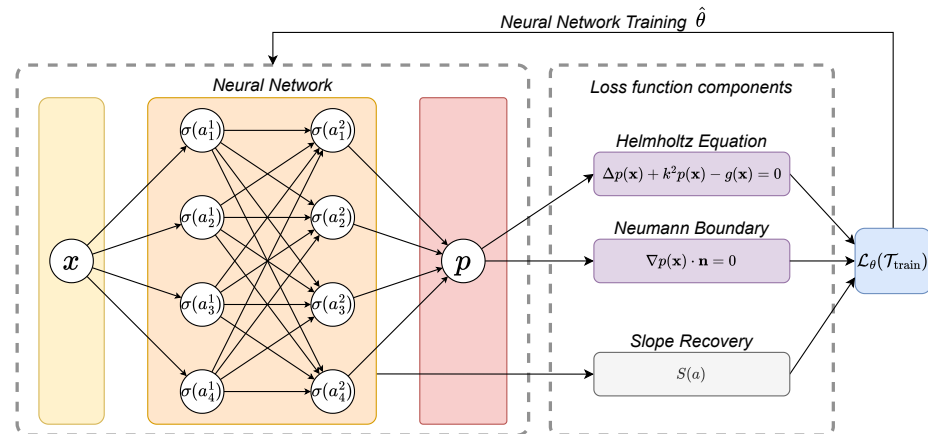


Figure 3. Schematic representation of a neuron-wise LAAF-PINN with slope recovery that learns the mapping $x \rightarrow p(x)$.

3.4. Multi-Scale Fourier Feature Networks

The multi-scale Fourier feature architecture (see Figure 4) was proposed in [28] to address the spectral bias and low convergence rates observed in conventional PINN models [47]. The spectral bias expresses that the target function is first learned along the eigendirections connected to the larger eigenvalues of the neural tangent kernel and gradually the remaining components of smaller eigenvalues. Translated to conventional, fully connected neural networks, the eigenvalues are lowered by the increasing frequency of the corresponding eigenfunctions. As a result, the convergence rate of high-frequency components of $p(x)$ is low [48,49]. This learning tendency is a significant challenge when dealing with multi-scale problems like localized acoustic sources featuring space-dependent high-frequency solution features. The challenge is addressed by embedding multiple Fourier features into the network to learn features across scales and to mitigate the spectral bias behavior. Firstly, the input coordinates are transformed using multiple Fourier feature mappings. The forward pass of M Fourier features with $m = 1, \dots, M$ is defined by

$$\begin{aligned}
 &\text{Input layer: } \mathcal{N}^0(\mathbf{x}) = \mathbf{x} \in \mathbb{R}^{d_{\text{in}}} \\
 &\text{Fourier layer: } \gamma^{(m)}(\mathbf{x}) = \begin{bmatrix} \cos(2\pi B^{(m)} \mathbf{x}) \\ \sin(2\pi B^{(m)} \mathbf{x}) \end{bmatrix} \in \mathbb{R}^{2d_{\text{in}} \times M} \\
 &\text{Firstst hidden layer: } \sigma(\mathcal{N}^{1,(m)}(\mathbf{x})) = \sigma(\mathbf{W}^1 \gamma^{(m)}(\mathbf{x}) + \mathbf{b}^1) \in \mathbb{R}^{N_1}, \\
 &\text{Hidden layers: } \sigma(\mathcal{N}^{\ell,(m)}(\mathbf{x})) = \sigma(\mathbf{W}^\ell \sigma(\mathcal{N}^{\ell-1,(m)}(\mathbf{x})) + \mathbf{b}^\ell) \in \mathbb{R}^{N_\ell}, \quad \text{for } 2 \leq \ell \leq L-1, \\
 &\text{Output layer: } \hat{p}(\mathbf{x}) = \mathbf{W}^L \cdot [\sigma(\mathcal{N}^{L,(1)}), \sigma(\mathcal{N}^{L,(2)}), \dots, \sigma(\mathcal{N}^{L,(M)})] + \mathbf{b}^L \in \mathbb{R}^{d_{\text{out}}};
 \end{aligned} \tag{21}$$

where $B^{(m)}$ represents the preselected matrices with entries randomly sampled from a Gaussian distribution $N(0, \hat{s}_m^2)$. The $\hat{s}_m$ values are chosen because they are additional

hyperparameters that may be optimized, and typical values should be in the range of the expected frequency (e.g., $k = 1/\lambda$) [28]. Overall, no additional training parameters θ are introduced by this architecture compared to a standard fully connected neural network.

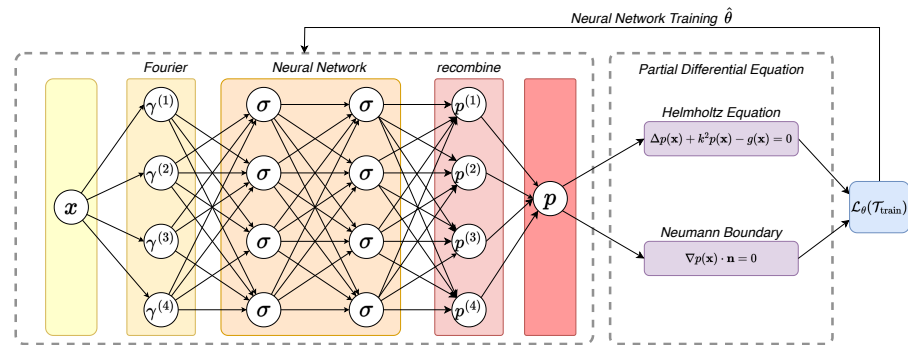


Figure 4. Schematic representation of a PINN with multi-scale Fourier feature architecture.

3.5. Validation and Testing Procedure

The validation and testing procedure is based on the standard convergence evaluation criteria for solving the partial differential in scientific computing [50]. The final physical validation (testing) of the PINN simulation was performed with pyCFS-data 0.0.7 [51] using the relative l^2 -error measure on an evenly spaced evaluation sample $\mathcal{T}_{\text{test}} = \{x_i\}_{i=1}^{|\mathcal{T}_{\text{test}}|}$

$$e_{\text{rel}} = \sqrt{\frac{\sum_{i=1}^{|\mathcal{T}_{\text{test}}|} (p_{i,\text{FEM}} - p_{i,\text{PINN}})^2}{\sum_{i=1}^{|\mathcal{T}_{\text{test}}|} (p_{i,\text{FEM}})^2}}, \quad (22)$$

where $|\mathcal{T}_{\text{test}}|$ is the number of evaluation points, $p_{i,\text{FEM}}$ is the FEM reference pressure result, and $p_{i,\text{PINN}}$ is the pressure result from the PINN. This error measure on a structured evaluation point sample is proportional to the L^2 error. The evenly spaced evaluation points coincide with the finite vertices of the FEM elements in the mesh.

4. Hypotheses

The core part of this article is the investigation of the influence on the prediction accuracy through narrowing the spread of the excitation source by decreasing the sharpness parameter s . In this preliminary section, the generalization capabilities of the standard neural network for confined sources are investigated. The source function $g(x)$ is modeled more by a gradually confined source in the center of the room modeled by Equation (3) and illustrated in Figure 5a. The sharpness of the source can be tuned by the parameter s in the exponential part of Equation (3). In Figure 5a, we see a range of s -parameters ranging from a distributed source for $s \rightarrow \infty$ to a relatively confined source $s = 0.02$ m. When sharpening the excitation $g(x)$, the neural network must model additional spatial modes and feature a multi-scale problem. As pointed out [28] and indicated by the modal description of a point source [52], it may be difficult for a standard fully-connected neural network to produce accurate results of the acoustic pressure. Figure 5b shows the results provided by the FEM analysis on the selected center line $y = z = 0.5$ m. The FEM analysis results show that independently of the source function, the acoustic pressure is primarily dominated by the wave number connected to the time-harmonic oscillation of 0.5 m.

