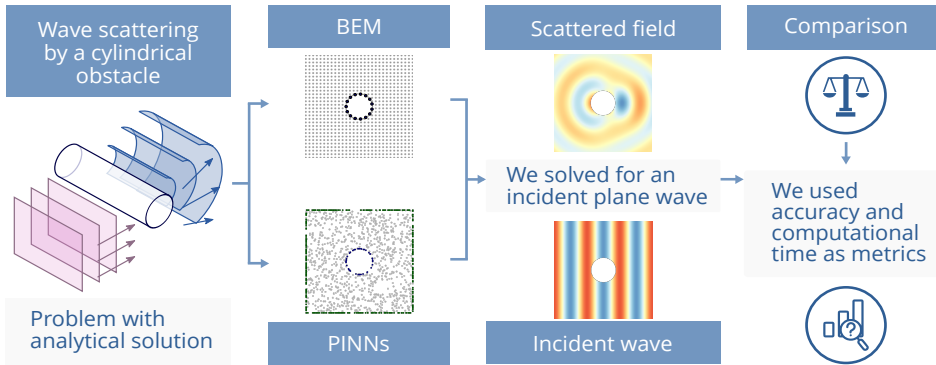


# Graphical Abstract

## Benchmarking Physics-Informed Neural Networks and Boundary Element Method: Accuracy–Efficiency Trade-offs in Wave Scattering

O. Rincón-Cardeno, G. Pérez-Bernal, S. Montoya-Noguera, N. Guarín-Zapata



# Benchmarking Physics-Informed Neural Networks and Boundary Element Method: Accuracy–Efficiency Trade-offs in Wave Scattering

O. Rincón-Cardeno<sup>a</sup>, G. Pérez-Bernal<sup>a</sup>, S. Montoya-Noguera<sup>b</sup> and N. Guarín-Zapata<sup>a,\*</sup>

<sup>a</sup>Mathematical Applications in Science and Engineering Research Group, School of Applied Sciences and Engineering, Universidad EAFIT, Medellín, Colombia

<sup>b</sup>Nature and Cities Research Group, School of Applied Sciences and Engineering, Universidad EAFIT, Medellín, Colombia

## ARTICLE INFO

### Keywords:

Physics-informed neural networks  
Scientific machine learning  
Boundary Element Method  
Numerical benchmarking  
Computational physics

## ABSTRACT

Physics-informed neural networks (PINNs) offer a flexible alternative to classical numerical solvers for partial differential equations, but their computational trade-offs remain incompletely quantified. We present a controlled benchmark of PINNs and the Boundary Element Method (BEM) for two-dimensional wave scattering problems governed by the Helmholtz equation, applying both approaches to the same problem under identical conditions. BEM solutions are obtained through boundary discretization and system assembly, while PINNs are trained by minimizing residuals of the governing equation and boundary conditions, with network hyperparameters selected through optimization. At comparable accuracy levels, BEM assembly and solution require on the order of  $10^{-2}$  s, whereas PINNs training incurs costs on the order of  $10^2$  s. Once trained, PINNs enable rapid field evaluation at interior points, with inference times comparable to or faster than BEM. These results establish a reproducible benchmarking procedure, providing quantitative guidance for comparing standard and neural network-based methods.

## Introduction

Partial differential equations (PDEs) are central to mathematical modeling and are used to represent physical processes across a wide range of scientific and engineering disciplines. In particular, wave scattering problems arise in a broad range of fields, including acoustics, electromagnetism, geophysics, and medical imaging. Numerical methods such as the Finite Element Method (FEM) and the Boundary Element Method (BEM) are commonly used to model these problems, as they provide accurate approximations of the solution [3, 5]. However, their computational cost increases significantly for complex geometries or high-frequency regimes, where fine spatial discretization is required to preserve accuracy.


These limitations have motivated the exploration of alternative approaches, including machine learning-based methods, which aim to reduce the need for fine discretization and scale more efficiently with the dimensionality of the problem [19]. Machine learning has increasingly been applied to the solution of forward problems governed by partial differential equations. One popular approach is Physics-Informed Neural Networks (PINNs) that incorporate the governing equations and available data into the training process [13, 6, 22]. PINNs have been used to approximate solutions of PDEs such as the Helmholtz equation [7, 18]. However, their performance is often assessed in isolation, without systematic comparison to established numerical methods or using limited and potentially biased benchmarks.

A systematic review in fluid mechanics reports that 79% of machine learning solvers claiming superior performance over traditional methods do not use adequate baselines [15].

Comparisons must be designed to avoid systematically underestimating or overestimating the performance of either approach. Results with these biases can be obtained by comparing solution times across methods with different accuracies, or by benchmarking against numerical methods that are not optimal for the specific problem. Therefore, the use of a strong, problem-appropriate baseline is essential to assess the true capabilities and limitations of machine learning methods for solving PDEs. For scattering problems governed by the Helmholtz equation, the BEM provides a particularly appropriate reference. In contrast to domain-based methods such as FEM, BEM enforces the radiation condition at infinity and requires discretization only on the boundary of the scatterer. This reduces the dimensionality of the problem and eliminates the need for artificial boundary truncation. For these reasons, BEM is an appropriate baseline for evaluating the performance of PINNs in the case of scattering problems.

An important characteristic of machine learning methods such as PINNs is their potential to approximate solutions beyond the stabilized training domain [4]. This allows the model to estimate the solution at locations different from the original training points. However, unlike BEM, PINNs do not directly enforce the Sommerfeld radiation condition, which can result in inaccuracies in the far field, such as incorrect decay of the scattered wave at large distances. To address this issue, domain truncation techniques such as Perfectly Matched Layers (PML) or Absorbing Boundary Conditions (ABC) are typically employed [2, 3, 12]. Therefore, a direct

\*Corresponding author

 nguarinz@eafit.edu.co (N. Guarín-Zapata)

ORCID(s): 0000-0002-5308-9710 (O. Rincón-Cardeno);

0009-0009-8586-7992 (G. Pérez-Bernal); 0000-0003-2490-9107 (S.

Montoya-Noguera); 0000-0002-9435-1914 (N. Guarín-Zapata)

comparison between methods is necessary to evaluate their performance and to understand how each approach handles the physical constraints associated with scattering problems.

Our study is framed within the broader question of: How can machine learning–based approaches for solving partial differential equations be compared with standard numerical methods? In this context, we are particularly interested in addressing the following research question: How do PINNs and the BEM compare in terms of accuracy and computational efficiency for wave scattering problem? In this work, we focus on this case, which serves as a canonical test in computational physics. While the numerical experiments are formulated for this specific problem, we propose a methodological framework for comparison that applies to other classes of partial differential equations and to both numerical and machine learning–based solvers. Addressing this question requires benchmarks that jointly account for accuracy, computational cost, and generalization behavior. Consistent evaluation criteria enable the identification of the conditions under which each method is better suited to a given problem.

