



Multiple scattering simulation via physics-informed neural networks

Siddharth Nair¹ · Timothy F. Walsh² · Greg Pickrell² · Fabio Semperlotti¹

Received: 2 February 2024 / Accepted: 23 July 2024

© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2024

Abstract

This work presents a physics-driven machine learning framework for the simulation of acoustic scattering problems. The proposed framework relies on a physics-informed neural network (PINN) architecture that leverages prior knowledge based on the physics of the scattering problem as well as a tailored network structure that embodies the concept of the superposition principle of linear wave interaction. The framework can also simulate the scattered field due to rigid scatterers having arbitrary shape as well as high-frequency problems. Unlike conventional data-driven neural networks, the PINN is trained by directly enforcing the governing equations describing the underlying physics, hence without relying on any labeled training dataset. Remarkably, the network model has significantly lower discretization dependence and offers simulation capabilities akin to parallel computation. This feature is particularly beneficial to address computational challenges typically associated with conventional mesh-dependent simulation methods. The performance of the network is investigated via a comprehensive numerical study that explores different application scenarios based on acoustic scattering.

Keywords Physics-informed neural network (PINN) · Multiple scattering · Superposition principle · Wave propagation

1 Introduction

The fundamental concept of scattering holds a critical role in the general area of wave physics including, but not limited to, photonics [1, 2], geophysics [3], bio-medicine [4], acoustics [5, 6], and particle physics [7, 8]. Within these fields, scattering finds applications in the most diverse areas including, but not limited to, non-destructive testing [9], metrology [10], advanced medical imaging [11, 12], remote sensing [13], material characterization [14, 15], and inverse material design [6, 16].

Over the years, various numerical approaches have been developed to tackle the complex problem of scattering simulations. Finite element methods (FEM) [17, 18], finite difference methods (FDM) [19], boundary element methods [20], and discontinuous Galerkin methods [21] have been among the most popular. Out of all these existing techniques, FEM

distinguished itself for its ability to handle complex geometries and irregular boundaries. However, being a mesh-based technique, FEM suffers from a strong dependence on the discretization parameters. This dependence renders the FE models very computationally intensive when applied to scattering problems involving a large number of degrees of freedom, such as those for high frequency, multiple scattering, and multiscale applications [22, 23]. In particular, when dealing with multiple scattering scenarios, a notable challenge arises as the scale and number of scatterers increases. In these situations, FEM simulations become very computationally intensive or even intractable (i.e. exceeding the resources of available supercomputers). In fact, as the number of scattering elements increases the mesh requirements become more stringent (in order to capture the interactions between different scatterers) and the number of degrees of freedom increases very rapidly. Further studies have developed meshfree approaches like meshless local Petrov-Galerkin [24], smoothed particle hydrodynamics [25], kernel-based meshless method [26], and element-free Galerkin method [27] to address these limitations of FEM. The ability of these methods to eliminate the need for complex mesh generation, which can be very time-consuming (particularly for problems involving irregular geometries or evolving domains) offers a significant advantage over mesh-based

✉ Fabio Semperlotti
fsemperl@purdue.edu

¹ Ray W. Herrick Laboratories, School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907, USA

² Sandia National Laboratories, Albuquerque, NM 87185, USA

approaches. However, meshless methods often increase the computational costs, compared to mesh-based methods, due to the need for large numbers of nodes to accurately represent the domain [28]. Moreover, ensuring stability and accuracy can be more complex in meshless approaches due to the lack of structured elements and their dependence on neighbor node connectivity, especially in problems involving discontinuities [28, 29].

More recently, the use of machine learning as a tool to perform computations has seen a rapid increase, ultimately leading to the formulation of several models and techniques for a variety of applications. In the field of scattering simulations, examples include supervised learning-based deep neural networks (DNNs) to estimate optical scattering produced by nanoparticles [30], and the solution of the Poisson's equation in 2D domains with simple scatterer shapes [31]. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have been employed to solve the forward finite difference time domain problems [32]. Recent studies have also introduced semi-supervised [6] and unsupervised [33] deep learning models for material design applications in acoustics and optics. A major drawback of these conventional DNNs is the need for comprehensive labeled datasets to achieve effective network training. Furthermore, these trained DNNs are inherently incapable of guaranteeing physically consistent results. Hence, various engineering applications have shifted their focus towards physics-driven deep learning models that weakens the data dependence. This latter process has led to the development of the concept of physics-driven machine learning [34], which aims at integrating the governing physics into conventional DNNs.

In physics-driven machine learning, the integration of the physics of the problem into a learning algorithm amounts to introducing appropriate biases in the DNN that can guide the network prediction towards physically consistent solutions. Among the different approaches that can be leveraged to embed the physics of the problem within the network [35], one of the most effective and extensively studied concept is that of physics-informed neural network (PINN) [36]. Since the dynamic behavior of most of the applications in the field of engineering is often described using governing laws expressed as a system of partial differential equations (PDEs), the ability of PINN to approximate numerical solutions of PDEs finds application in a wide range of domains. To-date, the literature shows that PINNs can address problems in a variety of fields, including fluid mechanics [37, 38], heat transfer [39], and solid mechanics [40] and more. Moreover, the use of PINNs offers two significant advantages over conventional DNN. First, PINNs can train without any need for labeled datasets, thus addressing the challenge of generating (either computationally or experimentally) extensive training data. Second, PINNs can learn to approximate results that

are consistent with the underlying physics, therefore overcoming the tendency of conventional DNNs (trained on labeled datasets) to produce physically inconsistent predictions. In addition, and not less important, unlike the conventional simulation approaches (e.g. FEM, FDM), PINNs reduce discretization dependence.

Recent studies have introduced PINNs for scattering applications across different fields. Chen et al. [41, 42] explored the problem of inverse parameter identification from optical scattered fields using PINNs. This work integrated the Maxwell equations into a PINN framework in order to extract the permittivity of the homogenized material based on the known scattered field. These parameters are crucial to characterize the scatterers in optical applications. Lee et al. [43] developed a physics-guided neural network model for the transient analysis of acoustic scattering problems due to a single scatterer. The approach presented was not fully physics-driven; instead, the model enforced a combination of governing equation-based and data-based loss functions.

PINNs were also developed as solvers for transient forward problems involving the seismic wave equation [44] and the full-wave inversion [45] for geophysical applications. Song et al. [46] developed a PINN to solve the frequency-domain acoustic wave equation for transversely isotropic (TI) media with a vertical axis of symmetry (VTI) for geophysics applications. Wang et al. [47] proposed a PINN model targeted to steady-state acoustic applications. The studies mentioned above developed PINN models to predict the scattered fields due to material heterogeneities whose influence was captured by an explicit material parameter within the governing PDE. In other terms, these previous studies investigated the scattered field due to scattering sources uniformly distributed across the domain, and typically employed homogenization techniques. However, a more complex and unexplored application of PINNs involves simulating the steady-state scattering response due to fully-resolved scatterers embedded within the domain. In this case, the scattered field is produced due to the wave interaction with the embedded scatterers, hence making it a function of the scatterer shape. It is important to note that, unlike the material parameters within the PDE, there is no equivalent parameter within the governing PDE that can explicitly account for the effect of the scatterer shape. In this latter scenario, the presence of rigid scatterers within the unbounded domain introduces additional (both internal and external) boundary conditions. These boundary conditions are independent of the governing PDE, and they play a crucial role in shaping the behavior of the scattered field. The proposed work focuses on a steady-state problem and does not address transient problems.

Therefore, the objective of this study is to establish a PINN framework capable of solving the scattered field

generated by rigid scatterers embedded within an unbounded homogeneous domain.

1.1 Major contributions

The primary goal of this study is to develop a class of PINNs tailored to address a wide range of acoustic scattering problems involving arbitrary shaped scatterers, and high-frequency excitations. In this context, the term “arbitrary shape” indicates a wide range of scatterers with both regular and irregular shapes, while the term “high frequency” indicates wavelengths smaller than the characteristic size of the scatterer. More specifically, we focus on the development of PINNs to find approximate solutions to steady-state harmonic acoustic problems (governed by the Helmholtz equation) involving arbitrary-shaped rigid scatterers distributed in an unbounded (infinite) acoustic domain. The key contributions of this study are strictly connected to the two essential components of the proposed PINN framework

1. **Baseline-PINN (*b* -PINN):** A PINN architecture conceived to simulate acoustic scattering problems involving either single or (a small number of) multiple rigid scatterers of arbitrary shapes. In contrast to some existing models, such as those in the DeepXDE library [48] that are suitable for enforcing one basic internal boundary shape, the baseline-PINN is specifically designed to simulate fields generated by arbitrary scatterer shapes (introducing boundaries internal to the domain). Furthermore, the *b*-PINN architecture integrated with a tailored optimization procedure (a combination of gradient-based and second-order optimization) is well suited to simulate steady-state scattered fields at high frequencies (i.e. wavelength much shorter than the characteristic scatterer size). It is important to note that the *b*-PINN is also able to solve multi-scatterer configurations (although with some limitations on the total number of scattering elements). Contrarily to classical solution methods (e.g. FEM), in this latter case the performance of *b*-PINN is not limited by the number of degrees of freedom. Nevertheless, a limitation is imposed by the error propagation within the network [49], as discussed below.
2. **Superposition-PINN (*s* -PINN):** A PINN architecture specifically conceived to address the limitations of *b*-PINNs when applied to large number of scatterers. Broadly speaking, the superposition-PINN is based on an architecture that leverages multiple *b*-PINNs. The uniqueness of this network lies in its dual-level integration of the underlying physics: 1) the *s*-PINN architecture embodies the superposition principle of linear acoustics, and 2) the governing PDE and boundary conditions are directly enforced through the network

loss function. This approach addresses the limitation of *b*-PINNs by providing extensive control over the local distribution of scatterers within the multiple scatterer domain. Specifically, the *s*-PINN implements a domain partitioning approach such that individual *b*-PINNs can be used to simultaneously handle separate sub-domains. In addition, the individual *b*-PINNs within the overall *s*-PINN architecture naturally enable parallel processing by assigning each *b*-PINN training to separate computational cores. This parallelization approach can significantly reduce the total simulation time for multiple scattering studies as the *s*-PINN runtime is only dominated by the largest *b*-PINN and is not the sum of the runtimes of all the *b*-PINNs.

It follows that, by virtue of its reduced spatial discretization dependence and its parallelization capabilities, the resulting physics-driven machine learning-based forward solver has the potential to effectively reduce the computational cost associated with multiple scattering simulations compared to traditional methods (based on the solution of systems of differential equations). While PINNs exhibit lower discretization dependence during the training process, the ability of the trained PINNs to approximate solutions on a new discretization makes the prediction process discretization-independent. However, note that this work focused on the development of the methodology and architecture and does not implement a parallel version of the proposed model; all results herein are obtained via serial computations. The primary focus of this work is on introducing the above-mentioned PINN methodologies to enable certain classes of simulations (such as multiple scattering), rather than optimizing the computational performance.

The paper is organized as follows. Section 2 introduces some background on PINNs. Section 3 presents the general problem setup for single scatterers, elaborates on the implementation of the *b*-PINN, and reports and discusses the results of the trained *b*-PINNs. Thereafter, Sect. 4 presents the general problem setup for multiple scattering, elaborates on the implementation of the *s*-PINN, and reports and discusses the results of the trained *s*-PINNs.

2 Basics of physics-informed neural network (PINN)

Multiple approaches can be followed to construct a deep learning model capable of simulating the physical response of a system. Data-driven deep learning models can capture the physical response of the system by optimizing the neural network hyperparameters that lead to the minimization of the error between the training data and the network predictions. A major drawback of the data-driven approach is that

it requires a large volume of training data and the predictions are often physically inconsistent (as the model can only capture the physical response represented by the training data). On the other hand, physics-informed neural networks (PINNs) belong to a class of deep learning algorithms that can integrate prior physical information about the problem to improve the performance of the learning algorithms. More specifically, PINNs can enforce the deep learning algorithm to learn the mathematical framework of the problem in the form of governing partial differential equations (PDEs) and constitutive equations. The two major advantages of using PINNs in comparison to conventional DNNs are: 1) PINNs present the capability to train without labeled datasets, and 2) PINNs can learn to predict physically consistent results.

In a general PINN framework, the unknown solution ϕ is computationally predicted by a neural network parameterized via a set of parameters $\theta = [\mathbf{W}, \mathbf{b}]$, where \mathbf{W} and \mathbf{b} are the sets of weights and biases of the network, respectively. The PINN approximation of the solution is $\hat{\phi}$ such that $\hat{\phi}(\theta) \approx \phi$. Moreover, the PINN learns to predict $\hat{\phi}$ by finding the optimal θ by minimizing a loss function $\mathcal{L}(\theta)$

$$\theta^* = \arg \min_{\theta} \mathcal{L}(\theta) \tag{1}$$

The PINN loss function can be represented as follows

$$\mathcal{L}(\theta) = w_B \mathcal{L}_B(\theta) + w_N \mathcal{L}_N(\theta) \tag{2}$$

where w_B and w_N are the weighting factors and

$$\begin{aligned} \mathcal{L}_B(\theta) &= \frac{1}{N_b} \sum_{i=1}^{N_b} |\mathcal{B}(\hat{\phi}, \mathbf{x}_b^i; \theta) - g^i|^2 \\ \mathcal{L}_N(\theta) &= \frac{1}{N_n} \sum_{i=1}^{N_n} |\mathcal{N}(\hat{\phi}, \mathbf{x}_n^i; \theta)|^2 \end{aligned} \tag{3}$$

where \mathcal{B} represents the boundary condition, \mathcal{N} represents the governing PDE, and g represents the boundary data. Further, $\mathcal{L}_B(\theta)$ and $\mathcal{L}_N(\theta)$ are the mean square errors (MSE) of the residual of the boundary condition and the governing PDE, respectively. Here, \mathbf{x}_b is an arbitrary point sampled from a distribution of N_b boundary training points and \mathbf{x}_n is an arbitrary point sampled from a distribution of N_n domain collocation points.