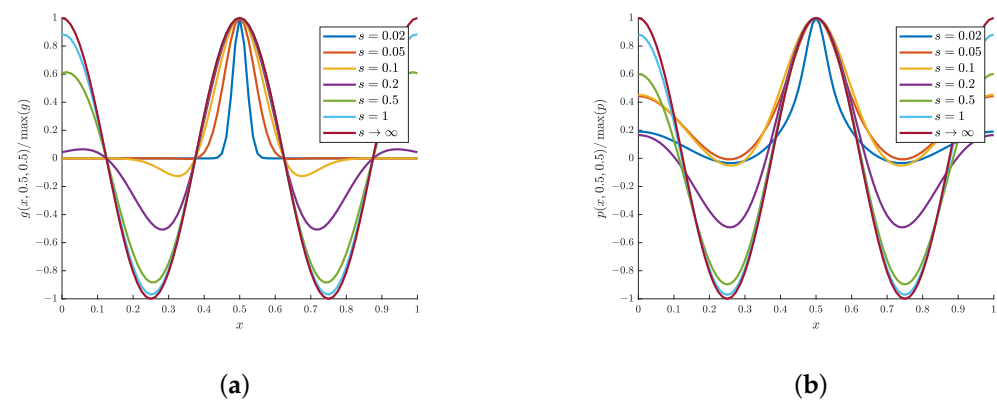


Figure 5. Higher order oscillation and spatial dependence of (a) the forcing and (b) the pressure results as a function of the source sharpness parameter s . (a) Forcing depending on the x coordinate at a line with $y = z = 0.5$ m. (b) FEM analysis pressure results depending on the x coordinate at a line with $y = z = 0.5$ m.

The maximum pressure amplitude is $\max(p) = \{1.01, 1.02, 1.11, 0.903, 0.505, 0.123\}$ Pa for $s = \{1, 0.5, 0.2, 0.1, 0.05, 0.02\}$ m, respectively. The lowest level of $s = 0.02$ was selected to respect the frequency considerations in Section 2.4 for accurate FEM solutions of the source concerning the FEM mesh. The standard network with hyperparameters $\theta_{\text{NN}} = \{L - 1 = 3, W = 180, w_{\text{NBC}} = 5, \sin, \text{PyTorch}\}$ (motivated by [5,28]), and after 200,000 iterations of training with a learning rate of 10^{-3} previously resulted in large relative errors [5]. The recalculation of the PINNs with respect to the converged FEM solution presented in Table 1 shows that the standard network with hyperparameters $\theta_{\text{NN}} = \{L - 1 = 3, W = 180, w_{\text{NBC}} = 5, \sin, \text{PyTorch}\}$ and after 200,000 iterations of training with a learning rate of 10^{-3} resulted in an error, with an increase in the relative error as the source sharpened $s \rightarrow 0$ (see Table 2).

Table 2. Summary of the relative errors of the standard PINN for various source sharpness parameters s compared to the FEM reference solution.

s in m	0.2	0.1	0.05	0.02
e_{rel} in %	4.52	5.72	7.05	no convergence

Therefore, the following hypotheses (H1–H4) were formed to investigate this behavior and improve prediction accuracy.

H1. *Tuning the hyperparameters positively influences the generalization and prediction accuracy.*

H2. *A layer-wise local adaptive activation function positively influences the generalization and prediction accuracy.*

H3. *Adding data points to the training dataset by loss-residual-based adaptive refinement positively influences the generalization and prediction accuracy.*

H4. *A multi-scale Fourier feature network positively influences the generalization and prediction accuracy.*

H1–H4 will be tested by the FEM reference solution.

5. Results

The recent hyperparameter study in [11] presented a framework for automatically analyzing a set of hyperparameters in a Gaussian processes-based Bayesian optimization. Both a 2D and a 3D scenario were studied with a distributed force $s \rightarrow \infty$, with a very detailed description of the 2D case (Dirichlet problem), which in turn is not directly translatable to the 3D case (Neumann problem). The representativeness of the study shows limited training runs for the expensive training runs. The hyperparameter optimization study for the 3D case found an optimal configuration with the learning rate 10^{-4} , a width of 207, a depth of 10, and a sine activation function and a boundary weight term of 1; the second best had a learning rate close to 10^{-3} , a width of 292, a depth of 3, and a sine activation function and a boundary weight term of 19.1. The diversity of these two nearly optimal solutions expresses the challenge and high sensitivity of optimizing the network structure for this application. It was reported that for increased wave numbers, the performance of the PINNs was low in the 2D and 3D cases. Furthermore, by uniformly increasing the density of collocation points, the performance of the network prediction capability improved, suggesting that adaptive refinement may lead to further improvements. From the results of [11], a structured hyperparameter exploration was conducted for the general 3D case with a finite source sharpness parameter s to evaluate the generalization regarding the excitation.

5.1. Explorative Hyperparameter Study for $s = 1\text{ m}$

Compared to the previous studies conducted for $s \rightarrow \infty$, the study investigates the networks' capability of calculating the results for a slightly distorted field based on $s = 1\text{ m}$. In this case, the amplitudes of the solution are marginally higher due to reflections at the sound-hard walls than for the simple example $s \rightarrow \infty$.

A relative error less than the relative l^2 deviations $e_{\text{rel}} < 3.14\%$ from the case $s \rightarrow \infty$ indicates that the network has learned a meaningful description of the field. The number of layers for a configuration improved the accuracy, as it can be seen in Table 3 and for the first cohort (separated by rules in the table). As a training speed accuracy compromise, the networks with $L - 1 = 2$ and $L - 1 = 3$ were selected for further investigations. This trend was observable for all hyperparameter tests provided in Table 3. In the second and third cohorts, a combination of switching the optimizer during the training was tested, and the effect of this on performance was examined. Interestingly, when using $w_{\text{NBC}} = 50$, the PDE and the BC loss were initially of the same level and caused an excellent prediction when combining it with the LBFGS optimizer to further iterate on the network trainable parameters (orange row in Table 3). In the final two cohorts, the boundary condition weights were tested. It was found that a relative weight increase from 5 to 500 improves the accuracy. As indicated by [53], an increasing relative boundary condition weight can positively influence the optimization by a more regular loss landscape.

In a subsequent study, the network width was tested, as was its influence on the accuracy (Table 4). As expected, narrow networks performed comparably poor. In some occasions for a layer width of 40 neurons, no convergence to the FEM results was obtained. In conclusion, the layer width between 120 and 240 gave acceptable results for this case. Compared to the network's depth, the width shows less sensitivity in terms of accuracy for a neuron range of 120–240 neurons. The accuracy was in the bounds for all tested backends (Table 5), with a somewhat superior performance of TensorFlow 1 leading to an error below 0.12% and the highest training speed of about 20 min (being in the range of the FEM simulation). Regarding the accuracy, the PINN using JAX as a backend led to a low relative L^2 error of 0.34%. This low relative error is comparable to the refined FEM simulation. Finally, the PINN was trained on an alternative GPU (NVIDIA A100-PCIE-40GB (NVIDIA,

Santa Clara, CA, USA)) with a newer tensor core technology, reducing the training time by a factor of more than 2.5. The execution times are in the range reported previously [11].

