

RIKEN AIP

Sound Scene Understanding Team
Kyoto University

Neural acoustic scattering fields simulation with Physics-Informed Neural Networks

For Head-Related Transfer Function estimation

Master's Thesis

Supervisor: Prof. Yoshii Kazuyoshi
Co-supervisor: Prof. Roland Badeau

by

Tancrède Martinez Boisseau

October 21, 2025

Abstract

ABSTRACT

This work presents a neural simulation framework for estimating the scattered acoustic pressure field produced by an incoming wave around arbitrary objects in continuous 2D and 3D domains. Traditionally, this task has been treated as a regression problem, training a physics-informed neural network (PINN) using pressure measurements at discrete positions. Such approaches require retraining the entire network for each incident wave direction, limiting efficiency. To overcome this, we propose a measurement-free simulator driven purely by the Helmholtz equation, incorporating Robin boundary conditions and the Sommerfeld radiation condition via a perfectly matched layer (PML) framework. Furthermore, we introduce a PHysics-Informed Scattering hypernetworK (PHISK), which leverages low-rank adaptation (LoRA) of PINNs trained on specific incident wave directions to generalize efficiently to arbitrary directions. Experiments demonstrate that our method accurately predicts scattered fields around various scatterers, including 3D objects, while maintaining minimal performance loss for unseen wave directions. Finally, the approach is applied to approximate head-related transfer functions (HRTFs) from complex human head meshes, highlighting its potential for practical acoustics applications.

Index Terms— Scattering field, physics-informed neural network (PINN), Helmholtz equation, perfectly matched layer (PML), low-rank adaptation (LoRA), head-related transfer function (HRTF).

Acknowledgements

I would like to begin by expressing my deepest gratitude to my supervisor, Prof. Yoshii Kazuyoshi and my co-supervisors, Dr. Nugraha Aditya Arie and especially Dr. Diego Di Carlo for introducing me to Physics-Informed Neural Networks. Their knowledge and innovative ideas have fundamentally helped me throughout my internship.

My sincere appreciation goes to my academic supervisor Prof. Roland Badeau, and Prof. Mathieu Fontaine for their guidance during this Master.

I am also deeply thankful to my intern colleagues, Emilio Picard and Samy Aimeur without whom I would not have enjoyed my trip to Japan this much.

Finally, I am grateful to my family, especially my mother, for their everyday support despite the physical distance between us.

Contents

Abstract	1
Acknowledgements	2
Notation	6
1 Introduction	8
1.1 A brief history of HRTF	9
1.2 Anatomy Principle	9
1.2.1 Internal anatomy	9
1.2.2 External Anatomy	10
1.3 Head-Related Transfer Function	11
1.3.1 Mathematical definition	11
1.3.2 Auralization	11
1.3.3 Link with body parts	11
1.4 Measuring HRTF	12
1.5 Contributions	14
1.6 Thesis structure	14
2 Physics-Informed Neural Networks	15
2.1 Principle	15
2.2 Input transformation	16
2.2.1 Random Fourier Features	16
2.3 Activation Function	17
2.3.1 Sinusoidal Function with Scaled Input: SIREN	17
2.3.2 Hyperbolic Tangent: \tanh	18
2.3.3 Quadratic Hyperbolic Tangent: QuadraticTanh	18
2.4 Optimizers	18
2.4.1 L-BFGS	19
2.4.2 Practical Considerations for L-BFGS in PINNs	20

3 Acoustic Scattering	22
3.1 Preliminaries	22
3.1.1 Fourier Transform	22
3.1.2 Discrete Fourier Transform (DFT) and Fast Fourier Transform (FFT)	23
3.2 Principle of acoustic scattering	23
3.3 Mathematical model	25
3.3.1 The wave equation	25
3.3.2 The Robin's boundary condition	26
3.3.3 The Sommerfeld Radiation Condition	26
3.4 Perfectly Matched Layer	27
3.4.1 Wave Equation with PML	28
4 Review of existing methods	29
4.1 Estimating HRTF based on head mesh	29
4.1.1 Mesh2HRTF	30
4.2 Scattering with PINN	32
5 Methodology	33
5.1 Proposed method	33
5.1.1 mathematical model	34
5.1.2 Perfectly matched layer parameter	35
5.2 Network definition	35
5.2.1 Loss definition	35
5.2.2 Reference PINN architecture	36
5.2.3 LoRA adapted reference PINN	37
5.3 Hypernetwork PHISK definition	38
5.3.1 Principle of PHISK	38
5.3.2 Loss consideration	39
5.4 Sampling method	40
5.4.1 Sampling on the boundary of the obstacle	41
5.4.2 Sampling in the extended PML domain	42
5.4.3 Direction sampling	43
6 Experiments	44
6.1 Experimental setup	44
6.1.1 Hyperparameters	44
6.1.2 Training setup	45
6.1.3 Selection of LoRA Rank Parameters	45
6.2 Evaluation method	45

6.2.1	Case $\dim(\Omega) = 2$	45
6.2.2	Case $\dim(\Omega) = 3$	46
6.2.3	Evaluation metrics	46
6.3	Scattering in a two dimensional space	47
6.3.1	Performance of the reference PINN	47
6.3.2	LoRA performances	50
6.3.3	PHISK	50
6.3.4	General Results	53
6.4	Scattering in a three dimensional space	54
6.4.1	Scattering around a sphere	54
6.4.2	LoRA performance	55
6.4.3	PHISK	55
6.5	Head-Related Transfer Estimation	55
6.5.1	Scaling to real condition	55
6.5.2	Complexity of the Ear Shape	56
6.5.3	Evaluation of HRTF	57
6.5.4	Coherence results	58
7	Conclusion	60
7.1	Overview of the study	60
7.2	Future Directions	61

Notation

The following notations are used throughout this thesis:

Symbol	Description
\mathcal{T}	A mesh defined by the triplet $\mathcal{T} = (V, E, F)$ of its vertices, edges and faces.
Ω	an open unbounded domain such that $\partial\Omega$ is compact.
Ω_c	an compact domain such that $\partial\Omega$ is compact.
$\tilde{\Omega}_c$	Perfectly matched layer computational domain derived from Ω_c .
f	The frequency.
ω	Angular frequency $\omega = 2\pi f$.
c	Speed of sound, $c = 343\text{m.s}^{-1}$.
k	Wavenumber, $k = \frac{\omega}{c}$.
ϵ	A small positive number.
ρ_0	Density of air under standard atmospheric conditions, $\rho_0 \approx 1.204\text{kg.m}^{-3}$
$\dim(\Omega)$	The dimension of the problem (e.g $\Omega \subset \mathbb{R}^{\dim(\Omega)}$).
\mathcal{F}	Fourier transform defined by $\mathcal{F} : u \rightarrow (U = \mathcal{F}(u) : \omega \rightarrow \int u(., t)e^{-i\omega t} dt)$. We denote the Fourier transform of a function by its capital letter.
$p(\mathbf{x}, t)$	Acoustic pressure at position \mathbf{x} at time t .
$P(\mathbf{x}, \omega)$	Fourier transform of the acoustic pressure at position \mathbf{x} at angular frequency ω .
$G(\mathbf{x}, \mathbf{x}_s, \omega)$	Green's function from source \mathbf{x}_s to \mathbf{x} at angular frequency ω .
h	The head-related impulse response (See CHAPTER XXX).
H	The head-related transfer function, $H = \mathcal{F}(h)$.
(r, θ, ϕ)	Spherical coordinates in \mathbb{R}^3 .
χ_A	Indicator function of the set A .
Γ	The Euler Gamma function defined by $\Gamma : z \in \mathbb{C} \rightarrow \int_0^\infty t^{z-1} e^{-t}$.
$\ \mathbf{x}\ $	Euclidean norm of \mathbf{x} .
Δ	Laplacian operator.
$\frac{\partial}{\partial \mathbf{n}}$	The normal derivative operator defined by $\frac{\partial}{\partial \mathbf{n}} : u \rightarrow (\mathbf{x} \rightarrow \nabla_{\mathbf{x}} u \cdot \mathbf{n}(\mathbf{x}))$.
j	The imaginary number such that $j^2 = -1$.
J	The number of LoRA adapted networks considered for interpolation.
T	Temperature parameter for interpolation.
α	Scaling parameter for coordinate transformation (Chapter 6).

Abbreviation	Description
HRTF	Head-related transfer function.
HRIR	Head-related impulse response.
PINN	Physical-Informed Neural Network.
BFGS	Broyden-Fletcher-Goldfarb-Shanno.
PML	Perfectly Matched Layer.
LoRA	Low-Rank adaptation.
FEM	Finite Element Method.
BEM	Boundary Element Method.
ITD	Interaural Time Difference.
ILD	Interaural Level Difference.
MLP	MultiLayer Perceptron.

Chapter 1

Introduction

With the recent advancements in augmented and virtual reality, accurately simulating human perception has become a key focus area for creating immersive experiences. Among the various sensory modalities, spatial hearing plays a crucial role in providing users with a sense of presence and realism within virtual environments.

Human spatial hearing is governed by complex auditory mechanisms that depend heavily on the unique anatomy of each individual. These mechanisms have been studied for over eight decades and are well represented by a set of functions known as *Head-Related Transfer Functions* (HRTFs). HRTFs describe how incoming sound from a specific point in space is filtered by the listener's torso, head, and outer ears before reaching the eardrums. This transformation enables the brain to localize sound sources in three dimensional space.

Beyond virtual reality, HRTFs have a broad range of applications, including 3D audio rendering, immersive gaming, teleconferencing, and the development of personalized hearing aids. However, accurately measuring or simulating HRTFs remains a challenging task due to the high degree of anatomical variation between individuals and the complexity of wave propagation around the human body.

The modeling of HRTFs has been studied for over fifty years, leading to an acoustic scattering formulation of the problem, grounded in physical laws.

Although current methods such as *Mesh2HRTF* [45] provide a powerful framework for estimating HRTFs based solely on a mesh representing a subject's head, they often rely on several approximations that cannot generalize to more complex scenarios, such as those involving multiple HRTFs using devices like the HoloLens.

This report explores a novel approach to HRTF estimation by providing a framework for solving scattering problems in both 2D and 3D.

1.1 A brief history of HRTF

The concept of the *Head-Related Transfer Function* (HRTF) emerged in the 1960s, particularly through the pioneering work of E. G. Shaw. Although ideas such as *Interaural Time Difference* (ITD) and *Interaural Level Difference* (ILD) had been investigated earlier—especially during World War II in studies of sound localization and psychoacoustics—the formal understanding of HRTFs as directional acoustic filters was established later.

E. G. Shaw conducted numerous experiments demonstrating the crucial role of anatomical features such as the torso, head, shoulders, and especially the pinnae in shaping the sound waves reaching the eardrums. His research revealed that the unique geometry of the pinnae produces direction-dependent filtering effects essential for vertical sound localization [34].

The development of physical dummy head models, such as the KEMAR mannequin introduced by Middlebrooks and Green [26], enabled more systematic research throughout the 1970s. These models allowed for consistent and repeatable measurements of HRTFs across different directions and acoustic environments.

According to Prepelita [30], the first computational simulations of HRTFs were carried out by Weinrich [41] using the *Boundary Element Method* (BEM). Weinrich's work successfully computed HRTFs up to frequencies of approximately 5475 Hz, achieving results within a few decibels of experimental measurements—an important milestone demonstrating the feasibility of numerical acoustic modeling for HRTF estimation.

1.2 Anatomy Principle

The HRTF estimation is a complex task as it depends heavily on individual features described in the following paragraph.

1.2.1 Internal anatomy

The ear canal has been shown to convey little directional information [37]; therefore, modeling efforts can primarily focus on the external ear. The external ear, comprising the pinna and the entrance of the ear canal, plays a crucial role in shaping incoming sound waves before they reach the eardrum. While the ear canal itself transmits sound with minimal directional filtering, the pinna introduces complex reflections and diffractions that create direction-dependent spectral cues, which are essential for vertical localization and front-back discrimination. These acoustic modifications form the basis of the HRTF.

Recent studies have also emphasized the contribution of the inner ear and bone conduction pathways, particularly relevant for emerging headphone technologies where

vibrations bypass the outer ear. In recent years, such internal auditory mechanisms have been studied in greater depth, notably due to the growing interest in bone conduction headphones, for which the role of the inner ear is fundamental.

1.2.2 External Anatomy

The torso

The torso influences sound propagation by reflecting and diffracting waves, particularly at low frequencies where wavelengths are comparable to body dimensions. This interaction generates direction-dependent cues that assist in horizontal localization and affect the spectral content of the sound. In the far field (sources several meters away), the torso primarily acts as a reflector that modifies the wavefront. In the near field (sources close to the listener), its influence becomes more complex, producing spatially varying interference patterns that affect perceived loudness and timbre.

The head

The head produces the well-known acoustic shadow effect, attenuating sounds arriving from the opposite side, which results in ILD. Its shape and size also lead to phase differences between the ears, contributing to ITD. In the far field, these cues remain stable and depend mainly on the azimuth angle; in the near field, however, they become distance-dependent and more complex as wavefront curvature and diffraction effects intensify.

The pinnae

The pinnae generate high-frequency filtering effects through multiple reflections and resonances, producing spectral notches and peaks that vary with sound direction—particularly with elevation and front-back position. These cues are critical for vertical localization. In the far field, pinna cues remain consistent and are dominated by direction-dependent filtering. In the near field, however, close sound sources may interact differently with the pinna geometry, altering these spectral patterns and complicating localization.

1.3 Head-Related Transfer Function

1.3.1 Mathematical definition

The *Head-Related Transfer Function* (HRTF) for a source positioned at $\mathbf{x}_s = (r, \theta, \phi)$ is defined as [6]:

$$H_{\text{ear}}(r, \theta, \phi, f) = \frac{P_{\text{ear}}(r, \theta, \phi, f)}{P_{\text{ref}}(r, \theta, \phi, f)} \quad (1.1)$$

where:

- $P_{\text{ear}}(r, \theta, \phi, f)$ is the sound pressure at the ear.
- $P_{\text{ref}}(r, \theta, \phi, f)$ is the reference sound pressure at the center of the head (pressure at the center of the head for the free field problem without any head as obstacle).

1.3.2 Auralization

As discussed previously, one of the main advantages of the HRTF is its ability to simulate sound spatialization through simple signal processing operations.

Given a signal $y(t)$ emitted from a position $\mathbf{x}_s = (r, \theta, \phi)$, the spatialized version of this signal for a listener whose HRTF has been computed for an ear located at ear, denoted by H_{ear} , is obtained as:

$$y_{\text{ear}}(t) = \mathcal{F}^{-1}[YH_{\text{ear}}](t) \quad (1.2)$$

Alternatively, by using the *Head-Related Impulse Response* (HRIR), defined as $h_{\text{ear}} = \mathcal{F}^{-1}(H_{\text{ear}})$, we can equivalently write:

$$y_{\text{ear}}(t) = (h_{\text{ear}} * y)(t) \quad (1.3)$$

When this process is applied to both the left and right ears simultaneously, the listener gains the ability to localize sound sources in space with an accuracy of approximately 5°, as reported by [6].

Such transformations make it possible to simulate realistic acoustic environments and create highly immersive auditory experiences.

1.3.3 Link with body parts

Studies such as those by Algazi. and al [1] have shown that the body shape was the main factor in the personalization of Head-Related Transfer Function. Moreover, researchers have shown that three main parts were responsible to explain more than 95% of the HRTF.

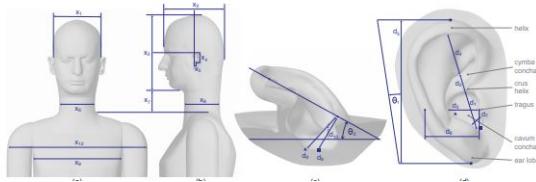


Figure 1.1: Essential anatomical features for HRTF. Adapted from the HUTUBS database by Brinkmann et al. [4], licensed under CC BY 4.0

These are :

- The torso, which operates as a resonance box especially for frequencies below 3kHz.
- The head shape which has great influence for frequency up to 6kHz.
- The pinnae whose thin details operate as small obstacle for the sound and explain the HRTF throughout every range of frequencies and are the main obstacles to consider regarding high frequencies.

The other body parts are either too far from the ear or too thin to have an influence on the HRTF.

To further understand why those body parts play a crucial role for specific range of frequencies, we can roughly use wave theory such that if L is the characteristical length of an obstacle, for an incoming wave of wavelength $\lambda = \frac{2\pi}{k} = \frac{c}{f}$, we should have $\lambda \approx L$ for the obstacle to really have a significant effect on the produced scattered field.

1.4 Measuring HRTF

The measurement of HRTFs is a complex process, as measurements simultaneously serve as ground truth for model validation while themselves being approximations subject to measurement uncertainties. Moreover, the measurement setup typically considers only one fixed source distance from the listener, which limits comparisons to far-field HRTFs. In most experimental configurations, the acoustic source is positioned at a distance greater than 1.5 meters from the subject to ensure an accurate far-field approximation of the sound field. For near-field HRTFs, additional measurements at shorter distances (typically between 0.2 m and 1 m) may be required to capture the effects of wavefront curvature and distance-dependent cues.

Measurement inaccuracies can arise due to reflections, environmental noise, or slight misalignments in the setup. To reduce such errors, multiple recordings are often taken and averaged, improving the signal-to-noise ratio and ensuring more consistent results across measurement angles and frequencies. In addition, the subject's hair is often

covered in order to obtain smoother 3D scan of the head.

Hair has been shown to have no significant effect on HRTF properties [30].

An important consideration when simulating or measuring HRTFs is the characterization of the sound source. Depending on the distance and intended application, the source can be modeled as a plane wave—assuming it is sufficiently far away to be considered as a point at infinity with appropriate amplitude scaling—or as a spherical point or velocity source to more accurately represent its physical placement.