In this work, we present a reproducible benchmarking procedure between BEM and PINNs. The objective of the comparison is to determine the range of conditions where each method performs best and to outline their respective advantages and limitations. Both approaches were applied to the problem of plane wave scattering by an obstacle. Particularly for PINNs, we adopt the original formulation introduced by Raissi et al. [22], as a baseline for comparison. We also implemented a hyperparameter optimization strategy to the Helmholtz equation in wave scattering problems. The behavior of each method was evaluated both within and outside the training domain, with an emphasis on the asymptotic decay of the scattered field. With this work, we aim to provide tools for comparative analysis in the field. By explicitly quantifying accuracy–efficiency trade-offs, this study provides practical guidance for the use scientific machine learning methods alongside classical numerical techniques.

## Wave Scattering and the Helmholtz Equation

We consider the classical problem of acoustic wave scattering by a single circular obstacle under a time-harmonic incident wave [16]. The problem involves solving the Helmholtz equation under boundary conditions that represent the interaction between the incident wave and the obstacle. The scattered wave field depends on the geometry of the obstacle and the properties of the surrounding medium, which in this case is unbounded. We illustrate this phenomenon in the case of a long cylinder in Figure 1 (left). In numerical methods based on spatial discretization, the infinite domain is typically approximated by truncating the domain at a finite distance from the obstacle and imposing special boundary conditions to minimize artificial reflections. We begin by considering the Helmholtz equation:

$$\Delta u(\mathbf{x}) + k^2 u(\mathbf{x}) = 0, \quad \mathbf{x} \in \Omega_p, \quad (1)$$

where  $\mathbf{x} \in \mathbb{R}^2$  is the spatial coordinate,  $u(\mathbf{x})$  is the unknown field,  $\Delta$  is the Laplacian operator,  $k = \frac{\omega}{c}$  is the wavenumber,  $\omega$  is the angular frequency,  $c$  is the speed of sound in the medium, and  $\Omega_p$  is the physical domain where the wave propagates. It is important to note that the Helmholtz equation models a time-harmonic regime. The formulation assumes a steady-state oscillation at a fixed frequency  $\omega$ , and the entire problem is posed in the frequency domain. Figure 1 (right) shows the representation of the scattering problem considering the domain truncation. The field is obtained by considering the incident and the scattered fields due to the interaction with the obstacle. Therefore, we write it as:

$$u(\mathbf{x}) = u_{\text{inc}}(\mathbf{x}) + u_{\text{sct}}(\mathbf{x}).$$

The incident wave is  $u_{\text{inc}}$ , which is typically a known function describing the incoming acoustic field. In our case, it is a plane wave that we can write in polar coordinates as:

$$u_{\text{inc}}(r, \theta) = A e^{ikr \cos \theta}, \quad (2)$$

where  $A$  is the amplitude and  $k$  is the wavevector. Using the Jacobi-Anger expansion, the variables of the right term can be separated and expressed in Equation 2 as:

$$A e^{ikr \cos \theta} = A \sum_{n=-\infty}^{\infty} i^n J_n(r) e^{in\theta}. \quad (3)$$

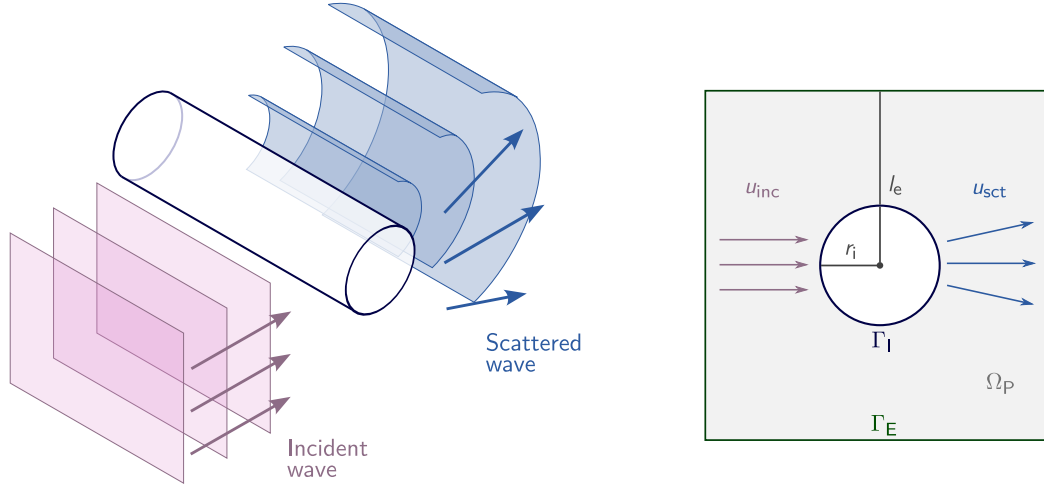
Here,  $J_n$  denotes the Bessel function of the first kind of order  $n$ . The scattered wave  $u_{\text{sct}}$  satisfies the Helmholtz equation in the exterior of the obstacle and is subject to a boundary condition at the obstacle surface. The specific form of the boundary condition on the surface (at  $r = r_i$ ) depends on the type of obstacle. For a sound-hard obstacle, the appropriate condition is of Neumann type:

$$\nabla u(\mathbf{x}) \cdot \mathbf{n} = 0, \quad \mathbf{x} \in \Gamma_I. \quad (4)$$

In this expression,  $\mathbf{n}$  denotes the outward unit normal to the boundary  $\Gamma_I$ . To ensure a well-posed problem in an unbounded domain, the solution must also satisfy the Sommerfeld radiation condition. In the case of a plane problem it reads [24]:

$$\lim_{r \rightarrow \infty} \left( r^{1/2} \left( \frac{\partial u(r, \theta)}{\partial r} - iku(r, \theta) \right) \right) = 0, \quad (5)$$

where  $r$  is the radial distance from the center and  $u(r, \theta)$  is the solution to the Helmholtz equation. This condition is mathematically required to ensure the uniqueness of the solution, and thus the well-posedness of the problem. In the case of domain-based methods, the radiation condition is imposed numerically by truncating the domain with an artificial boundary  $\Gamma_E$  and applying an ABC or PML to approximate outgoing waves. The total acoustic field  $u(\mathbf{x})$  must satisfy the



**Figure 1:** Schematic representations of acoustic wave scattering by an obstacle. (left) Three-dimensional illustration of an incident plane wave propagating toward a circular obstacle and generating a scattered field upon interaction with the interface. (right) Two-dimensional computational model, where the scattered field  $u_{\text{sct}}$  is generated by an incident wave  $u_{\text{inc}}$  in the domain  $\Omega_p$ , bounded internally by a circular interface  $\Gamma_I$  of radius  $r_i$  and externally by a square boundary  $\Gamma_E$  of semi-length  $l_e$ . The scattered field is evaluated within the computational domain  $\Omega_p$ .

Helmholtz equation (equation (1)) in the physical domain  $\Omega_p$ , the Neumann boundary condition on the surface of the obstacle  $\Gamma_I$  (equation (4)), and the Sommerfeld radiation condition as  $r \rightarrow \infty$  (equation (5)). Together, these three conditions fully define the mathematical formulation of the acoustic scattering problem in an unbounded domain.

## Methods

This section presents two distinct numerical approaches for solving the exterior Helmholtz problem: the BEM and PINNs. We provide implementation details and discuss numerical strategies for both methods. For PINNs, we detail the procedure used for hyperparameter optimization in order to define a suitable network configuration. Finally, we outline the criteria used to compare the performance of both approaches in terms of accuracy and computational cost. All code used in this study is available in the public repository at [github.com/oscar-rincon/comparative-bem-pinns](https://github.com/oscar-rincon/comparative-bem-pinns).