Table 3. Error e_{rel} in percentages for the number of layers and PDE/BC balancing weight hyperparameter of θ_{NN} compared to FEM results. Yellow-marked entries describe the best-performing parameter set in the direction of variation in the line search, and orange indicates the overall best value. For hyperparameters described as 5 + 50, 5 was chosen for the first step of the training sequence, with 90,000 Adams in this example, and then switched to 50 when continuing with the training by 15,000 iterations using LBFGS. The abbreviation "nc" stands for not converged.

$L - 1$	W	w_{NBC}	σ	$\mathcal{B}$	Training Sequence	e_{rel} in %
1	180	5	sin	P	90,000 Adams	nc
2	180	5	sin	P	90,000 Adams	0.77
3	180	5	sin	P	90,000 Adams	0.45
4	180	5	sin	P	90,000 Adams	0.14
5	180	5	sin	P	90,000 Adams	0.07
1	180	5	sin	P	90,000 Adams + 15,000 LBFGS	nc
2	180	5	sin	P	90,000 Adams + 15,000 LBFGS	0.61
3	180	5	sin	P	90,000 Adams + 15,000 LBFGS	0.42
4	180	5	sin	P	90,000 Adams + 15,000 LBFGS	0.13
5	180	5	sin	P	90,000 Adams + 15,000 LBFGS	0.07
1	180	5 + 50	sin	P	90,000 Adams + 15,000 LBFGS	nc
2	180	5 + 50	sin	P	90,000 Adams + 15,000 LBFGS	0.28
3	180	5 + 50	sin	P	90,000 Adams + 15,000 LBFGS	0.42
4	180	5 + 50	sin	P	90,000 Adams + 15,000 LBFGS	0.13
5	180	5 + 50	sin	P	90,000 Adams + 15,000 LBFGS	0.07
2	180	0.02	sin	P	90,000 Adams	24.15
2	180	0.2	sin	P	90,000 Adams	9.20
2	180	1	sin	P	90,000 Adams	7.19
2	180	5	sin	P	90,000 Adams	0.77
2	180	50	sin	P	90,000 Adams	0.70
2	180	500	sin	P	90,000 Adams	0.019
3	180	5	sin	P	90,000 Adams	0.45
3	180	50	sin	P	90,000 Adams	0.39
3	180	500	sin	P	90,000 Adams	0.041

Following up on the layer number and width study (Table 6), the activation functions were tested $\sigma = \{\text{sin}, \text{ELU}, \text{GELU}, \text{ReLU}, \text{SELU}, \text{sigmoid}, \text{SiLU}, \text{swish}, \text{and tanh}\}$. The solution oscillates around zero, yielding positive and negative values. The functions ELU, ReLU, SELU, and sigmoid performed poorly potentially due to their predominantly positive branch. It was confirmed in the tests that none of the networks using those activation functions reached the converged results with an error of less than 50%. The sinus function performed best; the GELU, swish, tanh, and SiLU activation reached similar errors, which were already quite large. In conclusion to H1, the hyperparameter optimization improved the accuracy from initially 0.45% to 0.019%.

Table 4. Error e_{rel} in percentages for the variable layer width as a hyperparameter compared to FEM results. Yellow-marked entries describe the best-performing parameter set in the direction of variation in the line search, and orange indicates the overall best value. The abbreviation “nc” stands for not converged.

$L - 1$	W	w_{NBC}	σ	$\mathcal{B}$	Training Sequence	e_{rel} in %
2	20	5	sin	P	90,000 Adams	6.78
2	40	5	sin	P	90,000 Adams	0.48
2	60	5	sin	P	90,000 Adams	3.49
2	120	5	sin	P	90,000 Adams	2.34
2	180	5	sin	P	90,000 Adams	0.77
2	240	5	sin	P	90,000 Adams	3.90
2	300	5	sin	P	90,000 Adams	5.57
3	20	5	sin	P	90,000 Adams	11.36
3	40	5	sin	P	90,000 Adams	2.35
3	60	5	sin	P	90,000 Adams	1.79
3	120	5	sin	P	90,000 Adams	1.32
3	180	5	sin	P	90,000 Adams	0.45
3	240	5	sin	P	90,000 Adams	0.18
3	300	5	sin	P	90,000 Adams	3.01
2	20	50	sin	P	90,000 Adams	nc
2	40	50	sin	P	90,000 Adams	0.76
2	60	50	sin	P	90,000 Adams	1.39
2	120	50	sin	P	90,000 Adams	1.81
2	180	50	sin	P	90,000 Adams	0.70
3	20	50	sin	P	90,000 Adams	5.44
3	40	50	sin	P	90,000 Adams	1.32
3	60	50	sin	P	90,000 Adams	2.02
3	120	50	sin	P	90,000 Adams	0.057
3	180	50	sin	P	90,000 Adams	2.37

Table 5. Error e_{rel} in percentages for the backend as a hyperparameter variation compared to FEM results. Yellow-marked entries describe the best-performing parameter set of the backend.

$L - 1$	W	w_{NBC}	σ	$\mathcal{B}$	Training Sequence	e_{rel} in %	Training Time in s
2	120	50	sin	P(A100)	90,000 Adams	1.59	591
2	120	50	sin	P	90,000 Adams	1.81	1502
3	120	50	sin	P	90,000 Adams	0.057	2304
2	120	50	sin	T1	90,000 Adams	0.70	765
3	120	50	sin	T1	90,000 Adams	0.13	1268
2	120	50	sin	T2	90,000 Adams	0.55	825
3	120	50	sin	T2	90,000 Adams	0.39	1263
2	120	50	sin	J	90,000 Adams	0.42	1168
3	120	50	sin	J	90,000 Adams	0.34	2046

Motivated by the improved convergence capabilities, the layer-wise LAAF was tested subsequently. The layer-wise LAAF is a promising approach with additional degrees of freedom and theoretically quick adjustments in learning the function approximation. For the layer-wise LAAF, only a few additional trainable parameters were added to the network (in total parameters at an amount of the number of hidden layers). Overall,

the performance of the $n = 1$ LAAF was superior compared to other n parameters tested (see Table 7). The accuracy was comparable to the one of the finite element method.

Table 6. Error e_{rel} in percentages for the activation function as a hyperparameter variation compared to FEM results. Yellow-marked entries describe the best-performing parameter set in the direction of variation in the line search.

$L - 1$	W	w_{NBC}	σ	$\mathcal{B}$	Training Sequence	e_{rel} in %
2	180	5	sin	P	90,000 Adams	0.77
2	180	5	ELU	P	90,000 Adams	nc
2	180	5	GELU	P	90,000 Adams	1.56
2	180	5	ReLU	P	90,000 Adams	nc
2	180	5	SELU	P	90,000 Adams	nc
2	180	5	sigmoid	P	90,000 Adams	nc
2	180	5	SiLU	P	90,000 Adams	9.77
2	180	5	swish	P	90,000 Adams	6.09
2	180	5	tanh	P	90,000 Adams	2.53

Table 7. Error e_{rel} in percentages for the layer-wise locally adaptive activation function network compared to FEM results. Yellow-marked entries describe the best-performing parameter set in the direction of variation in the line search.