3 Acoustic scattering in the frequency domain

This section introduces the benchmark problem used to test the b -PINN architecture. As previously indicated, this problem is based on the acoustic scattering due to rigid scatterers in the frequency domain. Initially, some theoretical preliminaries of acoustic scattering and the corresponding mathematical formulation are presented. Then, the concept of physics-informed neural network (PINN) formulated for the specific benchmark problem is introduced. Finally, the performance of the proposed network is studied through a series of numerical experiments.

3.1 Problem description

Consider the classical problem of 2D acoustic scattering in air due to a single rigid scatterer. Assuming harmonic excitation and steady state conditions, the acoustic scattering problem is governed by the Helmholtz equation defined on a 2D domain $\Omega_0 \subset \mathbb{R}^2$ as shown in Fig. 1

$$\mathcal{N}(p_s) := \nabla^2 p_s(\mathbf{x}) + k^2 p_s(\mathbf{x}) = 0, \quad \mathbf{x} \in \Omega_0 \tag{4}$$

with the following external boundary conditions enforced on $\Gamma_e^1, \Gamma_e^2, \Gamma_e^3$, and Γ_e^4

$$\begin{aligned} \mathcal{B}_e^1(p_s) &= g_1 \quad \mathbf{x} \in \Gamma_e^1, & \mathcal{B}_e^2(p_s) &= g_2 \quad \mathbf{x} \in \Gamma_e^2 \\ \mathcal{B}_e^3(p_s) &= g_3 \quad \mathbf{x} \in \Gamma_e^3, & \mathcal{B}_e^4(p_s) &= g_4 \quad \mathbf{x} \in \Gamma_e^4 \end{aligned} \tag{5}$$

where p_s is the scattered pressure, $\mathbf{x} = (x, y)$ is the spatial coordinate, the wavenumber $\mathbf{k} = \frac{2\pi f}{c_s} \hat{\mathbf{e}}_k$ is a function of frequency f and the speed of sound c_s , with unit normal vector $\hat{\mathbf{e}}_k = [1, 0]$, and boundary data g_j for $j = 1, 2, 3, 4$. The differential operator \mathcal{N} represents the governing PDE. The external boundary conditions prescribed by the boundary

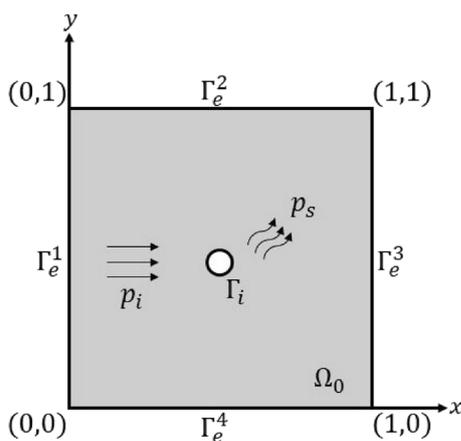


Fig. 1 Schematic of a 2D square acoustic domain Ω_0 of size $[0, 1]m \times [0, 1]m$. A rigid internal scatterer defines an internal boundary Γ_i , while the external boundaries of the domain are $\Gamma_e^1, \Gamma_e^2, \Gamma_e^3$, and Γ_e^4 . The incident pressure (p_i) and scattered pressure (p_s) are also indicated

operators \mathcal{B}_e^j for $j = 1, 2, 3, 4$ can take one of three possible forms: 1) Dirichlet boundary condition $\mathcal{B}_e^j = \mathbf{I}$, 2) Neumann boundary condition $\mathcal{B}_e^j = \frac{\partial}{\partial \mathbf{n}}$, or 3) impedance boundary condition $\mathcal{B}_e^j = \frac{\partial}{\partial \mathbf{n}} + ik\mathbf{I}$. To simulate an infinite acoustic domain, the Sommerfeld radiation condition (i.e. a throw-off boundary) is also enforced by means of the impedance boundary condition with $g_j = 0$ (called the absorption boundary) on $\Gamma_e^1, \Gamma_e^2, \Gamma_e^3$, and Γ_e^4 .

While Eqs. (4)–(5) introduce the governing PDE and the external boundary conditions, the definition of the internal boundary condition (embedded within Ω_0) of the rigid scatterer also plays a critical role. Moreover, the solution to Eq. (4) is determined by the pressure scattered by the rigid scatterer Γ_i and indicated by $p_s = \Re e(p_s) + i\Im m(p_s)$. For a unit amplitude ($p_0 = 1 Pa$) incident plane wave $p_i = p_0 e^{-ik \cdot \mathbf{x}}$ incident on a rigid scatterer Ω_0 , the boundary condition is enforced in the form of a Neumann boundary (sound hard boundary) as follows

$$\mathcal{B}_i(p_s) := \frac{\partial p_s(\mathbf{x})}{\partial \mathbf{n}} - ike^{-ik \cdot \mathbf{x}} = 0 \quad \mathbf{x} \in \Gamma_i \tag{6}$$

where \mathcal{B}_i is the internal boundary operator and \mathbf{n} is the surface normal. Equations (4)–(6) provide the mathematical description of the forward acoustic scattering problem due to a single rigid scatterer Γ_i .

3.2 Development of PINN for rigid body acoustic scattering

Based on the elements discussed in Sects. 2 and 3.1, a b -PINN framework capable of solving the rigid body acoustic scattering problem is presented. Note that this section introduces the b -PINN parameters and the loss function, while a detailed discussion on the network architecture will be presented in the following section.

The b -PINN model consists of a neural network taking the spatial coordinates \mathbf{x} describing the acoustic domain as the network input (see Fig. 2a). This deep learning model is designed to predict the unknown scattered pressure p_s using the network parameterized by a set of parameters θ . The approximate solution provided by the PINN \hat{p}_s such that $\hat{p}_s \approx p_s$. Moreover, the PINN learns to predict \hat{p}_s by finding the optimal θ via minimization of the loss function $\mathcal{L}(\theta)$. In the case of a single scatterer, the loss function \mathcal{L} is defined to enforce Eqs. (4)–(6) as follows

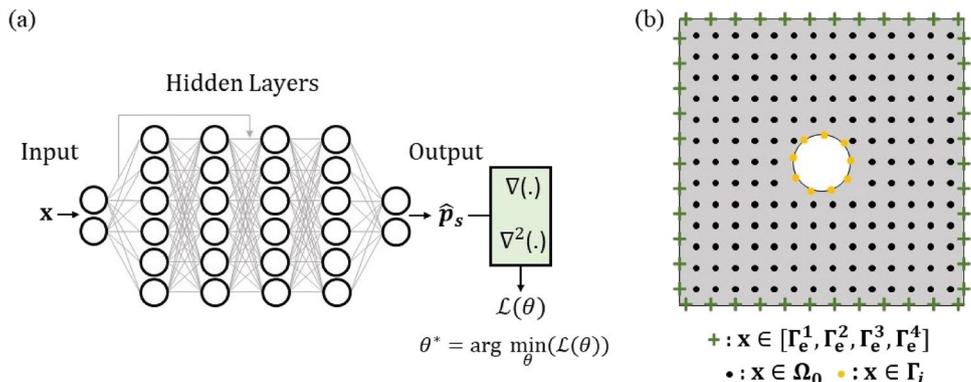
$$\mathcal{L}(\theta) = w_{\mathcal{B}_i} \mathcal{L}_{\mathcal{B}_i}(\theta) + w_{\mathcal{B}_e} \mathcal{L}_{\mathcal{B}_e}(\theta) + w_{\mathcal{N}} \mathcal{L}_{\mathcal{N}}(\theta) \tag{7}$$

where,

$$\begin{aligned} \mathcal{L}_{\mathcal{B}_i}(\theta) &= \frac{1}{N_{b_i}} \sum_{j=1}^{N_{b_i}} |\mathcal{B}_i(\hat{p}_s, \mathbf{x}_{b_i}^j; \theta)|^2 \\ &= \frac{1}{N_{b_i}} \sum_{j=1}^{N_{b_i}} \left| \frac{\partial \hat{p}_s(\mathbf{x}_{b_i}^j)}{\partial \mathbf{n}} - ike^{-ik \cdot \mathbf{x}_{b_i}^j} \right|^2 \\ \mathcal{L}_{\mathcal{B}_e}(\theta) &= \frac{1}{N_{b_e}} \sum_{j=1}^{N_{b_e}} \sum_{l=1}^4 |\mathcal{B}_e(\hat{p}_s, \mathbf{x}_{b_e}^{j,l}; \theta)|^2 \\ &= \frac{1}{N_{b_e}} \sum_{j=1}^{N_{b_e}} \sum_{l=1}^4 \left| \frac{\partial \hat{p}_s(\mathbf{x}_{b_e}^{j,l})}{\partial \mathbf{n}} + ik\hat{p}_s(\mathbf{x}_{b_e}^{j,l}) \right|^2 \\ \mathcal{L}_{\mathcal{N}}(\theta) &= \frac{1}{N_n} \sum_{j=1}^{N_n} |\mathcal{N}(\hat{p}_s, \mathbf{x}_n^j; \theta)|^2 \\ &= \frac{1}{N_n} \sum_{j=1}^{N_n} \left| \nabla^2 \hat{p}_s(\mathbf{x}_n^j) + k^2 \hat{p}_s(\mathbf{x}_n^j) \right|^2 \end{aligned} \tag{8}$$

where, $w_{\mathcal{B}_i}$, $w_{\mathcal{B}_e}$, and $w_{\mathcal{N}}$ are the weighting factors and $\mathcal{L}_{\mathcal{B}_i}$, $\mathcal{L}_{\mathcal{B}_e}$, and $\mathcal{L}_{\mathcal{N}}$ are the mean square errors (MSE) of the residual of the rigid internal boundary conditions (Eq. 6), the external boundary conditions (Eq. 5), and the governing PDE (Eq. 4), respectively. In addition, $\mathbf{x}_{b_i}^j \in \Gamma_i$ is the j^{th} coordinate from the N_{b_i} internal boundary training points,

Fig. 2 **a** High level schematic of the b -PINN architecture to simulate acoustic scattering. The spatial coordinates \mathbf{x} serve as the network input, and the network is trained to predict the scattered pressure \hat{p}_s as output. **b** Schematic illustration of the coordinates used to train the b -PINN and sampled on the external boundaries ($\Gamma_e^1, \Gamma_e^2, \Gamma_e^3, \Gamma_e^4$), inside the acoustic domain (Ω), and on the rigid internal boundary (Γ_i)



$\mathbf{x}_{b_e}^{j,l} \in \Gamma_e^l$ is the j^{th} coordinate on the l^{th} external boundary, where $l \in [1, 2, 3, 4]$, from the N_{b_e} external boundary training points, and $\mathbf{x}_n^j \in \Omega_0$ is the j^{th} coordinate from the N_n number of domain collocation points.

3.3 Baseline-PINN architecture

With the loss functions fully defined to integrate the governing differential equation of the acoustic domain and the corresponding boundary conditions, we can address the setup of the network architecture. In the following, we first elaborate on b -PINN architecture, and later we cover the details for network training.

While PINNs are commonly formed using fully connected neural network layers, recent studies [50, 51] have developed physics-driven deep learning models based on deep residual networks, also called ResNet architectures. These studies highlight the role of ResNets in improving training efficiency and prediction accuracy. Based on these previous findings, we choose a basic architecture for the PINN based on ResNet. Our ResNet architecture has n_r residual blocks, where each residual block contains n_l linear layers of n_w neurons. Note that the residual block is an arrangement of linear neural network layers such that the input to the residual block is combined with its output deeper within the block for better information flow to the next stage of the DNN. Moreover, each linear layer is followed by an adaptive *Sine* activation function. Adaptive activation functions include trainable parameters that are dynamically optimized during the training process to accelerate convergence and improve prediction accuracy in PINNs [52].

A schematic illustration of the proposed architecture is shown in Fig. 2a. The architecture passes the input coordinate $\mathbf{x} = (x, y)$ of size 2×1 through a linear layer of n_w neurons, then the transformed input is passed through the residual blocks to generate \hat{p}_s of size 2×1 as the output. Recall that, while \hat{p}_s is a complex number with real and imaginary components, DNNs cannot operate with complex numbers. Therefore, the two components of \hat{p}_s are stored as separate real numbers in the output vector, such that $\hat{p}_s = [\Re(\hat{p}_s), \Im(\hat{p}_s)]$. In addition, as the solution approximation \hat{p}_s contains both real and imaginary parts, the MSE losses in Eq. (7) are evaluated separately for the real and imaginary components. Automatic differentiation is used to calculate the gradients of \hat{p}_s with respect to the input coordinate \mathbf{x} in the loss function. The PINN architecture described above is called the *baseline-PINN* or b -PINN.