### Boundary Element Method Solution

To solve this problem using the BEM, it is necessary to formulate the integral equation for the exterior problem and to establish the boundary conditions for a circular obstacle under a plane incident wave [9, 10]. The following system of integral equations on the boundary is formulated:

$$\frac{1}{2}u^i + \sum_{j=1}^N \hat{H}_{ij} u^j = \sum_{j=1}^N G_{ij} u_n^j,$$

where  $u^j$  denotes the boundary solution at node  $j$ , and  $u_n^j$  its normal derivative with respect to the outward normal direction. The influence coefficients  $\hat{H}_{ij}$  and  $G_{ij}$  are defined

as

$$\hat{H}_{ij} = \int_{\Gamma_j} \frac{\partial v(p_i, q)}{\partial n_q} ds$$

and

$$G_{ij} = \int_{\Gamma_j} v(p_i, q) ds.$$

Here,  $p_i$  is the  $i$ -th collocation point on the boundary, and  $q$  is an integration point along the boundary element  $\Gamma_j$ . The  $v(p_i, q)$  being the Green's function for the 2D Helmholtz equation:

$$v = \frac{i}{4} H_0^{(1)}(kr),$$

and

$$\frac{\partial v}{\partial n_q} = -\frac{ik}{4} H_1^{(1)}(kr) \cos \phi.$$

where  $r = \|p_i - q\|$  is the distance between the collocation point and the integration point, and  $\phi$  is the angle between the outward normal vector at  $q$  and the vector connecting  $q$  to  $p_i$ . The solution at evaluation points is computed as:

$$\frac{1}{2}u^i_{\text{pred}} = \sum_{j=1}^N G_{ij} u_n^j - \sum_{j=1}^N \hat{H}_{ij} u^j.$$

The influence coefficients are estimated numerically for  $i \neq j$ , while for the case  $i = j$ , analytical corrections are applied due to the singularity of the kernel. We consider:

$$G_{jj} = \left( \int_{\Gamma_j} \frac{i}{4} H_0^{(1)}(kr) ds - \int_{\Gamma_j} \frac{1}{2\pi} \ln r ds \right) + \int_{\Gamma_j} \frac{1}{2\pi} \ln r ds, \quad \text{and} \quad H_{jj} = \frac{1}{2}.$$

To approximate the boundary integrals in the computation of the influence coefficients  $G_{ij}$  and  $\hat{H}_{ij}$ , we employ Gaussian quadrature over each boundary element  $\Gamma_j$ . Each element is mapped to a reference interval,  $[-1, 1]$ , and the integral is computed as a weighted sum of the integrand evaluated at specific quadrature points:

$$\int_{\Gamma_j} f(s) ds \approx \sum_{m=1}^{n_q} w_m f(s(\xi_m)) |J(\xi_m)|,$$

where  $\xi_m$  and  $w_m$  are the quadrature points and weights, respectively, and  $J(\xi)$  is the Jacobian of the transformation from the reference element to the physical coordinates. In this work, a Gaussian quadrature rule with  $n_q = 8$  points is employed for numerical integration.

First, the boundary was discretized into nodes and elements. Then, the solution was computed in the domain of interest, which consists of a uniform grid of evaluation points.

### Physics-Informed Neural Networks Solution

To solve the boundary value problem, we use PINNs to approximate the scattered field  $u_{\text{sct}}$ . Since the domain  $\Omega_P \subset \mathbb{R}^2$  is unbounded, an artificial boundary  $\Gamma_E$  is introduced to truncate the computational domain. On this boundary, ABC is imposed to approximate the Sommerfeld radiation condition (equation (5)):

$$\nabla u_{\text{sct}}(\mathbf{x}) \cdot \mathbf{n} + iku_{\text{sct}}(\mathbf{x}) = 0, \quad \mathbf{x} \in \Gamma_E.$$

The full problem is given by the system:

$$\Delta u_{\text{sct}}(\mathbf{x}) + k^2 u_{\text{sct}}(\mathbf{x}) = 0, \quad \mathbf{x} \in \Omega_P, \quad (6)$$

$$\nabla u_{\text{sct}}(\mathbf{x}) \cdot \mathbf{n} + \nabla u_{\text{inc}}(\mathbf{x}) \cdot \mathbf{n} = 0, \quad \mathbf{x} \in \Gamma_1, \quad (7)$$

$$\nabla u_{\text{sct}}(\mathbf{x}) \cdot \mathbf{n} + iku_{\text{sct}}(\mathbf{x}) = 0, \quad \mathbf{x} \in \Gamma_E. \quad (8)$$

Equations (6)–(8) define the physical constraints used to construct the loss function in the PINNs method. A neural network is constructed to approximate the solution  $u_{\text{sct}} : \mathbb{R}^2 \rightarrow \mathbb{C}$ , using two input variables ( $x, y$ ), and two output variables corresponding to the real and imaginary parts of the scattered wave:  $\Re(u_{\text{sct}}(x, y))$  and  $\Im(u_{\text{sct}}(x, y))$ . The network architecture is a multilayer perceptron implemented in PyTorch [20], with weights initialized using the Glorot uniform initializer to promote stable training. Training is accelerated on a GeForce RTX 4060 GPU.

The loss function  $L_T$  is constructed by enforcing the residual of each governing condition at collocation points distributed throughout the domain and its boundaries. In our implementation, all components of the loss function are weighted equally, i.e., no additional weighting coefficients are introduced:

$$L_T = L_{\Omega_P} + L_{\Gamma_1} + L_{\Gamma_E}.$$

The loss function is defined as the sum of squared residuals evaluated at collocation points located in the interior

domain  $\Omega_P$ , on the obstacle boundary  $\Gamma_1$ , and on the artificial boundary  $\Gamma_E$ :

$$L_{\Omega_P} = \frac{1}{N_{\Omega_P}} \sum_{i=1}^{N_{\Omega_P}} \left\| \mathcal{R}_{\Omega_P}(\mathbf{x}_{P,i}) \right\|^2,$$

$$L_{\Gamma_1} = \frac{1}{N_{\Gamma_1}} \sum_{i=1}^{N_{\Gamma_1}} \left\| \mathcal{R}_{\Gamma_1}(\mathbf{x}_{1,i}) \right\|^2,$$

$$L_{\Gamma_E} = \frac{1}{N_{\Gamma_E}} \sum_{i=1}^{N_{\Gamma_E}} \left\| \mathcal{R}_{\Gamma_E}(\mathbf{x}_{E,i}) \right\|^2,$$

where the residuals are given by:

$$\mathcal{R}_{\Omega_P}(\mathbf{x}) := \Delta u_{\text{sct}}(\mathbf{x}) + k^2 u_{\text{sct}}(\mathbf{x}),$$

$$\mathcal{R}_{\Gamma_1}(\mathbf{x}) := \nabla u_{\text{sct}}(\mathbf{x}) \cdot \mathbf{n} + \nabla u_{\text{inc}}(\mathbf{x}) \cdot \mathbf{n},$$

$$\mathcal{R}_{\Gamma_E}(\mathbf{x}) := \nabla u_{\text{sct}}(\mathbf{x}) \cdot \mathbf{n} + iku_{\text{sct}}(\mathbf{x}).$$