$L - 1$	W	w_{NBC}	σ	$\mathcal{B}$	Training Sequence	e_{rel} in %
2	180	5	sin	P	90,000 Adams	0.77
2	180	5	LAAF-n 1	T1	90,000 Adams	0.086
2	180	5	LAAF-n 2	T1	90,000 Adams	0.370
2	180	5	LAAF-n 4	T1	90,000 Adams	0.120
2	180	5	LAAF-n 8	T1	90,000 Adams	0.096

In conclusion to this experimental hyperparameter study (H1), we found that a layer number of 2 or 3 will be a reasonable choice for the standard network. The layer width between 120 and 240 is preferable, with the sin activation function in conjunction with switching the training from initial Adams iterations of about 100,000 with a boundary weight of 5 and then switching to L-BFGS using a boundary weight of 50. The layer-wise LAAF PINN structure enhanced the accuracy, and we were able to hold H2. Overall, the accuracy compared to the FEM seems good for $s = 1$ m. In the next step, we investigate the potential of adaptive refinement (H3) in terms of the network accuracy and if further data are needed for this application example to improve $s = 1$ m and the cases of $s = \{0.2, 0.1, 0.05, 0.02\}$ m, respectively.

5.2. Adaptive Refinement of Training Set

The pre-trained standard network architecture with hyperparameters $\theta_{\text{NN}} = \{L - 1 = 3, W = 180, w_{\text{NBC}} = 5, \text{sin}, \text{PyTorch}\}$, and a learning rate of 10^{-3} , after 200,000 iterations, is used for the adaptive fine-tuning of the model. The idea that adaptive refinement may improve the results emerged by looking at Figure 6a and the overlaid black dots, which seemed to be distributed nicely but were relatively rare compared to a FEM mesh. The adaptive refinement procedure works in the following steps: in the first data refinement step, $j = 1, \dots, 10$, the domain was sampled uniformly for a new dataset $\mathcal{T}_{\text{samp}, \Omega, j}$. At these data points, the residual of the loss (15) was evaluated, and the arbitrary choice of 150 data points with the most significant residual were added to $\mathcal{T}_{\text{train}, \Omega, j} = \mathcal{T}_{\text{train}, \Omega} \cup (\mathcal{T}_{\text{samp}, \Omega, j} \mid \max_{i=1 \dots 150} (\mathcal{L}(x_i)))$, $\forall x_i \in \mathcal{T}_{\text{samp}, \Omega, j}$. After the uniform refinement step, another 25 data refinement steps $j = 11, \dots, 35$ with random data point sampling were added to the training

data points. In each refinement cycle, 1000 iterations of the optimizer were carried out, with early stopping activated.

The relative error results (see Equation (22)) for the case $s \rightarrow \infty$ present a reduction in the error from 0.35% to 0.33% in the first refinement stage and then another reduction to 0.28% after the whole procedure of adaptive refinement. This reduction of 0.07% points is an improvement of the network's prediction accuracy by 20%. Figure 6a illustrates the pressure field predicted by the PINN in the evaluation plane ($z = 0.5$ m) with a colormap representing the pressure magnitude. Overlaid black dots highlight the training points close to the evaluation plane. Overlaid red dots highlight the training points that the adaptive refinement procedure added and indicate regions where the training error was high before the refinement. Figure 6b shows the pressure field predicted by the PINN compared to the analytic solution on the evaluation line ($y = z = 0.5$ m). Notably, the graphs do coincide qualitatively.

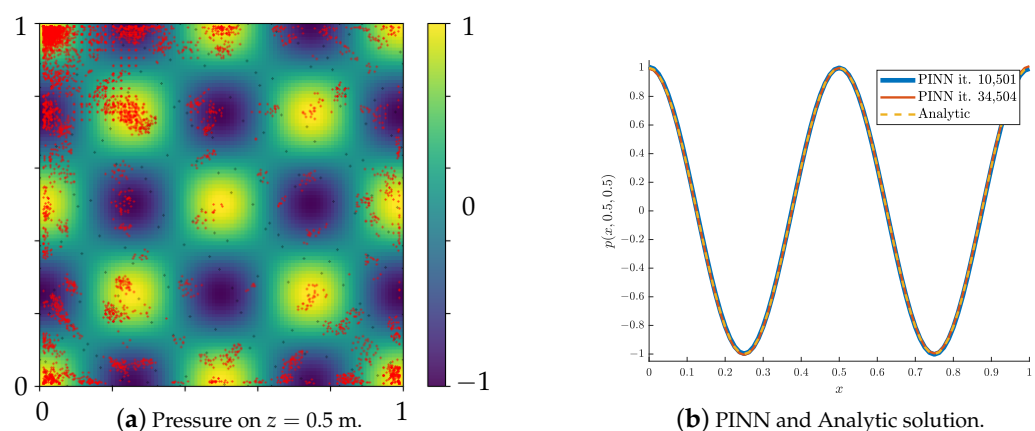


Figure 6. The added points based on adaptive refinement for $s/m \rightarrow \infty$ are highlighted in (a) in red, whereas the initial data points are the tiny black points. (b) shows the results of the acoustic pressure on the line $y = z = 0.5$ m in comparison to the analytic reference solution.

In the case of $s = 1$ m, the relative error results in (22) were reduced from 0.45% to 0.21% in the first refinement stage and then further decrease to 0.18% after the whole procedure of adaptive refinement. Overall, the improvement was 60% with an improvement of 53.3% after the first refinement stage.

Figure 7 summarizes the adaptive refinement procedure and the added data points to the training set. Notably, more training points were added in the central areas of the plot compared to the case of $s \rightarrow \infty$. Interestingly, in Figure 7b, the pressure results of the PINN are in good agreement with the FEM.

Since the relative errors were significant for the cases of $s < 1$ m [5] (error of nearly 100%), we expect improvements by using the slightly damped material model for the reference FEM case to obtain non-resonant and hence a bounded solution (see Table 1). For this benchmark, the case $s = 0.2$ m is discussed, featuring an error of nearly 4.52%. We expect further improvement to the results from the adaptive refinement procedure. In the adaptive refinement step, the error was reduced to 0.76%, being an acceptable amount of error compared to the FEM results (Table 1).

Due to the adaptive refinement, the point distribution showed a remarkably uniform behavior, and the loss is significant in a large area (see Figure 8). Furthermore, a strong cluster of points close to the source location at $x = y = z = 0.5$ m was visible and expected. Overall, the FEM result was captured by the neural network solution, which is illustrated in Figure 8. The findings suggest that the network's capabilities to solve the forward problem in 3D room acoustics is feasible. In summary to H3, adaptive refinement improves the

PINN prediction's accuracy while the overall training time increases significantly with the number of adaptive refinement iteration cycles.

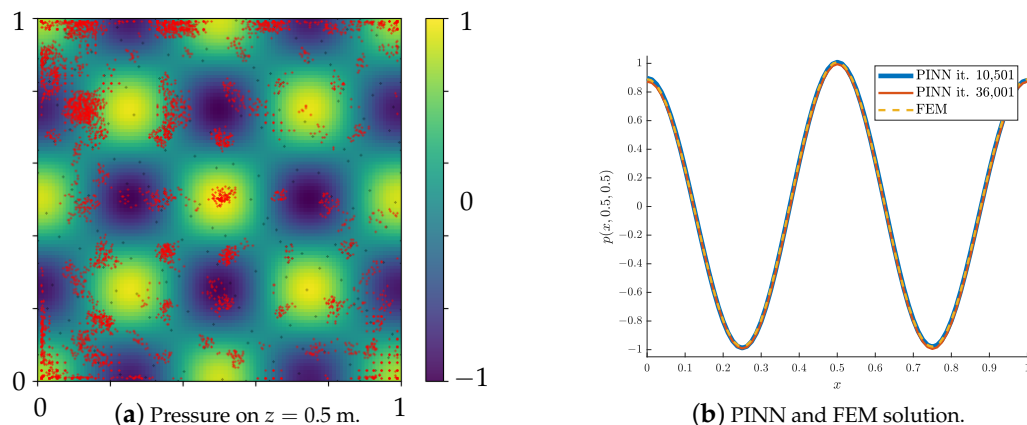


Figure 7. The added points based on adaptive refinement for $s = 1$ m are highlighted in (a) in red, whereas the initial data points are the tiny black points. (b) shows the results of the acoustic pressure on the line $y = z = 0.5$ m compared to the FEM reference solution.