3.3.1 Training

In this section, we will discuss the network training details for the proposed b -PINN. It is important to note that while this section introduces the general parameters and their functions in network training, the specific details regarding the precise numerical values of these network parameters will be described as a part of the numerical experiments presented in the next section.

In order to train the b -PINN, the spatial coordinates \mathbf{x}_{b_i} , \mathbf{x}_{b_e} , and \mathbf{x}_n serve as the input training points on Γ_i , the training points on $[\Gamma_e^1, \Gamma_e^2, \Gamma_e^3, \Gamma_e^4]$, and the collocation points in Ω_0 , respectively. Subsequently, the b -PINN is trained on mini-batches of the input data by sampling input coordinates $\mathbf{x} \in [\mathbf{x}_{b_i}, \mathbf{x}_{b_e}, \mathbf{x}_n]$ from their corresponding distribution of points as shown in Fig. 2b. The network is trained using a two-step optimization process. Initially, we employ the Adam optimizer with a learning rate l_r and train the network for N_e number of epochs. Subsequently, we switch to the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) optimization method, a second-order optimization algorithm, to achieve faster convergence in the later stages of training. Moreover, the PINN is trained until $\mathcal{L}(\theta)$ converges by finding the optimal value of θ through back-propagation via automatic differentiation. The b -PINN is trained and implemented in Python 3.8 using Pytorch API on NVIDIA A100 Tensor Core GPU with 80GB memory.

3.4 Numerical experiments and results

This section studies the performance of the proposed b -PINN through a series of numerical experiments. More specifically, the evaluation focuses on assessing the capabilities of the proposed PINNs in simulating two applications. The first application focuses on examining the performance of the b -PINN in approximating the scattered acoustic field due to an arbitrary rigid scatterer. The second study assesses the ability of the b -PINN to perform high-frequency simulations, which is a common limitation of the existing PINNs.

The performance of the PINNs is measured by their ability to accurately approximate the scattered pressure fields. This assessment involves a direct comparison between the PINN predictions (\hat{p}_s) and their corresponding ground truth (p_s), which is obtained through finite element analysis. In this process, the forward rigid body acoustic scattering problem is simulated using COMSOL Multiphysics®, a FEM software, to evaluate p_s . To ensure the accuracy of the forward simulation model in FE, the rigid internal boundary is enforced through Neumann boundary conditions and the infinite acoustic boundaries are replicated by enforcing absorbing boundary conditions using a perfectly matched layer (PML). It is also important to note that since we are

addressing a steady-state problem, the comparison between network predictions and the FEM ground truth is conducted at individual frequencies. To achieve this, for a characteristic length a_0 of a scatterer, forward simulations with a planar incident wave input are performed at different frequency (or wavelength) ranges corresponding to $ka_0 < 1$, $ka_0 \approx 1$, and $ka_0 > 1$, where the wavenumber $k = \frac{2\pi f}{c_s}$ is a function of frequency f and speed of sound in air $c_s = 343.21 \text{ m/s}$. Broadly speaking, the frequencies are categorised into three ranges, namely, the low-frequency range ($ka_0 < 1$), the mid-frequency range ($ka_0 \geq 1$), the high-frequency range ($ka_0 \gg 1$). Although the current study deals with shapes of different sizes, we have evaluated a suitable reference of $a_0 = 0.1 \text{ m}$ as an average characteristic length across the different scatterer shapes. Based on this, the current study categorizes $f < 500 \text{ Hz}$ ($ka_0 < 1$) as low-frequency range, $f = 500 \text{ Hz} - 1 \text{ kHz}$ ($ka_0 \geq 1$) as mid-frequency range, and $f > 1 \text{ kHz}$ ($ka_0 \gg 1$) as high-frequency range. Given the relative straightforwardness of low-frequency applications in the context of a forward solver, this study emphasizes the use of the proposed PINNs for simulating rigid acoustic scattering specifically for applications in the mid- and high-frequency ranges.

Further, we introduce the following metrics to assess the overall performance of the networks

1. **Relative L_2 -error:** The following error metric is used to assess the relative prediction quality in an average sense

$$L_2 = \frac{\left(\sum_{i=1}^N |p_s^i - \hat{p}_s^i|^2\right)^{1/2}}{\left(\sum_{i=1}^N |p_s^i|^2\right)^{1/2}} \tag{9}$$

where, p_s^i and \hat{p}_s^i are the ground truth pressure value and the predicted pressure value, respectively, at the i^{th} location in the corresponding fields. Additionally, N represents the number of locations where the pressure

fields are compared. The lower the L_2 -error better the network prediction.

2. **R^2 -score:** The coefficient of determination also called the R^2 -score is used to assess the accuracy of the network prediction. The R^2 -score measures the average variation in the predicted field (\hat{p}_s) with respect to the ground truth field (p_s) as follows

$$R^2 = 1 - \frac{\sum_{i=1}^N (p_s^i - \hat{p}_s^i)^2}{\sum_{i=1}^N (p_s^i - \bar{p}_s)^2} \tag{10}$$

The maximum score of $R^2_{max} = 1$ is achieved when the prediction matches exactly with the ground truth. Hence, the closer the R^2 -score is to R^2_{max} , the higher the prediction accuracy.

3. **Point-wise error:** The following error metric assesses the point-wise prediction quality of the pressure field

$$\mathcal{E}_p = |p_s - \hat{p}_s| \tag{11}$$

Here, \mathcal{E}_p measures the absolute error to provide an error map highlighting variation between p_s and \hat{p}_s fields across the spatial domain.

3.4.1 Application to arbitrary-shaped rigid scatterers

In this first application, we study the performance of the b -PINN by investigating the prediction accuracy of the network for a single scatterer embedded within Ω_0 . More specifically, we choose to train the b -PINN to simulate the forward scattering problem for arbitrary rigid scatterer shapes. Table 1 introduces the values of the network parameters chosen for this application denoted as Case 1.

Figure 3 illustrates the prediction accuracy of b -PINN for three random scatterer shapes. Within Fig. 3A–C, the section marked (a) highlights the forward geometry model used for the simulation, (b) indicates the corresponding loss variation with training epochs using both Adam and L-BFGS optimizer, and (c) illustrates the prediction accuracy comparison for both real and imaginary scattered pressure

Table 1 Summary of key network parameters used to develop the b -PINN architecture for Case 1: arbitrary rigid scatterer simulation and Case 2: high-frequency simulation applications at $f = 2 \text{ kHz}$, 3.5 kHz , and 5 kHz

Parameter	Case 1	Case 2	Case 2	Case 2
		$f = 2 \text{ kHz}$	$f = 3.5 \text{ kHz}$	$f = 5 \text{ kHz}$
Hidden layer neurons (n_w)	50	50	100	125
Number of residual blocks (n_r)	2	2	2	3
Number of layers in each residual block (n_l)	3	3	5	5
Training points on Γ_i (N_{b_i})	100	100	100	100
Training points on Γ_e (N_{b_e})	1000	1000	1000	1000
Collocation points in Ω_0 (N_n)	10,000	10,000	10,000	10,000
Learning rate (l_r)	1e-4	1e-4	1e-4	1e-4

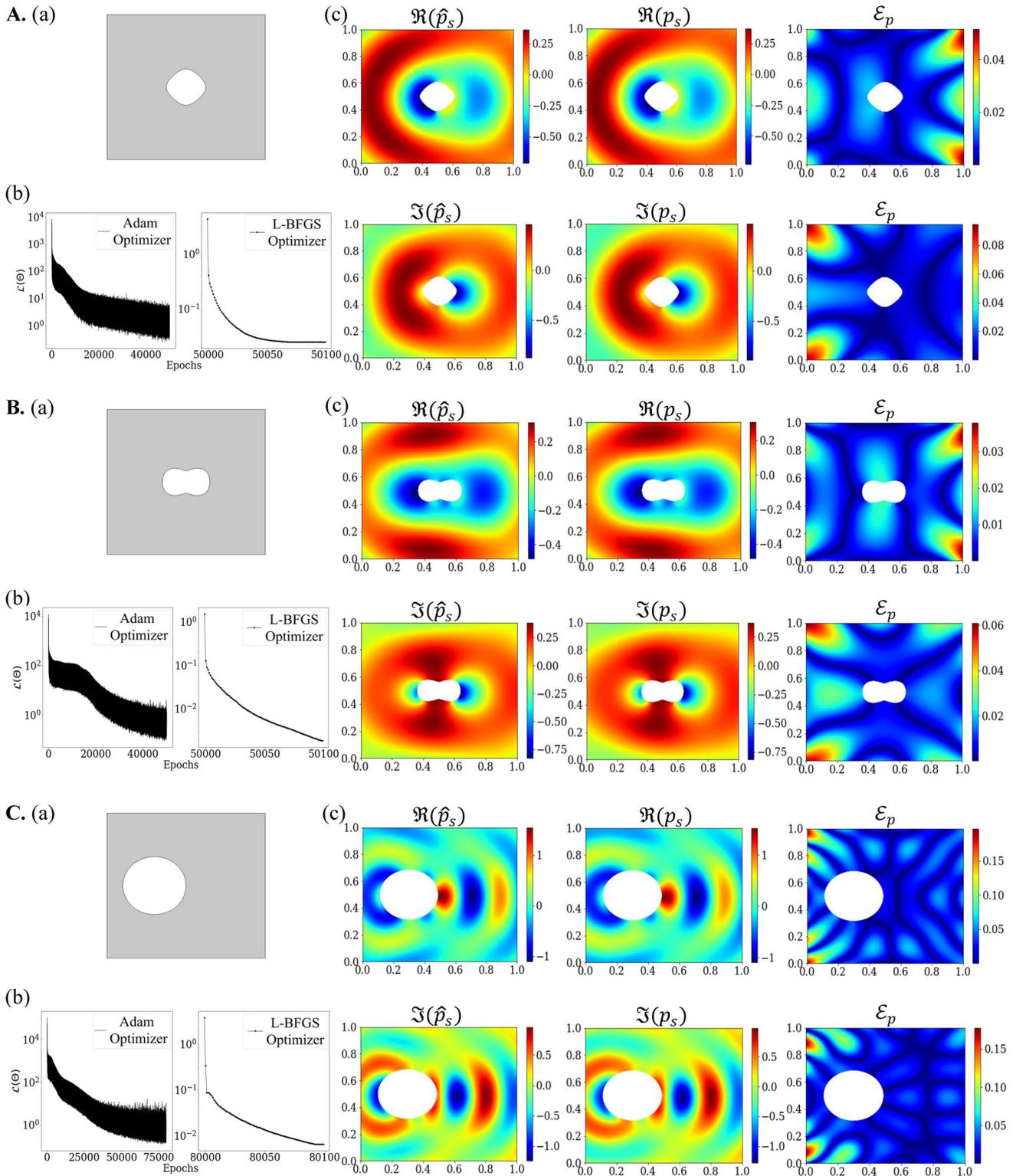


Fig. 3 Case 1- Arbitrary-shaped rigid scatterers: Performance analysis of *b*-PINN for three scatterers described by an arbitrary shaped boundary (see A–C). The simulations in A, B, and C are performed for harmonic incident plane waves at $f = 500 \text{ Hz}$, $f = 500 \text{ Hz}$, and $f = 1 \text{ kHz}$, respectively. For each sample shape, the subplots a, b, and c represent: a the geometry of the acoustic domain and of the embed-

ded rigid scatterer; b plots of the variation in the loss function with the training epochs (loss variations obtained using both Adam and L-BFGS optimizers are presented); c amplitude maps of the real (top) and imaginary (bottom) components of the predicted pressure field (\hat{p}_s), true pressure field (p_s), and the point-wise error (\mathcal{E}) between \hat{p}_s and p_s

components. Note that this is the general format used for all the figures in the rest of the numerical experiments as well. While Fig. 3A and B compare the prediction accuracy for scattered fields simulated by the b -PINN trained with $N_e = 50,000$ epochs at $f = 500$ Hz, Fig. 3C illustrates the performance of the b -PINN trained with $N_e = 80,000$ epochs in simulating a scattered pressure field at $f = 1$ kHz. The scattered fields generated for the shapes in Fig. 3A and B calculates [$L_2 = 0.0725, R^2 = 0.9937$] and [$L_2 = 0.0749, R^2 = 0.9928$], respectively. It is important to highlight that both the L_2 -error and R^2 -score are evaluated by averaging the L_2 and R^2 values of the real and imaginary scattered pressure components for each simulation. The high R^2 -score and low L_2 -error for the simulations in Fig. 3A and B highlights the ability of the b -PINN to accurately approximate the scattered field due to an arbitrary scatterer shape. Further, the performance evaluation for the simulation at $f = 1$ kHz in Fig. 3C calculates [$L_2 = 0.095, R^2 = 0.9909$]. This also showcases the ability of the b -PINN to accurately simulate the wavefield for different incident wave frequencies in the mid-frequency range. However, a comparison of the performance metrics between the simulations at $f = 500$ Hz and $f = 1$ kHz indicates a marginally higher prediction error at higher frequencies. This can be attributed to the inability of the same number of network weights (n_w) to capture finer wavefield features at higher frequencies.