The **SONICOM dataset** [12] provides HRTF measurements for over 200 individuals, offering five different data formats:

- **Raw:** Direct measurements without any compensation filters applied.
- **Windowed:** Measurements processed with time-domain windowing to isolate the direct sound and minimize reflections.
- **Free-field compensated:** HRTFs corrected for the influence of the loudspeaker and measurement setup to represent an ideal free-field condition.
- **Free-field compensated with a minimum-phase filter:** Same as the previous format, but processed to enforce a minimum-phase response, ensuring a physically consistent impulse response.
- **Mesh2HRTF estimated HRTF:** HRTFs simulated using 3D head scans of the same individuals via the *Mesh2HRTF* software [45].

In this dataset, the acoustic source is placed at a distance of 1.5 meters from the listener, corresponding to far-field conditions. Additionally, 3D scan of the head are provided, allowing us to use them throughout this study.

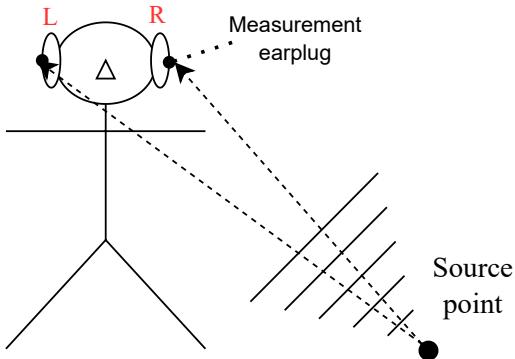


Figure 1.2: Typical HRTF measurement setup. The subject is often immobilized to have more accurate measurement.

1.5 Contributions

In this study, we investigate the solution of acoustic scattering problems using *Physics-Informed Neural Networks* (PINNs), in both two dimensional and three dimensional domains. Each problem instance corresponds to a specific incident wave direction (or source point position) and frequency.

Furthermore, we introduce a novel hypernetwork architecture, termed *PHISK* (**P**Hysics-**I**nformed **S**cattering **h**ypernetwor**K**), designed to solve the scattering problem for a fixed frequency across all possible source directions. This approach leverages the interpolation of *Low-Rank Adaptation* (LoRA) networks trained on different incident directions from a common reference PINN, allowing efficient generalization over directional inputs.

We apply this framework to several scattering scenarios, including head-related acoustic scattering, with the goal of estimating the HRTF for a given frequency. In addition, we provide practical insights and guidelines for the effective design and training of PINNs in the context of scattering field estimation.

1.6 Thesis structure

This thesis is organized as follows:

- An introductory chapter on PINNs (2), providing an overview of their design principles as well as their limitations.
- A preliminary chapter (3) on the physical equations underlying the acoustic scattering problem.
- A literature review (4) summarizing existing methods for HRTF estimation and acoustic scattering modeling.
- A methodology chapter (5) describing our proposed approach, including the different architectures and additional considerations for the problem.
- An experimental chapter (6) presenting results that illustrate the robustness and accuracy of our approach.
- A concluding chapter (7) discussing overall findings and suggesting directions for future research on this topic.

Chapter 2

Physics-Informed Neural Networks

This chapter describes the principle of *Physics-Informed Neural Networks* (PINNs) as well as the techniques used in their general use and in our study.

2.1 Principle

PINNs have emerged as a promising alternative to traditional numerical methods. Whereas the Finite Element Method (FEM) [18] computes solutions on a fixed-resolution grid, and the Boundary Element Method (BEM) [21] incurs high inference times for individual problem instances, PINNs offer a flexible and efficient approach to overcome these limitations.

PINNs aim to solve systems governed by partial differential operators of the form $\mathcal{N}(u) = 0$, where \mathcal{N} encodes the underlying physical laws (e.g., Eqs. (3.18) and (3.13)), and u represents the solution [32]. The objective is to learn a neural network Φ_θ , parameterized by θ , that approximates the solution u for $\mathbf{x} \in \Omega$. This is achieved by embedding the governing equations \mathcal{N} into the loss function, using small batches of points \mathcal{B} and minimizing the residual:

$$\sum_{\mathbf{x} \in \mathcal{B}} |\mathcal{N}(\Phi_\theta)(\mathbf{x})|^2, \quad (2.1)$$

with derivatives evaluated via automatic differentiation [35].

This framework allows the integration of both data-driven information (e.g., measured pressure points) and physical knowledge (the governing equations), resulting in solutions that are more interpretable and often generalize better beyond the computational domain.

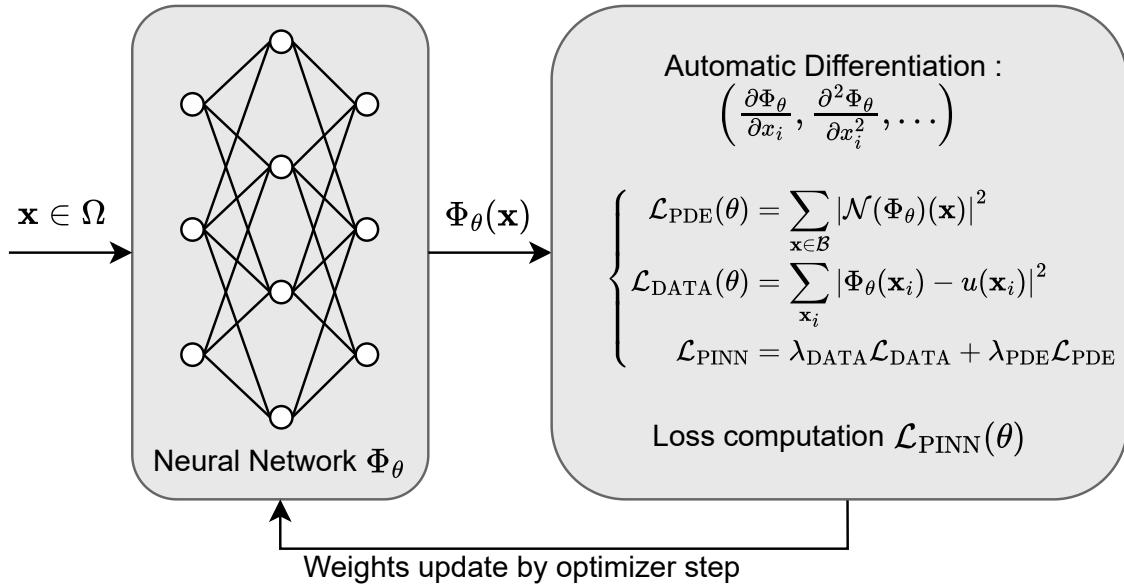


Figure 2.1: Principle of PINNs physics-informed

2.2 Input transformation

Physically-defined functions typically take inputs from a low-dimensional space $\Omega \subset \mathbb{R}^d$, with $\dim(\Omega) \leq 4$.

However, in the context of PINNs, such coordinates—despite their clear physical meaning—tend to bias the network toward low-frequency representations of the output [39]. To mitigate this intrinsic bias, various techniques have been developed to map the input into a higher-dimensional space more suitable for capturing complex, high-frequency patterns [38].

2.2.1 Random Fourier Features

The concept of *Random Fourier Features* (RFFs), rooted in kernel theory [31], involves projecting the input into a sinusoidal basis to better capture oscillatory and high-frequency components of the output.

Mathematically, assuming the input space Ω has dimension $\dim(\Omega)$, we define the number of features M , a scale parameter σ , and a random matrix $\mathbf{B} \in \mathbb{R}^{M \times \dim(\Omega)}$ such that:

$$\forall (i, j) \in [\![0, M]\!] \times [\![0, \dim(\Omega)]\!], \quad B_{ij} \sim \mathcal{N}(0, \sigma^2) \quad \text{i.i.d.} \quad (2.2)$$

The Random Fourier Features transformation is then defined as:

$$\mathbf{x} \in \Omega \mapsto [\cos(\mathbf{B}\mathbf{x}), \sin(\mathbf{B}\mathbf{x})]. \quad (2.3)$$

Such oscillatory basis functions are particularly useful in the context of acoustic scattering, as they more effectively capture the propagating behavior of waves.

2.3 Activation Function

Since Physics-Informed Neural Networks (PINNs) aim to produce a continuous and highly differentiable representation of the target function, commonly used activation functions such as ReLU are not well-suited due to their lack of smoothness and poor behavior of higher-order derivatives.

Although modern alternatives such as Gabor functions or H-SIREN have emerged [14, 33], two main activation functions have traditionally been used in PINNs.

2.3.1 Sinusoidal Function with Scaled Input: SIREN

The SIREN network is a multilayer perceptron (MLP) that employs a sinusoidal activation function with a specific weight initialization scheme [35]. The activation includes a frequency scaling parameter ω_0 , which is typically chosen based on the problem resolution, with empirical studies often recommending $\omega_0 = 30$.

The SIREN activation function is defined as:

$$\text{Sine} : x \in \mathbb{R} \longmapsto \sin(\omega_0 x), \quad (2.4)$$

and has the following properties:

- $\text{Sine} \in \mathcal{C}^\infty$,
- Higher-order derivatives:

$$\begin{aligned} \text{Sine}^{(2k)}(x) &= \omega_0^{2k}(-1)^k \sin(\omega_0 x), \\ \text{Sine}^{(2k+1)}(x) &= \omega_0^{2k+1}(-1)^k \cos(\omega_0 x), \end{aligned}$$

- The sinusoidal behavior of the activation function facilitates the representation of oscillatory solutions.

The resolution parameter ω_0 plays a crucial role in enhancing the representational capacity of the network. However, its presence in higher-order derivatives requires careful weight initialization, as improper settings can lead to convergence difficulties [35].

2.3.2 Hyperbolic Tangent: \tanh

The hyperbolic tangent function has been a staple activation for PINNs since their inception, often used in combination with feature-scaling methods.

It is defined as:

$$\tanh : x \in \mathbb{R} \longmapsto \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad (2.5)$$

with key properties including:

- $\tanh \in \mathcal{C}^\infty$,
- All derivatives of \tanh are bounded. In particular, the zeroth, first, and second-order derivatives lie within $[0, 1]$,
- Every derivative of \tanh can be expressed as a polynomial in \tanh itself.

The hyperbolic tangent is widely adopted in PINNs due to its smoothness, bounded output range, and polynomial derivative structure. When combined with input transformation techniques such as Random Fourier Features, it enables accurate modeling of fine-scale structures in the target solution.

2.3.3 Quadratic Hyperbolic Tangent: `QuadraticTanh`

Although less commonly used in the literature, the quadratic hyperbolic tangent `QuadraticTanh` has shown promise in capturing oscillatory behaviors in our experiments. This function modifies the standard \tanh by including a quadratic term in its argument [43]:

$$\text{QuadraticTanh} : x \in \mathbb{R} \longmapsto \tanh(x + \alpha x^2), \quad (2.6)$$

where the parameter α can either be learned during training or fixed manually. It is also possible to assign different α values to each layer, although this flexibility may reduce the interpretability of the network.

2.4 Optimizers

Physics-Informed Neural Networks (PINNs) are often challenging to optimize, and convergence issues can arise during training [40].

A commonly adopted training scheme for PINNs involves a two-stage process: an initial phase using the Adam optimizer, followed by a second phase employing the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm.

The Adam phase serves primarily as a robust initialization, guiding the optimization toward a region where L-BFGS can converge more effectively. In our work, we follow a similar two-step strategy (with variations discussed in a later chapter), consisting of an Adam phase followed by an L-BFGS phase with a strong Wolfe line search.

As highlighted in [20], L-BFGS combined with a strong Wolfe line search offers a good compromise for our problem setting, particularly because we employ small batch sizes relative to the overall size of the computational mesh (see Section 6 for details on sampling).

We do not provide a detailed discussion of the Adam optimizer here, as it has become a standard tool in machine learning since its introduction. For further details, we refer the reader to the original paper [19].

2.4.1 L-BFGS

The L-BFGS algorithm is a quasi-Newton method that approximates the inverse Hessian of the objective function using a limited number of correction vectors (denoted by the memory length m) [22].

It is particularly effective for large-scale optimization problems and performs well when applied to large batches of data points.

A pretraining step using the Adam optimizer often improves convergence, as the strong second-order updates of L-BFGS can lead to divergence if the network is poorly initialized.

The algorithm proceeds as follows:

Algorithm 1 L-BFGS Algorithm

- 1: **Input:** Initial point x_0 , memory length m , tolerance ϵ
 - 2: Set $k = 0$
 - 3: **while** $\|\nabla f(x_k)\| > \epsilon$ **do**
 - 4: Compute gradient $g_k = \nabla f(x_k)$
 - 5: Determine search direction $p_k = -H_k g_k$ using two-loop recursion (see Algorithm 2)
 - 6: Perform line search to find step size α_k satisfying the **strong Wolfe conditions**
 - 7: Update position: $x_{k+1} = x_k + \alpha_k p_k$
 - 8: Compute $s_k = x_{k+1} - x_k$ and $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$
 - 9: Store (s_k, y_k) in memory (retain only the last m pairs)
 - 10: Increment k
 - 11: **end while**
 - 12: **Return:** Approximate minimizer x_k
-

The search direction p_k is computed using the two-loop recursion method, which avoids explicit computation of the Hessian inverse:

Algorithm 2 Two-Loop Recursion for $q = H_k g_k$

```

1: Input: Gradient  $g_k$ , stored vectors  $\{(s_i, y_i)\}$  for  $i = k - 1, \dots, k - m$ 
2:  $q \leftarrow g_k$ 
3: for  $i = k - 1$  down to  $k - m$  do
4:    $\rho_i \leftarrow 1/(y_i^\top s_i)$ 
5:    $\alpha_i \leftarrow \rho_i s_i^\top q$ 
6:    $q \leftarrow q - \alpha_i y_i$ 
7: end for
8:  $H_k^0 \leftarrow \frac{s_{k-1}^\top y_{k-1}}{y_{k-1}^\top y_{k-1}} I$                                  $\triangleright$  Initial Hessian approximation
9:  $r \leftarrow H_k^0 q$ 
10: for  $i = k - m$  to  $k - 1$  do
11:    $\beta_i \leftarrow \rho_i y_i^\top r$ 
12:    $r \leftarrow r + s_i(\alpha_i - \beta_i)$ 
13: end for
14: Return:  $p_k = -r$ 

```

Strong Wolfe Line Search.

To ensure sufficient descent and numerical stability, L-BFGS typically employs a line search procedure satisfying the *strong Wolfe conditions*:

$$f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k \nabla f(x_k)^\top p_k, \quad (2.7)$$

$$|\nabla f(x_k + \alpha_k p_k)^\top p_k| \leq c_2 |\nabla f(x_k)^\top p_k|, \quad (2.8)$$

where $0 < c_1 < c_2 < 1$ are typically chosen as $c_1 = 10^{-4}$ and $c_2 = 0.9$. These conditions ensure both sufficient decrease in the objective function and appropriate control over curvature along the search direction.

2.4.2 Practical Considerations for L-BFGS in PINNs

When applying L-BFGS to Physics-Informed Neural Networks, several practical aspects should be considered to ensure stable and efficient training, particularly in the context of acoustic scattering and HRTF estimation:

- **Memory length m :** The number of correction pairs stored in memory directly impacts the quality of the Hessian approximation. While larger m values improve the approximation, they also increase computational cost and memory usage. For typical PINN applications, m is often chosen between 5 and 20. We set $m = 20$ in this study.

- **Batch size:** L-BFGS performs best when gradients are computed over the full dataset or sufficiently large batches. In acoustic scattering problems, using small batch sizes relative to the computational mesh can introduce noise into the gradient, which may slow convergence or cause instability. A pretraining phase with Adam mitigates this issue by providing a better initial parameter region.
- **Line search and numerical stability:** The strong Wolfe line search ensures sufficient decrease and curvature control, but tuning the constants c_1 and c_2 may be necessary depending on the oscillatory nature of the wavefield. For HRTF estimation, where wave solutions contain high-frequency components, strict Wolfe conditions often yield more stable optimization.

In summary, successful application of L-BFGS in PINNs for acoustic scattering relies on a combination of appropriate memory length, sufficiently large batch sizes, careful weight initialization, and robust line search strategies. Pretraining with a first-order optimizer such as Adam is highly recommended to enhance stability and convergence. However, in some particular case such as LoRA, L-BFGS may be sufficient as our result suggest (Chapter 6).

Chapter 3

Acoustic Scattering

3.1 Preliminaries

We first introduce the Fourier transform that we will use extensively throughout this work, using mainly the Fourier domain for computation.

3.1.1 Fourier Transform

The Fourier Transform is a mathematical tool used to decompose a signal into its constituent frequencies and was introduced by Jean-Baptiste Joseph Fourier in 1807. It transforms a function from the time domain to the frequency domain.

The continuous Fourier Transform of a function x is defined as:

$$\mathcal{F} : x \longmapsto \left(\omega \in \mathbb{R} \mapsto X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt \right), \quad (3.1)$$

where ω is the angular frequency, and $X(\omega)$ represents the frequency content of the signal.

The inverse Fourier Transform is:

$$\mathcal{F}^{-1} : X \longmapsto \left(t \in \mathbb{R} \mapsto x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{j\omega t} d\omega \right). \quad (3.2)$$

Convolution Property

A key property of the Fourier Transform is its behavior under convolution. For two functions x and y , the convolution $(x * y)$ is defined as:

$$(\cdot * \cdot) : (x, y) \longmapsto \left(t \in \mathbb{R} \mapsto (x * y)(t) = \int_{-\infty}^{\infty} x(\tau) y(t - \tau) d\tau \right). \quad (3.3)$$

The Fourier Transform of a convolution satisfies:

$$\mathcal{F}\{x * y\} = \mathcal{F}\{x\} \cdot \mathcal{F}\{y\}. \quad (3.4)$$

This means that *convolution in the time domain becomes multiplication in the frequency domain*, which is especially useful in signal processing and systems analysis.

3.1.2 Discrete Fourier Transform (DFT) and Fast Fourier Transform (FFT)

In practical applications, signals are often sampled and represented as discrete sequences. The *Discrete Fourier Transform (DFT)* is used to analyze such signals in the frequency domain.