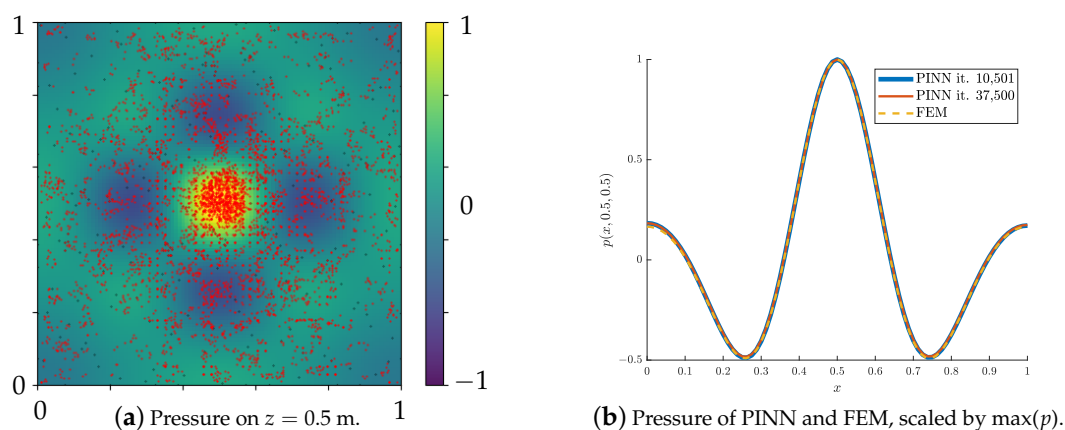


Figure 8. The added points based on adaptive refinement for $s = 0.2$ m are highlighted in (a) in red, whereas the initial data points are the tiny black points. (b) shows the results of the acoustic pressure on the line $y = z = 0.5$ m compared to the FEM reference solution.

5.3. Multi-Scale Fourier Feature Networks

The prior knowledge of spatial scales in the system is strongly driven by the behavior of the wave and the emissions coinciding with the wavelength λ . Furthermore, it is expected that based on the sharpening of the force, the spatial frequency content gets more dispersed over the wave number and can lead to approximation difficulties using a modal field [52]. This was formally shown in Section 2.2. We selected an architecture with the hyperparameters $\theta_{\text{NN}} = \{L - 1 = 2, W = 80, w_{\text{NBC}} = 5, \sin, \text{TensorFlow 1}\}$ and a learning rate of 10^{-3} , and after 200,000 iterations, the errors were evaluated. The Fourier features were selected from a normal distribution with a standard deviation of $\hat{s}_1 = 0.5$ m and $\hat{s}_1 = 1.0$ m, which is in the range of the wavelength based on the selected wave number. In the case of $s \rightarrow \infty$, the relative error (22) with 1.4% was in the range of the standard feed-forward network with the same number of trainable parameters. This was also observed for the case $s = 1$ m with an error of 1.93%. We observed no convergence for all the repeated test runs for $s < 1$ m, which was unexpected. In a second test, the activation function of the network neurons was altered to tanh to avoid the possible poor performance of a

combination of Fourier feature inputs and sinus activation functions. Similarly, the relative error measure was 1.4%, 2.3%, and no convergence for $s \rightarrow \infty$, for $s = 1$ m, and $s = 0.2$ m, respectively. The simulation time was evenly distributed for all tests, around 2500 s.

Overall, the multi-scale Fourier feature networks performed similarly to the standard model, and the means of H4 did not improve prediction accuracy. A possible explanation is based on the exact linearization of the input–output relation due to the wave’s dispersion relation [29]. In the precise form, the cosine and sine transformations for the s -dependent FEM results are illustrated in Figure 9.

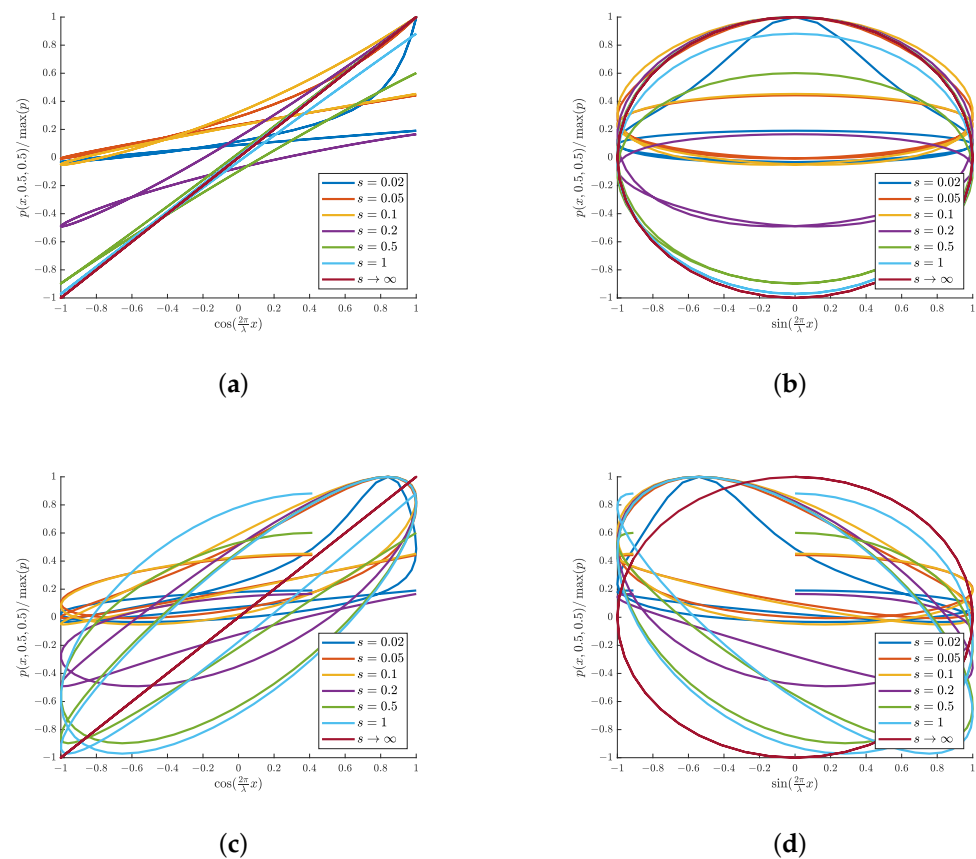


Figure 9. Input feature generation depending on the wavelength and the results for various source sharpness parameter s . (a) FEM pressure results depending on the $\cos(\frac{2\pi}{\lambda}x)$ coordinate at $y = z = 0.5$ m. (b) FEM pressure results depending on the $\sin(\frac{2\pi}{\lambda}x)$ coordinate at $y = z = 0.5$ m. (c) FEM pressure results depending on the $\cos(\frac{2\pi}{1.1\lambda}x)$ coordinate at $y = z = 0.5$ m. (d) FEM pressure results depending on the $\sin(\frac{2\pi}{1.1\lambda}x)$ coordinate at $y = z = 0.5$ m.