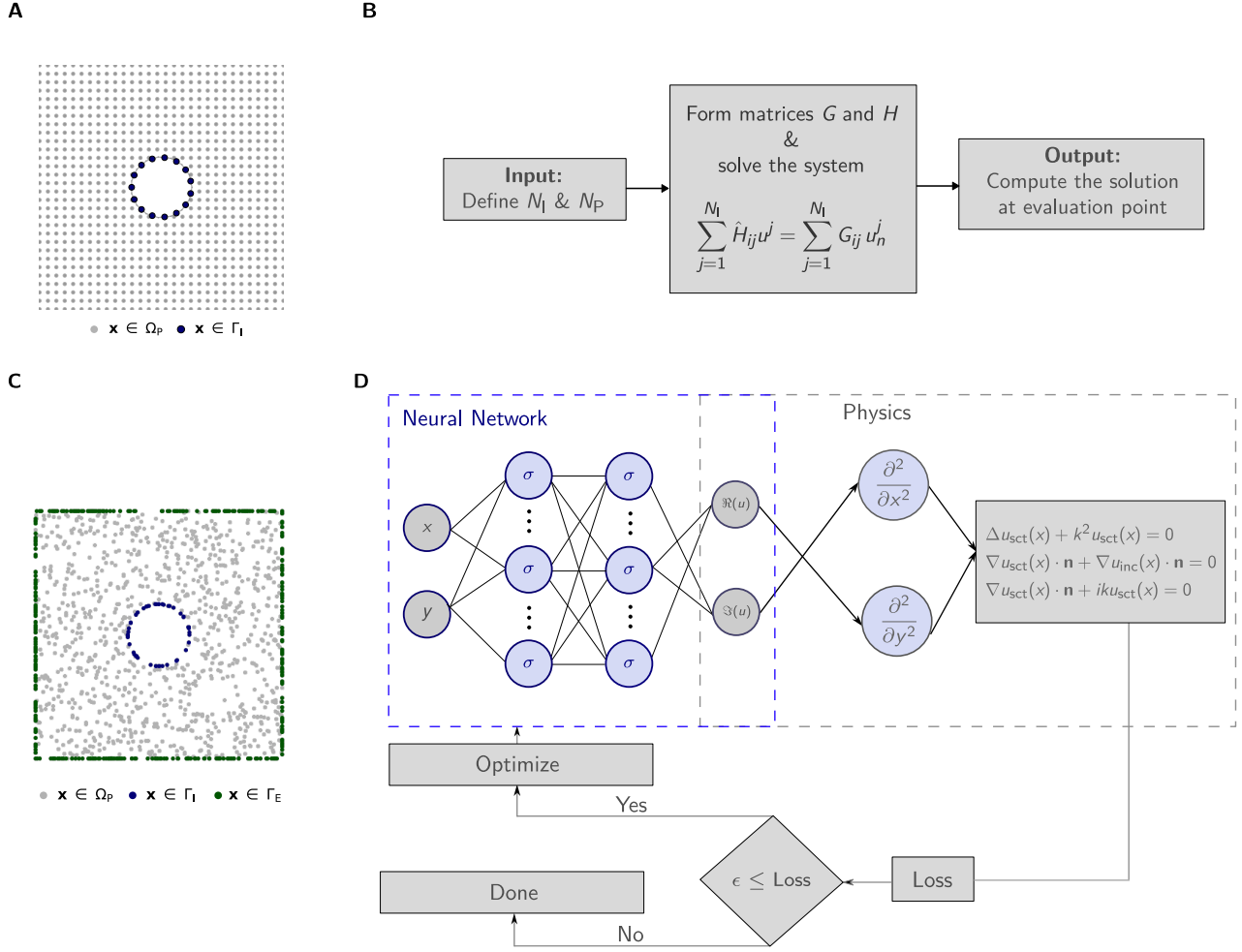
Here,  $\mathbf{x}_{P,i}$ ,  $\mathbf{x}_{1,i}$ , and  $\mathbf{x}_{E,i}$  denote collocation points in the interior, on the obstacle boundary, and on the artificial boundary, respectively. The collocation points are distributed using a Latin hypercube sampling strategy. An analysis of the sampling strategy, including a comparison with an adaptive approach, is provided in Supplementary Material S4 (PINNs sampling points analysis).

Both methods differ in their sampling strategies and overall workflows used to estimate the solution. Figure 2 illustrates the implementation of BEM and PINNs for the scattering problem. In the case of BEM, the elements are uniformly distributed along the interior boundary of the physical domain. The estimation process in BEM occurs in a single computational cycle: it begins with the selection of the number of integration and evaluation points, followed by the assembly and solution of the boundary integral equations. Finally, the solution is evaluated at the specified evaluation points. In contrast, PINNs use randomly distributed training points throughout the physical domain, including the interior, the interior boundary, and the exterior boundary. The estimation process involves multiple optimization cycles, during which the neural network coefficients are iteratively updated to minimize the residuals of the governing equations and boundary conditions.

The training process involves several hyperparameters, including the learning rate  $\alpha$ , the number of hidden layers  $L$ , the number of neurons per layer  $N$ , and the activation function  $\sigma$ . These are collectively described by the hyperparameter vector

$$\lambda = [\alpha, N, L, \sigma],$$

where  $\lambda \in \Lambda$  and  $\Lambda$  denotes the Cartesian product of the predefined hyperparameter ranges. The Adam optimizer relies on gradient updates and is highly sensitive to the choice of learning rate, which directly influences stability and convergence speed during the initial training stage. The



**Figure 2:** Illustration of the BEM and PINNs implementations for the scattering problem. (A) Distribution of boundary nodes and elements for BEM, along with the spatial arrangement of evaluation points within the domain. Gray dots represent evaluation points, while blue dots indicate collocation points on the boundary  $\Gamma_I$ . (B) Sequential workflow of the BEM implementation. The inputs are the number of elements for the boundary condition  $u_n$  on  $\mathbf{x} \in \Gamma_I$  and the number of evaluation points  $N_p$ . The boundary integral equation is solved to obtain the scattered field, which is then evaluated at the points  $\mathbf{x} \in \Omega_p$ . (C) Sampling strategy in the physical domain for PINNs, including points in the interior  $\Omega_p$  and on the boundaries  $\Gamma_I$  and  $\Gamma_E$ . Gray dots represent collocation points for the Helmholtz equation, while blue dots denote training points for the interior boundary condition on  $\Gamma_I$ . (D) Optimization process of the PINNs implementation for the Helmholtz equation. The neural network takes spatial coordinates  $(x, y)$  as input and outputs the real and imaginary parts of the scattered field  $u_{\text{sct}}$ . The loss function enforces the physics through the Helmholtz equation in  $\Omega_p$ , the Neumann boundary condition on  $\Gamma_I$ , and the Sommerfeld radiation condition on  $\Gamma_E$ . The optimization loop updates the network parameters until the prescribed tolerance  $\epsilon$  is reached.

network depth  $L$  and width  $N$  control the representational capacity of the network to approximate more complex functions. However, larger values of  $L$  and  $N$  increase the number of optimization coefficients, which makes the training process more computationally demanding and can slow down convergence if not properly regularized. Finally, the activation function  $\sigma$  affects both the expressiveness of the network and the propagation of gradients, thereby influencing the efficiency of the optimization process. Since it governs the type of nonlinearities introduced at each layer, its choice is particularly important in wave problems such as the Helmholtz equation, where oscillatory behavior must be

represented accurately.