Given a sequence $\mathbf{x}[n]$ of length N , the DFT is defined as:

$$DFT : \mathbf{x} \longmapsto \left(k \in \llbracket 0, N - 1 \rrbracket \mapsto \mathbf{X}[k] = \sum_{n=0}^{N-1} \mathbf{x}[n] e^{-j \frac{2\pi}{N} kn} \right). \quad (3.5)$$

The inverse DFT is given by:

$$DFT^{-1} : \mathbf{X} \longmapsto \left(n \in \llbracket 0, N - 1 \rrbracket \mapsto \mathbf{x}[n] = \frac{1}{N} \sum_{k=0}^{N-1} \mathbf{X}[k] e^{j \frac{2\pi}{N} kn} \right). \quad (3.6)$$

The *Fast Fourier Transform (FFT)* is an efficient algorithm to compute the DFT. It reduces the computational complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$, enabling fast and large-scale frequency analysis.

The convolution property also holds for the DFT:

$$DFT(\mathbf{x} * \mathbf{y}) = DFT(\mathbf{x}) \cdot DFT(\mathbf{y}). \quad (3.7)$$

Note: In the discrete case, convolution is typically *circular convolution* unless zero-padding is applied to simulate linear convolution.

3.2 Principle of acoustic scattering

Acoustic scattering describes the physical phenomenon in which an incident acoustic wave interacts with an obstacle or medium inhomogeneity, generating a scattered wave. The total acoustic field is then composed of the incident and scattered components. This process is governed by the Helmholtz equation in the frequency domain, which arises from the time-harmonic wave equation under the assumption of steady-state oscillations.

When an acoustic wave encounters a boundary, part of the wave energy is reflected, transmitted, or absorbed, depending on the properties of the boundary. In scattering problems, we are typically interested in computing the scattered field given a known incident field and boundary conditions. These conditions may be of Dirichlet, Neumann, or Robin type, depending on whether the pressure, velocity, or impedance characteristics of the surface are known.

A fundamental requirement for modeling scattering problems is the enforcement of the Sommerfeld radiation condition, which ensures that waves radiate outward and do not return from infinity. This condition guarantees the uniqueness of the solution in unbounded domains.

However, since such condition are difficult to compute numerically, alternate methods such has *perfectly matched layer* or *boundary element method* are used in practice to efficiently tackle this physical constraint.

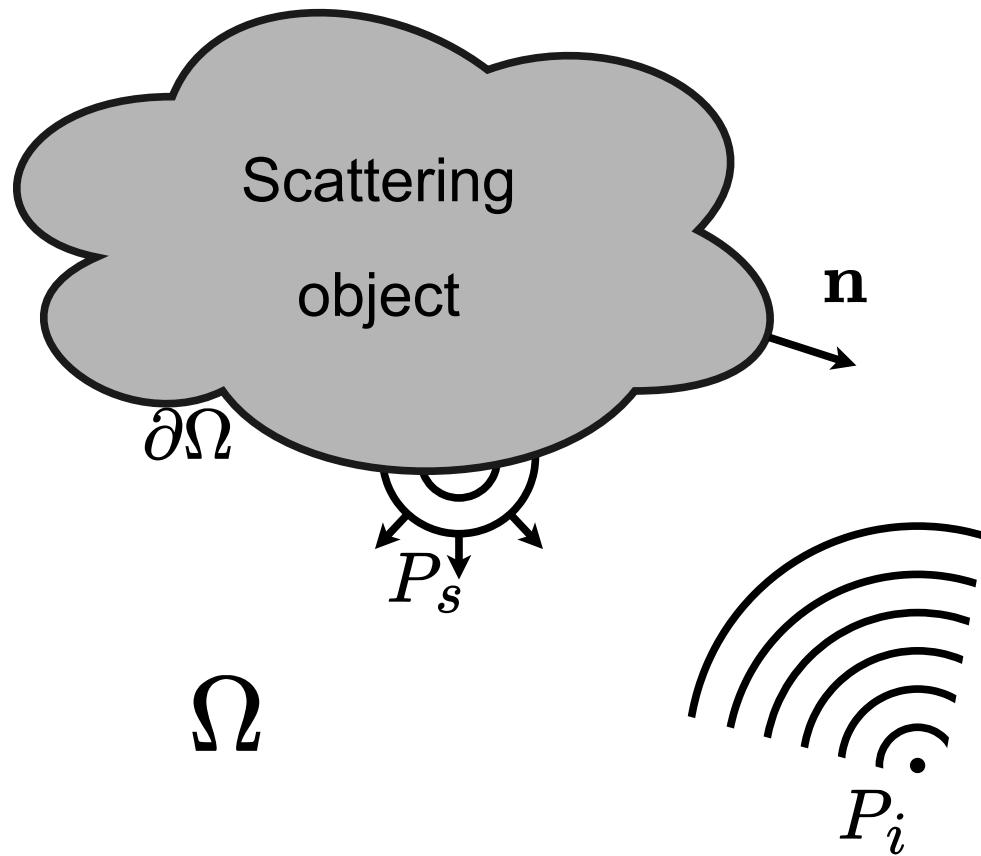


Figure 3.1: Illustration of acoustic scattering for a point source incoming wave

3.3 Mathematical model

In this work, we focus on the problem of acoustic scattering.

An incoming wave, denoted by P_i , reflects off an obstacle $\partial\Omega$, producing a scattered wave P_s .

The resulting total wave is given by:

$$P = P_i + P_s. \quad (3.8)$$

In the frequency domain, the governing equations are:

$$\begin{cases} \Delta P + k^2 P = 0, \\ \frac{\partial P}{\partial \mathbf{n}} + \frac{j\omega\rho_0}{Z} P = 0, \\ \frac{\partial P_s}{\partial \|\mathbf{x}\|}(\mathbf{x}, \cdot) - jkP_s(\mathbf{x}, \cdot) = o_{\|\mathbf{x}\| \rightarrow \infty}(\|\mathbf{x}\|^{-\frac{\dim(\Omega)-1}{2}}). \end{cases} \quad (3.9)$$

The above system admits a unique solution for a given P_i .

3.3.1 The wave equation

Given in its general form as $\Delta P + k^2 P = S_{source}$, the wave equation describes the oscillatory behavior of wave phenomena.

It admits plane waves of the form $u_d(\mathbf{x}) = e^{j\mathbf{k}\mathbf{d}\cdot\mathbf{x}}$, where \mathbf{d} is a unit direction vector, as a basis of solutions of the homogeneous equation.

Another important solution of the wave equation is the *Green's function* in the free field problem, which represents the field generated by a point source located at \mathbf{x}_s and emitting at time $t = 0$.

It satisfies:

$$\Delta G(\mathbf{x}, \mathbf{x}_s, \omega) + k^2 G(\mathbf{x}, \mathbf{x}_s, \omega) = \delta(\mathbf{x} - \mathbf{x}_s), \quad (3.10)$$

together with the Sommerfeld radiation condition:

$$\frac{\partial G}{\partial \|\mathbf{x}\|}(\mathbf{x}) - jkG(\mathbf{x}) = o\left(\|\mathbf{x}\|^{-\frac{\dim(\Omega)-1-1}{2}}\right), \quad \text{as } \|\mathbf{x}\| \rightarrow \infty. \quad (3.11)$$

Green's function for the free field problem is defined as:

$$G(\mathbf{x}, \mathbf{y}, \omega) = \begin{cases} \frac{e^{jk\|\mathbf{x}-\mathbf{y}\|}}{\|\mathbf{x}-\mathbf{y}\|} & \text{if } \dim(\Omega) = 3 \\ \frac{j}{4} H_0^{(1)}(k \|\mathbf{x} - \mathbf{y}\|) & \text{if } \dim(\Omega) = 2 \end{cases} \quad (3.12)$$

Where :

- J_0 and Y_0 both satisfies $\forall z \in \mathbb{C}, z^2 W''(z) + z W'(z) + z^2 W(z) = 0$
- $J_\nu(z) = \sum_{m=0}^{\infty} \frac{(-1)^m}{m! \Gamma(m+\nu+1)} \left(\frac{z}{2}\right)^{2m+\nu}$, the Bessel function of the first kind of order ν .
- $H_\nu^{(1)} = J_\nu + j Y_\nu$ is the Hankel function of the first kind and order ν .
- $Y_\nu(z) = \lim_{\nu \rightarrow 0} \frac{J_0(z) \cos(\nu\pi) - J_{-\nu}(z)}{\sin(\nu\pi)}$, the Bessel function of the second kind of order ν .

3.3.2 The Robin's boundary condition

The *Robin boundary condition* is a general form of boundary condition that encapsulates both Neumann and Dirichlet types.

It is given, on the boundary of the scatterer $\partial\Omega$, by:

$$\alpha \frac{\partial P}{\partial \mathbf{n}} + \beta P = g, \quad (3.13)$$

which reduces to:

- Neumann condition when $\beta = 0$
- Dirichlet condition when $\alpha = 0$

In the SONICOM dataset [12], where simulated HRTFs are generated using Mesh2HRTF [5], Neumann boundary conditions are used.

In this work, we can set the impedance parameter Z for HRTF, considering it as the impedance of bone tissue [8]. This leads to a Robin condition written as:

$$\frac{\partial P}{\partial \mathbf{n}} + \frac{j\omega\rho_0}{Z} P = 0. \quad (3.14)$$

However, due to its really high value, $Z = +\infty$ can be used for HRTF, retrieving Neumann boundary condition.

3.3.3 The Sommerfeld Radiation Condition

The *Sommerfeld radiation condition* is a mathematical constraint imposed on the scattered wave field to ensure the physical behavior of outgoing waves at infinity. It guarantees that:

- The wave radiates outward from the source.
- There is no incoming energy from infinity (i.e., no reflections from artificial boundaries).

- The total wave energy remains finite.

Formally, for a time-harmonic scattered wave $P_s(\mathbf{x})$, the Sommerfeld condition is written as:

$$\frac{\partial P_s}{\partial \|\mathbf{x}\|}(\mathbf{x}) - jkP_s(\mathbf{x}) = \lim_{\|\mathbf{x}\| \rightarrow \infty} o\left(\|\mathbf{x}\|^{-\frac{\dim-1}{2}}\right). \quad (3.15)$$

This condition holds in dimensions $\dim = 2$ and $\dim = 3$, where the scattered wave satisfies the Helmholtz equation.

Physically, this expresses that the scattered field behaves like an *outgoing spherical or cylindrical wave* at large distances. This condition is essential for the uniqueness of solutions to scattering problems.

3.4 Perfectly Matched Layer

The Perfectly Matched Layer layer was introduced by Berenger as a simulation tool for unbounded problem in Electromagnetism in 1994 [2]. It has been particularly used in Finite-Element-Method (FEM) [11] and has started to be used in PINN [36]. It consists of changing a rectangle shaped bounded domain $\Omega_c \subset \Omega$ of computation into another domain $\tilde{\Omega}_c$ ($\Omega_c \subset \tilde{\Omega}_c$) where we will consider a Laplace stretching of coordinate to simulate an infinite domain on the Perfectly Matched Layer boundary.

This transformation attenuates outgoing waves, simulating an infinite domain and preventing artificial reflections at the boundary.

Mathematically, it corresponds to the change of variable $x \mapsto \tilde{x}$ described by the following transform [11]:

$$(x_n)_{1 \leq n \leq \dim(\Omega)} \in \tilde{\Omega}_c \longmapsto (\tilde{x}_n = \int_0^{x_n} s_{z_n} dz_n)_{1 \leq n \leq \dim(\Omega)}, \quad (3.16)$$

with :

- $s_{z_n} = \gamma_{z_n} (1 + \frac{d_{z_n}}{s + \alpha_{z_n}})$
- d_{z_n} the damping factor equal to 0 inside Ω_c (we keep coordinate non-stretched in the domain of interest since we want to control only behavior at infinity)
- γ_{z_n} the grid compression factor equal to 1 inside Ω_c .
- α_{z_n} the complex frequency shift.

These parameters control the attenuation of waves outside the domain of interest, suppressing non-physical solutions and preventing reflections from infinity. Inside Ω_c , no stretching is applied to preserve the original physics.

In the PDEs, this leads to a change of derivative operators:

$$\frac{\partial}{\partial x_n} \mapsto \frac{1}{s_{x_n}} \frac{\partial}{\partial x_n}. \quad (3.17)$$

3.4.1 Wave Equation with PML

Applying the previous transform to the equations described in the previous section, we can deduce the PML-Wave Equation which can be formulated as follow :

$$\nabla_x [\mathbf{R}_x \mathbf{S}_x^{-1} \nabla_x P] + k^2 |\mathbf{S}_x| P = 0, \quad (3.18)$$

with :

$$\begin{cases} \mathbf{S}_x = \text{diag} \left((s_{x_n})_{1 \leq n \leq \dim(\Omega)} \right), \\ \mathbf{R}_x = \text{diag} \left(\left(\prod_{m \neq n} s_{x_m} \right)_{1 \leq n \leq \dim(\Omega)} \right). \end{cases} \quad (3.19)$$

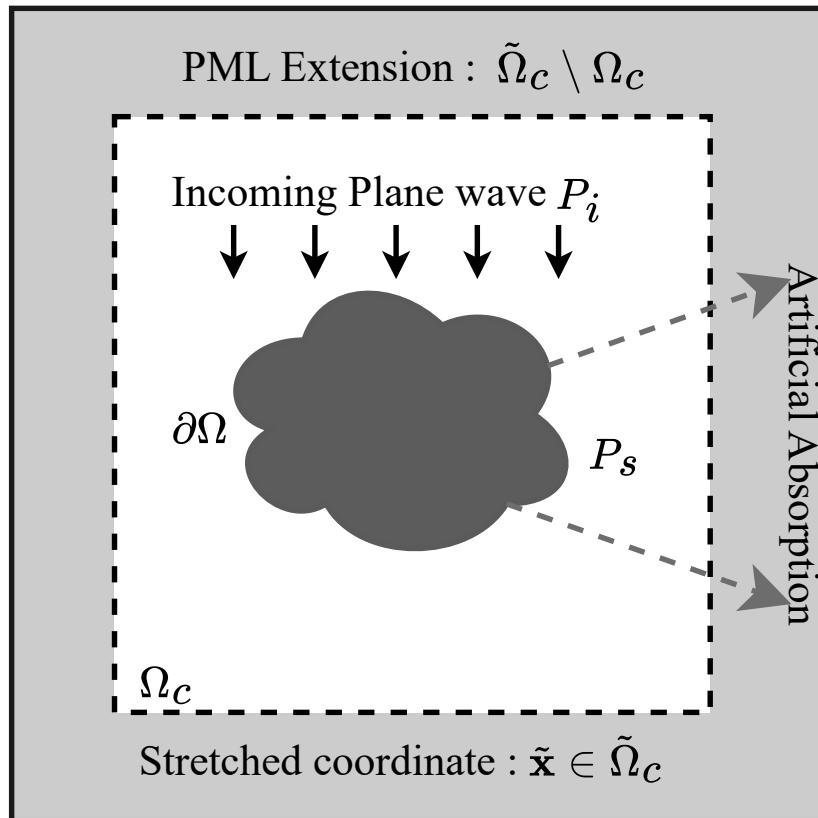


Figure 3.2: Illustration of PML domain extension

Chapter 4

Review of existing methods

4.1 Estimating HRTF based on head mesh

Most modern HRTF simulation models rely on two fundamental principles:

- The reciprocity principle
- The boundary equation formula

Leveraging these powerful tools, models such as Mesh2HRTF have achieved results closely matching measured HRTFs [5].

However, these methods still exhibit certain limitations, most notably high computational cost. For example, using full-resolution scanned head meshes remains impractical. To satisfy the widely accepted six-elements-per-wavelength criterion [24], adaptive (non-uniform) meshing is often necessary. This requirement significantly increases computational complexity and typically excludes the torso from the simulation due to its larger scale.

The *reciprocity principle*, originally introduced by Lord Rayleigh, states that in a homogeneous, linear, and time-invariant medium, if a source at point A produces a response at point B, then placing the same source at point B would yield the same response at point A—assuming identical boundary conditions surround both points. This principle is widely used in HRTF simulation to reduce computational cost by simulating a single receiver (typically in the ear canal) while placing sources at various positions around the head. However, its application in HRTF estimation can be debated. In particular, the reciprocity-based approach assumes symmetry in the acoustic transfer function, which may overlook certain path-dependent effects—especially in complex environments or when only the direct path is modeled. For instance, subtle differences in reflected paths, pinna diffraction, or asymmetric anatomical features might not be fully captured under the reciprocity assumption.

Therefore, while the principle greatly simplifies computation and is often a reasonable approximation, its limitations should be considered when high-fidelity or perceptually accurate HRTFs are required.

4.1.1 Mesh2HRTF

Mesh2HRTF is a software for HRTF estimation that has been updated and enhanced since its release in 2015 [45, 46]. Recent developments include modeling adaptations [5] and the release of a general module for scattering problems [21]. It has also been adopted as the reference simulation framework in the SONICOM dataset [12].

The main assumptions in this model are:

- the *reciprocity principle*,
- the *Green's function separability approximation*.