3.4.2 Application to high-frequency scattering

This application studies the performance of the b -PINN by investigating the prediction accuracy of the network for a single scatterer embedded within Ω_0 at high frequency. More specifically, we choose to train the b -PINN to simulate the forward scattering problem for arbitrary rigid scatterer shapes at randomly chosen frequencies $f = 2$ kHz and 3.5 kHz in the high-frequency range. Table 1 introduces the values of some of the essential network parameters chosen for this application denoted as Case 2.

In the context of our discussion in the previous section, the increase in network prediction error with frequency is more evident in the high-frequency range. To address this, we observed that an increase in network prediction accuracy can be achieved by introducing more n_w and adjusting N_e . Therefore, the b -PINN is customized for specific frequencies by modulating n_w and N_e , as detailed in Table 1. Subsequently, the b -PINN is trained to simulate the scattered pressure field for an arbitrary shape at $f = 2$ kHz with $n_w = 50$ and $N_e = 100,000$ as shown in Fig. 4A. Similarly,

the b -PINN is also trained to simulate the scattered pressure field for another arbitrary shape at $f = 3.5$ kHz with $n_w = 100$ and $N_e = 200,000$ as shown in Fig. 4B. While the performance metrics for Fig. 4A are calculated as [$L_2 = 0.0682, R^2 = 0.9953$], the performance metrics for Fig. 4B are calculated as [$L_2 = 0.04675, R^2 = 0.9978$]. The low L_2 -error and high R^2 -score for both simulations highlight the key ability of the b -PINN to approximate scattered pressure fields in the high-frequency range. Moreover, the constant number of training and collocation points used for all the numerical experiments in the mid- and high-frequency ranges highlight the discretization-independence of the proposed PINN across frequency ranges. However, it is important to highlight that as we move to higher frequencies, the scattering simulations use up larger computational resources as we have to design deeper networks (Table 1) to capture extensive scattered field features at high frequencies.

Furthermore, we conducted simulations at $f = 5$ kHz by training b -PINN for $N_e = 200,000$ epochs by sampling from 10,000 collocation points and computed performance metrics to be [$L_2 = 0.2595, R^2 = 0.9327$]. As we study very high frequencies, i.e. $f \geq 5$ kHz ($ka_0 \geq 10$), it is observed that an increased sampling distribution size is essential for better performance, thereby needing a finer sampling space to capture more detailed scattered field features. In summary, this study demonstrated the capability of the b -PINN to simulate in the high-frequency range. However, the prediction accuracy reduces as we transition to very high frequency values, as indicated by the high error metrics for the simulation at $f = 5$ kHz.

4 Multiple acoustic scattering

This section introduces the benchmark problem used to develop and test the superposition-PINN or s -PINN architecture. First, some theoretical and mathematical preliminaries of multiple scattering in acoustics are presented. Then, the concept of physics-informed neural network (PINN) formulated for the specific benchmark problem is introduced. Finally, the performance of the proposed network is studied through a series of numerical experiments.

4.1 Problem description

A natural extension of the previous acoustic scattering problem involves the case of multiple rigid scatterers. Consider a domain Ω_0 including multiple scatterers with rigid internal boundaries $\Gamma_i^1, \Gamma_i^1, \dots, \Gamma_i^{N_s}$. The scattered field radiated

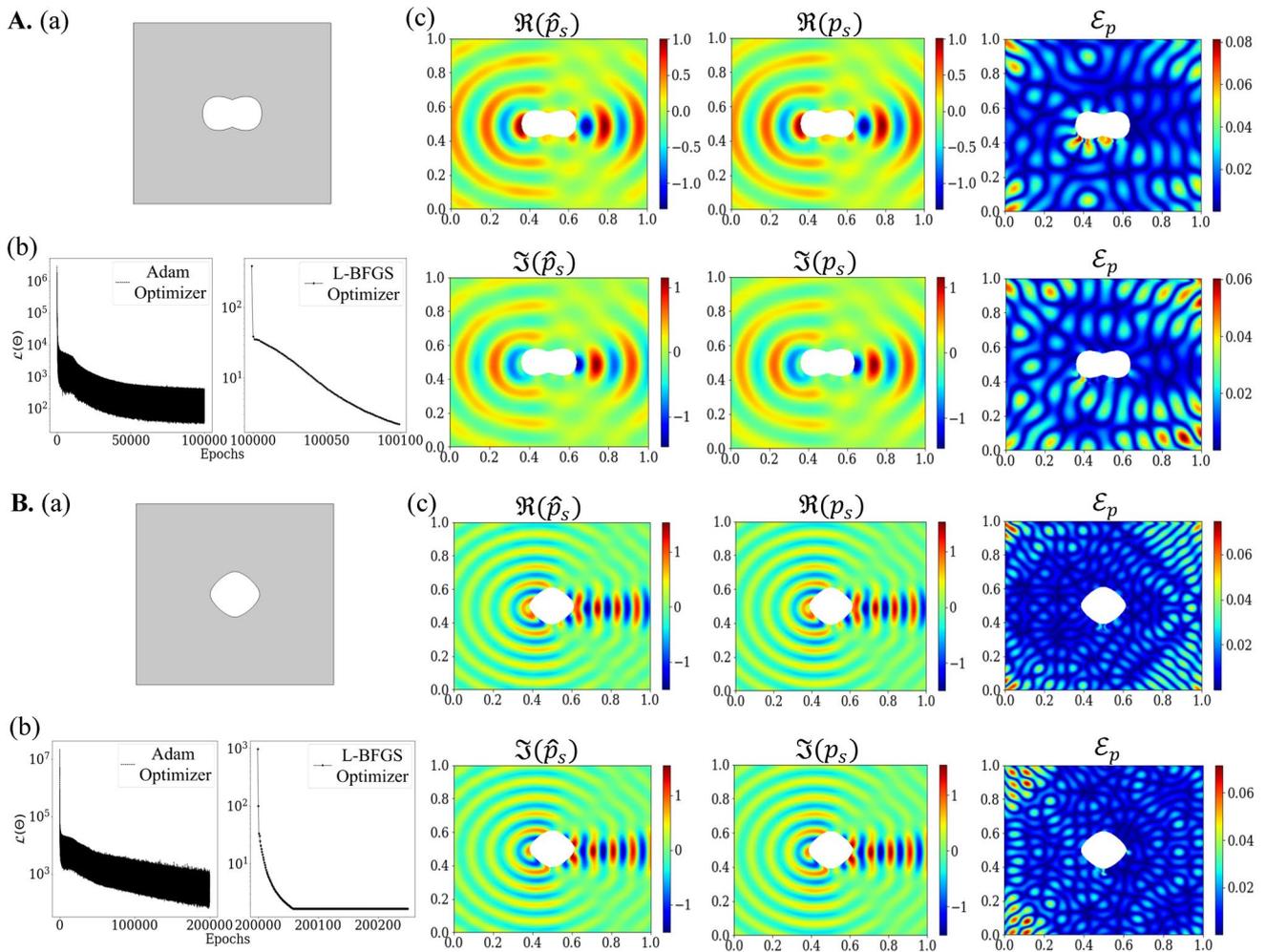


Fig. 4 Case 2- High frequency: Performance analysis of *b*-PINN for high-frequency applications (see **A–B**). The simulations in **A** and **B** are performed for harmonic incident plane waves at $f = 2 \text{ kHz}$ and $f = 3.5 \text{ kHz}$, respectively. For each sample shape, the subplots **a**, **b**, and **c** represent: **a** the geometry of the acoustic domain and of the embedded rigid scatterer; **b** plots of the variation of the loss function

outward from each scatterer is $p_s^1, p_s^2, \dots, p_s^{N_s}$. A sample illustration including four scatterers ($N_s = 4$) is shown in Fig. 5. According to the *superposition principle*, the scattered pressure field at each point in the acoustic domain is the sum of the fields due to each scatterer [53, 54]. According to the theory of multiple scattering, the total scattered pressure (p_s) within Ω_0 is

$$p_s = \sum_{j=1}^{N_s} p_s^j \tag{12}$$

where p_s^j is the scattered field radiated outward from the j^{th} scatterer. By substituting Eq. (12) in Eqs. (4)–(6), we can generalize the governing equations to the case of N_s scatterers. Leveraging the properties of linear differential

with the training epochs (loss variations obtained using both Adam and L-BFGS optimizers are presented); **c** amplitude maps of the real (top) and imaginary (bottom) components of the predicted pressure field (\hat{p}_s), true pressure field (p_s), and the point-wise error (\mathcal{E}) between \hat{p}_s and p_s

operators, the governing Helmholtz equation for the multiple scattering case is

$$\begin{aligned} \mathcal{N}\left(\sum_{j=1}^{N_s} p_s^j\right) &:= \nabla^2\left(\sum_{j=1}^{N_s} p_s^j(\mathbf{x})\right) + k^2\left(\sum_{j=1}^{N_s} p_s^j(\mathbf{x})\right) \\ &\equiv \sum_{j=1}^{N_s} \left(\nabla^2 p_s^j(\mathbf{x}) + k^2 p_s^j(\mathbf{x})\right) = 0 \end{aligned} \tag{13}$$

The radiation boundary conditions are enforced on the external boundaries $\Gamma_e^1, \Gamma_e^2, \Gamma_e^3$, and Γ_e^4

$$\begin{aligned}
 \mathcal{B}_e^1(\sum_{j=1}^{N_s} p_s^j) &:= \sum_{j=1}^{N_s} \left(\frac{\partial}{\partial \mathbf{n}} p_s^j(\mathbf{x}) + ikp_s^j(\mathbf{x}) \right) = 0 \quad \mathbf{x} \in \Gamma_e^1 \\
 \mathcal{B}_e^2(\sum_{j=1}^{N_s} p_s^j) &:= \sum_{j=1}^{N_s} \left(\frac{\partial}{\partial \mathbf{n}} p_s^j(\mathbf{x}) + ikp_s^j(\mathbf{x}) \right) = 0 \quad \mathbf{x} \in \Gamma_e^2 \\
 \mathcal{B}_e^3(\sum_{j=1}^{N_s} p_s^j) &:= \sum_{j=1}^{N_s} \left(\frac{\partial}{\partial \mathbf{n}} p_s^j(\mathbf{x}) + ikp_s^j(\mathbf{x}) \right) = 0 \quad \mathbf{x} \in \Gamma_e^3 \\
 \mathcal{B}_e^4(\sum_{j=1}^{N_s} p_s^j) &:= \sum_{j=1}^{N_s} \left(\frac{\partial}{\partial \mathbf{n}} p_s^j(\mathbf{x}) + ikp_s^j(\mathbf{x}) \right) = 0 \quad \mathbf{x} \in \Gamma_e^4
 \end{aligned} \tag{14}$$

The Neumann boundary conditions are imposed on the rigid internal boundaries of each scatterer

$$\begin{aligned}
 \mathcal{B}_i(\sum_{j=1}^{N_s} p_s^j(\mathbf{x})) &:= \frac{\partial}{\partial \mathbf{n}} \left(\sum_{j=1}^{N_s} p_s^j(\mathbf{x}) \right) - ike^{-ik \cdot \mathbf{x}} = 0 \quad \mathbf{x} \in \Gamma_i \\
 &:= \sum_{j=1}^{N_s} \frac{\partial p_s^j(\mathbf{x})}{\partial \mathbf{n}} - ike^{-ik \cdot \mathbf{x}} = 0
 \end{aligned} \tag{15}$$

where Γ_i represents the ensemble of rigid internal boundaries such that $\Gamma_i = \Gamma_i^1 \cup \Gamma_i^2 \dots \cup \Gamma_i^{N_s}$. In summary, Eqs. (13)–(15) provide the mathematical description of the forward multiple scattering problem.