The ranges for each hyperparameter were selected based on previous related studies in the PINNs literature, ensuring that they encompass configurations known to provide stable and accurate solutions for Helmholtz type problems [7, 18]. To determine the optimal configuration, we employed Optuna [1], an automatic hyperparameter optimization framework compatible with PyTorch. This procedure explored variations in the learning rate, network depth and width, activation functions, and boundary loss weights. The final configuration was selected as the one that minimized the

objective loss, which is given by the relative error of the prediction.

### Comparison strategy

We conducted numerical experiments to compare the performance of the BEM and PINNs, focusing on solution accuracy and computational cost. We followed a similar approach to that of Grossmann et al. [8]. Both methods were applied to solve the Helmholtz equation and were systematically compared using a ground-truth solution for the scattering obstacle provided in the Section S1 of the Supplementary Material (Analytical Solution).

We evaluated the accuracy of BEM as a function of boundary discretization by varying the number of elements. Smaller element sizes lead to higher accuracy but increase computation time. For PINNs, we controlled model capacity by adjusting the number of layers and neurons per layer, and assessed the corresponding effect on solution accuracy. The expressiveness of a neural network depends on both its depth and width.

We also estimated the computational time required for both methods. In the case of BEM, this included the time required to assemble and solve the boundary integral system, as well as the time required to evaluate the solution at interior points, while for PINNs, we measured training and evaluation times separately. To compare computational efficiency under similar conditions, we selected the BEM configuration whose accuracy most closely matched that of the best-performing PINN model obtained through hyperparameter tuning.

We distinguish between solution time and evaluation time because the BEM and PINNs approximate the solution in different ways. In BEM, the solution is obtained by discretizing the boundary and solving a boundary integral system, while values in the domain are computed afterwards through the numerical evaluation of boundary integral representations. In contrast, in PINNs, a neural network is trained using the collocation points distributed in the domain and on the boundaries. Once the network is trained, the evaluation at new points requires only a forward pass through the network, which is faster than the training phase. For this reason, we report separately the training and evaluation times in PINNs.

To analyze the generalization ability of PINNs beyond the training domain and to assess how well each method handles the radiation condition, we perform an additional study. In this analysis, we evaluate the scattered field in a region that extends beyond the training domain by computing the numerical solution at a fixed propagation angle while varying the radial distance from the scatterer. The results obtained from both methods are compared against the analytical solution to assess their accuracy in the far field. We examine the radial decay of the wave amplitude and the relative error as functions of distance, with particular

attention given to the role of domain truncation in the PINN framework and its impact on solution accuracy at large distances. Based on the results detailed in Section S2 of the Supplementary Material (Hyperparameter Optimization), the configuration  $\lambda^* = \{\alpha = 10^{-2}, N = 25, L = 3, \sigma = \text{Sine}\}$  was selected. The hyperparameters were optimized using Optuna over 50 trials, considering the following search ranges:  $\alpha \in \{10^{-2}, 10^{-3}, 10^{-4}\}$ ,  $L \in \{1, 2, 3\}$ ,  $N \in \{25, 50, 75\}$ , and  $\sigma \in \{\text{Tanh}, \text{Sigmoid}, \text{Sine}\}$ . The trained PINN models used in this study are publicly available in a Zenodo repository [23].

## Results

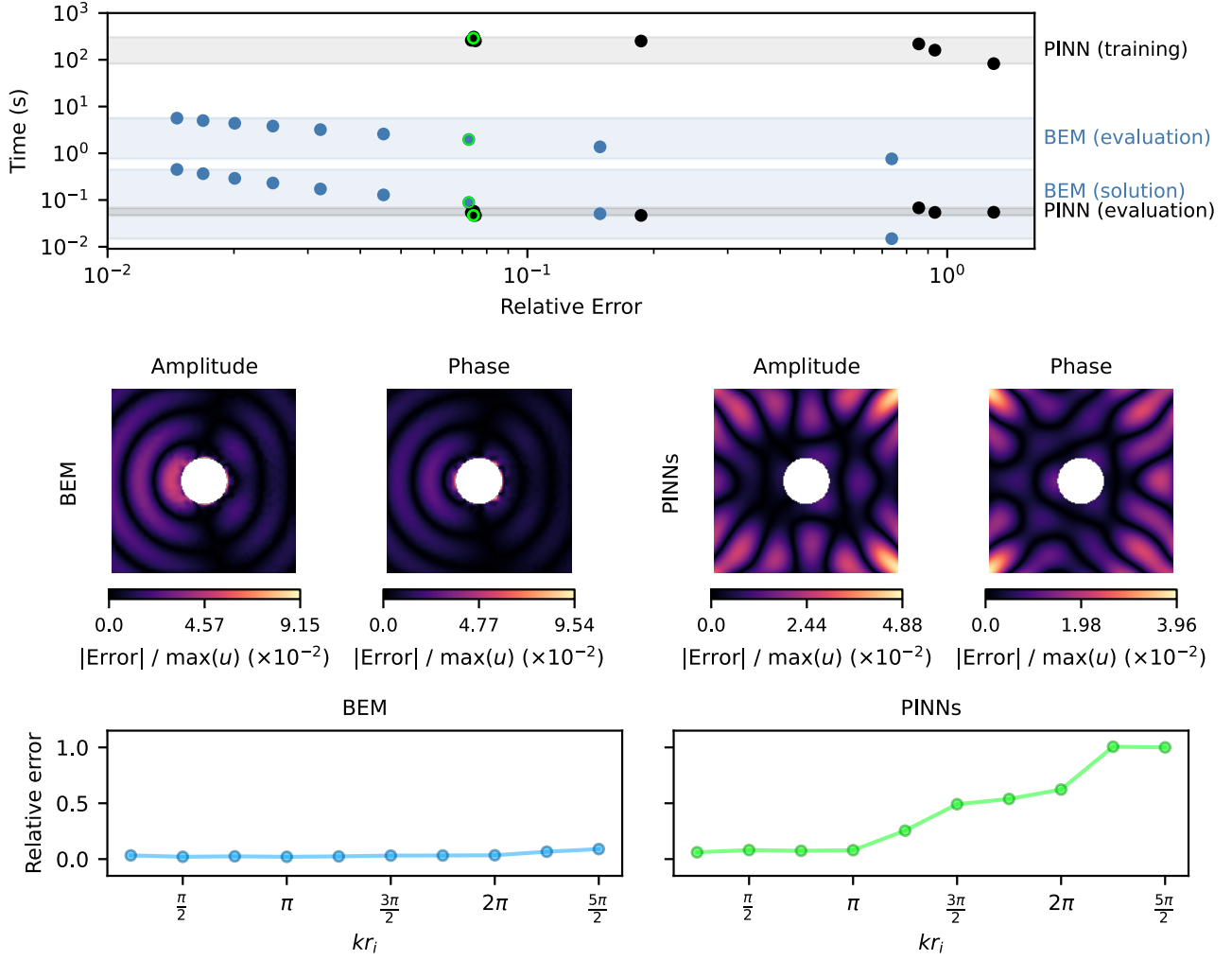
In this section, we present the results obtained from the analytical solution, BEM, and PINNs for the 2D Helmholtz scattering problem. We vary the number of elements in BEM to achieve comparable accuracy, enabling a fair comparison of computational time. Then, we evaluate the accuracy of both methods outside the training domain to assess their generalization capability. The results highlight the performance of each approach in estimating the scattered field within and beyond the training domain. All experiments were performed on machines equipped with 10 CPU cores / 20 threads (13<sup>th</sup> Gen Intel Core i9-13900H @ 3.40 GHz) and an NVIDIA GeForce RTX 4060 Laptop GPU.