The source can be modeled as one of the following:

1. Source point :

We define for a source location at \mathbf{x}_s with a source amplitude p_0 ($p_0 = 0.1Pa$ in Mesh2HRTF) the incident wave as :

$$P_i(\mathbf{x}) = p_0 \frac{e^{jk\|\mathbf{x}-\mathbf{x}_s\|}}{4\pi \|\mathbf{x} - \mathbf{x}_s\|}, \quad (4.1)$$

2. Volume velocity source :

We define for an emitting source from an area A with a velocity ν_0 ($\nu_0 = 0.1m.s^{-1}$ in Mesh2HRTF). ρ_0 is the air density, the incident wave as :

$$P_i(\mathbf{x}) = -j\rho_0 ck\nu_0 A \frac{e^{jk\|\mathbf{x}-\mathbf{x}_s\|}}{4\pi \|\mathbf{x} - \mathbf{x}_s\|}, \quad (4.2)$$

3. Plane Wave :

We define for an emitting source in a direction $-\mathbf{d}$ coming from infinity:

$$P_i(\mathbf{x}) = p_0 e^{jk\mathbf{x}\cdot\mathbf{d}}, \quad (4.3)$$

Since every quantity is computed at a fixed frequency, we omit the dependance in the frequency ω in the notation. Mesh2HRTF implements a the Burton Miller Method applied on the velocity potential $\phi(\mathbf{x}) = \frac{p(\mathbf{x})}{j\omega\rho}$ and its normal derivative $v(\mathbf{x}) = \frac{\partial\phi}{\partial\mathbf{n}}(x)$. It is based on both the Green's representation derived from the wave equation and

boundary condition of our problem :

$$\forall \mathbf{x} \in \bar{\Omega}, \lambda(\mathbf{x})\phi(\mathbf{x}) - \int_{\partial\Omega} H(\mathbf{x}, \mathbf{y})\phi(\mathbf{y})d\mathbf{y} + \int_{\partial\Omega} G(\mathbf{x}, \mathbf{y})v(\mathbf{y})d\mathbf{y} = \phi_i(\mathbf{x}), \quad (4.4)$$

with :

$$\begin{cases} H(\mathbf{x}, \mathbf{y}) = \frac{\partial G}{\partial \mathbf{n}(\mathbf{y})}(\mathbf{x}, \mathbf{y}) = \left(jk - \frac{1}{\|\mathbf{x} - \mathbf{y}\|} \right), \\ \lambda(\mathbf{x}) = 1 - \frac{\chi_{\partial\Omega}}{2}. \end{cases} \quad (4.5)$$

The Burton–Miller method consists in adding the normal derivative of (4.4), multiplied by a coupling factor $\gamma = \frac{i}{k}$:

$$\forall \mathbf{x} \in \partial\Omega, \frac{1}{2}v(\mathbf{x}) - \int_{\partial\Omega} E(\mathbf{x}, \mathbf{y})\phi(\mathbf{y})d\mathbf{y} + \int_{\partial\Omega} H'(\mathbf{x}, \mathbf{y})v(\mathbf{y})d\mathbf{y} = v_i(\mathbf{x}), \quad (4.6)$$

with :

$$\begin{cases} E(\mathbf{x}, \mathbf{y}) = \frac{\partial^2 G}{\partial \mathbf{n}(\mathbf{x}) \partial \mathbf{n}(\mathbf{y})}(\mathbf{x}, \mathbf{y}), \\ H'(\mathbf{x}, \mathbf{y}) = \frac{\partial G}{\partial \mathbf{n}(\mathbf{x})}(\mathbf{x}, \mathbf{y}). \end{cases} \quad (4.7)$$

On a discrete mesh where $V = \{\mathbf{x}_i\}$, $F = \{F_j\}$, after adding previous equation multiplied by the coupling factor $\gamma = \frac{i}{k}$ into the mix, this consists in solving:

$$\frac{\boldsymbol{\phi} - \gamma \mathbf{v}}{2} - (\mathbf{H} - \gamma \mathbf{E}) + (\mathbf{G} - \gamma \mathbf{H}') = \mathbf{v} = \boldsymbol{\phi}_i - \gamma \mathbf{v}_i, \quad (4.8)$$

with bold notation matrix consisting in $\mathbf{A}_{ij} = \int_{F_j} A(\mathbf{x}_i, \mathbf{y})d\mathbf{y}$.

Computing the matrices \mathbf{G} , \mathbf{H} , \mathbf{H}' , and \mathbf{E} can be computationally expensive and requires special care when handling singularities in the Green's function and its derivatives—issues for which Mesh2HRTF provides dedicated numerical strategies.

To accelerate computation, a *Fast Multipole Method* (FMM) is used, which approximates:

$$G(\mathbf{x}, \mathbf{y}) \approx G_1(\mathbf{x} - \mathbf{z}_1) G_2(\mathbf{z}_1 - \mathbf{z}_2) G_3(\mathbf{z}_2 - \mathbf{y}), \quad (4.9)$$

enabling a separation of computations over F_j and \mathbf{x}_i and reducing the complexity from $\mathcal{O}(N^2)$ to approximately $\mathcal{O}(N \log N)$.

This approach achieves remarkable results, and is one of the few simulation methods able to closely match measured HRTFs over a wide frequency range. However, despite its relatively optimized computational cost, simulations typically require several hours on a single CPU and demand careful preprocessing of the mesh (often performed manually) to properly manage Green's function singularities.

While torso modeling can be included, it is often omitted due to additional computational cost. Moreover, the inference time remains $\mathcal{O}(N)$, since evaluating the HRTF in one direction still requires solving a boundary integral. Finally, the method does not

perform well for near-field sources—primarily due to the limitations of the reciprocity principle.

4.2 Scattering with PINN

Multiple scattering simulation via physics-informed neural networks

This paper introduces two architectures *b-PINN* and *s-PINN* to simulate scattering fields in a subdomain $\Omega \subset \mathbb{R}^2$ by solving the problem (3.9) with an enforcement of the Sommerfeld condition as $\frac{\partial P_s}{\partial \mathbf{n}} - jkP_s = 0$ on the exterior boundary on the computational domain [27] in addition to Neumann conditions and the wave equation.

This condition is problematic because a solution minimizing the loss of the model is not solution of the original problem (3.9).

Their model, however, provides interesting results for their resnet based PINN (*b-PINN*) and its generalization to multiple scatterer (*s-PINN*).

They also provides further explanation on the size of the used network regarding the frequency of the incoming wave.

Unfortunatly, the code of their work and the setup for each provided scatterer plot is not provided, so no comparison will be done with this work (only one figure will be provided).

Physics-Informed Neural Networks for Estimation of Scattered Sound Fields with Boundary Condition

This paper uses some a few number of microphone recordings to estimates the scattered field while knowing the scatterer shape in \mathbb{R}^3 [28].

Here the impedance Z is used as a regularization, assuming the scatterer possess a uniform impedance around its boundary. This results in minimizing the difference of estimated impedance :

$$\int_{\partial\Omega \times \partial\Omega} |Z(\mathbf{x}) - Z(\mathbf{x}')|^2 d\mathbf{x} d\mathbf{x}' \quad (4.10)$$

with $Z(\mathbf{x}) = \frac{p(\mathbf{x})}{v(\mathbf{x})}$. This is a sort of reformulation of the Robin's boundary condition (3.14), where Z is the unknown variable.

Such modelisation allows to learn the scattered fields even when Z is unknown. However, the main drawback of this method is that a sufficient number of pressure points need to be measured.

Even though, those two works prove that solving the scattering problem with PINN can be possible and that we can benefit of the powerful PINN framework for this problem.

Chapter 5

Methodology

In this chapter, we described the proposed method in this works which consists in a PML-based model of the scattering problem incorporated in the loss of a PINN. A Random Fourier Features based PINN architecture is proposed to be the support of a hypernetwork PHISK that output continuous representation of the scattering field for every direction in $\mathbb{S}^{\dim(\Omega)}$ using the power of Low-Rank Adaptation.

5.1 Proposed method

Although the reciprocity principle appears theoretically sound, its idealized assumptions may not hold in practice for HRTF estimation:

- The head, even if often considered as a sound-hard obstacle, is, as measured in [8], a penetrable obstacle with non-homogeneous medium.
- The symmetry of the boundary condition can be contested especially with the complex effects introduced by the Pinnae and the Head, especially for a source in the Near-field.

Also, the two independent source assumption of HRTF may be broken when we simulate more ears by adding microphones, such as in HoloLens.

Understanding these issues, we took a forward approach to HRTF estimation, placing the source at its real position and simulating the full field from there.

To tackle this problem, we proposed a physics-based approach for training our network, proposing an architecture denoted 'Reference PINN' to solve the scattering problem for an instance of the problem defined by a direction \mathbf{d} (or a source position \mathbf{x}_s) and a frequency f .

Furthermore, we propose a hypernetwork architecture "PHysics-Informed Scattering hypernetworK" (PHISK) to enable the production of scattering field for every direction $\mathbf{d} \in \mathbb{S}^{\dim(\Omega)} = \{u \in \mathbb{R}^{\dim(\Omega)} | \|u\| = 1\}$ using a reference PINN trained for a

single instance (f, \mathbf{d}) of the scattering problem and LoRA [17] adapted networks for $(f, \mathbf{d}_j)_{j \leq J}$ where J is the number of adapted direction.

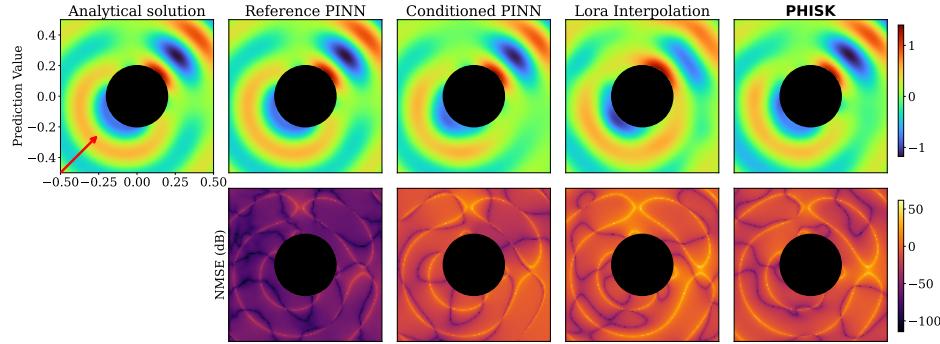


Figure 5.1: Visual comparison of the estimated scattered field by our models for incident direction $\mathbf{d} = (0.7, 0.7)$ at $f = 1\text{kHz}$.

5.1.1 Mathematical model

As presented in the previous chapter (3), we adopt a perfectly matched layer representation of the scattering problem with Robin's condition, which consists in finding the solution of :

$$\begin{cases} \nabla_{\mathbf{x}}[\mathbf{R}_{\mathbf{x}}\mathbf{S}_{\mathbf{x}}^{-1}\nabla_{\mathbf{x}}P] + k^2|\mathbf{S}_{\mathbf{x}}|P = 0 \\ \frac{\partial P}{\partial \mathbf{n}} + \frac{j\omega\rho_0}{Z}P = 0 \end{cases} \quad (5.1)$$

In fact, due to our only interest in the scattering field P_s , we can reformulate the previous (5.1) in function of P_i as :

$$\begin{cases} \nabla_{\mathbf{x}}[\mathbf{R}_{\mathbf{x}}\mathbf{S}_{\mathbf{x}}^{-1}\nabla_{\mathbf{x}}P_s] + k^2|\mathbf{S}_{\mathbf{x}}|P_s = 0 \\ \frac{\partial P_s}{\partial \mathbf{n}} + \frac{j\omega\rho_0}{Z}P_s = -\frac{\partial P_i}{\partial \mathbf{n}} - \frac{j\omega\rho_0}{Z}P_i \end{cases} \quad (5.2)$$

As, we only use two types of incident wave in our work (plane wave and point source based), the right hand side of Robin's boundary condition of (5.2), can be simplified as :

- $-(jk\mathbf{n} \cdot \mathbf{d} + \frac{j\omega\rho_0}{Z})e^{ik\mathbf{d} \cdot \mathbf{x}}$, if $P_i(\mathbf{x}) = e^{ik\mathbf{d} \cdot \mathbf{x}}$ describes a plane wave.
- $-p_0 \frac{e^{jk\|\mathbf{x}-\mathbf{x}_s\|}}{4\pi\|\mathbf{x}-\mathbf{x}_s\|} [(jk - \frac{1}{\|\mathbf{x}-\mathbf{x}_s\|}) \frac{(\mathbf{x}-\mathbf{x}_s) \cdot \mathbf{n}}{\|\mathbf{x}-\mathbf{x}_s\|} + \frac{j\omega\rho_0}{Z}]$, if $P_i(\mathbf{x}) = p_0 \frac{e^{jk\|\mathbf{x}-\mathbf{x}_s\|}}{4\pi\|\mathbf{x}-\mathbf{x}_s\|}$ describes an incoming wave coming from a point source located at \mathbf{x}_s .

To simplify the description, we will define two operator :

- **Robin's boundary operator:**

$$\mathcal{R} : U \in L^2(\partial\Omega) \longmapsto \frac{\partial U}{\partial \mathbf{n}} + \frac{j\omega\rho_0}{Z}U, \quad (5.3)$$

such that a function U satisfying the Robin's boundary condition verifies :

$$\mathcal{R}(U) = 0. \quad (5.4)$$

- **PML-Wave equation operator:**

$$\mathcal{W}^{\text{pml}} : U \in L^2(\tilde{\Omega}_c) \longmapsto \nabla_{\mathbf{x}}[\mathbf{R}_{\mathbf{x}} \mathbf{S}_{\mathbf{x}}^{-1} \nabla_{\mathbf{x}} U] + k^2 |\mathbf{S}_{\mathbf{x}}| U, \quad (5.5)$$

such that a function U satisfying the PML-Wave equation verifies :

$$\mathcal{W}^{\text{pml}}(U) = 0. \quad (5.6)$$

With those two operator, (5.2) can be written simply as :

$$\begin{cases} \forall \mathbf{x} \in \tilde{\Omega}_c, \mathcal{W}^{\text{pml}}(P_s)(\mathbf{x}) = 0 \\ \forall \mathbf{x} \in \partial\Omega, \mathcal{R}(P_s + P_i)(\mathbf{x}) = 0 \end{cases} \quad (5.7)$$

5.1.2 Perfectly matched layer parameter

We specify the parameter of the perfectly matched layer using a quadratic decay outside the domain of interest, making the wave evanescent, $s_{x_n} = 1 + j\frac{\sigma_{x_n}}{\omega}$ [9] with :

$$\begin{cases} \sigma_{x_n} = a_0 \omega \left(\frac{l_{x_n}}{L_{\text{PML}}} \right)^2 & \text{if } \mathbf{x} \in \tilde{\Omega}_c \setminus \Omega_c \\ = 0 & \text{otherwise} \end{cases} \quad (5.8)$$

5.2 Network definition

We will denote a neural network as a function Φ_θ defined by its architecture and its parameters θ . For an input $\mathbf{x} \in \tilde{\Omega}_c$, the network produces the output $\Phi_\theta(\mathbf{x})$. The different architecture evaluated in this works will be explained in the following section after defining the loss of our network.

5.2.1 Loss definition

As the equations (5.7) fully describes the scattering problem, a function U satisfying $\int_{\tilde{\Omega}_c} |\mathcal{W}^{\text{pml}}(U)(\mathbf{x})|^2 d\mathbf{x} + \int_{\partial\Omega} |\mathcal{R}(U + P_i)(\mathbf{x})|^2 d\mathbf{x} = 0$ should satisfies the initial problem as

$U = P_s$. As for, for every finite batch of points $\mathcal{B} \subset \tilde{\Omega}_c$, $\mathcal{B}_{bc} \subset \partial\Omega$, the network should minimized the residuals of the governing equation, defining the loss as :

$$\mathcal{L}_{\text{PINN}}(\theta) = \lambda_{\text{pml}} \mathcal{L}_{\text{pml}}(\theta) + \lambda_{\text{bc}} \mathcal{L}_{\text{bc}}(\theta) \quad (5.9)$$

with :

$$\mathcal{L}_{\text{pml}}(\theta) = \frac{1}{|\mathcal{B}|} \sum_{\mathbf{x} \in \mathcal{B}} |\mathcal{W}^{\text{pml}}(\Phi_\theta)(\mathbf{x})|^2, \quad (5.10)$$

$$\mathcal{L}_{\text{bc}}(\theta) = \frac{1}{|\mathcal{B}_{\text{bc}}|} \sum_{\mathbf{x} \in \mathcal{B}_{\text{bc}}} |\mathcal{R}(\Phi_\theta + P_i)(\mathbf{x})|^2. \quad (5.11)$$

The loss weights ($\lambda_{\text{pml}}, \lambda_{\text{bc}}$) are fixed prior to the train as results suggests that using adaptive weights as in the paper from Xiang and al. [44] may create divergence in the training especially during L-BFGS training phase (2.4.1).

5.2.2 Reference PINN architecture

As for the reference PINN architecture, we explored many possibilities of architectures whose performance would be detailed in the next chapter (6). The first question that came was the structure of the output :

- Should the output of our network Φ_θ represents the polar decomposition (magnitude and phase) of the complex pressure field or should it output the real and imaginary part ?

Experiments seems to show that the later representation works better for the scattering problem due to the high correlation between real and imaginary part for scattering fields.

For the architecture, ablation studies show that the number of hidden neurons per layer is much more important than the depth of the network itself. The proposed reference PINN architecture begins with a Random Fourier Features (2.3) embedding if the activation function is Tanh based, followed by alternating layers of MLPs and activation functions.

Choice of activation function

The choice of activation function is quite complicated as tanh (2.5) requires a Random Fourier Features embedding to be really effective, adding computational cost compared to Sine (2.4). Sine function however, despite working really well if $\dim(\Omega) = 2$ encountered convergence issues in some case when $\dim(\Omega) = 3$.

We found that QuadraticTanh was a great choice, allowing faster and stable convergence of the network in every dimension while also giving the best results when

compared to ground truth.

As a consequence, when the activation is not detailed in the experiments chapter (6), QuadraticTanh is the activation function with a fixed α parameter.

Training procedure

The training procedure consists at taking a batch of points in the exterior of the obstacle \mathcal{B} and a batch of points on the boundary of the obstacle \mathcal{B}_{bc} and then, doing a certain number of optimization steps, wanting to minimize \mathcal{L}_{PINN} (5.9) for the chosen batches. This process is repeated at first with a certain number of batch and an Adam optimizers. The latest training phase is done with the same procedure but using an L-BFGS optimizer (2.4).