The cosine transformation in Figure 9a analytically fulfills the boundary condition of sound–hard walls and yields approximately linear slopes. These linear and nearly linear behaviors can be predicted easily by the neural network. However, for a slightly shifted wavelength (by 10%), one can identify that the cosine and sine both transform into ellipses, making it comparably hard to solve the wave equation and estimate the solution. As soon as there is a deviation in the wave number, inherently present in the multi-scale network by the random draw of the Fourier features, it is fundamentally hard for the network architecture to find the solution of the wave equation. These facts motivate us to form hypothesis 5 (H5) based on the given angular frequency ω , wave number k , and wavelength λ .

H5. Deterministic Fourier features as the input [54,55] that satisfy the dispersion relation of the wave will improve the prediction accuracy and the performance of the PINN.

5.4. Input Feature Generation

The ordinary input x and additionally the transformed input are fed into the neural network. We select $\lambda = 0.5$ m so that the underlying application's acoustic wave number frequency relation is respected. The transform is motivated by Green's function of the waves in a box. It can be modeled by a complete orthonormal solution basis as a product of solutions for each Cartesian coordinate

$$\psi(x) = \prod_{x_i \in x, y, z} \cos\left(\frac{2\pi}{\lambda_i} x_i\right). \quad (23)$$

It is important to note that an additive ansatz would fail and not reduce the complexity of the PINN problem. We selected an architecture with hyperparameters $\theta_{\text{NN}} = \{L - 1 = 2, W = 10, w_{\text{NBC}} = 5, \sin, \text{TensorFlow 1}\}$ and a learning rate of 10^{-3} , and, after 20,000 iterations, the errors were evaluated. In doing so, the following errors were reached: $2 \times 10^{-4}\%$, 0.20%, 17%, and 45%, $s \rightarrow \infty$, $s = 1$ m, $s = 0.2$ m, and $s = 0.02$ m, respectively. This significantly improves the other network structures by ingraining expert knowledge. For $s \rightarrow \infty$, the errors are in the range of the FEM method, and training took a few seconds, which is even faster than the FEM solution. The execution time is in the range reported in [11] for the 2D simulations without fixed boundary conditions (Quadro RTX 8000 Nvidia (Santa Clara, CA, USA)). We obtained the convergence of the amplitudes in the case of $s \rightarrow \infty$ and $s = 1$ m. Discrepancies in the amplitudes and shapes for the cases of $s = 0.2$ m and $s = 0.02$ m are given. In Figure 10, the pressure prediction of the PINN compared to the analytic or FEM results is illustrated for $s \rightarrow \infty$, $s = 1$ m, $s = 0.2$ m, and $s = 0.02$ m. In the case of $s = 1$ m, the pressure agrees well with the FEM results. For $s = 0.2$ m, additional features with wave numbers of $0.5k_0$, $1.5k_0$, $2k_0$, and $2.5k_0$ are necessary to obtain convergence as illustrated in Figure 10c. The graph of the pressure starts to deviate at the domain's boundaries from the FEM solution. In the case of $s = 0.02$ m, the pressure signal does not adapt to the reference FEM solution.

As a result of the fast training speed (below a minute and 20 times faster than the FEM simulation) in the feature-engineered neural network architecture, an automated hyperparameter optimization using a Gaussian processes-based Bayesian optimization [11] was performed in scikit-learn 1.5.2. Five hundred automated hyperparameter model iterations were performed. The learning rate, the network depth, the network width, the activation function, and the boundary condition loss weight were investigated. In doing so for $s = 1$ m, the relative error was reduced from 0.2% to 0.1938%. In contrast to the claimed strong improvements by automated hyperparameter optimization in the loss reduction [11], only a slight improvement in the accuracy of the model was obtained (see Table 8). The fastest model took 50 s to train and had an error of 0.204%. Regarding hypothesis H5, it was confirmed that the feature-engineered network outperformed the other architectures presented in that article for $s > 1$.

Table 8. Summary of the relative errors and network parameters of the automated hyperparameter optimization with $s = 1$.

	Overall Best	Best Sin	Best Sigmoid	Fastest
e_{rel} in %	0.1938	0.195	0.21	0.204
Training time in s	621	404	89	50
Learning rate	0.0009	0.00077	0.025	0.001
$L - 1$	9	3	10	2
W	80	80	12	35
σ	tanh	sin	sigmoid	sin
w_{NBC}	1	0.1	4.7	5

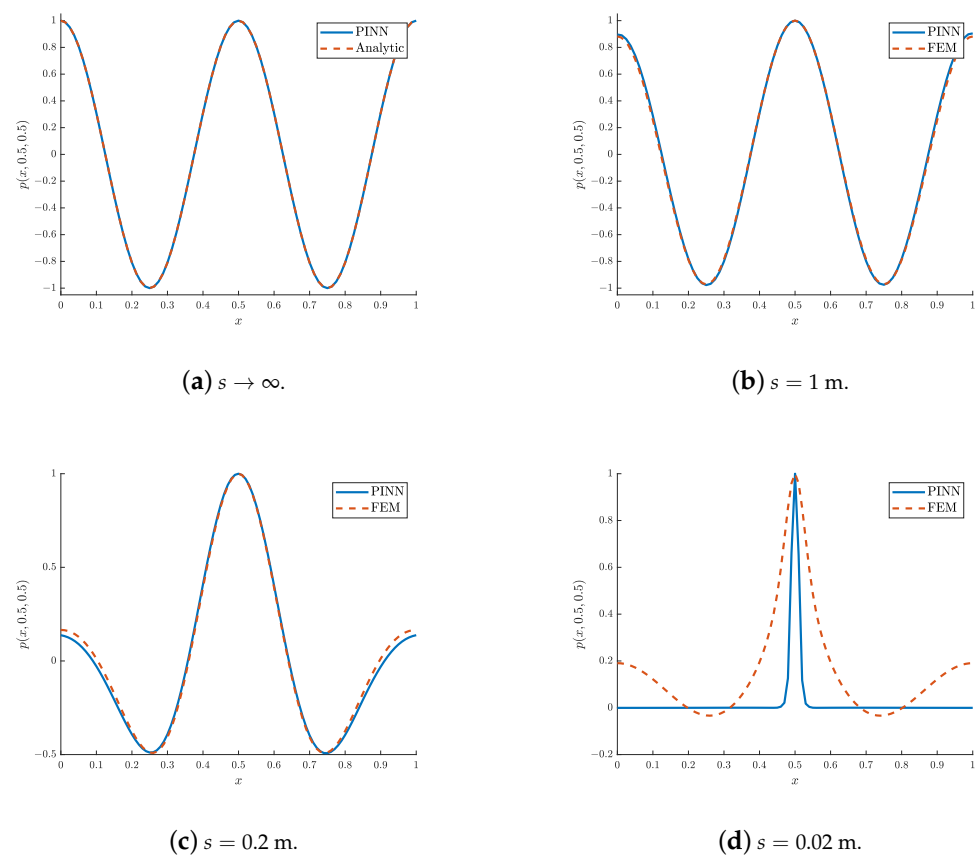


Figure 10. Pressure results normalized by the maximum pressure of the reference (FEM, analytic) depending on the x coordinate at a line with $y = z = 0.5$ m inside the domain.

