PHYSICS-INFORMED NEURAL NETWORK FOR ROBUST TIME-OF-FLIGHT EXTRACTION IN COMPLEX WAVEFIELDS: APPLICATIONS IN GUIDED-WAVE-BASED STRUCTURAL HEALTH MONITORING

SONG YANG

PhD

The Hong Kong Polytechnic University

2026

The Hong Kong Polytechnic University

Department of Mechanical Engineering

Physics-Informed Neural Network for Robust Time-of-Flight Extraction in Complex Wavefields: Applications in Guided-Wave-based Structural Health Monitoring

Song Yang

A Thesis Submitted in Partial Fulfilment of the Requirement for the Degree of Doctor of Philosophy

September 2025

Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

September 2025

Abstract

In wave-based sensing technologies, accurately extracting Time-of-Flight (ToF) is a universally challenging scientific problem, especially in fields like acoustics, electromagnetics, and elastodynamics. In complex environments, multipath interference and scattering from boundaries, obstacles, and various structural features cause severe signal superposition. This makes it difficult for traditional signal processing and purely data-driven methods to precisely isolate the target wave packet, significantly reducing the sensing system's accuracy and reliability. This thesis focuses on guided-wave technology in Structural Health Monitoring (SHM) as a key application and systematically explores a series of innovative Physics-Informed Machine Learning (PIML) frameworks to address the challenge of robust ToF extraction in complex wavefields.

The research in this thesis progresses systematically. First, to address the primary source of interference in SHM, boundary reflections, a Multi-Scale Spatio-Temporal (MSST) fusion network is introduced. Using a physics-guided architectural design, this network combines the temporal information of signals with the spatial geometry of sensors, effectively learning to remove the influence of boundary reflections on damage-scattered signals. Interpretability analyses, including deconvolution-based visualization, confirm the network's ability to effectively separate the target signal,

providing a practical solution to expand the effective inspection area of structures.

To further improve model interpretability and explicitly simulate the physical processes of wave phenomena, this dissertation introduces the Physics-informed Ray-tracing augmented Graph Neural Network (PhysRayGNN). This framework attempts to use a Graph Neural Network (GNN) to model wave propagation, attenuation, and superposition in SHM. By integrating a physics-based ray-tracing algorithm to build a high-fidelity wave-path graph and employing an attention-based GNN to simulate wave dynamics on this graph, the model's internal decision-making process aligns closely with physical laws. The learning process is also regularized by physics-informed constraints derived from the wave equation and wavelet features, ensuring the physical consistency of the predictions. Experimental results show that the framework can accurately disentangle highly superimposed signals, achieving centimeter-level damage localization precision.

Finally, to tackle the challenge of real-world engineering structures with many complex scatterers, such as stiffeners and bolts, a more comprehensive and resilient multimodal framework, WaveSenseNet, is developed. This framework features several key innovations: (1) a new parametric activation function designed to address the inherent "low-frequency bias" of typical neural networks, effectively capturing high-frequency wave packet features essential for ToF; (2) a Gaussian positional encoding map that represents complex environmental geometry and scatterer physics as spatial features, enabling an in-depth multimodal fusion of time-domain signals and spatial data; and (3) the use of Monte Carlo Dropout to provide uncertainty estimates for ToF predictions, giving the final localization outcomes probabilistic confidence. In complex experimental settings with multiple scatterers, WaveSenseNet outperforms existing top models, further confirming the benefits of a deep fusion of physics and data.

In summary, this thesis offers a systematic, interpretable, and high-precision approach to ToF extraction in complex wavefields through a series of increasingly advanced PIML frameworks. The results not only significantly improve the reliability and intelligence of guided-wave SHM technology but also introduce a new paradigm for integrating prior physical knowledge into future machine learning models, providing substantial theoretical and practical value for the broader wave-based sensing field.