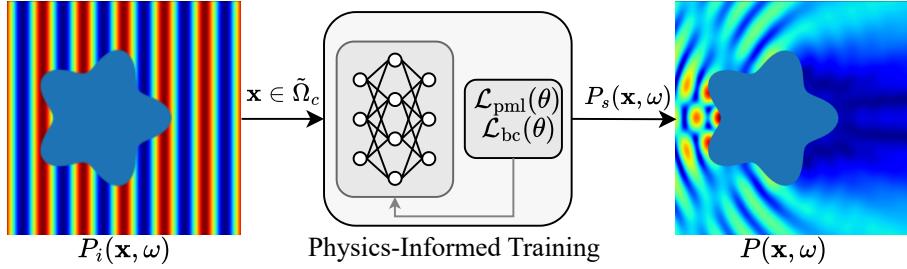


Figure 5.2: Example of our network output for a star shaped obstacle at $f = 2\text{kHz}$

5.2.3 LoRA adapted reference PINN

LoRA principle

Low-Rank Adaptation (LoRA) [17] has been introduced has a method to efficiently fine-tune large pre-trained models by injecting trainable low-rank matrices into each layer of the network. Instead of updating the full set of model weights, LoRA freezes the original parameters and learns a much smaller number of parameters in a low-rank decomposition, significantly reducing memory and computational costs during fine-tuning.

Mathematically, if we denote the layer i weights of a trained network as $\mathbf{W}_i \in \mathbb{R}^{N \times M}$ (and the bias $\mathbf{b}_i \in \mathbb{R}^N$), LoRA consists in finding a couple of low rank R matrix $(\mathbf{A}_i, \mathbf{B}_i) \in \mathbb{R}^{N \times R} \times \mathbb{R}^{R \times M}$ such that the network defined by the layers $(\mathbf{W}_i + \mathbf{A}_i \mathbf{B}_i, \mathbf{b}_i)_{i < N_{layer}}$ is fine-tuned to a novel instance of a problem without much time of training needed.

Using this method, one can assume that the reference PINN has captured enough information from one instance (f, \mathbf{d}) of the scattering problem, so that finding the solution to a novel instance (f, \mathbf{d}') can be done by only adapting the reference PINN with LoRA

weights such adapted network will be denoted LoRA PINN.

LoRA Interpolation

We propose the idea of LoRA Interpolation in the spatial domain.

Assuming, we have trained a reference PINN for an instance (f, \mathbf{d}^{ref}) of the scattering problem, we can estimate other solution of the scattering problem for J other directions $(\mathbf{d}^j)_{j < J}$ using LoRA scheme and denoting the resulting LoRA weights for direction \mathbf{d}^j as $(\mathbf{A}_i^j, \mathbf{B}_i^j)_{i < N_{layers}, j < J}$.

Let's defined the interpolated LoRA-InterPINN network for a given direction $\mathbf{d} \in \mathbb{S}^{\dim(\Omega)}$ as the network defined by :

$$\left(\mathbf{W}_i + \sum_{j=0}^{J-1} \alpha_j(\mathbf{d}) \mathbf{A}_i^j \mathbf{B}_i^j, \mathbf{b}_i \right)_{i < N_{layers}} \quad (5.12)$$

with $(\alpha_j)_{j < J}$ the weights function verifying the following properties:

- $\alpha_j : \mathbb{S}^{\dim(\Omega)} \mapsto [0, 1]$ are continuous and we only define positive weights.
- $\forall \mathbf{d} \in \mathbb{S}^{\dim(\Omega)}, \sum_{j=0}^{J-1} \alpha_j(\mathbf{d}) = 1$, the α_j corresponds to a barycenter in $\mathbb{S}^{\dim(\Omega)}$.

We can see that if we define $(\alpha_j)_{j < J}$ such that $\forall j < J, \alpha_j(\mathbf{d}_j) = 1$ then the LoRA-InterPINN should give a correct solution for at least the directions $(\mathbf{d}_j)_j$.

In fact such interpolation can give really coarse version of a solution for simple scatterer as we will see in the experiment part (Chapter 6). Multiple interpolation method can be used which give versatility to the given framework.

5.3 Hypernetwork PHISK definition

The LoRA-InterPINN is a way to estimate the scattering fields continuously in $\mathbb{S}^{\dim(\Omega)}$ without needing to retrain or adapt the reference network for new direction.

However such interpolation is really coarse and the field became too badly estimated, especially when we go far from the trained direction $(\mathbf{d}_j)_j$ (in $\mathbb{S}^{\dim(\Omega)}$ distance). To tackle this issue, we propose to 'correct' the interpolation using a hypernetwork (PHISK) that will predict interpolation correction in order to enhance the produced pressure field.

5.3.1 Principle of PHISK

A hypernetwork is a neural network that output weights of another network for a given input.

Applied to the problem of direction conditioned scattering, the hypernetwork PHISK,

$\mathcal{H} : \mathbf{d} \in \mathbb{S}^{\dim(\Omega)} \mapsto (\mathbf{C}_i(\mathbf{d}), \mathbf{c}_i(\mathbf{d}))_{i < N_{layers}}$ such that the corrective weights \mathbf{C}_i and \mathbf{c}_i enables to adjust LoRA-InterPINN network.

As a consequence, for a given direction \mathbf{d} , the network $\Phi_\theta^{\text{PHISK}}(\mathbf{d})$ predicting the scattering field would be given by :

$$\left(\mathbf{W}_i + \mathbf{C}_i(\mathbf{d}) + \sum_{j=0}^{J-1} \alpha_j(\mathbf{d}) \mathbf{A}_i^j \mathbf{B}_i^j, \mathbf{b}_i + \mathbf{c}_i(\mathbf{d}) \right)_{i < N_{layers}} \quad (5.13)$$

5.3.2 Loss consideration

The loss used for PHISK does not differ so much from the one presented for the reference network (5.9), the only difference is the addition of a regularization loss and a computation of the loss over multiple directions, in order to constraint the output of PHISK to give weights that are not prevalent over the original structure.

This results in the following loss, defintion, using the same notation as in (5.2.1) :

$$\mathcal{L}_{\text{PHISK}}(\theta) = \lambda_{\text{pml}} \mathcal{L}_{\text{pml}}(\theta) + \lambda_{\text{bc}} \mathcal{L}_{\text{bc}}(\theta) + \lambda_{\text{hyper}} \mathcal{L}_{\text{hyper}}(\theta) \quad (5.14)$$

where, using the same batches $\mathcal{B}, \mathcal{B}_{\text{bc}}$ aswell as a small direction batch $\mathcal{B} \subset \mathbb{S}^{\dim(\Omega)}$:

$$\mathcal{L}_{\text{pml}}(\theta) = \frac{1}{|\mathcal{B}|} \sum_{\mathbf{d} \in \mathcal{B}_{\text{hyper}}} \sum_{\mathbf{x} \in \mathcal{B}} |\mathcal{W}^{\text{pml}}(\Phi_\theta^{\text{PHISK}}(\mathbf{d}))(\mathbf{x})|^2 \quad (5.15)$$

$$\mathcal{L}_{\text{bc}}(\theta) = \frac{1}{|\mathcal{B}_{\text{bc}}|} \sum_{\mathbf{d} \in \mathcal{B}_{\text{hyper}}} \sum_{\mathbf{x} \in \mathcal{B}_{\text{bc}}} |\mathcal{R}(\Phi_\theta^{\text{PHISK}}(\mathbf{d}) + P_i)(\mathbf{x})|^2 \quad (5.16)$$

$$\mathcal{L}_{\text{hyper}}(\theta) = \sum_{\mathbf{d} \in \mathcal{B}_{\text{hyper}}} \|\mathcal{H}(\mathbf{d})\| \quad (5.17)$$

Training procedure

The training procedure consists at taking a batch of points in the exterior of the obstacle \mathcal{B} , a batch of points on the boundary of the obstacle \mathcal{B}_{bc} and a batch of directions $\mathcal{B}_{\text{hyper}}$ then, doing a certain number of optimization step, wanting to minimize $\mathcal{L}_{\text{PHISK}}$ (5.14) for the chosen batches.

The batches $\mathcal{B}, \mathcal{B}_{\text{bc}}$ and $\mathcal{B}_{\text{hyper}}$ are changed in an asynchronous way in order for a same batch of points to see multiple batch of directions.

This process is done with Adam optimizer, since L-BFGS does not improve the results so much after an Adam training phase.

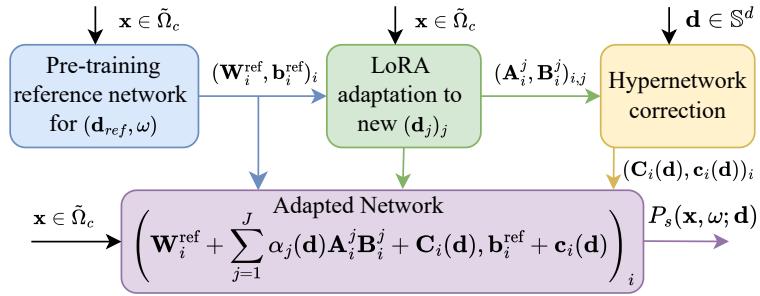


Figure 5.3: Training process of the hypernetwork *PHISK*

5.4 Sampling method

Sampling is of relative importance for PINN as many paper have proven that the effect on learning capacity was critical [16]. We can see below the effect of different sampling method on an example of an expanding wave from a gaussian source point :

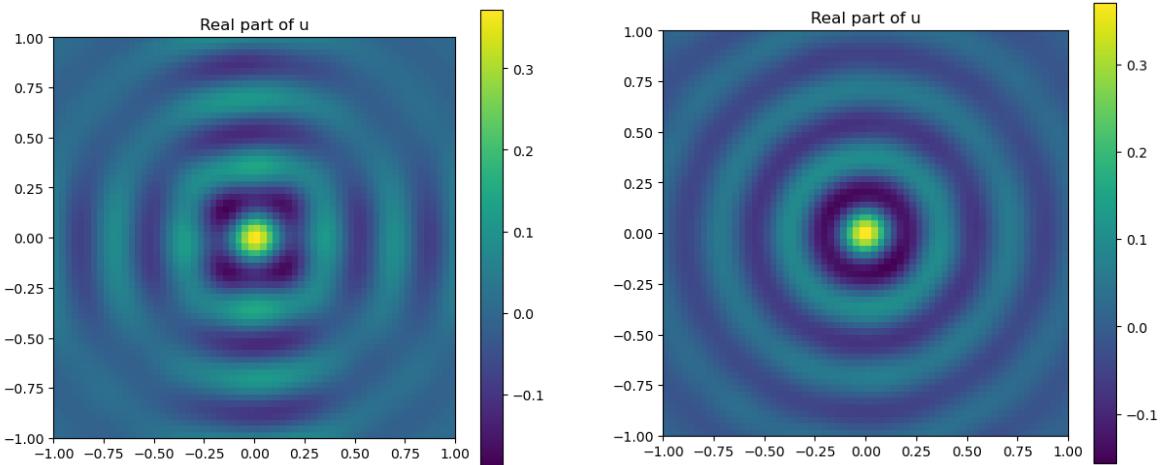


Figure 5.4: Sampling on the whole domain at each epochs

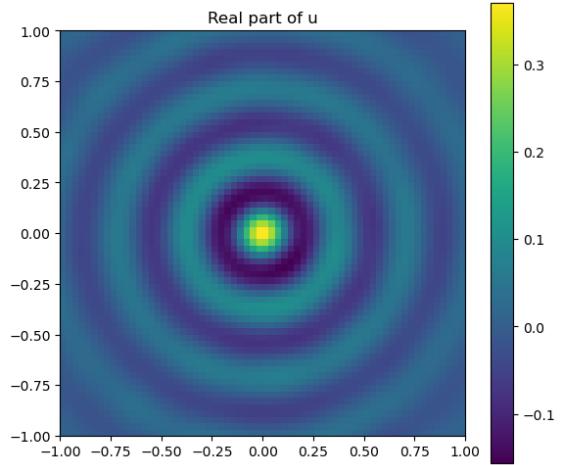


Figure 5.5: Sampling in increasing disk as a function of the epoch

Figure 5.6: Comparison of training results for an expanding wave problem, using different sampling method and a Gaussian source point.

We can see that the sampling method is critical in the training process that is why we will discuss the method used for PHISK and reference PINN training in this section.

5.4.1 Sampling on the boundary of the obstacle

The boundary of the obstacle is described as a mesh $\mathcal{T} = (V, E, F)$, $V = \{\mathbf{x}_i, i \leq |V|\}$ the set of ordered vertices. The first sampling strategy used in this is to sample a batch of random indices $\text{Rand}\mathcal{B}$ according to uniform law on $\llbracket 0, |V| \rrbracket$, such that $\mathcal{B}_{bc} = \{\mathbf{x}_i | i \in \text{Rand}\mathcal{B}\}$ defines our random batch on the boundary of the scatterer.

This method works well for objects where \mathcal{T} has uniform edge lengths (*e.g.* $\forall e \in E, \|e\| \approx K$) however for non uniform meshing the probability to sample a lot around the dense part of the mesh is high (which is the case with HRTF head meshes where ears have a more refined resolution than the rest of the head).

To tackle this, we employ Poisson disk sampling, a technique designed to produce more evenly spaced sample points over the surface of the mesh.

Unlike uniform random sampling over vertices—which can lead to clustering in densely meshed regions—Poisson sampling enforces a minimum distance between sample points, promoting better spatial coverage and avoiding over-representation of areas with high vertex density.

Formally, a Poisson disk sampling strategy generates a set of points $\mathcal{B}_{bc} \subset \partial\Omega$ such that for any two points $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{B}_{bc}$, the geodesic or Euclidean distance satisfies:

$$\text{dist}(\mathbf{x}_i - \mathbf{x}_j) \geq r_{min} \quad (5.18)$$

for some fixed radius r_{min} , ensuring samples are well-distributed over the surface. This approach is particularly effective for boundary sampling on non-uniform meshes (*e.g.*, high-resolution HRTF head models), where it reduces sampling bias caused by mesh density and leads to improved training stability and generalization in physics-informed neural networks (PINNs).

However, Poisson sampling is computationally expensive, especially on large or high-resolution meshes where neighbor queries are costly and we need to sample a lot of points for our training.

To retain good spatial coverage while significantly reducing computational overhead, we instead adopt a Halton sequence-based sampling strategy—a deterministic, low-discrepancy alternative that approximates blue noise distributions :

Halton sampling [42].

Halton sequences provide quasi-random sampling with low discrepancy, meaning they fill the sampling domain more uniformly than purely random samples. In our implementation, two 1D Halton sequences (*e.g.*, in bases 2 and 3) are combined and transformed into sample indices over the boundary mesh:

$$\text{index}_i = \lfloor (\text{Halton}_2(i) + \text{Halton}_3(i) \cdot \gamma) \cdot |V| \rfloor, \quad (5.19)$$

where γ is an irrational constant (*e.g.* $\gamma \approx 0.754877$) used to decorrelate dimensions.

To introduce variation over training epochs, an offset based on the golden ratio is added:

$$\text{Halton}_2(i) \leftarrow (\text{Halton}_2(i) + \text{epoch} \cdot \phi) \bmod 1, \quad (5.20)$$

where $\phi \approx 0.618034$ is the golden ratio.

This method provides:

- **Fast sampling:** avoiding rejection sampling or neighborhood queries,
- **Determinism:** making training reproducible,
- **Better distribution** than purely random sampling, especially on irregular meshes,
- **Epoch-wise diversity** without significant overhead.

As a result, this Halton-based sampling method serves as a drop-in, scalable alternative to Poisson sampling, particularly suited for our scattering problem networks loops where speed and memory efficiency are critical.

5.4.2 Sampling in the extended PML domain

Sampling when $\dim(\Omega) = 2$

In 2D, a sufficient batch size allows to cover the whole domain at every epochs, allowing to take random sample uniformly in $\dim(\Omega) = 2$.

This is not the case in 3D where a different approach allows to have better results whereas changing our approach in 2D made close to no difference.

Sampling when $\dim(\Omega) = 3$

In 3D, the domain being much larger, a grid of 8096, for example in a $1m^3$ cube, allows only to have a resolution of $5cm$ which is really coarse whereas it will be around $1cm$ if the same batch size was used in a 2D square. This is even more true for our problem since the scatterer occupies a 'bigger' space relatively to the measure of the domain in 2D than in 3D.

To tackle, this we propose to sample uniformly for a part of our batch in the extended PML region of $\tilde{\Omega}_c$ (*e.g.* $\tilde{\Omega}_c \setminus \Omega_c$) allowing our network to learn PML behavior. On the other hand, for the other part of the batch inside Ω_c , we sample in a spherical way according to a radius uniform distribution.

We use Marsaglia's method [25] to generate points uniformly on $\mathbb{S}^{\dim(\Omega)}$ that we then multiply by the desired radius, using as minimum radius the one of the biggest sphere contained in the scatterer. This does not corresponds exactly to uniform sampling with cartesian coordinate since we are more likely to reject high radius and low

radius points that could be outside of Ω_c or inside the scatterer. The distribution of observed radius confirm this behavior allowing to have a evenly spread representation of points in space.

All points inside $\dim(\Omega) = 3$ are pre-computed before training to benefit from optimized numpy librairies.

5.4.3 Direction sampling

For direction sampling, we uses a simple strategy as our direction batches are quite small (with J directions), so we deterministically choose J directions $(d^l)_{l \leq J}$ evenly spread in $\mathbb{S}^{\dim(\Omega)}$ that we perturbate with a small noise ϵ^l defined by :

$$\epsilon^l \sim \mathcal{N}(0, \max(C \frac{N_{\text{current epoch}}}{N_{\text{epochs total}}} + \epsilon, 1)\sigma^2) \quad (5.21)$$

where C is a multiplicative factor and ϵ a minimal noise factor such that the resulting direction batch consists in :

$$\mathcal{B}_{\text{hyper}} = \{d^l + \epsilon^l | l \leq J\} \quad (5.22)$$

Chapter 6

Experiments

6.1 Experimental setup

6.1.1 Hyperparameters

- a_0 is set to be 5 in every simulation. The influence of this parameter is essential for the network to learn. A too high value of a_0 will 'shred' the produced field making it have low values almost everywhere. A too low value will 'disable' the PML Layer making the network find one of the many solution of the solution of the problem without the Sommerfeld condition (3.15) solution since uniqueness is not guarantee in the absence of PML. From the experiment, a value between 1 and 10 is recommended.
- L_{pml} can be adapted to approximately $0.25L$. The size of L_{pml} depends on the importance we want to give to the PML Layer. A too small L_{pml} could neglect the effect of PML, making the PINN create non physical solution whereas a too huge L_{pml} would make the network learn on an unnecessary broad domain, reducing the effectiveness of each training step or requiring bigger batch size.
- The minimal noise ϵ is set to 0.05 in every experiment.
- The evolution of noise variance factor C is set to 8 in every experiment.
- The batch size for the Adam phase is set to 4096 when $\dim(\Omega) = 2$ and 8096 when $\dim(\Omega) = 3$.
- The temperature for PHISK will be discussed in the appropriate section.
- $\lambda_{\text{pml}} = 1, \lambda_{\text{bc}} = 100, \lambda_{\text{hyper}} = 0.001$, the set of parameters weights.