4.2 Development of PINN for multiple scattering

Based on the discussions in Sects. 2 and 4.1, this section presents a neural network framework, with a parameter set θ , capable of solving the multiple scattering problem in acoustics. For the multiple scattering case, the loss function \mathcal{L} is defined to enforce Eqs. (13)–(15) as follows

$$\mathcal{L}(\theta) = w'_{B_i} \mathcal{L}'_{B_i}(\theta) + w'_{B_e} \mathcal{L}'_{B_e}(\theta) + w'_{\mathcal{N}} \mathcal{L}'_{\mathcal{N}}(\theta) \tag{16}$$

where, \mathcal{L}'_{B_i} , \mathcal{L}'_{B_e} , and $\mathcal{L}'_{\mathcal{N}}$ are the mean square errors (MSE) of the residual of the rigid internal boundary conditions (Eq. 15), the external boundary conditions (Eq. 14), and the governing PDE (Eq. 13), with w'_{B_i} , w'_{B_e} , and $w'_{\mathcal{N}}$ as the corresponding weighting factors. Here,

$$\begin{aligned}
 \mathcal{L}'_{B_i}(\theta) &= \frac{1}{N_{b_i}} \sum_{j=1}^{N_{b_i}} \left| \mathcal{B}_i \left(\sum_{q=1}^{N_s} \hat{p}_s^q, \mathbf{x}_{b_i}^j; \theta \right) \right|^2 \\
 &= \frac{1}{N_{b_i}} \sum_{j=1}^{N_{b_i}} \left| \frac{\partial}{\partial \mathbf{n}} \left(\sum_{q=1}^{N_s} \hat{p}_s^q(\mathbf{x}_{b_i}^j) \right) - ike^{-ik \cdot \mathbf{x}_{b_i}^j} \right|^2 \\
 \mathcal{L}'_{B_e}(\theta) &= \frac{1}{N_{b_e}} \sum_{j=1}^{N_{b_e}} \sum_{l=1}^4 \left| \mathcal{B}_e \left(\sum_{q=1}^{N_s} \hat{p}_s^q, \mathbf{x}_{b_e}^{j,l}; \theta \right) \right|^2 \\
 &= \frac{1}{N_{b_e}} \sum_{j=1}^{N_{b_e}} \sum_{l=1}^4 \left| \sum_{q=1}^{N_s} \left(\frac{\partial \hat{p}_s^q(\mathbf{x}_{b_e}^{j,l})}{\partial \mathbf{n}} + ik \hat{p}_s^q(\mathbf{x}_{b_e}^{j,l}) \right) \right|^2 \\
 \mathcal{L}'_{\mathcal{N}}(\theta) &= \frac{1}{N_n} \sum_{j=1}^{N_n} \left| \mathcal{N} \left(\sum_{q=1}^{N_s} \hat{p}_s^q, \mathbf{x}_n^j; \theta \right) \right|^2 \\
 &= \frac{1}{N_n} \sum_{j=1}^{N_n} \left| \sum_{q=1}^{N_s} \left(\nabla^2 \hat{p}_s^q(\mathbf{x}_n^j) + k^2 \hat{p}_s^q(\mathbf{x}_n^j) \right) \right|^2
 \end{aligned} \tag{17}$$

4.3 Superposition-PINN architecture

The b -PINN architecture (Fig. 2a), introduced in Sect. 3, is capable of calculating the scattered pressure field due to either single or multiple scatterers by minimizing the loss functions presented in Eq. (7) (for the single scattering case) and in Eq. (16) (for the multiple scattering case). In a multiple scattering scenario, our numerical results revealed that the prediction accuracy of the b -PINN decreases as the number of scatterers increases (see the comparative study in Sect. 4.4). This reduction in accuracy, accompanied also by an increase in computational time, could be attributed to the limited ability of the single network b -PINN to minimize \mathcal{L} . Indeed, when the number of scatterers increases, the high

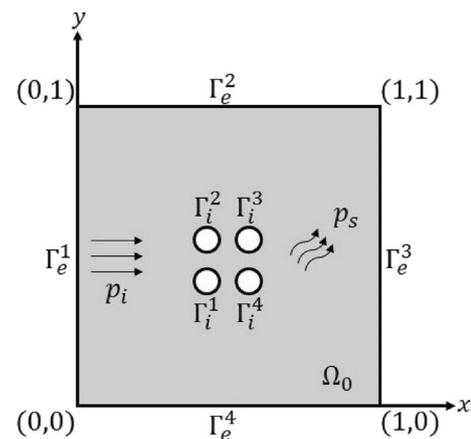


Fig. 5 Schematic of 2D square acoustic domain Ω_0 of size $[0, 1]m \times [0, 1]m$ embedded with multiple scatterers. The domain represents rigid internal boundary $\Gamma_i = \Gamma_i^1 \cup \Gamma_i^2 \cup \Gamma_i^3 \cup \Gamma_i^4$ and external boundaries $\Gamma_e^1, \Gamma_e^2, \Gamma_e^3$, and Γ_e^4 . Further, the incident pressure (p_i) and scattered pressure (p_s) are also indicated

prediction error observed during the initial network training phase, as characterized by high \mathcal{L}'_{B_i} values in \mathcal{L} , propagates across the spatial domain. Consequently, there is a limited ability to control the error propagation while using a single network. To address this issue, we introduce a complementary PINN architecture, that leverages the superposition principle in linear acoustics; in the following, this network will be referred to as the *superposition-PINN* or *s-PINN*.

The main goal of the *s-PINN* architecture is to overcome the limitations of the *b-PINN* when used in the context of multiple scattering. The basic concept at the foundation of *s-PINN* is to leverage a domain decomposition approach such that separate *b-PINNs* can be used simultaneously to obtain the response in each scatterer sub-domain. Subsequently, the scattered fields approximated by the individual *b-PINNs* are integrated using the superposition principle. Using this domain decomposition approach, each *b-PINN* within the *s-PINN* is constrained to simulate only a specific subset of the scatterers, thereby preventing high error propagation within each *b-PINN*. In a multiple scattering scenario, an increase in the total number of scatterers is

accommodated by introducing additional *b-PINNs* without increasing the maximum number of scatterers each *b-PINN* can handle. Consequently, the *s-PINN* is not affected by the inaccuracies due to the error propagation. However, the scalability of *s-PINN* depends on the number of *b-PINNs* that can be simulated simultaneously on a given hardware configuration. The rest of this section elaborates on the implementation of the *s-PINN*.

We define a scatterer sub-domain as an acoustic domain Ω_0 including n_s scatterers, which represent a subset of the total number of scatterers N'_s . The number of scatterers in a scatterer sub-domain is typically $n_s \geq 1$, but it should be chosen so that the *b-PINNs* can provide predictions with the desired accuracy without deteriorating the computational cost to train the network. Fig. 6 illustrates a *s-PINN* architecture to simulate a lattice of $N'_s = 16$ rigid scatterers divided across four sub-domains, each containing $n_s = 4$ rigid scatterers. The *s-PINN* learns to approximate the total scattered field by simultaneously training all the *b-PINNs*. A *s-PINN* with j *b-PINNs* is parameterized by a training parameter set θ such that $\theta = \theta^1 \cup \theta^2 \dots \cup \theta^j$, where θ^j denotes the parameter set corresponding to the j^{th} *b-PINN*. Further,

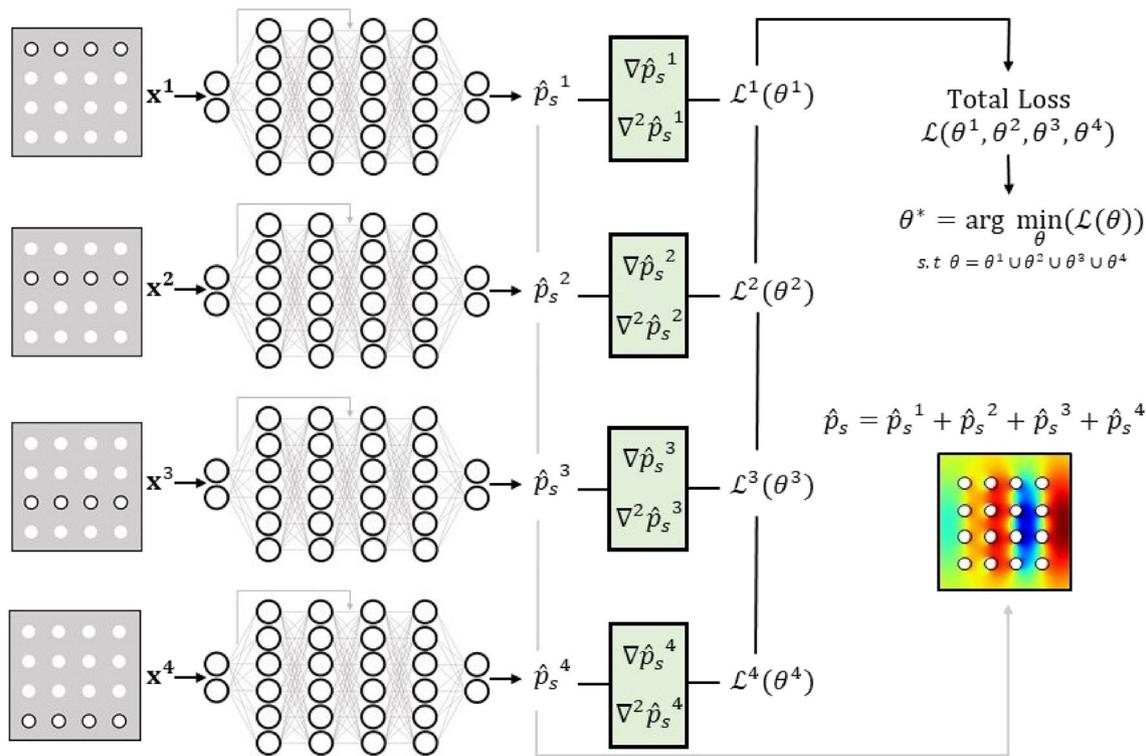


Fig. 6 A schematic representation of the *s-PINN* architecture to simulate acoustic scattering in domains with multiple scatterers. In particular, this framework simulates a 4×4 lattice by simultaneously training four *b-PINNs*, each containing four rigid scatterers. Every j^{th} *b-PINN* is trained to predict the scattered pressure \hat{p}_s^j for the corresponding input coordinates \mathbf{x}^j by finding the optimal parameter set θ^j

through minimization of the total loss $\mathcal{L}(\theta)$ which is a function of the loss components $\mathcal{L}^j(\theta^j)$ evaluated by the individual *b-PINN*. Finally, based on the concept of superposition in linear acoustics, the scattered pressure is evaluated as the sum of the scattered pressures simulated by the individual *b-PINNs*, i.e. $\hat{p}_s = \hat{p}_s^1 + \hat{p}_s^2 + \hat{p}_s^3 + \hat{p}_s^4$

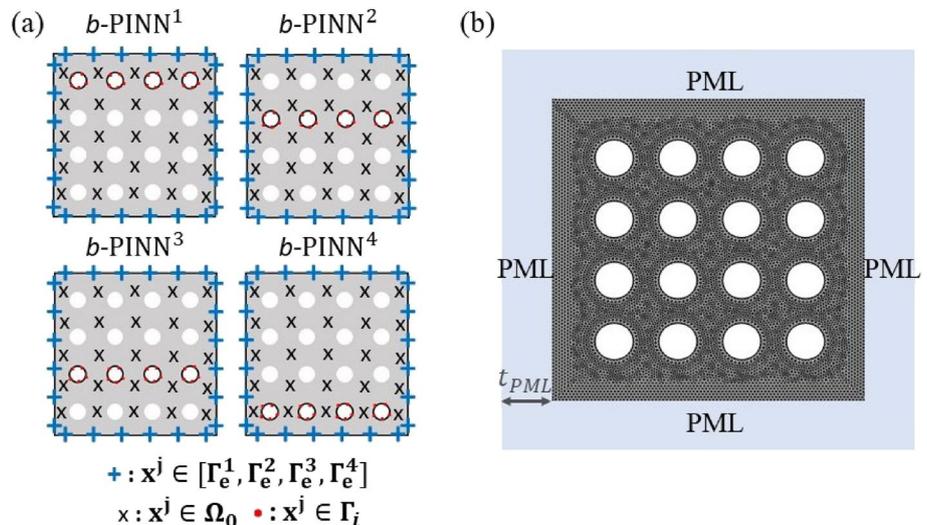
the optimal θ is obtained by minimizing the superposition theorem-based total loss \mathcal{L} defined in Eq. (16). Note that N_s in Eq. (16) is generalized as the number of b -PINNs within the s -PINN. Although N_s was defined as the total number of scatterers in Sect. 4.1, in s -PINN N_s can be less than or equal to the total number of scatterers, i.e. $N_s \leq N'_s$. After training, each b -PINN learns to predict the corresponding scattered field \hat{p}_s^j , where the index j represents j^{th} sub-domain. Thereafter, based on Eq. (12), the total scattered pressure (\hat{p}_s) due to the N'_s scatterers is evaluated as the superposition of the scattered fields predicted by N_s b -PINNs, i.e. $\hat{p}_s = \sum_{j=1}^{N_s} \hat{p}_s^j$.

To further clarify the basic principle of s -PINN, we present the example in Fig. 6. Consider the 4×4 lattice of scatterers ($N'_s = 16$) as described above. The s -PINN contains $N_s = 4$ b -PINNs corresponding to an equal number of sub-domains, each containing $n_s = 4$ rigid scatterers. In this approach, individual b -PINN is responsible for simulating the scattered field due to a subset of four scatterers within Ω_0 , as depicted in Fig. 6. Alternatively, each b -PINN enforces the rigid internal boundary condition for the selected number of scatterers ($n_s = 4$), as highlighted in bold within the scatterer sub-domains input to the individual b -PINNs in Fig. 6. The proposed approach can effectively address the challenges of simulating multiple scattering by reducing the error propagation within each b -PINN. This is a significant improvement compared to training a single b -PINN to simulate the scattered field due to all 16 rigid scatterers. Finally, by applying the superposition theorem in linear acoustics, the trained s -PINN can estimate the total scattered pressure field \hat{p}_s as the summation of the scattered fields $\hat{p}_s^1, \hat{p}_s^2, \hat{p}_s^3, \hat{p}_s^4$ approximated by the b -PINNs corresponding to the scatterer sub-domains with input coordinates $\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^4$, i.e. $\hat{p}_s = \hat{p}_s^1 + \hat{p}_s^2 + \hat{p}_s^3 + \hat{p}_s^4$.