### Comparison of Accuracy and Computational Efficiency Between Methods

We conducted a comparative analysis of the BEM and PINNs for solving the two-dimensional Helmholtz equation, focusing on both solution accuracy and computational efficiency under varying parameters. To ensure reproducibility of the accuracy metrics, a fixed random seed was applied during model initialization and training. To account for runtime variability, each experiment was repeated 10 times and the average execution time was reported.

We tested BEM with the number of elements ranging from 5 to 45 in increments of 5, and PINNs with network configurations varying from 1 to 3 hidden layers and 25 to 75 neurons per layer in increments of 25. These ranges are sufficient to capture the error reduction trends and to identify the achievable accuracy scales without making the computation excessively expensive. The remaining hyperparameters of the PINN were fixed according to the optimal configuration identified during the hyperparameter optimization stage.

Figure 3 (top) illustrates the relation between relative error and computational time for BEM and PINNs. As expected, finer discretization in BEM improves accuracy but also increases computational cost. The assembly and solution time for BEM spans from the order of  $10^{-1}$  to  $10^{-2}$  s across the tested values of  $n$ , while the evaluation time required to compute the solution at interior points is of  $10^{-1}$  to  $10^0$  s. For PINNs, training times varied between  $10^1$  and  $10^2$  s, whereas evaluation times were on the order



**Figure 3:** Performance evaluation of BEM and PINNs. (Top) Relative error versus computational time for BEM and PINNs. For BEM, the reported times correspond to the assembly and solution of the boundary integral system and the subsequent evaluation of the solution at interior points. For PINNs, training time and evaluation time are reported separately. Shaded horizontal bands indicate the ranges between the minimum and maximum times for each category. Labels on the right identify the four groups: PINN (training), BEM (evaluation), BEM (solution), and PINN (evaluation). The green highlighted points correspond to the selected configurations, ( $L = 3, N = 25$ ) for the PINN and  $n = 15$  for the BEM, used for direct comparison of computational costs. (Middle) Spatial distributions of the relative error in amplitude and phase for the selected BEM and PINN configurations. (Bottom) Relative error as a function of  $kr_i$  for both methods.

of  $10^{-2}$  s.

For the computational time comparison, we selected the BEM configuration that achieved the closest accuracy to the optimal PINN solution. Specifically, the PINN with 3 layers and 25 neurons reached a relative error of  $7.44 \times 10^{-2}$ , while the closest BEM result was obtained with  $n = 15$ , yielding  $7.25 \times 10^{-2}$ . Figure 3 (middle) shows the field error for both methods using these configurations. For BEM, the error tends to increase near the interior boundary of the obstacle, whereas for PINNs, errors are mainly concentrated in regions closer to the exterior boundary, where the domain was truncated. We extended the analysis to different frequencies Figure 3 (bottom). While BEM maintains consistently low errors across the range of  $kr_i$ . PINNs accuracy exhibits a

stronger sensitivity to increasing wavenumber.

In the selected case, the assembly and solution of the BEM system required a computational time on the order of  $10^{-2}$  s, making it approximately four orders of magnitude faster than the training time of the PINN, which was on the order of  $10^2$  s. This highlights the high computational cost of PINNs associated with the optimization process during training. However, once trained, the PINN model can be evaluated in about  $10^{-2}$  s, which is approximately two orders of magnitude faster than the evaluation of the BEM solution at interior points, which requires on the order of  $10^0$  s. This makes PINNs particularly well suited for applications requiring repeated simulations or near real-time inference. When considering the total computational cost, the PINN

requires 289.92 s (training and evaluation), while the BEM requires 2.06 s (assembly and evaluation), confirming that the overall cost remains dominated by the PINN training stage. Additional analyses complementing these results are provided in Supplementary Material S3.

A central objective of this work was to compare the performance and characteristics of traditional numerical methods and machine learning-based approaches for solving wave scattering problems using a fair baseline. In comparing BEM to PINNs, we addressed two common pitfalls identified in the literature on ML-based PDE solvers [15].

First, we ensured that both methods were evaluated at comparable accuracy levels. In the case of the PINN approach, this required an hyperparameter tuning procedure to identify a configuration yielding optimal predictive performance. Once this reference accuracy was established, the number of elements in the BEM solution was progressively increased until a comparable level of accuracy was achieved. Subsequently, the computational cost of both methods was assessed, thereby satisfying the requirement of comparing methods at either equal accuracy or equal computational cost.

Second, to avoid misleading conclusions due to the use of suboptimal numerical baselines, we employed BEM as the reference method, a well-established solver for wave scattering problems. BEM is particularly effective for problems in unbounded domains. In the case they are governed by the Helmholtz equation, it allows the exact enforcement of the Sommerfeld radiation condition through the use of fundamental solutions. Additionally, BEM reduces the dimensionality of the problem by requiring discretization only on the boundary, which is computationally advantageous in unbounded domains. This choice ensures that the traditional method is not only accurate but also representative of state-of-the-art efficiency for the class of problems considered.

By fulfilling these conditions, our comparative study provides a fair and informative assessment of the trade-offs between both solvers. While BEM demonstrates superior performance in terms of runtime for forward problems, PINNs offer a flexible framework that can be applied to a wide range of scientific and engineering problems, particularly due to the availability of open-source implementations. PINNs employ a mesh-free formulation in which the residuals of the governing equations are minimized at randomly sampled collocation points, allowing flexibility in spatial resolution. In contrast, classical methods such as BEM require mesh refinement to achieve higher accuracy. The proposed PINN framework also presents limitations. In particular, the trained model is problem-specific and depends on the geometry considered during training. Therefore, it cannot be directly evaluated on other geometry without retraining the model. This limitation is further compounded by the fact that retraining PINNs can be computationally

expensive due to long training times, as shown in Figure 3, which restricts their practical applicability. Nonetheless, in scenarios that require repeated evaluations of the same PDE, such as in inverse problems, the relatively fast evaluation time of trained PINNs can be advantageous.