6.1.2 Training setup

We fix the number of epochs for each training phase prior to the training, based on empirical observation. Learning the scattered field at higher frequencies may require more epochs to converge compared to low frequency. The defining point is the product fL which controls what is high frequency or low frequency for our network. As a result, the network will learn with the same accuracy a problem where $L' = 0.5L$ and $f' = 2f$ with same relative scatterer size.

We always set $L = 0.5$ in every provided experiment so that the predicted field fits in a $1\text{m}^{\dim(\Omega)}$ box allowing to compare the results to other studies [28].

6.1.3 Selection of LoRA Rank Parameters

The rank of LoRA is set to $R = 12$ such that each couple $(\mathbf{A}_i^j, \mathbf{B}_i^j) \in \mathbb{R}^{N_i \times R} \times \mathbb{R}^{R \times N_{i+1}}$ is expressive enough to capture the structure of the spherical harmonics in the plane $z = 0$ [23].

This number may be dropped to $R = 4$ when $\dim(\Omega) = 2$ while keeping relatively good performance, however since our main interest is $\dim(\Omega) = 3$, we have not further investigated if setting $R = 4$ was physically meaningful in this two dimensional case.

6.2 Evaluation method

6.2.1 Case $\dim(\Omega) = 2$

For evaluation, we compare the scattering field estimated by our network with the analytical solution for a sound-hard circular obstacle. In this case, the obstacle is a perfectly rigid cylinder (i.e., Neumann boundary condition), and the solution is derived from classical scattering theory [15].

For a circle of radius a centered at the origin, the scattered field in polar coordinates is given by:

$$P_s(r, \psi) = \sum_{n=-\infty}^{\infty} j^n \frac{H_n^{(1)\prime}(ka)}{J_n'(ka)} J_n(kr) e^{jn(\psi - \psi_d)}, \quad (6.1)$$

where:

- J_n and $H_n^{(1)}$ are the Bessel and Hankel functions of the first kind of order n as defined in (3.12),
- J_n' and $H_n^{(1)\prime}$ denote derivatives with respect to the argument,
- ψ_d is the angle of arrival of the incident wave \mathbf{d} (e.g., $\psi_d = \arctan 2(d_y, d_x)$).

This series converges rapidly for $r > a$, so a finite number of terms is sufficient to approximate the analytical solution for comparison. The error between the predicted field and the analytical solution is then evaluated in the exterior domain Ω_c .

Additionally, visualizations of the scattered field are provided to assess the spatial coherence for different scatterer shapes. In the two dimensional case, one may also compare the results with finite element method (FEM) solutions.

6.2.2 Case $\dim(\Omega) = 3$

For 3D, we also compare the estimated scattered pressure from our network with the analytical scattered pressure for a sphere scatterer of radius a with Neumann condition [15] given by, in spherical coordinates:

$$P_s(r, \psi, \phi) = \sum_{n \in \mathbb{N}} \sum_{m=-n}^n 4\pi j_n^n \frac{j'_n(ka)}{h_n^{(1)\prime}(ka)} Y_n^m(\psi, \phi) Y_n^m(\psi_d, \phi_d) \quad (6.2)$$

where:

- j_n and $h_n^{(1)}$ are the spherical Bessel and spherical Hankel functions of the first kind, respectively, of order n ,

$$j_n(z) = \sqrt{\frac{\pi}{2z}} J_{n+\frac{1}{2}}(z), \quad h_n^{(1)}(z) = \sqrt{\frac{\pi}{2z}} H_{n+\frac{1}{2}}^{(1)}(z).$$

- j'_n and $h_n^{(1)\prime}$ denote their derivatives with respect to the argument.
- $Y_n^m(\psi, \phi)$ are the spherical harmonics.
- (ψ_d, ϕ_d) define the angle of arrival of the incident wave in spherical coordinates, $(e.g. \psi_d = \arccos\left(\frac{d_z}{\|d\|}\right), \phi_d = \arctan 2(d_y, d_x))$.

This series converges rapidly for $r > a$, so in practice it is truncated to a finite number of terms N , depending on the desired accuracy.

In the three dimensional case, one may also compare the results with FEM or BEM solutions.

6.2.3 Evaluation metrics

We evaluate difference between two different pressure field with the normalized mean square error (NMSE) and the normalized cross correlation [10].

The NMSE and NCC between two pressure fields P_s and \hat{P}_s over a grid \mathcal{G} are defined by:

$$\text{NMSE}(P_s, \hat{P}_s) = \frac{\|P_s - \hat{P}_s\|_{\mathcal{G}}^2}{\|P_s\|_{\mathcal{G}}^2} \quad (6.3)$$

$$\text{NCC}(P_s, \hat{P}_s) = \frac{\sum_{\mathbf{x} \in \mathcal{G}} \hat{P}_s \cdot \bar{P}_s}{\|P_s\|_{\mathcal{G}} \| \hat{P}_s \|_{\mathcal{G}}} \quad (6.4)$$

where $\|\cdot\|_{\mathcal{G}} : A \in \ell^2(\mathcal{G}) \mapsto \sum_{\mathbf{x} \in \mathcal{G}} |A(\mathbf{x})|^2$.

For readability, we also report the *cosine distance*, defined in terms of the NCC as:

$$\text{CosDist}(P_s, \hat{P}_s) = 1 - \text{NCC}(P_s, \hat{P}_s). \quad (6.5)$$

The NMSE quantifies how well the network approximates the true solution relative to a trivial zero solution, which corresponds to $\text{NMSE} = 1$. This normalization reduces the influence of the absolute amplitude of the solution, which can otherwise lead to large MSE values even for accurate approximations. Results are typically reported in decibels (dB) using the transformation $A \mapsto 10 \log_{10}(A)$, where an NMSE of 0 dB corresponds to a null approximation.

The cosine distance takes values in $[0, 2]$ and measures the similarity between two fields, with values close to 0 indicating high similarity.

6.3 Scattering in a two dimensional space

6.3.1 Performance of the reference PINN

In this section, we will show and discuss the results of the reference PINN. The input direction if not specified otherwise is $\mathbf{d} = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$.

In fact the input direction is not so important as our network hold similar results for every direction. The architecture if not specified otherwise consists in 3 hidden layers with a hidden dimension of 64.

Reference PINN activation function comparison

To analyze the estimated solution in detail, we report NMSE and CosDist separately for the real and imaginary parts rather than directly on the complex field. For each model, we use 25,000 Adam steps followed by 250 L-BFGS steps (as for the experiment shown below).

For networks operating in polar coordinates, the output is decomposed into magnitude and phase. The internal architecture predicts three features (g_1, g_2, g_3) such that the network output is [7]:

$$\Phi_\theta = \exp(g_1) \exp(j \arctan 2(g_2, g_3)). \quad (6.6)$$

However, polar networks struggle to learn boundary conditions, which prevents convergence during the L-BFGS training phase.

The *Double Sine* network refers to a model composed of two independent SIRENs: one for the real part and one for the imaginary part of the estimated solution.

For tanh-based networks, we use 256 Random Fourier Features as input transformation. The radius of the circular scatterer is set to $a = 0.2$.

Architecture	NMSE (Real)	NMSE (Imag)	Cosine Dist. (Real)	Cosine Dist. (Imag)	Training Time (s)
QuadraticTanh	-34.9	-32.8	1×10^{-4}	2×10^{-4}	1264
QuadraticTanh (with 500 L-BFGS epochs)	-42.1	-41.7	2×10^{-5}	3×10^{-5}	2056
Tanh	-9.2	-9.4	5×10^{-2}	5×10^{-2}	1242
Sine	-48.1	-44.6	6×10^{-6}	1×10^{-5}	688
Double Sine	-57.8	-54.8	6×10^{-7}	1×10^{-6}	1068

Table 6.1: Comparison of the performance field estimation for circular scatterer at $f = 2\text{kHz}$

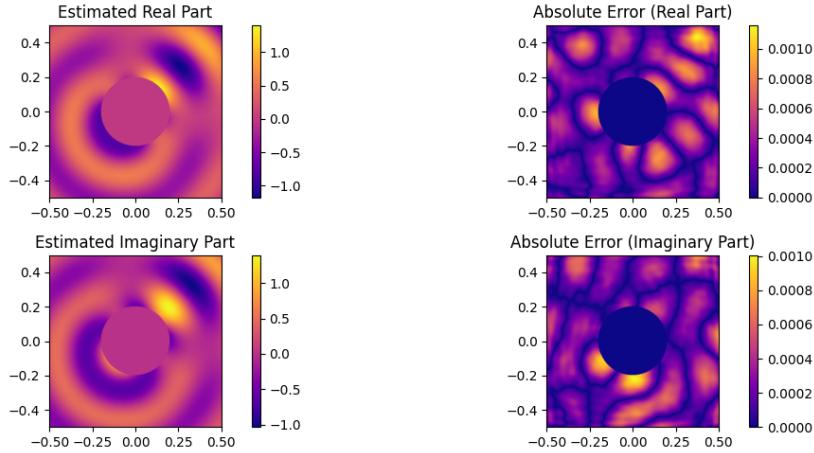


Figure 6.1: Estimated pressure field by the reference network at $f = 1\text{kHz}$

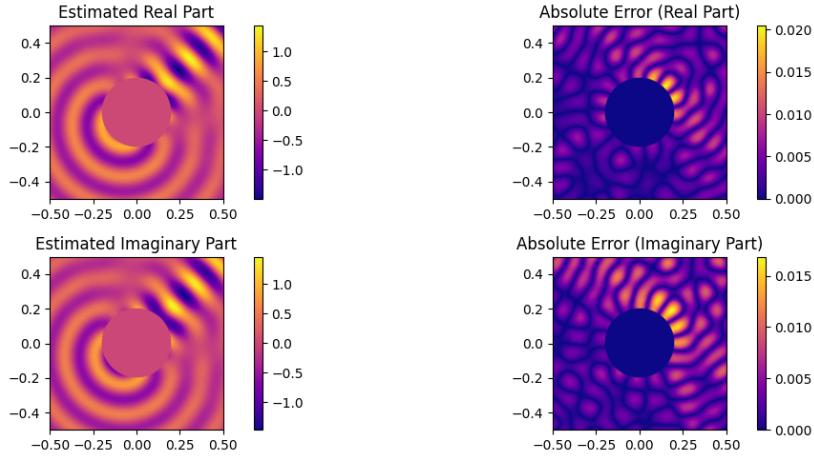


Figure 6.2: Estimated pressure field by the reference network at $f = 2\text{kHz}$

Number of Layers

We do not provide detailed numerical results regarding the effect of the number of layers; however, our experiments indicate that 3 or 4 hidden layers provide a good balance between training time and performance. The improvement from 3 to 4 layers is generally minor, even for higher-frequency problems that require more capacity to capture complex behavior [27].

A more critical factor is the number of hidden neurons, which plays a key role in accurately representing high-frequency features and is especially important when extending the network to 3D problems.

Comments

As observed, the SIREN network performs better and converges faster than the quadratic activation in 2D. However, this faster convergence comes at the cost of training instability and poorer results in 3D, as we will see in the next section. For this reason, the QuadraticTanh activation is chosen as our default function, even though it requires longer training to achieve comparable performance metrics.

As a general guideline, for problems involving scatterers with irregular structures, we recommend using QuadraticTanh for stability, whereas SIREN is more efficient and accurate for simpler scatterers. Increasing the L-BFGS batch size can help capture higher-frequency behavior, but this approach can be computationally intensive.

For example, when using SIREN at a frequency of $f = 4500\text{Hz}$, a batch size of 128,000 is required to ensure convergence. Here, the reported performance indicates * as the multiple of the original batch size of 32,784. The number of Adam steps is slightly adjusted depending on the frequency to optimize training.

Frequency	NMSE (Real)	NMSE (Imag)	Cosine Dist. (Real)	Cosine Dist. (Imag)	Training Time (s)
$k = 30/ f = 1636 \text{ Hz}$	-56.1	-56.7	3×10^{-7}	4×10^{-7}	688
$k = 50/ f = 2728 \text{ Hz (2*)}$	-34.7	-34.7	6×10^{-5}	7×10^{-5}	1028
$k = 60/ f= 3274 \text{ Hz (4*)}$	-28.4	-29.2	5×10^{-4}	4×10^{-4}	1985
$k = 80/ f = 4378 \text{ Hz (4*)}$	-17.8	-17.4	4×10^{-3}	3×10^{-3}	2309

Table 6.2: Evolution of the performance for a SIREN architecture regarding frequency.

6.3.2 LoRA performances

The performance drop introduced by LoRA is significant (around -15 dB); however, it enables fast adaptation and interpolation. Since the accuracy of our base model is very high at low frequencies, LoRA can still be effectively used for adaptation.

A more guided LoRA approach—such as incorporating physical structure into the decomposition, similar to Non-Negative Matrix Factorization [13]—could help improve performance at higher frequencies.

Training with LoRA is approximately 3 to 4 times faster than full training. Adapting LoRA on models trained for a specific direction slightly improves performance, but the gain is minor (less than 1 dB).

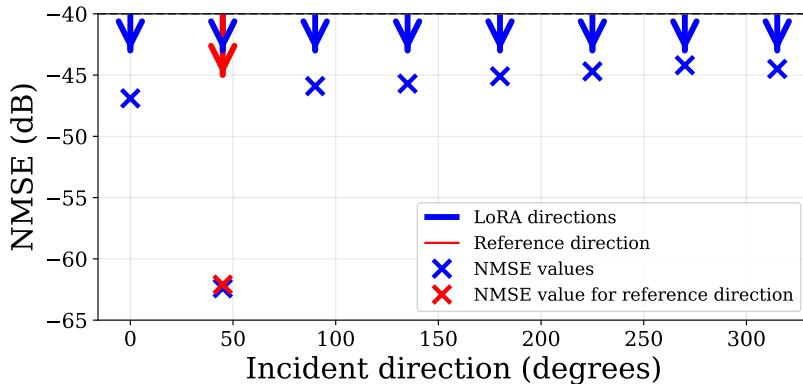


Figure 6.3: LoRA adaptation NMSE for directions denoted by blue arrow at $f = 1\text{kHz}$

6.3.3 PHISK

LoRA Interpolation

Since the training of the network is quite long, we have not perform an optimal study for the ideal parameters to choose (J, T) . We have performed the study regarding those parameters when $\dim(\Omega)$ to facilitate the comprehension as well as the visualisation. Those parameters are highly related, a high number of LoRA adapted directions (d_j) increase the interpolation accuracy globally for sharp values of T so that the selection of LoRA for interpolation is highly correlated with neighbouring directions, ignoring far directions interpolated LoRA.

Even though this approach allows to have good interpolation on average without training PHISK (.e.g., setting correction to zero), it then after hinder the training by having a not smooth enough interpolation. The tradeoff is made by choosing to have a good performance for close to every direction (.e.g., average temperature T) or keeping an excellent performance around LoRA directions and having poor results between interpolated directions (.e.g., low temperature T).

Temperature T	NMSE (Complex)	Cosine Dist. (Complex)
0.5 (with $J = 4$)	-3.5 ± 2.23	0.42 ± 0.22
0.5	0.55 ± 4.33	0.82 ± 0.41
0.25	-0.66 ± 4.33	0.64 ± 0.32
0.1	-4.7 ± 5	0.30 ± 0.26
0.05	-9.52 ± 8.25	0.18 ± 0.23
0.01	-10.65 ± 8.6	0.11 ± 0.13
0.001	-10.7 ± 8.5	0.10 ± 0.08

Table 6.3: Comparison of raw LoRA Interpolation with $J = 8$ (except first row) and $f = 1\text{kHz}$

As shown in Table 6.3, the temperature parameter is essential for accurate interpolation. A temperature of $T = 0$ corresponds to selecting only the closest direction, with weight $\alpha_j = 1$.

With 8 LoRA-adapted directions, selecting the closest direction performs reasonably well on average, although it increases the standard deviation. However, this approach fails when fewer directions are used, creating significant imbalance in NMSE across directions: some directions near trained LoRA points achieve very low NMSE, while others far from these points exhibit very high NMSE.

A scheme that appears to balance accuracy and LoRA performance is $T = 2/J$. While better schemes may exist, this choice provides a relatively low standard deviation while ensuring sufficient directional interpolation for PHISK to correct residual errors.

Since LoRA training is fast, increasing the number of adapted directions J is generally beneficial, as it improves interpolation quality. The additional computational cost during PHISK training is negligible because gradients are not backpropagated through the LoRA weights.