Comparison to other domain decomposition based approaches

Since s -PINN is introduced as a domain decomposition approach, we briefly compare s -PINN to other methods in this category. The extended physics-informed neural network (XPINN) [55] is likely the most representative method leveraging a domain decomposition-based framework. In XPINN, the governing equations and continuity conditions at the sub-domain interfaces are directly enforced. Unlike XPINN, the s -PINN enforces dual-level integration of underlying physics as it implicitly interconnects the sub-domains through the network architecture that implements the superposition theorem and enforces the governing equations. This concept differs from XPINN, which interconnects sub-domains by enforcing the continuity conditions at the interfaces along with the governing equations in a single loss function. The key differences in the implementation of XPINN and the s -PINN are summarized as follows: 1) *Larger networks*: XPINN requires larger networks to handle finer training discretization due to the need for additional sample points on the sub-domain interfaces. Moreover, careful network hyperparameter tuning is necessary. 2) *Complex network loss implementation*: Implementing network loss with additional terms capturing the interface continuity is more challenging compared to the straightforward implementation of s -PINN. The use of multiple terms in the loss function introduces several challenges: (i) The additional loss terms represent competing objectives that must be minimized simultaneously. (ii) Each additional loss term adds to the complexity of the optimization landscape, making it highly non-convex. This complexity makes it harder for the optimization algorithm to converge to a global minimum. (iii) The loss terms often operate on different scales, necessitating the use of dynamic weighting factors to balance them effectively. 3) *Careful domain separation*: XPINN training

Fig. 7 **a** Schematic illustration of the coordinates sampled on the external boundaries ($\Gamma_e^1, \Gamma_e^2, \Gamma_e^3, \Gamma_e^4$), inside the acoustic domain (Ω), and rigid internal boundary (Γ_i) for training a s -PINN with 4 scatterer sub-domains. **b** The meshed geometry for a sample lattice with multiple rigid scatterers enforcing PML boundaries for finite element (FE) analysis



requires careful splitting into sub-domains without overlaps, whereas s -PINN only separates the different scatterers, not the entire domain.

4.3.1 Training

The s -PINN learns by simultaneously training N_s b -PINNs. Every j^{th} b -PINN is trained with the spatial coordinates $\mathbf{x}_{b_i}^j$, \mathbf{x}_{b_e} , and \mathbf{x}_n as the input training points on Γ_i^j , the training points on $[\Gamma_e^1, \Gamma_e^2, \Gamma_e^3, \Gamma_e^4]$, and the collocation points in Ω_0 , respectively. Subsequently, the b -PINN is trained on mini-batches of the input data by sampling input coordinates $\mathbf{x}^j \in [\mathbf{x}_{b_i}^j, \mathbf{x}_{b_e}, \mathbf{x}_n]$ from their corresponding distribution of points. For example, Fig. 7a illustrates the sampling strategy for a s -PINN with $N_s = 4$ b -PINNs. Note that within the inputs to every j^{th} b -PINN, only the coordinates of the Γ_i of the rigid scatterers vary, while the Γ_e and Ω_0 remain the same for all b -PINNs. The s -PINN is also trained with a learning rate l_r using the Adam optimizer followed up with the L-BFGS optimization method. Moreover, the s -PINN is trained until \mathcal{L}'_{B_i} , \mathcal{L}'_{B_e} , and \mathcal{L}'_N converges by finding the optimal value of $\theta = \theta^1 \cup \theta^2 \dots \theta^j$ for all the b -PINNs through backpropagation.

The s -PINN is trained and implemented in Python 3.8 using Pytorch API on NVIDIA A100 Tensor Core GPU with 80GB memory.

4.4 Numerical experiments and results

This section explores and evaluates the performance of the proposed s -PINN architecture through a series of numerical experiments. The ability of the s -PINN to address multiple scattering scenarios is thoroughly examined, with a specific focus on its effectiveness in simulating the scattered acoustic pressure fields within acoustic lattice structures. Acoustic lattices (such as those at the basis of acoustic metamaterials) play an important role in practical applications because they are typically used to achieve wavefront manipulation and control. These structures often employ scatterers in periodic arrangements forming grids or lattices. The specific characteristics of the individual scatterers (e.g. shape, size, etc.) and of the grid (e.g. spacing, arrangement, etc.) are critical to determine the equivalent medium properties of the resulting assembly. These properties can be targeted and tailored to manipulate the acoustic wavefront as it propagates through the material. The periodic lattice example is selected to present the characteristics of the s -PINN.

As introduced in Sect. 3.4, the performance assessment of PINNs involves a direct comparison between the PINN predictions (\hat{p}_s) and their corresponding ground truth (p_s), which is obtained through finite element analysis. In this

process, the forward rigid body acoustic scattering problem is simulated using COMSOL Multiphysics® to evaluate p_s . The FEM simulation setup for a sample multiple scattering scenario is shown in Fig. 7b.

In these numerical experiments, we investigate the performance of the proposed networks when simulating acoustic lattices in the mid-frequency range. Specifically, we assess the performance of both the b -PINN and s -PINN for the same multiple scattering simulations. Additionally, we compare the performances of both networks to emphasize the significance of the superposition network. Note that while the s -PINN architecture can be implemented using parallel computations, the multiple scattering simulations presented below are not evaluated using a parallel implementation.

4.4.1 Multiple scattering simulation using b -PINN

The schematics in Fig. 8 illustrate the prediction accuracy of the b -PINN for multiple scattering simulations. Fig. 8A highlights the prediction assessment for a multiple scattering problem with two scatterers. Here, in addition to the network parameters represented in Table 2, the b -PINN is trained for $N_e = 40,000$ epochs with a step learning rate of $l_r = 1e - 3$ for the first 20,000 epochs and $l_r = 1e - 4$ for the rest of the training with Adam Optimizer (Fig. 8A(b)). The comparison between \hat{p}_s and p_s in Fig. 8A evaluates [$L_2 = 0.072, R^2 = 0.9942$], indicating a good prediction accuracy of the b -PINN for a multiple scattering problem with two scatterers. Further, the b -PINN is trained to simulate a multiple scattering problem with four scatterers embedded in Ω_0 as shown in Fig. 8B(a). Here, the b -PINN is trained for $N_e = 50,000$ epochs with a step learning rate of $l_r = 1e - 3$ for the first 20,000 epochs and $l_r = 1e - 4$ for the rest of the training with Adam Optimizer. A direct comparison between the \hat{p}_s and p_s fields in Fig. 8B(c) presents low accuracy in the b -PINN approximation for multiple scattering with four scatterers. Moreover, the error

Table 2 Summary of key network parameters used to develop the b -PINN and s -PINN architecture for multiple scattering simulation due to a 4×1 scatterer lattice at $f = 500$ Hz

Parameter	Case 3 b -PINN	Case 3 s -PINN			
		PINN ¹	PINN ²	PINN ³	PINN ⁴
4×1 lattice					
n_w	100	100	100	100	100
n_r	2	2	2	2	2
n_l	5	4	4	4	4
N_{b_i}	100	100	100	100	100
N_{b_e}	1000	1000	1000	1000	1000
N_n	10,000	10,000	10,000	10,000	10,000
l_r	1e-3, 1e-4	1e-3, 1e-4			

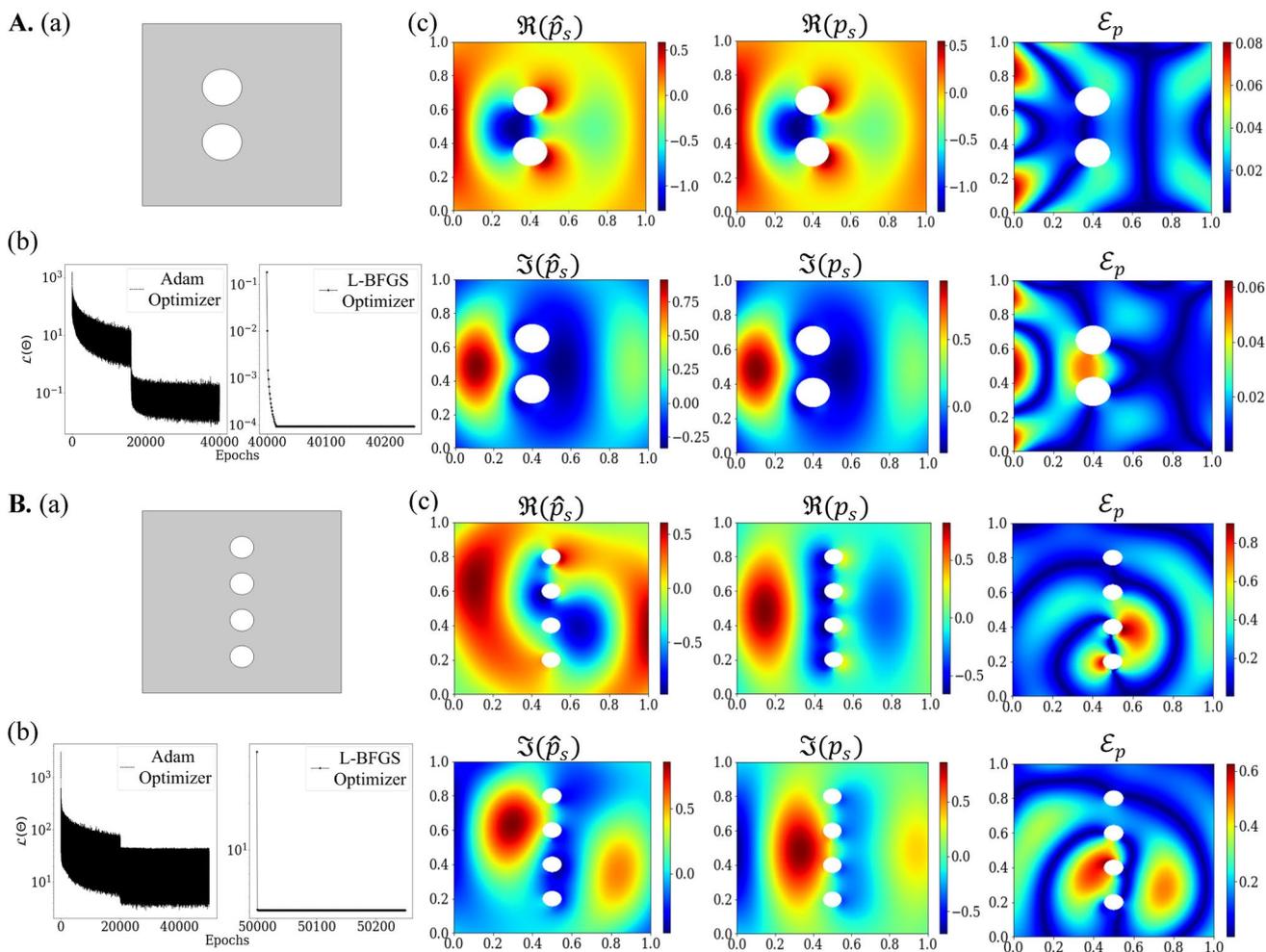


Fig. 8 Case 3- *b*-PINN: Illustration of the performance analysis of *b*-PINN for multiple scattering applications in **A–B**. The simulations in **A** and **B** are performed for planar wavefields at $f = 500$ Hz incident on four scatterers and four scatterers, respectively. In each case, schematic: **a** Represents the geometry of the acoustic domain embedded with a rigid scatterer. **b** Plots the variation in the loss function

with training epochs. Note that the plots include the loss variation using both Adam and L-BFGS optimizers. **c** Amplitude maps of the real (top) and imaginary (bottom) components of the predicted pressure field (\hat{p}_s), true pressure field (p_s), and the point-wise error (\mathcal{E}) between \hat{p}_s and p_s

metrics values [$L_2 = 0.2816, R^2 = 0.8319$], also indicate a high prediction error. This inaccuracy can be directly attributed to the inability of the *b*-PINN to handle the specific four scatterers configuration. More specifically, the *b*-PINN finds it challenging to learn to approximate the simulated field due to four scatterers as indicated by the fact that the value of the training loss reaches a plateau (see Fig. 8B(b)). A subsequent analysis reveals that this high prediction error is directly caused by the rigid scatterers boundaries as indicated by the high $\mathcal{L}_{B_i}(\theta)$ values. This observation reinforces our assumption that as the number of scatterers increases, the high prediction error observed during the initial network training phase propagates across the spatial domain throughout the entire training process.

Therefore, while the *b*-PINN demonstrates good performance and prediction accuracy when simulating a few scatterers, it encounters limitations when solving multiple scattering problems with an increasing number of scatterers.