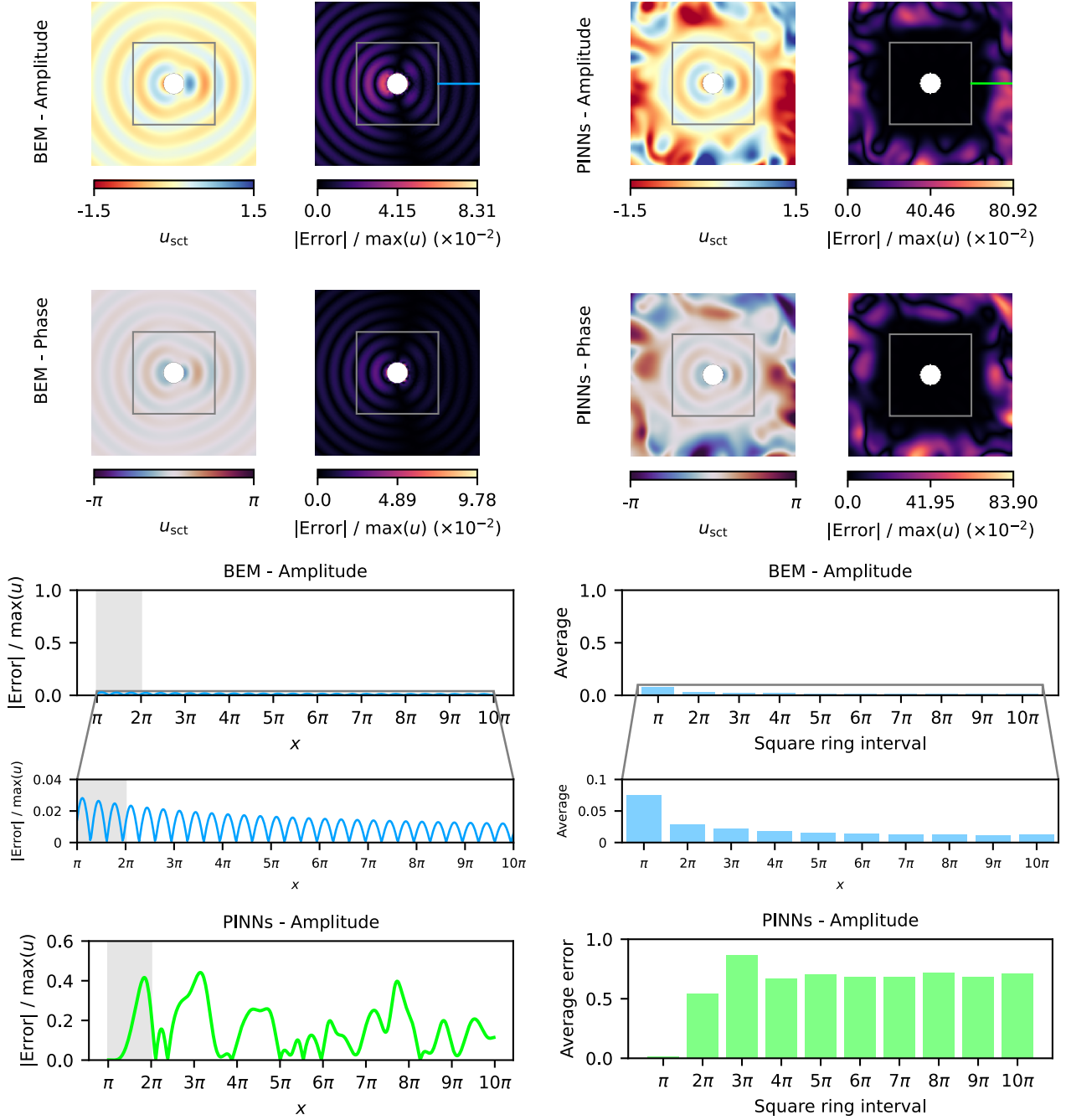
It is also important to acknowledge a limitation of this comparison regarding the measurement of computational cost. In this work, efficiency was assessed primarily through wall-clock time, which, although widely adopted in the literature, inherently depends on the characteristics of the hardware and system configuration used to perform the experiments. This makes the results sensitive to differences across machines and may reduce the generalizability of the reported results. A hardware independent approach would be relevant to complement wall-clock time with algorithmic complexity analysis or with estimates of the number of floating-point operations (FLOPs), thereby providing a clearer separation between the intrinsic efficiency of the methods and the capabilities of the underlying hardware. Nevertheless, reporting execution time remains a practical and frequently used benchmark for empirical comparisons, and thus it was employed in this study.

The hyperparameter tuning performed presented in the Section S2 of the Supplementary Material (Hyperparameter Optimization) provides further insight into potentially suitable sets of hyperparameters for this problem. However, it is important to note that the performance of PINNs also depends on additional hyperparameters not considered in this work, such as the total number of training iterations and the relative weights of the loss terms. Even with dedicated tuning procedures, the high dimensionality of the design space introduces variability and limits the reproducibility and comparability of results.

These factors imply that comparisons between BEM and PINNs should be interpreted in the context of specific problem setups and computational constraints. While BEM yields consistent solutions with relatively low sensitivity to parameters, PINNs require additional effort in model selection and optimization to achieve competitive accuracy. A possible alternative to further improve accuracy would be the incorporation of Fourier features, which have been found appropriate to mitigate spectral bias; however, the purpose of this work was to implement the method as originally proposed. An additional possible approach would be to adopt a multi-objective perspective, taking into account not only accuracy but also computational complexity.

### Generalization Beyond the Training Domain

To evaluate the generalizability of both methods in an extended region of the domain, we computed the solution beyond the original training area. Figure 4 shows the amplitude and phase of the predicted fields, as well as the spatial distribution of the relative error. For the BEM, the



**Figure 4:** Scattered field computed by BEM and PINNs outside the training region. The first two rows show the amplitude (top) and phase (second row) of the scattered field. For each method (BEM on the left, PINNs on the right), the left panel displays the predicted field, while the right panel shows the relative error normalized by the maximum of the reference solution. The square indicates the training domain, corresponding to  $l_e = \pi$ . The third row corresponds to BEM and the fourth row to PINNs. In each case, the left panel shows the relative amplitude error along the line  $y = 0$  as a function of  $x \in [\pi, 10\pi]$ , while the right panel presents the average relative error computed over square ring intervals with semi-lengths ranging from  $\pi$  to  $10\pi$ . For BEM, a zoomed-in view is included to highlight small error variations.

relative error increased to  $7.27 \times 10^{-2}$  in the extended region. The error remained uniformly distributed, suggesting that the method maintains consistent accuracy even far from the obstacle. Additionally, an exponential decay of the field amplitude was observed with increasing distance from the

scatterer.

This behavior reflects the global nature of BEM, which inherently enforces the boundary conditions over the entire domain. In contrast, PINNs experienced a marked reduction

in accuracy when evaluated outside the training domain. The relative error increased to 7.09, and the error distribution clearly showed high values in the region outside the training domain.

An important distinction between BEM and PINNs lies in how each method handles the radiation condition, which governs the behavior of the scattered field at large distances. In BEM, this condition is satisfied exactly by construction through the use of the fundamental solution in the boundary integral formulation. This allows for an accurate computation of the far-field response without requiring additional assumptions. In contrast, PINNs do not explicitly enforce the radiation condition. Instead, they approximated it using domain truncation techniques, such as the ABC used in this work, which may not fully capture the correct asymptotic behavior of outgoing waves.