Results of Hypernetwork correction

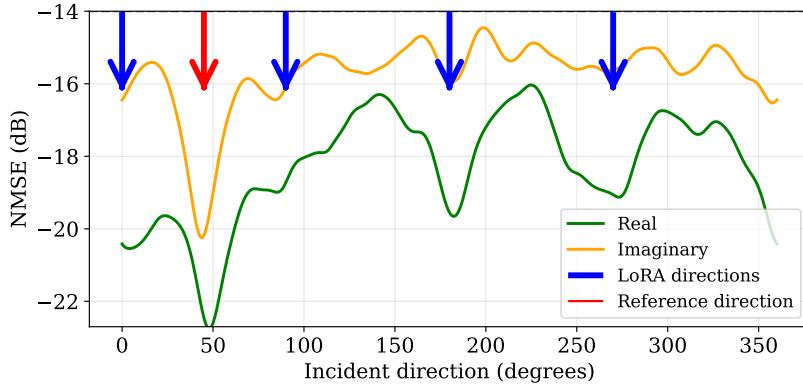


Figure 6.4: Hypernetwork results with $J = 4$ and $T = 0.5$ at $f = 1\text{kHz}$

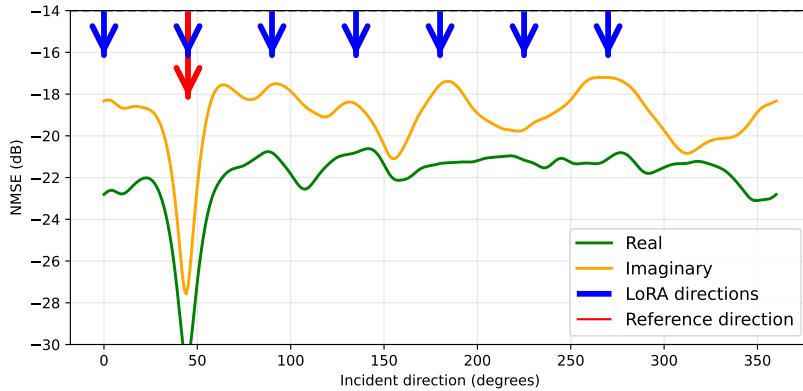


Figure 6.5: Hypernetwork results with $J = 8$ and $T = 0.5$ at $f = 1\text{kHz}$

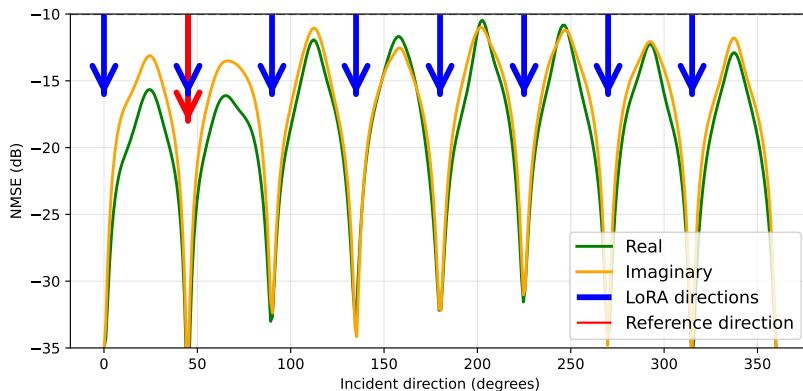


Figure 6.6: Hypernetwork results with $J = 8$ and $T = 0.05$ at $f = 1\text{kHz}$

We observe that, despite suboptimal LoRA interpolation due to a high temperature (Figure 6.5), the network is able to extract significant information from this representation. Conversely, a very sharp temperature (Figure 6.6) makes correcting the initial interpolation much more difficult.

Network training is stopped after 100,000 Adam steps, corresponding to approximately 18 hours of training. However, reasonable performance (around -12 dB average NMSE) can already be achieved after only 4–5 hours.

Comparison with Conditioned PINN

We compare these results with a “naive” conditioned PINN approach at $f = 1\text{kHz}$, consisting of a network with the same architecture as the reference PINN, but with doubled input dimensions (coordinates plus direction). To train this conditioned PINN, we use a two-phase training scheme to aid convergence.

In the first phase, we fix the directional information to train the network as a reference PINN, providing a strong initialization for the subsequent phase, where the direction varies. The conditioned PINN requires roughly half the training time of PHISK, but its performance is lower on average (NMSE: -11.9 dB, Cosine Dist.: 3×10^{-2}). By contrast, PHISK achieves significantly better performance: for 4 LoRA directions, (NMSE: -16.8 dB, Cosine Dist.: 8×10^{-3}) and for 8 LoRA directions, (NMSE: -20.2 dB, Cosine Dist.: 2×10^{-3}), demonstrating the efficiency of our approach.

6.3.4 General Results

We provide only visual evaluation for other scatterer shapes, as explained previously. Nonetheless, the network produces coherent scattered fields even with multiple scatterers, in scenarios where the baseline b-PINN described in [27] struggles to perform accurately (see Figure 6.7).

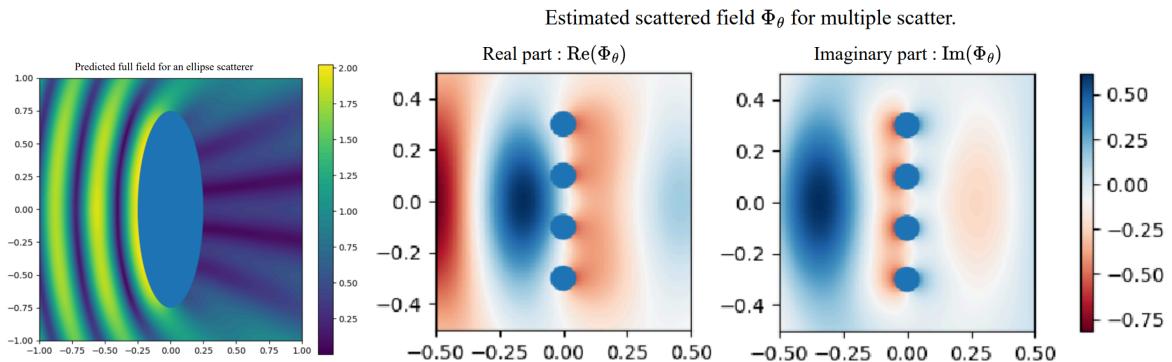


Figure 6.7: Illustration of estimated field for more complex shape than a regular circle

We can also look at the derivative of the estimated pressure to see the coherence of the produced field in \mathbb{H}_1 .

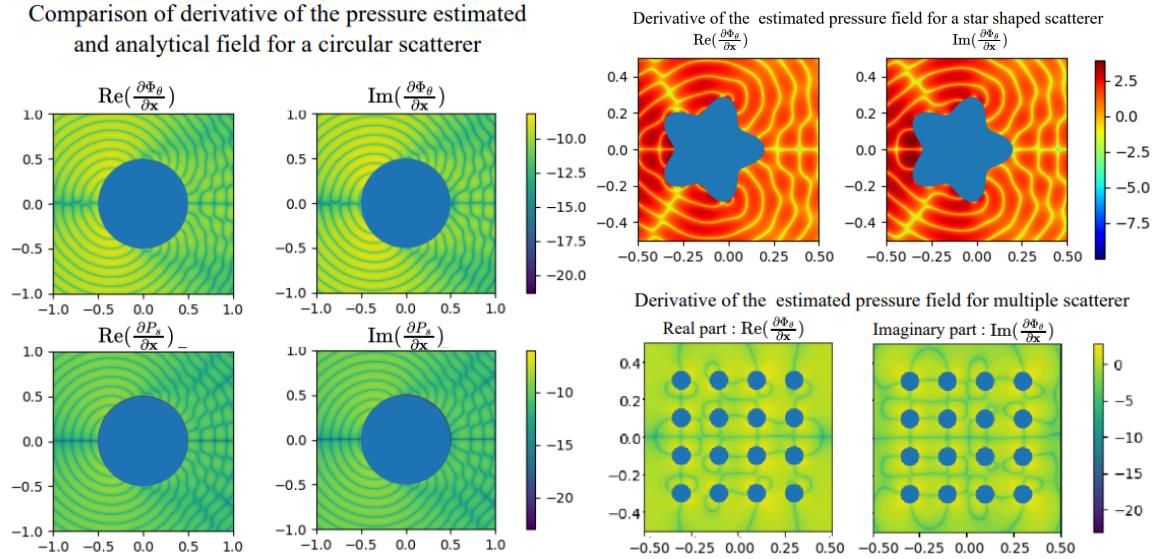


Figure 6.8: Illustration of different estimate of the pressure field x-derivative

We can even see that the network learn symmetry even for complex scenario with multiple scatter Figure 6.8, confirming the reliability of our architecture.

6.4 Scattering in a three dimensional space

6.4.1 Scattering around a sphere

Scattering on a 3D sphere highlights the potential of our QuadraticTanh approach, which allows scaling the hidden dimension of the network up to 512 while incurring minimal performance loss compared to 2D.

However, training time increases significantly with this larger network. We do not report results for the SIREN architecture with a hidden dimension of 512 in Table 6.4, as such an architecture diverges even during the Adam phase.

Moreover, even with smaller architectures, the QuadraticTanh model continues to achieve excellent results.

Architecture	NMSE (Real)	NMSE (Imag)	Cosine Dist. (Real)	Cosine Dist. (Imag)	Training Time (min)
QuadraticTanh	-52.2	-52.0	1×10^{-6}	1×10^{-6}	171
QuadraticTanh (hidden dimension 64)	-30.7	-30.2	7×10^{-4}	7×10^{-4}	57.1
Tanh	-35.8	-33.4	1×10^{-4}	2×10^{-4}	158
Sine (hidden dimension 64)	-29.7	-28.6	3×10^{-4}	6×10^{-4}	30.2

Table 6.4: Comparison of the performance for sphere scatter field estimation at $f = 1\text{kHz}$ (a QuadraticTanh 2D approach for the same frequency produce a **NMSE** of -62.1 dB and a Cosine Distance of 3×10^{-7} in complex domain)

6.4.2 LoRA performance

The performance drop for LoRA in 3D is around 15–20 dB, which is slightly higher than in 2D (10–15 dB), despite the LoRA dimension being much smaller relative to the network’s hidden dimension in 3D.

For example, LoRA adds approximately 20% more weights in 2D, whereas in 3D it adds only about 3%. As a result, LoRA training is even faster compared to the original full model.

A phenomenon observed in 2D, and even more pronounced in 3D, is that LoRA directions far from the initial trained direction tend to be adapted less accurately, with performance drops around 20 dB.

Frequency	Reference NMSE	Reference Cosine Dist.	LoRA adapted NMSE	LoRA adapted Cosine Dist.	Training Time (min)
$f = 1\text{kHz}$	−50.1	4×10^{-6}	−28.1	4×10^{-6}	48
$f = 2\text{kHz}$	−39.2	6×10^{-4}	−17.2	3×10^{-2}	68

Table 6.5: Average performance of LoRA on a 3D scatter

6.4.3 PHISK

For 3D, we present results with $J = 6$, where the corresponding directions are the canonical basis of \mathbb{R}^3 and their opposites.

Fibonacci sampling can be used to generate well-spaced directions for different values of J . The interpolation parameter is set to $T = 0.5$.

Training does not require significantly more time despite the larger number of parameters, as the network is conditioned in a similar LoRA fashion.

Improved results could likely be achieved by increasing the number of LoRA directions to 12, or even 24. However, since LoRA training time becomes more substantial in 3D, such changes must be carefully considered if overall training time is a constraint.

Architecture	NMSE	Cosine Dist.	Training Time (min)
LoRA Interpolation	−4.3	8×10^{-1}	N/A
Phisk	−11.7	1×10^{-2}	568

Table 6.6: Results of PHISK components in estimating scattering field for a sphere at $f = 1\text{kHz}$.

6.5 Head-Related Transfer Estimation

6.5.1 Scaling to real condition

As our network takes as input normalized (between $-1 - L_{\text{pml}}, 1 + L_{\text{pml}}$) coordinates, a coordinate change is done in the preprocessing as $\mathbf{x} \rightarrow \alpha\mathbf{x} = \tilde{\mathbf{x}}$ corresponding to the

partial derivative update $\frac{\partial}{\partial x_i} \rightarrow \alpha \frac{\partial}{\partial x_i} = \frac{\partial}{\partial \tilde{x}_i}$ (*e.g.* $\nabla_{\mathbf{x}} = \alpha \nabla_{\tilde{\mathbf{x}}}$).

Furthermore, since s_{x_n} is independant of the scale representation (both value in the ratio are scaled), $S_{\mathbf{x}} = \tilde{S}_{\tilde{\mathbf{x}}}, R_{\mathbf{x}} = \tilde{R}_{\tilde{\mathbf{x}}}$ where \tilde{S}, \tilde{R} are just incorporating scaled value of the coordinates, we can deduce by replacing the previous formula in (5.2) and using that $P_s(\mathbf{x}) \approx \Phi_{\theta}(\tilde{\mathbf{x}})$, we get:

In fact due to our only interest in the scattering field P_s , we can reformulate the previous (5.1) in function of P_i as :

$$\begin{cases} \nabla_{\tilde{\mathbf{x}}}[\tilde{R}_{\tilde{\mathbf{x}}}\tilde{S}_{\tilde{\mathbf{x}}}^{-1}\nabla_{\tilde{\mathbf{x}}}\Phi_{\theta}](\tilde{\mathbf{x}}) + (\alpha k)^2|\tilde{S}_{\tilde{\mathbf{x}}}| \Phi_{\theta}(\tilde{\mathbf{x}}) = 0 \\ \alpha \frac{\partial \Phi_{\theta}}{\partial \mathbf{n}(\tilde{\mathbf{x}})}(\tilde{\mathbf{x}}) + \frac{j\omega\rho_0}{Z} \Phi_{\theta}(\tilde{\mathbf{x}}) = -\frac{\partial P_i}{\partial \mathbf{n}(\mathbf{x})}(\mathbf{x}) - \frac{j\omega\rho_0}{Z} P_i(\mathbf{x}) \end{cases} \quad (6.7)$$

Understanding the real-world scaling, the only factor we can control in the predicted field is how much space the scatter object occupies within Ω .

We choose this to be 0.75 times the maximum dimension of the scatter along one of the Cartesian directions (x, y, or z).

6.5.2 Complexity of the Ear Shape

Unfortunately, we face challenges in capturing the scattered field around the ear. In this particular area, the network tends to produce very high values or strong variations. This is due to the complex geometry of the ear, which can be described using Bézier curves [29], making it difficult for the network to accurately learn the rapidly varying contours.

To illustrate the influence of the ear on the predicted scattered field, we provide a visual comparison of the estimated pressure with a full head versus a head with the ears manually reduced using Blender [3]. As observed in Figure 6.10, the variations in the field are much higher when the ear is present compared to the simplified head.

Further analysis of this effect will be presented in Section 6.5.4.

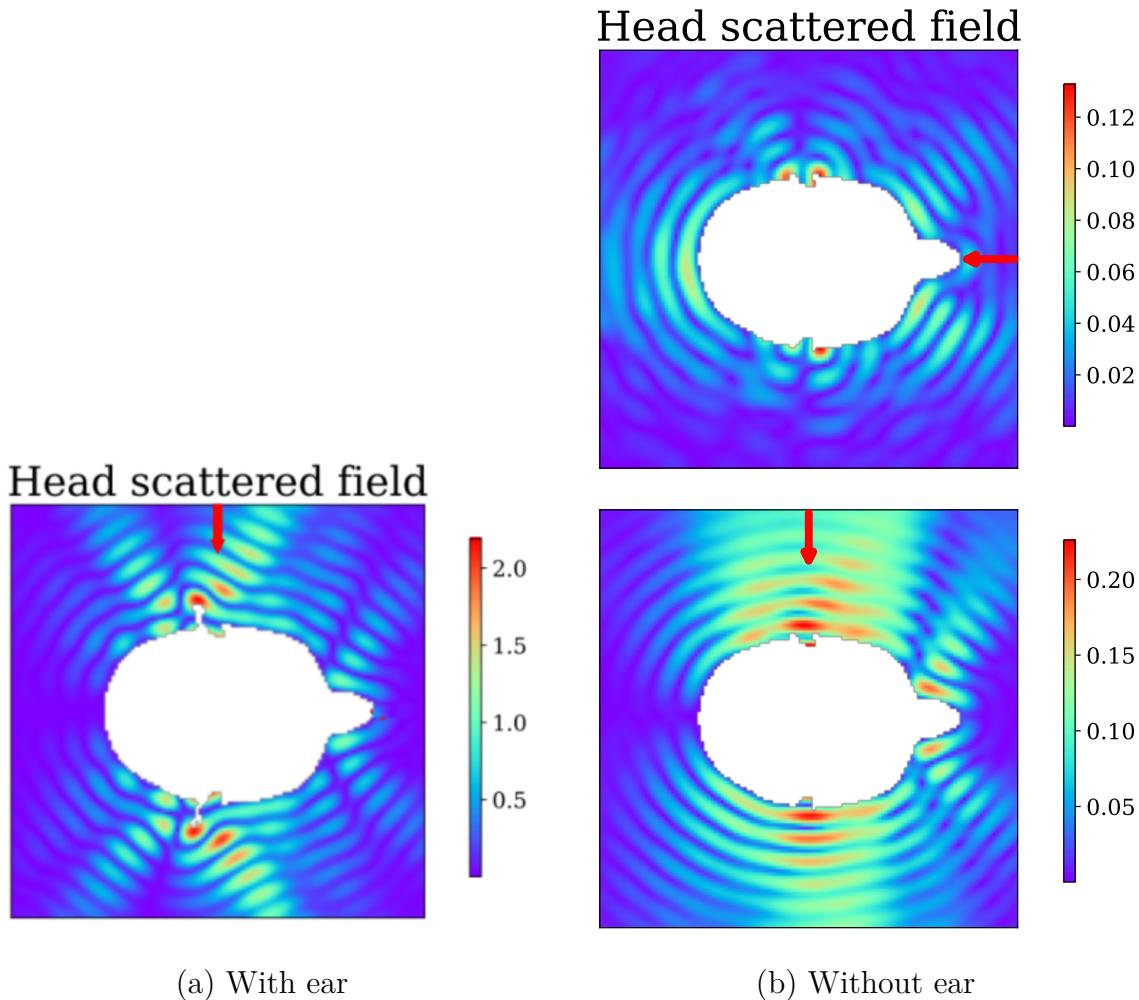


Figure 6.9: Produced field comparison for a head with and without ear, for direction $\mathbf{d} = (0, 1, 0)$.

6.5.3 Evaluation of HRTF

We also have two ways of evaluating the total pressure field once the network is trained, one of which involves using the boundary integral equation. This approach is also useful for verifying the validity of the estimated field by examining the residuals of the equation.