4.4.2 Multiple scattering simulation using *s*-PINN

As the *s*-PINN is designed to address the limitations of the *b*-PINN in the context of multiple scattering simulations, we first compare the performance of the two networks (i.e. *s*-PINN and *b*-PINN). Specifically, we compare the performance of the *b*-PINN (see Fig. 8B) and *s*-PINN (see Fig. 9A) for a 4×1 acoustic lattice simulated at $f = 500$ Hz. Similar to the other schematics shown in this

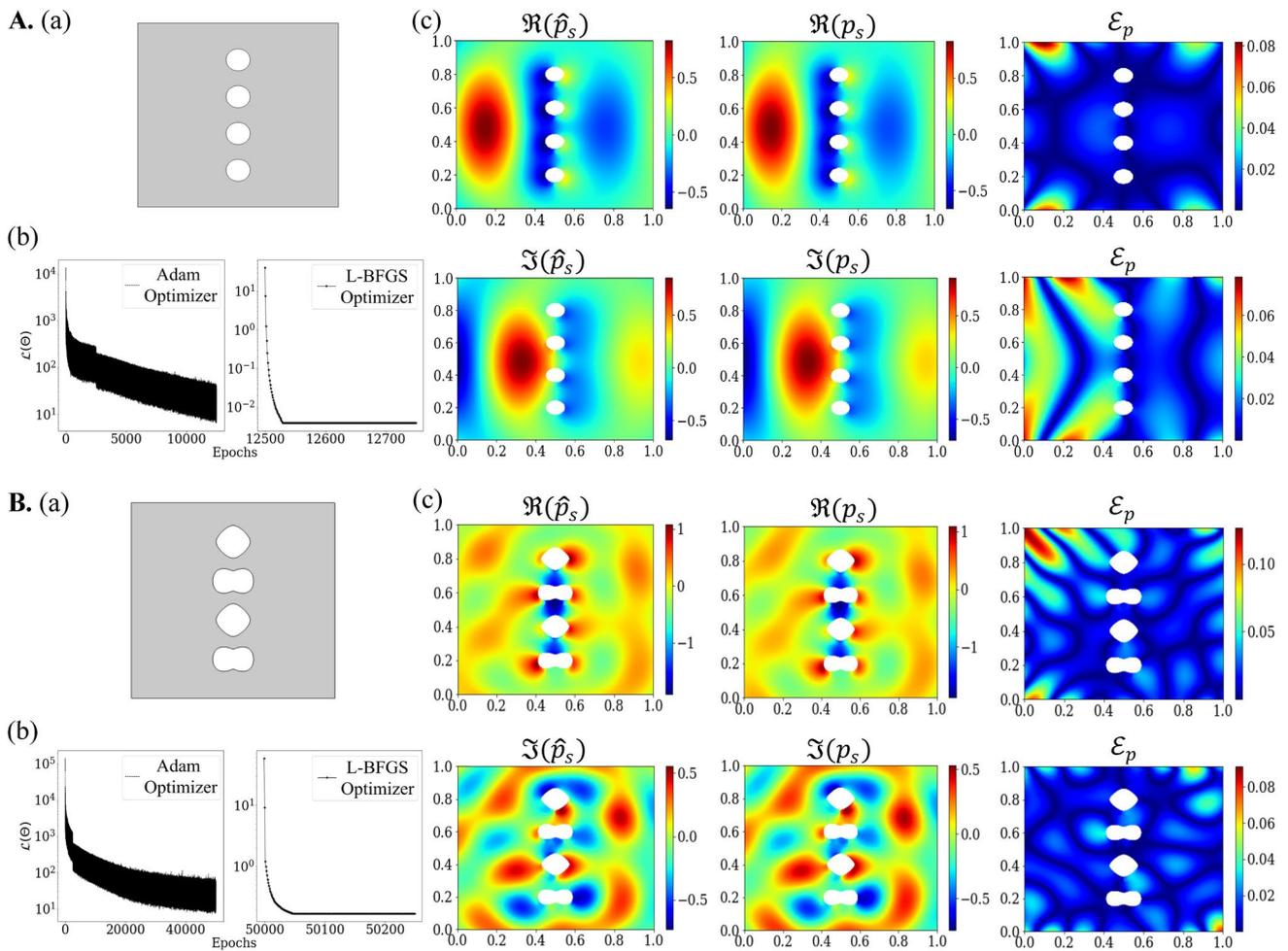


Fig. 9 Case 3- *s*-PINN: Illustration of the performance analysis of *s*-PINN for multiple scattering applications in **A–B**. The simulations in **A** and **B** are performed for planar wavefields incident on scatterer lattices with scatterers arranged in 4×1 grid with circular scatterers at $f = 500 \text{ Hz}$ and 4×1 grid with arbitrary scatterers at $f = 1 \text{ kHz}$, respectively. In each case, schematic: **a** represents the geometry of the

acoustic domain embedded with a rigid scatterer. **b** Plots the variation in the loss function with training epochs. Note that the plots include the loss variation using both Adam and L-BFGS optimizers. **c** Amplitude maps of the real (top) and imaginary (bottom) components of the predicted pressure field (\hat{p}_s), true pressure field (p_s), and the point-wise error (\mathcal{E}) between \hat{p}_s and p_s

Table 3 Summary of key network parameters used to develop the *s*-PINN architecture for multiple scattering simulation due to a 4×4 scatterer lattice at $f = 500 \text{ Hz}$ and $f = 1 \text{ kHz}$

Parameter	Case 3 <i>s</i> -PINN				Case 3 <i>s</i> -PINN			
	$(f = 500 \text{ Hz})$				$(f = 1 \text{ kHz})$			
	PINN ¹	PINN ²	PINN ³	PINN ⁴	PINN ¹	PINN ²	PINN ³	PINN ⁴
n_w	100	100	100	100	125	125	125	125
n_r	2	2	2	2	3	3	3	3
n_l	4	4	4	4	4	4	4	4
N_{b_i}	100	100	100	100	100	100	100	100
N_{b_e}	1000	1000	1000	1000	1000	1000	1000	1000
N_n	10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000
l_r	1e-3, 1e-4				1e-3, 1e-4			

section, the (a), (b), and (c) parts of Fig. 8B for b -PINN and Fig. 9A for s -PINN represent the simulated domain, loss variation, and network prediction accuracy, respectively. A direct comparison of the variation in training losses between the b -PINN and the s -PINN highlights the ability of the s -PINN to learn to approximate the scattered fields for multiple scattering simulations. Further, the s -PINN predictions return [$L_2 = 0.0694, R^2 = 0.9946$], which represents a 75% increase in the prediction accuracy compared to the b -PINN prediction for the same simulation domain. Subsequently, we have performed additional simulations to further validate the performance of s -PINN for multiple scatters of arbitrary shapes. Fig. 9B provides the results illustrating the performance of s -PINN for an acoustic scatterer lattice arranged in 4×1 grid with arbitrary scatterer shapes at $f = 1 \text{ kHz}$. The comparison between predictions and ground truth provides the following parameters [$L_2 = 0.082, R^2 = 0.9929$]. The \mathcal{E} plot along with low L_2 -error and high R^2 -score demonstrates the ability of s -PINN to accurately simulate multiple scattering due to arbitrary scatterer shapes.

Note that Table 2 presents the values of key network parameters selected for each b -PINN within the s -PINN to simulate the 4×1 scatterer lattice. Here, the s -PINN is trained with a total number of $N'_s = 4$ scatterers using $N_s = 4$ b -PINNs each with $n_s = 1$ scatterer. Moreover, this s -PINN is trained for $N_e = 12,500$ epochs with a step learning rate of $l_r = 1e - 3$ for the first 2500 epochs and $l_r = 1e - 4$ for the rest of the training with Adam Optimizer. Although the individual b -PINN within the s -PINN is designed with lower n_l and N_e compared to when trained with a single b -PINN, the much higher prediction accuracy of the s -PINN indicates the remarkable ability of the network architecture to enforce the superposition principle along with the minimization of the governing PDE and boundary conditions on individual scatterer sub-domains. Another possible comparison is between a b -PINN and s -PINN to solve the same multiple scattering problem with the same total number of network weights or parameters. However, it was proven by Jagtap et al. [49, 55] that using the same number of total weights in a single network (similar to b -PINN) cannot manage cases with high initial error propagation as well as a domain decomposition-based multiple network (similar to s -PINN) with the same number of weights. This is because the high initial error in the total loss is influenced by the increase in the number of internal boundaries, making it challenging for the optimizer to reduce the total loss, which is biased by this high error term. This high initial error propagates through the network training regardless of the increase in network parameters. In contrast, this issue can be managed by reducing the initial error in prediction by allowing separate sets of network parameters to handle different sub-scatterer domains.

As the above results established the superior performance of the s -PINN over the b -PINN for multiple-scattering simulations, in the following we investigate the ability of s -PINN to simulate more complex multiple scattering cases. Table 3 records the values of the key network parameters used to develop the s -PINN for a more elaborate scattering scenario. Specifically, Fig. 10A indicates the network performance assessment for a 4×4 lattice with a total number of $N'_s = 16$ scatterers simulated at $f = 500 \text{ Hz}$ by using $N_s = 4$ b -PINNs each with $n_s = 4$ scatterers. The comparison of the predictions and ground truth in Fig. 10A(c), and of the metrics [$L_2 = 0.1076, R^2 = 0.9861$] highlights the good prediction accuracy of the s -PINN. Moreover, this s -PINN is trained for $N_e = 40,000$ epochs with a step learning rate. In addition, we also simulate the 4×4 scatterer lattice at $f = 1 \text{ kHz}$ and study the performance at higher frequencies as shown in Fig. 10B with a step learning rate for $N_e = 120,000$ epochs. Based on the prediction comparison in Fig. 10B(c), the accuracy metrics are evaluated as [$L_2 = 0.0883, R^2 = 0.9913$]. The low L_2 -error and high R^2 -score highlight the performance of s -PINN when simulating complex multiple scattering problems at different frequencies in the low- and mid-frequency ranges. Note that as we move into the high-frequency range, the prediction accuracy of the current s -PINN begins to deteriorate due to the inability of the current network to capture finer wavefield features with existing spatial sampling. Therefore, although the proposed network model has lower discretization dependence in comparison to mesh-based approaches like FEM, the high-frequency simulations require finer discretization to capture more accurate results.

5 Conclusions

This study presented a physics-informed machine learning framework designed to simulate 2D acoustic scattering from arbitrary-shaped objects embedded in an unbounded domain. The framework encompassed two distinct physics-informed neural network (PINN) models, referred to as the baseline-PINN (b -PINN) and the superposition-PINN (s -PINN). The b -PINN was developed to perform acoustic scattering simulations when in presence of either a single or a small number of rigid scatterers. The two key attributes of the b -PINN are its ability to simulate the scattered field due to arbitrary-shaped rigid scatterers and to address acoustic scattering problems with incident wavelengths shorter than the characteristic scatterer sizes (i.e. towards the geometric scattering regime). However, while the b -PINN can handle multiple scattering, it was shown that its applicability is restricted to scenarios with only a few rigid scatterers due to error propagation. To overcome this limitation, an

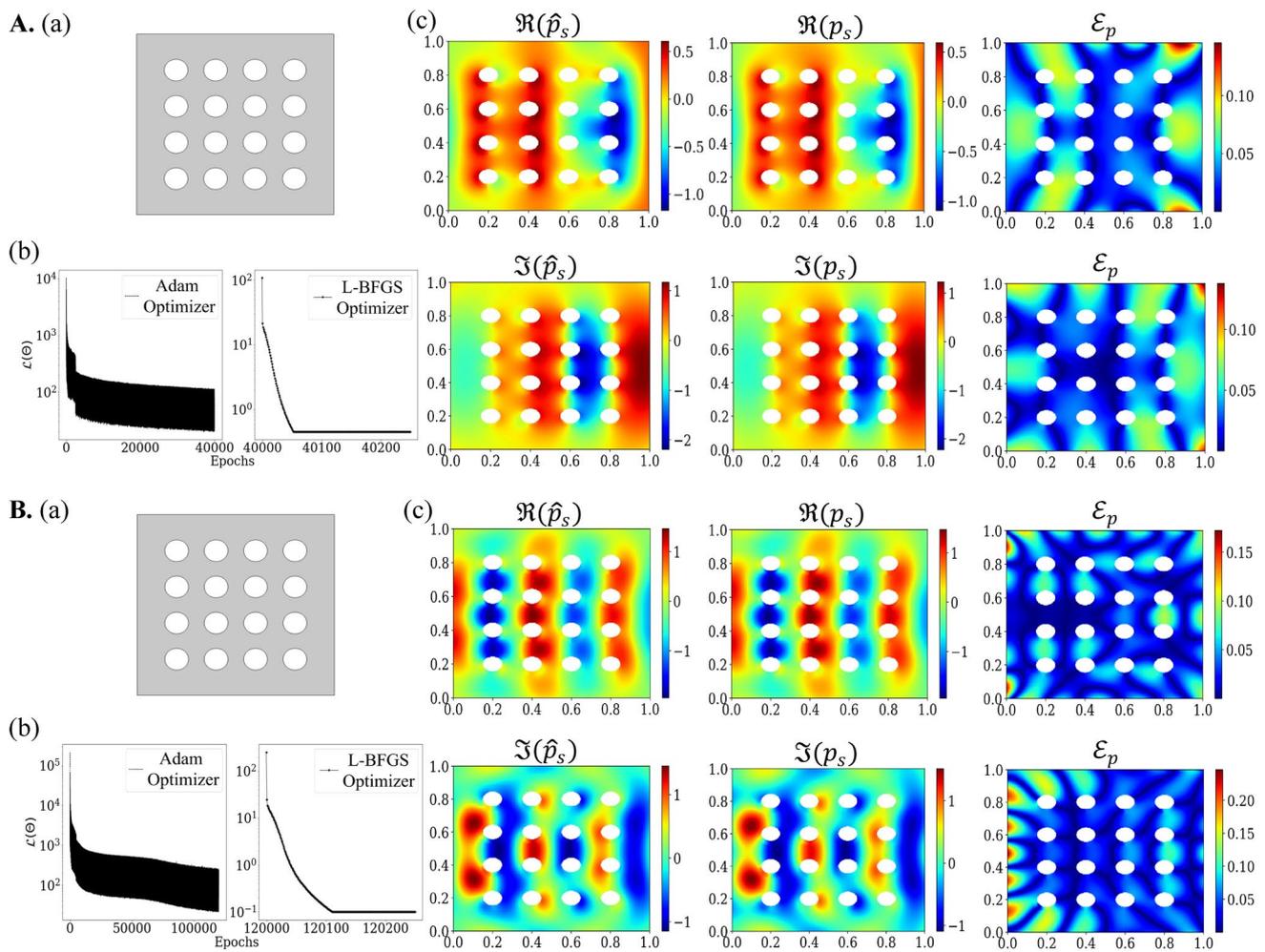


Fig. 10 Case 3- s -PINN: Performance analysis of s -PINN for multiple scattering applications. The simulations in **A** and **B** are performed under planar wavefields incident on a lattice of 4×4 scatterers at $f = 500 \text{ Hz}$ and $f = 1 \text{ kHz}$, respectively. **a** schematic representing the geometry of the acoustic domain embedded with a rigid scatterer.

b Plots the loss function with training epochs. Note that the plots include the loss variation using both Adam and L-BFGS optimizers. **c** Amplitude maps of the real (top) and imaginary (bottom) components of the predicted pressure field (\hat{p}_s), true pressure field (p_s), and the point-wise error (\mathcal{E}) between \hat{p}_s and p_s .

alternative configuration, denominated the superposition PINN, was developed.