This limitation directly affects the generalization capacity of PINNs beyond the training region. In the current implementation, collocation points are sampled randomly and uniformly within the physical domain. While this strategy is straightforward, it does not prioritize regions where the radiation condition is most critical, particularly near the artificial boundary. As a result, PINNs may not accurately represent the far-field response, especially outside the domain used during training. This lack of generalization is a known limitation of neural network-based models, which approximate solutions only within the region where they are trained. Alternative sampling strategies, such as a generative point sampling strategy [11], or boundary-focused distributions or adaptive collocation schemes, could help mitigate this issue and improve the enforcement of far-field behavior.

## Conclusions

In summary, while BEM ensures accuracy through the direct satisfaction of physical conditions and dimensional reduction, PINNs offer flexible implementation through the configuration of their loss function, but require careful hyperparameter tuning and strategic sampling to achieve reliable results. Although BEM outperforms PINNs when considering solution time and accuracy, the strengths of each method are complementary, motivating the exploration of hybrid approaches that combine the robustness of boundary-based formulations with the adaptability of neural networks.

Recent advances in computational modeling have integrated neural networks as function approximators, supported by automatic differentiation and GPU acceleration. These machine learning techniques offer advantages such as mesh-free implementation, efficient inference, and flexibility to incorporate data into the solution process.

Our comparative analysis demonstrates that PINNs require significantly longer training times than BEM to achieve comparable accuracy. This computational cost limits their

practical use for forward simulations. However, hybrid approaches that combine the strengths of both frameworks may offer improved performance, particularly in inverse problems, where repeated model evaluations and parameter estimation are required.

In future work, we aim to investigate hybrid modeling methods that combine the physical accuracy of BEM with the flexibility of neural networks, building on recent developments in boundary-based and physics-informed approaches [17, 21, 14]. Specifically, we propose to replace the differential operator in the PINN loss function with a boundary integral formulation. For the Helmholtz equation, this approach enables the direct incorporation of the Sommerfeld radiation condition into the training process, improving the model's ability to generalize beyond the training domain and ensuring accurate predictions in unbounded regions. Moreover, this formulation is particularly advantageous in inverse problems, where enforcing correct far-field behavior is essential for achieving stable and physically consistent reconstructions.

## References

- [1] Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M., 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. doi:10.48550/arXiv.1907.10902.
- [2] Berenger, J.P., 1994. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of Computational Physics* 114, 185–200. doi:10.1006/jcph.1994.1159.
- [3] Bermúdez, A., Hervella-Nieto, L., Prieto, A., Rodríguez, R., 2007. An optimal perfectly matched layer with unbounded absorbing function for time-harmonic acoustic scattering problems. *Journal of Computational Physics* 223, 469–488. doi:10.1016/j.jcp.2006.09.018.
- [4] Bonfanti, A., Santana, R., Ellero, M., Gholami, B., 2024. On the generalization of PINNs outside the training domain and the hyperparameters influencing it. *Neural Computing and Applications* 36, 22677–22696. doi:10.1007/s00521-024-10178-2.
- [5] Chandler-Wilde, S.N., Graham, I.G., Langdon, S., Spence, E.A., 2012. Numerical-asymptotic boundary integral methods in high-frequency acoustic scattering. *Acta Numerica* 21, 89–305. doi:10.1017/S0962492912000037.
- [6] Dissanayake, M.W.M.G., Phan-Thien, N., 1994. Neural-network-based approximations for solving partial differential equations. *Communications in Numerical Methods in Engineering* 10, 195–201. doi:10.1002/cnm.1640100303.
- [7] Escapil-Inchauspé, P., Ruz, G.A., 2023. Hyper-parameter tuning of physics-informed neural networks: Application to Helmholtz problems. *Neurocomputing* 561, 126826. doi:10.1016/j.neucom.2023.126826.
- [8] Grossmann, T.G., Komorowska, U.J., Latz, J., Schönlieb, C.B., 2024. Can physics-informed neural networks beat the finite element method? *IMA Journal of Applied Mathematics* 89, 143–174. doi:10.1093/imamat/hxae011.
- [9] Katsikadelis, J.T., 2016. *The Boundary Element Method for Engineers and Scientists, Second Edition: Theory and Applications*. 2nd edition ed., Academic Press, London.
- [10] Kirkup, S., 2007. *The Boundary Element Method in Acoustics*. Journal Abbreviation: *Journal of Computational Acoustics* Publication Title: *Journal of Computational Acoustics*.
- [11] Kochliaridis, V., Dilmeris, I., Palaskos, A., Vlahavas, I., 2025. Generative point sampling strategies for physics-informed neural

- networks. *Engineering with Computers* 41, 3219–3239. doi:10.1007/s00366-025-02158-4.
- [12] Komatitsch, D., Martin, R., 2007. An unsplit convolutional Perfectly Matched Layer improved at grazing incidence for the seismic wave equation. *Geophysics* 72. doi:10.1190/1.2757586.
- [13] Lagaris, I., Likas, A., Fotiadis, D., 1998. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks* 9, 987–1000. doi:10.1109/72.712178. conference Name: IEEE Transactions on Neural Networks.
- [14] Li, R., Ye, W., Liu, Y., 2025. A boundary-based fourier neural operator (B-FNO) method for efficient parametric acoustic wave analysis. *Engineering with Computers* 41, 2393–2410. doi:10.1007/s00366-024-02103-x.
- [15] McGreivy, N., Hakim, A., 2024. Weak baselines and reporting biases lead to overoptimism in machine learning for fluid-related partial differential equations. *Nature Machine Intelligence* 6, 1256–1269. doi:10.1038/s42256-024-00897-5.
- [16] Morse, P.M., Ingard, K.U., 1987. *Theoretical Acoustics*. Princeton University Press, Princeton.
- [17] Nagy-Huber, M., Roth, V., 2024. Physics-informed boundary integral networks (PIBI-Nets): A data-driven approach for solving partial differential equations. *Journal of Computational Science* 81, 102355. doi:10.1016/j.jocs.2024.102355.
- [18] Nair, S., Walsh, T.F., Pickrell, G., Semperlotti, F., 2024. Multiple scattering simulation via physics-informed neural networks. *Engineering with Computers* doi:10.1007/s00366-024-02038-3.
- [19] Nganyu Tanyu, D., Ning, J., Freudenberg, T., Heilenkötter, N., Rademacher, A., Iben, U., Maass, P., 2023. Deep learning methods for partial differential equations and related parameter identification problems. *Inverse Problems* 39, 103001. doi:10.1088/1361-6420/ace9d4. publisher: IOP Publishing.
- [20] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*.
- [21] Qu, W., Gu, Y., Zhao, S., Wang, F., Lin, J., 2024. Boundary integrated neural networks and code for acoustic radiation and scattering. *International Journal of Mechanical System Dynamics* 4, 131–141. doi:10.1002/msd2.12109.
- [22] Raissi, M., Perdikaris, P., Karniadakis, G.E., 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics* 378, 686–707. doi:10.1016/J.JCP.2018.10.045. publisher: Academic Press.
- [23] Rincón, O.A., Pérez-Bernal, G., Montoya-Noguera, S., Guarín-Zapata, N., 2026. Trained physics-informed neural network (pinn) models for the 2d helmholtz equation. doi:10.5281/zenodo.18351598.
- [24] Schot, S.H., 1992. Eighty years of Sommerfeld's radiation condition. *Historia Mathematica* 19, 385–401. doi:10.1016/0315-0860(92)90004-U.