The field should satisfy the following equation for $\mathbf{x} \in \partial\Omega$:

$$P(\mathbf{x}) = 2P_i(\mathbf{x}) + 2 \oint_{\partial\Omega} d\mathbf{y} \left(P(\mathbf{y}) \frac{\partial}{\partial n(\mathbf{y})} G(\mathbf{x}, \mathbf{y}) - G(\mathbf{x}, \mathbf{y}) \frac{\partial P}{\partial n(\mathbf{y})}(\mathbf{y}) \right) \quad (6.8)$$

Equation (6.8) can be used to compute $P(\mathbf{x})$ at the ear points, providing an alternative estimate of the HRTF by leveraging multiple network evaluations.

The main challenge of this approach is handling potential singularities in the Green's function G . As a result, the more straightforward approach using the formula in Eq. (1.1) from the first section is often preferred.

6.5.4 Coherence results

Unfortunately, our network is not able to perfectly learn the scattered field around the full head shape, despite converging during training. The main challenge lies in the rapidly varying boundaries of certain regions of the head, particularly the ears. To better understand this effect, we also compare results for a head without ears (manually removed using Blender) to isolate the influence of the ears.

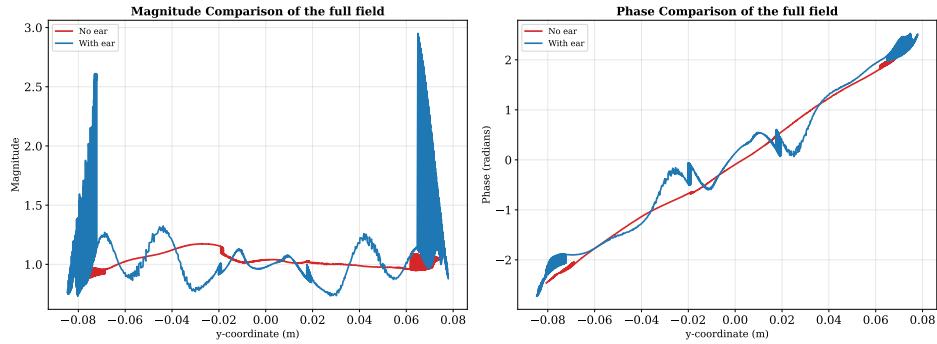


Figure 6.10: Comparison of a scattered field estimation with ear and without ear for $\mathbf{d} = (0, 1, 0)$

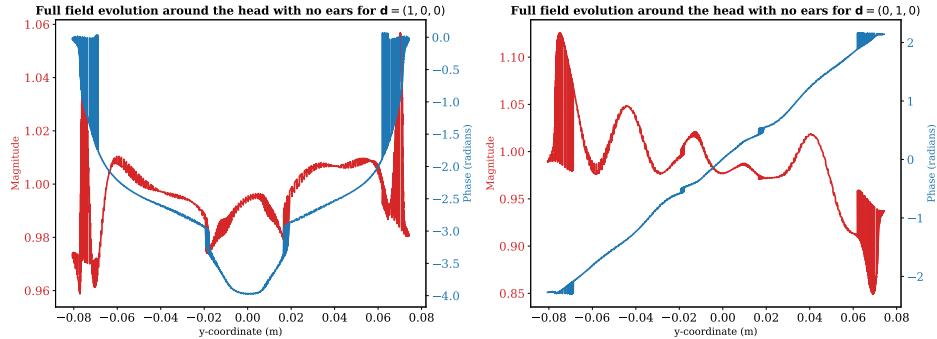


Figure 6.11: Comparison of network field around the head for $z = 0$

For the head with ears, the network produces extremely fast-varying fields around the ear region, making predictions unreliable there. In contrast, for the head without ears, the network output is much more coherent. For example, for an incident wave $\mathbf{d} = (1, 0, 0)$ (propagating toward the nose), the resulting field is nearly symmetric, consistent with the symmetry of the face.

For an incident wave coming from the left-ear direction, the network predicts a higher magnitude on the left than on the right, with a smooth linear phase variation around the head. Interestingly, this phase trend is also present for the full-head case, indicating that the network captures an overall coherent field even when local regions, such as the ears, are challenging to model accurately.

PHISK Results

Unfortunately, due to the high computational cost and the challenges associated with accurately modeling the ear shape, we do not provide any PHISK estimations for the head in this study. We hope that future research will address these limitations and enable reliable PHISK-based HRTF predictions for complex head geometries.

Chapter 7

Conclusion

7.1 Overview of the study

In this study, we explored the application of Physics-Informed Neural Networks (PINNs) for scattering field estimation using a purely physics-driven approach. We proposed a perfectly matched layer (PML) method to handle the Sommerfeld radiation condition, coupled with flexible Robin boundary conditions. This approach provided a solid mathematical foundation for the scattering estimation problem, enabling the neural network to learn the field solely from physical principles.

We discussed the architecture and training framework for such neural networks, emphasizing the importance of each design choice in acoustic scattering field estimation. Key factors include coordinate embeddings, sampling methods, activation functions, and training strategies, providing practical guidelines for similar problems in both 2D and 3D. Our experiments highlighted the effectiveness of the QuadraticTanh activation function and Halton sampling for accurately representing scatterer boundaries, even when the number of sampled points is less than 1% of the mesh vertices (as in head meshes).

To generalize scattering estimation across multiple directions—a requirement for Head-Related Transfer Function (HRTF) computation—we proposed the Hypernetwork architecture PHISK. PHISK leverages the empirical observation that interpolating the weights of LoRA-adapted networks provides a reasonable initial estimate of the scattered field for a set of adapted directions. It then refines this interpolation using a continuous directional embedding to generate corrective weights, allowing the network to produce accurate scattered fields for any given direction. Our results demonstrate that PHISK outperforms a naive approach that attempts to learn the scattered field simultaneously in both spatial and directional domains.

Overall, although ground truth HRTF results were not obtained, to the best of our knowledge, our work advances the state of the art in PINN-based scattering estimation

for simple shapes, achieving high accuracy without relying on additional data such as pressure measurements.

7.2 Future Directions

The proposed study offers many paths to explore in order to validate, enhance, and generalize the framework presented above.

First, due to the lack of sufficient open-source and simple-to-use FEM or BEM solvers, results on shapes other than simple circles have not been evaluated quantitatively, only visually. Although the coherence of the results seems reasonable, a comparison with traditional FEM or BEM could definitively validate our approach. In addition, despite providing a general framework that also works with Robin boundary conditions, as the scattered field seems to demonstrate visually, a comparison with traditional FEM would certify the ability of the proposed architecture to capture more complex boundary conditions than Neumann ones, including absorption inside the scatterer.

As we used mesh-based representations in this work, another idea would be to use a neural representation of the mesh beforehand, in order to capture the fully continuous structure of the studied geometry. We did not investigate this approach in this study, as using a mesh representing a circle or the equation of a circle to sample points produced results with insignificant differences but this might not be the case for more complex geometries.

The reference PINN defined in this study also struggles to capture higher frequencies when L_f becomes larger, with batch size increase being the only attempted solution, even though the QuadraticTanh partially solves this issue. Further studies may look at the impact of network size, which helps handle higher frequencies, as explained in Section 6, as a function of the frequency. Additionally, the impact of the α parameter inside the QuadraticTanh or the ω parameter for SIREN may play a crucial role in handling higher frequencies. In this study, α was fixed beforehand, and experiments showed that a learnable α only oscillated around its initial value for our problem. However, expert-informed α could drastically help the network learn the scattering field, potentially simplifying the governing equation. For example, using a single SIREN layer with $\omega = k$ could simplify the frequency dependency in the wave equation, but such studies are left for future exploration.

Regarding the proposed Hypernetwork, its general principle allows for many upgrades, even with adaptation in the frequency domain (in addition to the directional domain). First, the temperature dependence T comes from a compromise between sharpness and average performance. A learnable T , or one based on external knowledge, may help the network find the optimal balance for interpolation, with the main

challenge being maintaining stable optimization with a varying T . On the other hand, the proposed interpolation method may not be the most suitable, and alternative approaches modifying the weight forms could be explored. Attention-based weights were tried, showing slightly worse results than fixed-weight interpolation. The main challenge with attention weights was maintaining stable optimization, as the network could diverge if attention weights changed too much during a single optimization step. A potential solution could involve using two separate optimizers: one for attention weights with stricter control and another for corrective network weights. Furthermore, attention-based interpolation sometimes loses the directional information provided by LoRA in its respective directions (similar to the effect of too high a temperature), which is why attention was not used in the final architecture. Constraints-based attention or kernel-based interpolation of LoRA could be promising directions for future work. In addition, physical structure could potentially be encoded into one of the LoRA matrices, in a Non-Negative Matrix Factorization fashion, furthermore increasing explainability of the architecture.

For HRTF estimation, we invite researchers to explore using PHISK for HoloLens applications and to perform an evaluation of the acoustical cues using the HRTF estimated by this model. Additionally, although incorporating the torso was straightforward, it was omitted in our experiments for faster training and evaluation. Small experiments show that including the torso with only a watertight mesh (not fully processed as in Mesh2HRTF) may still yield coherent results.

Finally, although it was not the original purpose of this study, our network handles multiple scatterers, which suggests that an evaluation of produced acoustic scenes in 3D for outdoor acoustics could be a promising application.

Code and Data Availability

The source code implementing the methods described in this thesis is available at:

<https://github.com/Dankradino/Acoustic-Scattering-PINN>

Bibliography

- [1] V. Ralph Algazi, Richard O. Duda, Ramani Duraiswami, and Nail A. Gumerov. Approximating the head-related transfer function using simple geometric models of the head and torso. *Journal of the Acoustical Society of America*, 112(5):2053–2064, 2002.
- [2] Jean-Pierre Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of Computational Physics*, 114(2):185–200, 1994.
- [3] Blender Online Community. Blender - a 3d modelling and rendering package. *Blender Foundation, Amsterdam, The Netherlands*, 2025.
- [4] Fabian Brinkmann, Manoj Dinakaran, Robert Pelzer, Jan Joschka Wohlgemuth, Fabian Seipel, Daniel Voss, Peter Grosche, and Stefan Weinzierl. The hutubs head-related transfer function (hrtf) database. *DepositOnce*, 2019.
- [5] Fabian Brinkmann, Wolfgang Kreuzer, Jeffrey Thomsen, Sergejs Dombrovskis, Katharina Pollack, Stefan Weinzierl, and Piotr Majdak. Recent advances in an open software for numerical HRTF calculation. *J. Audio Eng. Soc*, 71(7/8):502–514, 2023.
- [6] Valeria Bruschi, Loris Grossi, Nefeli A. Dourou, Andrea Quattrini, Alberto Vancheri, Tiziano Leidi, and Stefania Cecchi. A review on head-related transfer function generation for spatial audio. *Applied Sciences*, 14(23):11242, 2024.
- [7] Diego Di Carlo, Aditya Arie Nugraha, Mathieu Fontaine, and Kazuyoshi Yoshii. Neural steerer: Novel steering vector synthesis with a causal neural field over frequency and source positions. *arXiv preprint arXiv:2305.04447*, 2024.
- [8] You Chang, Namkeun Kim, and Stefan Stenfelt. Simulation of bone-conducted sound transmission in a three-dimensional finite-element model of a human skull. In *AIP Conference Proceedings*, volume 1703, page 060014. AIP Publishing, 2015.
- [9] Zhongying Chen, Dongsheng Cheng, Wei Feng, and Tingting Wu. An optimal 9-point finite difference scheme for the helmholtz equation with pml. *International Journal of Numerical Analysis and Modeling*, 10:389–410, 01 2013.

- [10] Stefano Damiano and Toon van Waterschoot. Sound field reconstruction using physics-informed boundary integral networks. In *2025 European Signal Process. Conf.*, pages 26–30. IEEE, 2025.
- [11] Kenneth Duru and Gunilla Kreiss. The perfectly matched layer (pml) for hyperbolic wave propagation problems: A review. *arXiv preprint arXiv:2201.03733*, 2022.
- [12] Isaac Engel, Rapolas Daugintis, Thibault Vicente, Aidan Hogg, Johan Pauwels, Arnaud Tournier, and Lorenzo Picinali. The SONICOM HRTF dataset. *Journal of the Audio Engineering Society*, 71:241–253, 2023.
- [13] Rémi Flamary, Cédric Févotte, Nicolas Courty, and Valentin Emiya. Optimal spectral transportation with application to music transcription. *arXiv preprint arXiv:1609.09799*, 2016.
- [14] Rui Gao and Rajeev K. Jaiman. H-siren: Improving implicit neural representations with hyperbolic periodic functions. *arXiv preprint arXiv:2410.04716*, 2024.
- [15] Oleg A. Godin. Rayleigh scattering of a spherical sound wave. *The Journal of the Acoustical Society of America*, 133(2):709–720, 01 2013.
- [16] Jia Guo, Haifeng Wang, and Chenping Hou. A novel adaptive causal sampling method for physics-informed neural networks. *arXiv preprint arXiv:2210.12914*, 2022.
- [17] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.
- [18] Frank Ihlenburg. *Finite Element Analysis of Acoustic Scattering*, volume 132 of *Applied Mathematical Sciences*. Springer New York, NY, 1998.
- [19] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2017.
- [20] Elham Kiyani, Khemraj Shukla, Jorge F. Urbán, Jérôme Darbon, and George Em Karniadakis. Which optimizer works best for physics-informed neural networks and kolmogorov-arnold networks? *arXiv preprint arXiv:2501.16371*, 2025.
- [21] Wolfgang Kreuzer, Katharina Pollack, Fabian Brinkmann, and Piotr Majdak. Numcalc: An open-source bem code for solving acoustic scattering problems. *Engineering Analysis with Boundary Elements*, 161:157–178, 2024.

- [22] Dong C. Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45(1):503–528, 1989.
- [23] Fei Ma, Thushara D Abhayapala, Prasanga N Samarasinghe, and Xingyu Chen. Spatial upsampling of head-related transfer functions using a physics-informed neural network. *arXiv preprint arXiv:2307.14650*, 2023.
- [24] STEFFEN MARBURG. Six boundary elements per wavelength: Is that enough? *Journal of Computational Acoustics*, 10(01):25–51, 2002.
- [25] George Marsaglia. Choosing a Point from the Surface of a Sphere. *The Annals of Mathematical Statistics*, 43(2):645–646, 1972.
- [26] James C. Middlebrooks and Dennis M. Green. Kemar: A manikin for acoustic research. *The Journal of the Acoustical Society of America*, 52(5B):1456–1466, 1972.
- [27] Siddharth Nair, Timothy F. Walsh, Greg Pickrell, and Fabio Semperlotti. Multiple scattering simulation via physics-informed neural networks. *Engineering with Computers*, 41(1):31–50, 2024.
- [28] Ryosuke Onizawa, Gen Sato, Izumi Tsunokuni, and Yusuke Ikeda. Physics-informed neural networks for estimation of scattered sound fields with boundary condition. In *Proc. APSIPA ASC*, pages 1–5, 2024.
- [29] Felix Perfler, Florian Pausch, Katharina Pollack, Nicki Holighaus, and Piotr Majdak. Parametric model of the human pinna based on b  ezier curves and concave deformations. *Computers in Biology and Medicine*, 188:109817, 2025.
- [30] Sebastian Prepelita. *Verification and validation of wave based simulations of head-related transfer functions*. PhD thesis, 05 2020.
- [31] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 20, pages 1177–1184, 2007.
- [32] M. Raissi, P. Perdikaris, and G.E. Karniadakis. 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, 2019.
- [33] Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G. Baraniuk. Wire: Wavelet implicit neural representations. *arXiv preprint arXiv:2301.05187*, 2023.

- [34] E. A. G. Shaw. Acoustic response of external ear with progressive wave source. *The Journal of the Acoustical Society of America*, 51:150–150, 01 1972.
- [35] Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *arXiv preprint arXiv:2006.09661*, 2020.
- [36] Chao Song, Tariq Alkhalifah, and Umair Bin Waheed. A versatile framework to solve the helmholtz equation using physics-informed neural networks. *Geophysical Journal International*, 228(3):1750–1762, 2021.
- [37] T. Walsh, L. Demkowicz, and R. Charles. Boundary element modeling of the external human auditory system. *The Journal of the Acoustical Society of America*, 115(3):1033–1043, 2004.
- [38] Sifan Wang, Shyam Sankaran, Hanwen Wang, and Paris Perdikaris. An expert’s guide to training physics-informed neural networks. *arXiv preprint arXiv:2308.08468*, 2023.
- [39] Sifan Wang, Hanwen Wang, and Paris Perdikaris. On the eigenvector bias of fourier feature networks: From regression to solving multi-scale pdes with physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 384:113938, 2021.
- [40] Sifan Wang, Xinling Yu, and Paris Perdikaris. When and why pinns fail to train: A neural tangent kernel perspective. *J. of Computational Physics*, 449:110768, 2022.
- [41] S. G. Weinrich. *Sound Field Calculations Around the Human Head*. Phd thesis, Technical University of Denmark, 1984.
- [42] Tien-Tsin Wong, Wai-Shing Luk, and Pheng-Ann Heng. Sampling with hammer-sley and halton points. *Journal of Graphics Tools*, 2(2):9–24, 1997.
- [43] Yanqi Wu, Hossein S Aghamiry, Stephane Operto, and Jianwei Ma. Wave simulation in non-smooth media by pinn with quadratic neural network and pml condition. *arXiv preprint arXiv:2208.08276*, 2022.
- [44] Zixue Xiang, Wei Peng, Xu Liu, and Wen Yao. Self-adaptive loss balanced physics-informed neural networks. *Neurocomputing*, 496:11–34, jul 2022.
- [45] H. Ziegelwanger, W. Kreuzer, and P. Majdak. Mesh2HRTF: An open-source software package for the numerical calculation of head-related transfer functions. In *Proc. ICSV*, page 42583, 2015.

- [46] H. Ziegelwanger, P. Majdak, and W. Kreuzer. Numerical calculation of head-related transfer functions and sound localization: Microphone model and mesh discretization. *The Journal of the Acoustical Society of America*, 138:208–222, 2015.