6. Conclusions

This study demonstrates the feasibility of using PINNs to model 3D sound fields in a room with sound-hard walls governed by the inhomogeneous Helmholtz equation. The prediction accuracy was assessed by a validated reference FEM simulation in openCFS, which took, on average, 15 min and 30 s to compute on a standard notebook CPU. Three different network architectures and feature engineering were used to improve the pressure field prediction. The most significant improvement was reached by ingraining the characteristics of the Green's function based on modal expansion via feature engineering as additional input to the PINN. An explorative hyperparameter study provides intuition about the behavior of the standard PINN training concerning prediction accuracy. The network depth and width, activation functions, training routine, fine-tuning with different optimizers, backends, and learning rate adjustment improved the prediction capability for a simple, fully connected neural network architecture from an error of 0.45% to 0.019% ($s = 1$ m). The backend TensorFlow 1 on the Tesla T4 GPU with Cuda 12.2 and Python 3.10 had the best performance. The layer-wise local adaptive activation function network improved the prediction accuracy of the tested case. However, the neuron-wise algorithm, introducing more additional degrees of freedom, might further improve the capabilities.

Residual-based adaptive refinement improved the prediction accuracy. The refinement workflow connected both structured and randomly unstructured refinement. In each stage, 150 data points were added, adding about 1.5% additional data points. The convergence to the accurate solution reached relative errors as low as 0.28% for distributed sources from the initial 0.35%. In the case of $s = 1$ m, the relative error improved by 60%, using adaptive

refinement. In the case of $s = 0.2$ m, the relative error was reduced drastically by adaptive refinement being similar than the error computed by the mesh study of the FEM results.

Using random Fourier features in the multi-scale Fourier feature network did not yield a higher accuracy. Studying the transformation behavior of the input by random Fourier features revealed visibly that the transformation does not reduce the complexity of the prediction. These visualizations led to the idea that using dispersion relation-based Fourier features may significantly enhance the prediction capabilities and the generalization of the neural network architecture concerning alternative source distributions. This feature-engineered architecture substantially reduces the number of trainable parameters that reach 5% and the training time on the GPU to less than a minute. This fast prediction outperforms the FEM analysis. Also, the accuracy is improved by three orders of magnitude for $s \rightarrow \infty$, ($10^{-3}\%$). However, this is still two orders of magnitude less accurate than the pressure result of the FEM solution ($10^{-5}\%$). This work underscores the potential of PINNs as memory-efficient alternatives to traditional numerical methods for low-frequency room acoustic simulations. For the case of $s = 1$ m, the errors were reduced from initially 0.20% to 0.19% by a Gaussian process-based Bayesian hyperparameter optimization.

This article presented significant progress regarding the generalization of PINNs for confined sources. Challenges regarding the explainability of the network learning capabilities [53] and the generalization to multi-source and different room geometries remain. These challenges will be subject to a follow-up study. Transitional behavior for very localized sources will be a focus of upcoming work, including explainability.

Funding: This scientific work was significantly supported by the COMET project ECHODA (Energy Efficient Cooling and Heating of Domestic Appliances). ECHODA is funded within the framework of COMET—Competence Centers for Excellent Technologies—by BMK, BMAW, the province of Styria as well as SFG. The COMET programme is managed by FFG. Supported by TU Graz Open Access Publishing Fund.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The simulation data presented in this study are available on request from the author.

Acknowledgments: The computational results presented have been achieved in part using the Vienna Scientific Cluster (VSC). The author thanks F. Kraxberger for his help in the study published previously to this article. Open Access Funding by the Graz University of Technology.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Jeong, C.H. Room acoustic simulation and virtual reality—Technological trends, challenges, and opportunities. *J. Swed. Acoust. Soc. (Ljudbladet)* **2022**, *1*, 27–30.
2. Vorländer, M. Computer simulations in room acoustics: Concepts and uncertainties. *J. Acoust. Soc. Am.* **2013**, *133*, 1203–1213. [[CrossRef](#)] [[PubMed](#)]
3. Schmid, J.D.; Bauerschmidt, P.; Gurbuz, C.; Eser, M.; Marburg, S. Physics-informed neural networks for acoustic boundary admittance estimation. *Mech. Syst. Signal Process.* **2024**, *215*, 111405. [[CrossRef](#)]
4. Kraxberger, F.; Kurz, E.; Weselak, W.; Kubin, G.; Kaltenbacher, M.; Schoder, S. A Validated Finite Element Model for Room Acoustic Treatments with Edge Absorbers. *Acta Acust.* **2023**, *7*, 48. [[CrossRef](#)]
5. Schoder, S.; Kraxberger, F. Feasibility study on solving the Helmholtz equation in 3D with PINNs. *arXiv* **2024**, arXiv:2403.06623.
6. Lu, L.; Pestourie, R.; Yao, W.; Wang, Z.; Verdugo, F.; Johnson, S.G. Physics-informed neural networks with hard constraints for inverse design. *SIAM J. Sci. Comput.* **2021**, *43*, B1105–B1132. [[CrossRef](#)]
7. Chen, Y.; Lu, L.; Karniadakis, G.E.; Dal Negro, L. Physics-informed neural networks for inverse problems in nano-optics and metamaterials. *Opt. Express* **2020**, *28*, 11618–11633. [[CrossRef](#)]