While the s -PINN builds upon the basic b -PINN architecture, it overcomes the error propagation issue by using a domain partitioning approach. Multiple copies of the b -PINN used to simulate individual parts of the domain and then they are combined together by means of an architecture that embodies the superposition principle of linear acoustics. In addition to its ability to simulate multiple scattering problems, the results highlighted the discretization-independent nature of the proposed network that enables handling simulations of arbitrarily shaped scatterers and a wide range of frequencies using the same spatial discretization. Furthermore, the (yet unproven) potential of s -PINN to facilitate parallel computations by assigning sub-domain simulations to separate computational nodes could significantly reduce

the total computational time for complex multiple scattering simulations. These properties propel the s -PINN as a capable forward solver that can address the challenges associated with the high computational cost in traditional mesh-based methods like finite element method (FEM) for complex multiple scattering simulations.

In contrast to the existing data-driven DNNs, the proposed PINN framework is trained by explicitly enforcing the governing equations and boundary conditions of the problem. This approach enables it to generate accurate and physically consistent solutions without relying on labeled datasets for network training, thereby reducing the data generation cost by 100%. Furthermore, the performance of the proposed network models was evaluated by comparing their predictions to FEM ground truth for various acoustic scattering applications. The numerical investigations highlighted the

ability of the network models to accurately simulate acoustic scattering scenarios across applications involving arbitrary scatterer shapes, high frequencies, and lattices of scatterers. Additional areas needing more research and development exist in the *s*-PINN architecture. As an example, *s*-PINN has the potential to be set up and trained using parallel computations, therefore its implementation and performance should be explored. In addition, the current *s*-PINN architecture becomes ineffective at capturing multi-scattered fields with nonlinear behavior due to substantially higher acoustic pressure amplitudes, e.g., shock waves. Therefore, developing a PINN architecture to handle multiple scattering phenomenon while accommodating the corresponding nonlinear behavior is an interesting area of potential future research.

Acknowledgements This work was supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Declarations

Conflict of interest The authors declare no Conflict of interest.

References

- Bayati E, Pestourie R, Colburn S, Lin Z, Johnson SG, Majumdar A (2020) Inverse designed metalenses with extended depth of focus. *ACS Photonics* 7:873–878
- Pestourie R, Pérez-Arancibia C, Lin Z, Shin W, Capasso F, Johnson SG (2018) Inverse design of large-area metasurfaces. *Opt Express* 26:33732–33747
- Speziale S, Marquardt H, Duffy TS (2014) Brillouin scattering and its application in geosciences. *Rev Mineral Geochem* 78(1):543–603
- Yaman F, Yakhno VG, Potthast R (2013) A survey on inverse problems for applied sciences. *Math Prob Eng* 1–19:2013
- Wu RT, Jokar M, Jahanshahi MR, Semperlotti F (2022) A physics-constrained deep learning based approach for acoustic inverse scattering problems. *Mech Syst Signal Process* 164:108190
- Nair S, Walsh TF, Pickrell G, Semperlotti F (2023) GRIDS-Net: inverse shape design and identification of scatterers via geometric regularization and physics-embedded deep learning. *Comput Methods Appl Mech Eng* 414:116167
- Faddeev LD, Merkuriev SP (1993) Quantum scattering theory for several particle systems, vol 11. Springer Science & Business Media, Berlin
- Amsler C, Doser M, Bloch P, Ceccucci A, Giudice GF, Höcker A, Mangano ML, Masoni A, Spanier S, Törnqvist NA et al (2008) Review of particle physics. *Phys Lett B* 667(1–5):1–6
- Altpeter I, Becker R, Dobmann G, Kern R, Theiner W, Yashan A (2002) Robust solutions of inverse problems in electromagnetic non-destructive evaluation. *Inverse Prob* 18:1907
- Krivoshchev AI, Barkov FL, Konstantinov YA, Belokrylov ME (2022) State-of-the-art methods for determining the frequency shift of Brillouin scattering in fiber-optic metrology and sensing. *Instrum Expe Tech* 65(5):687–710
- Bertero M, Piana M (2006) Inverse problems in biomedical imaging: modeling and methods of solution. *Complex systems in biomedicine*. Springer, Berlin, pp 1–33
- Freudiger CW, Min W, Saar BG, Lu S, Holtom GR, He C, Tsai JC, Kang JX, Xie XS (2008) Label-free biomedical imaging with high sensitivity by stimulated Raman scattering microscopy. *Science* 322(5909):1857–1861
- Entekhabi D, Nakamura H, Njoku EG (1994) Solving the inverse problem for soil moisture and temperature profiles by sequential assimilation of multifrequency remotely sensed observations. *IEEE Trans Geosci Remote Sens* 32:438–448
- Waseda Y (1984) Novel application of anomalous (resonance) X-ray scattering for structural characterization of disordered materials. Springer, Berlin
- Carvalho PM, Felício MR, Santos NC, Gonçalves S, Domingues MM (2018) Application of light scattering techniques to nanoparticle characterization and development. *Front Chem* 6:237
- Kim K, Kang S, Yoo J, Kwon Y, Nam Y, Lee D, Kim I, Choi Y, Jung Y, Kim S (2018) Deep-learning-based inverse design model for intelligent discovery of organic molecules. *npj Comput Mater* 4:1–7
- Volakis JL, Chatterjee A, Kempel LC (1994) Review of the finite-element method for three-dimensional electromagnetic scattering. *JOSA A* 11(4):1422–1433
- Ihlenburg F (1998) Finite element analysis of acoustic scattering. Springer, Berlin
- Morgan MA (2013) Finite element and finite difference methods in electromagnetic scattering. Elsevier, Amsterdam
- Fahy FJ (2007) Sound and structural vibration: radiation, transmission and response. Elsevier, Amsterdam
- Giorgiani G, Modesto D, Fernández-Méndez S, Huerta A (2013) High-order continuous and discontinuous Galerkin methods for wave problems. *Int J Numer Meth Fluids* 73(10):883–903
- Thompson LL (2006) A review of finite-element methods for time-harmonic acoustics. *J Acoust Soc Am* 119(3):1315–1330
- Liu G, Karamanlidis D (2003) Mesh free methods: moving beyond the finite element method. *Appl Mech Rev* 56(2):B17–B18
- Atluri SN, Zhu T (1998) A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Comput Mech* 22(2):117–127
- Monaghan JJ (1992) Smoothed particle hydrodynamics. *Ann Rev Astron Astrophys* 30(1):543–574
- Schaback R (2007) Convergence of unsymmetric kernel-based meshless collocation methods. *SIAM J Numer Anal* 45(1):333–351
- Belytschko T, Lu YY, Gu L (1994) Element-free Galerkin methods. *Int J Numer Meth Eng* 37(2):229–256
- Nguyen VP, Rabczuk T, Bordas S, Duflot M (2008) Meshless methods: a review and computer implementation aspects. *Math Comput Simul* 79(3):763–813
- Liew KM, Zhao X, Ferreira AJM (2011) A review of meshless methods for laminated and functionally graded plates and shells. *Compos Struct* 93(8):2031–2041
- Peurifoy J, Shen Y, Jing L, Yang Y, Cano-Renteria F, DeLacy BG, Joannopoulos JD, Tegmark M, Soljačić M (2018) Nanophotonic particle simulation and inverse design using artificial neural networks. *Sci Adv* 4(6):eaar4206
- Shan T, Tang W, Dang X, Li M, Yang F, Xu S, Wu J (2020) Study on a fast solver for Poisson's equation based on deep learning technique. *IEEE Trans Antennas Propag* 68(9):6725–6733
- Yao HM, Jiang LJ (2018) Machine learning based neural network solving methods for the FDTD method. In: 2018 IEEE

- international symposium on antennas and propagation & USNC/URSI national radio science meeting, pp 2321–2322. IEEE
33. Liu Z, Zhu D, Rodrigues SP, Lee K, Cai W (2018) Generative model for the inverse design of metasurfaces. *Nano Lett* 18(10):6570–6576
 34. Karpatne A, Kannan R, Kumar V (2022) Knowledge guided machine learning: accelerating discovery using scientific knowledge and data. CRC Press, Boca Raton
 35. Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L (2021) Physics-informed machine learning. *Nature Rev Phys* 3:422–440
 36. Raissi M, Perdikaris P, Karniadakis GE (2019) Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J Comput Phys* 378:686–707
 37. Raissi M, Yazdani A, Karniadakis GE (2020) Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations. *Science* 367(6481):1026–1030
 38. Jin X, Cai S, Li H, Karniadakis GE (2021) NSFnets (Navier–Stokes flow nets): physics-informed neural networks for the incompressible Navier–Stokes equations. *J Comput Phys* 426:109951
 39. Zobeiry N, Humfeld KD (2021) A physics-informed machine learning approach for solving heat transfer equation in advanced manufacturing and engineering applications. *Eng Appl Artif Intell* 101:104232
 40. Samaniego E, Anitescu C, Goswami S, Nguyen-Thanh VM, Guo H, Hamdia K, Zhuang X, Rabczuk T (2020) An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. *Comput Methods Appl Mech Eng* 362:112790
 41. Chen Y, Lu L, Karniadakis GE, Dal Negro L (2020) Physics-informed neural networks for inverse problems in nano-optics and metamaterials. *Opt Express* 28(8):11618–11633
 42. Chen Y, Dal Negro L (2022) Physics-informed neural networks for imaging and parameter retrieval of photonic nanostructures from near-field data. *APL Photonics* 7(1)
 43. Lee SY, Park C, Park K, Lee HJ, Lee S (2023) A physics-informed and data-driven deep learning approach for wave propagation and its scattering characteristics. *Eng Comput* 39(4):2609–2625
 44. Karimpouli S, Tahmasebi P (2020) Physics informed machine learning: Seismic wave equation. *Geosci Front* 11(6):1993–2001
 45. Rasht-Behesht M, Huber C, Shukla K, Karniadakis GE (2022) Physics-informed neural networks (PINNs) for wave propagation and full waveform inversions. *J Geophys Res Solid Earth* 127(5):e2021JB023120
 46. Song C, Alkhalifah T, Waheed UB (2021) Solving the frequency-domain acoustic VTI wave equation using physics-informed neural networks. *Geophys J Int* 225(2):846–859
 47. Wang H, Li J, Wang L, Liang L, Zeng Z, Liu Y (2023) On acoustic fields of complex scatters based on physics-informed neural networks. *Ultrasonics* 128:106872
 48. Lu L, Meng X, Mao Z, Karniadakis GE (2021) DeepXDE: a deep learning library for solving differential equations. *SIAM Rev* 63(1):208–228
 49. Jagtap AD, Kharazmi E, Karniadakis GE (2020) Conservative physics-informed neural networks on discrete domains for conservation laws: applications to forward and inverse problems. *Comput Methods Appl Mech Eng* 365:113028
 50. Qin T, Wu K, Xiu D (2019) Data driven governing equations approximation using deep neural networks. *J Comput Phys* 395:620–635
 51. Karumuri S, Tripathy R, Bilionis I, Panchal J (2020) Simulator-free solution of high-dimensional stochastic elliptic partial differential equations using deep neural networks. *J Comput Phys* 404:109120
 52. Jagtap AD, Kawaguchi K, Karniadakis GE (2020) Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *J Comput Phys* 404:109136
 53. Koopmann GH, Song L, Fahline JB (1989) A method for computing acoustic fields based on the principle of wave superposition. *J Acoust Soc Am* 86(6):2433–2438
 54. Pierce AD (2019) *Acoustics: an introduction to its physical principles and applications*. Springer, Berlin
 55. Jagtap AD, Karniadakis GE (2020) Extended physics-informed neural networks (xpinnns): a generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. *Commun Comput Phys* 28(5)

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.