8. Cuomo, S.; Di Cola, V.S.; Giampaolo, F.; Rozza, G.; Raissi, M.; Piccialli, F. Scientific machine learning through physics-informed neural networks: Where we are and what's next. *J. Sci. Comput.* **2022**, *92*, 88. [\[CrossRef\]](#)
9. Lu, L.; Meng, X.; Mao, Z.; Karniadakis, G.E. DeepXDE: A deep learning library for solving differential equations. *SIAM Rev.* **2021**, *63*, 208–228. [\[CrossRef\]](#)
10. Song, C.; Alkhalifah, T.; Waheed, U.B. Solving the frequency-domain acoustic VTI wave equation using physics-informed neural networks. *Geophys. J. Int.* **2021**, *225*, 846–859. [\[CrossRef\]](#)
11. Escapil-Inchauspé, P.; Ruz, G.A. Hyper-parameter tuning of physics-informed neural networks: Application to Helmholtz problems. *Neurocomputing* **2023**, *561*, 126826. [\[CrossRef\]](#)
12. Gladstone, R.J.; Nabian, M.A.; Meidani, H. FO-PINNs: A First-Order formulation for Physics Informed Neural Networks. *arXiv* **2022**, arXiv:2210.14320. [\[CrossRef\]](#)
13. Wu, Y.; Aghamiry, H.S.; Operto, S.; Ma, J. Helmholtz-equation solution in nonsmooth media by a physics-informed neural network incorporating quadratic terms and a perfectly matching layer condition. *Geophysics* **2023**, *88*, T185–T202. [\[CrossRef\]](#)
14. Jagtap, A.D.; Shin, Y.; Kawaguchi, K.; Karniadakis, G.E. Deep Kronecker neural networks: A general framework for neural networks with adaptive activation functions. *Neurocomputing* **2022**, *468*, 165–180. [\[CrossRef\]](#)
15. Rasht-Behesht, M.; Huber, C.; Shukla, K.; Karniadakis, G.E. Physics-informed neural networks (PINNs) for wave propagation and full waveform inversions. *J. Geophys. Res. Solid Earth* **2022**, *127*, e2021JB023120. [\[CrossRef\]](#)
16. Shukla, K.; Di Leoni, P.C.; Blackshire, J.; Sparkman, D.; Karniadakis, G.E. Physics-informed neural network for ultrasound nondestructive quantification of surface breaking cracks. *J. Nondestruct. Eval.* **2020**, *39*, 61. [\[CrossRef\]](#)
17. Khan, M.R.; Zekios, C.L.; Bhardwaj, S.; Georgakopoulos, S.V. A Physics-Informed Neural Network-Based Waveguide Eigenanalysis. *IEEE Access* **2024**, *12*, 120777–120787. [\[CrossRef\]](#)
18. Pan, Y.Q.; Wang, R.; Wang, B.Z. Physics-informed neural networks with embedded analytical models: Inverse design of multilayer dielectric-loaded rectangular waveguide devices. *IEEE Trans. Microw. Theory Tech.* **2023**, *72*, 3993–4005. [\[CrossRef\]](#)
19. Schoder, S.; Museljac, E.; Kraxberger, F.; Wurzinger, A. Post-processing subsonic flows using physics-informed neural networks. In Proceedings of the 2023 AIAA Aviation Forum, AIAA, San Diego, CA, USA, 12–16 June 2023; pp. 1–8. [\[CrossRef\]](#)
20. Markidis, S. The old and the new: Can physics-informed deep-learning replace traditional linear solvers? *Front. Big Data* **2021**, *4*, 669097. [\[CrossRef\]](#)
21. Xu, Z.Q.J.; Zhang, Y.; Luo, T.; Xiao, Y.; Ma, Z. Frequency principle: Fourier analysis sheds light on deep neural networks. *arXiv* **2019**, arXiv:1901.06523.
22. Kraxberger, F.; Kurz, E.; Merkel, L.; Kaltenbacher, M.; Schoder, S. Finite Element Simulation of Edge Absorbers for Room Acoustic Applications. In Proceedings of the Fortschritte der Akustik—DAGA 2023, Hamburg, Germany, 6–9 March 2023; pp. 1292–1295.
23. Schoder, S.; Spieser, É.; Vincent, H.; Bogey, C.; Bailly, C. Acoustic modeling using the aeroacoustic wave equation based on Pierce's operator. *AIAA J.* **2023**, *61*, 4008–4017. [\[CrossRef\]](#)
24. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 8024–8035.
25. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. Available online: <https://tensorflow.org> (accessed on 20 October 2024).
26. Bradbury, J.; Frostig, R.; Hawkins, P.; Johnson, M.J.; Leary, C.; Maclaurin, D.; Necula, G.; Paszke, A.; VanderPlas, J.; Wanderman-Milne, S.; et al. JAX: Composable Transformations of Python+NumPy Programs. 2018. Available online: <https://jax.readthedocs.io> (accessed on 15 November 2024).
27. Jagtap, A.D.; Kawaguchi, K.; Em Karniadakis, G. Locally adaptive activation functions with slope recovery for deep and physics-informed neural networks. *Proc. R. Soc. A* **2020**, *476*, 20200334. [\[CrossRef\]](#) [\[PubMed\]](#)
28. Wang, S.; Wang, H.; Perdikaris, P. 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.* **2021**, *384*, 113938. [\[CrossRef\]](#)
29. Pierce, A.D. *Acoustics: An Introduction to Its Physical Principles and Applications*; Springer: Berlin/Heidelberg, Germany, 2019.
30. Schoder, S.; Roppert, K. openCFS: Open Source Finite Element Software for Coupled Field Simulation—Part Acoustics. *arXiv* **2022**, arXiv:2207.04443. [\[CrossRef\]](#)
31. Ainsworth, M. Discrete dispersion relation for hp-version finite element approximation at high wave number. *SIAM J. Numer. Anal.* **2004**, *42*, 553–575. [\[CrossRef\]](#)
32. Kaltenbacher, M. *Numerical Simulation of Mechatronic Sensors and Actuators: Finite Elements for Computational Multiphysics*, 3rd ed.; Springer: Berlin/Heidelberg, Germany, 2015. [\[CrossRef\]](#)
33. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [\[CrossRef\]](#)

34. Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev.* **1958**, *65*, 386. [\[CrossRef\]](#)
35. Haykin, S. *Neural Networks: A Comprehensive Foundation*; Prentice Hall PTR: Hoboken, NJ, USA, 1998.
36. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
37. Liu, D.C.; Nocedal, J. On the limited memory BFGS method for large scale optimization. *Math. Program.* **1989**, *45*, 503–528. [\[CrossRef\]](#)
38. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [\[CrossRef\]](#)
39. Baydin, A.G.; Pearlmutter, B.A.; Radul, A.A.; Siskind, J.M. Automatic differentiation in machine learning: A survey. *J. Mach. Learn. Res.* **2018**, *18*, 1–43.
40. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010; Proceedings of Machine Learning Research; Teh, Y.W., Titterton, M., Eds.; Volume 9, pp. 249–256.
41. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [\[CrossRef\]](#)
42. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv* **2017**, arXiv:1708.07747.
43. Clanuwat, T.; Bober-Irizar, M.; Kitamoto, A.; Lamb, A.; Yamamoto, K.; Ha, D. Deep learning for classical japanese literature. *arXiv* **2018** arXiv:1812.01718.
44. Srl, B.T.; Brescia, I. *Semeion Handwritten Digit Data Set*; Semeion Research Center of Sciences of Communication: Rome, Italy, 1994.
45. Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A.Y. Reading digits in natural images with unsupervised feature learning. In Proceedings of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning, Granada, Spain, 12–14 December 2011; Volume 2011, p. 4.
46. Krizhevsky, A.; Hinton, G. Learning Multiple Layers of Features from Tiny Images. 2009. Available online: <https://www.semanticscholar.org/paper/Learning-Multiple-Layers-of-Features-from-Tiny-Krizhevsky/5d90f06bb70a0a3dced62413346235c02b1aa086> (accessed on 15 November 2024).
47. Cao, Y.; Fang, Z.; Wu, Y.; Zhou, D.X.; Gu, Q. Towards understanding the spectral bias of deep learning. *arXiv* **2019**, arXiv:1912.01198.
48. Rahaman, N.; Baratin, A.; Arpit, D.; Draxler, F.; Lin, M.; Hamprecht, F.; Bengio, Y.; Courville, A. On the spectral bias of neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; PMLR: Cambridge MA, USA, 2019; pp. 5301–5310.
49. Ronen, B.; Jacobs, D.; Kasten, Y.; Kritchman, S. The convergence rate of neural networks for learned functions of different frequencies. *Adv. Neural Inf. Process. Syst.* **2019**, *32*.
50. Rucz, P.; Ulveczki, M.Á.; Heinz, J.; Schoder, S. Analysis of the co-rotating vortex pair as a test case for computational aeroacoustics. *J. Sound Vib.* **2024**, *587*, 118496. [\[CrossRef\]](#)
51. Wurzinger, A.; Schoder, S. pyCFS-data: Data Processing Framework in Python for openCFS. *arXiv* **2024**, arXiv:2405.03437.
52. Allen, J.B.; Berkley, D.A. Image method for efficiently simulating small-room acoustics. *J. Acoust. Soc. Am.* **1979**, *65*, 943–950. [\[CrossRef\]](#)
53. Krishnapriyan, A.; Gholami, A.; Zhe, S.; Kirby, R.; Mahoney, M.W. Characterizing possible failure modes in physics-informed neural networks. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 26548–26560.
54. Zeng, C.; Burghardt, T.; Gambaruto, A.M. Training dynamics in Physics-Informed Neural Networks with feature mapping. *arXiv* **2024**, arXiv:2402.06955.
55. Sallam, O.; Fürth, M. On the use of Fourier Features-Physics Informed Neural Networks (FF-PINN) for forward and inverse fluid mechanics problems. *Proc. Inst. Mech. Eng. Part M J. Eng. Marit. Environ.* **2023**, *237*, 846–866. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.