Publications arising from the thesis

Journal papers:

[1] **Song, Y.**, Shan, S., Zhang, Y., & Cheng, L. (2025). *Physics-guided neural network for structural health monitoring with lamb waves through boundary reflection elimination*. Structural Health Monitoring, 14759217241305050.

[2] **Song, Y.**, Sun, P., Shan, S., Liu, Z., Li, L., & Cheng, L. *PhysRayGNN: Physics-informed ray tracing augmented graph neural network for structural health monitoring*. (manuscript under review).

[3] **Song, Y.**, Shan, S., Zhang, Y., Liu, Z., Pi, R., Li, L., & Cheng, L. *WaveSenseNet: A multimodal multi-scale attention fusion network for accurate time of flight extraction in complex environments.* (manuscript under review).

Conference presentations:

[1] **Song, Y.**, Shan, S., Zhang, Y., & Cheng, L. *Physics-informed neural network-assisted SHM with guided waves through boundary reflection elimination.* In Proceedings of the 11th European Workshop on Structural Health Monitoring (EWSHM 2024).

[2] Pi, R., **Song, Y.**, Li, L., Yu, X., & Cheng, L. (2024). *TSSL: Trusted sound source localization*. In Proceedings of the 53rd International Congress & Exposition on Noise Control Engineering (Inter-Noise 2024).

Awards:

The best student paper in 11th European Workshop on Structural Health Monitoring (EWSHM 2024)

Acknowledgements

First and foremost, I would like to express my most sincere gratitude to my supervisor, Prof. Cheng Li. I am deeply grateful for his insightful guidance and unwavering support throughout my doctoral journey. Prof. Cheng's profound knowledge and enthusiasm for research have been a constant source of inspiration, leading me through every critical stage of this thesis with wisdom and vision.

I offer my deepest gratitude to my parents. The journey from my master's to doctoral studies has been long and often taxing, and it was your unconditional love that allowed me to pursue my academic dreams. During my most difficult times, especially when research hit a bottleneck and I was filled with self-doubt, you were always there for me. You did not just offer words, but truly stood by my side and gave me the strength to keep going when I was at my lowest. I know that this thesis is as much your achievement as it is mine. For all the sacrifices you made, I will spend a lifetime repaying your kindness.

Next, I want to thank my wife, Ms. ZHU Ye. You are not only my devoted partner in life, but also my very best friend. Your understanding and companionship were what kept me grounded through countless late nights and the inevitable anxieties of research. I am grateful that I could always turn to you to share my smallest joys and my deepest frustrations, knowing that you would listen with a patience and empathy that no one else could provide. Thank you for your courage in exploring the infinite possibilities of our future together, for your willingness to move to new places, and for your commitment to pursuing what we love as a team. To walk alongside you, sharing every challenge and every dream, is the greatest fortune of my life. I love you.

My journey was also enriched by the friends and colleagues who offered their help along the way. I would particularly like to thank Dr. Shengbo Shan for his generous help and insightful discussions.

I am also grateful to the group members for their helps in study and life, including but not limited to Dr. Fuzhen Wen, Dr. Linli Zhang, Dr. Lei Xiao, Dr. Linfeng Li, Dr. Shuowei An, Dr. Xiang Sun, Dr. Ze Liu, Yuanman Zhang, Chen Gong, Sihui Li, Tingjian Li, Jinze Li, Hangxing Li, Le Chang, Zengshen Yue, Zhiyuan Li, Rendong Pi, Zhaohua Li, Yuanze Li, Yuting Yang. Additionally, I would like to extend my heartfelt thanks to my friends outside the lab, especially Dr.Shouguo Peng, Dr.Yi Yang, Dr.Zekun Li, Dr.Ziran Wang, Zhuoyuan Liu, Binglong Zhao, Run Zheng, Shichang Feng, Xin Xu and Yifei Wu, for their constant encouragement and the joy they brought to my life throughout this journey.

The completion of this doctoral thesis marks the culmination of years of dedicated effort and the beginning of a new journey. I am profoundly grateful for the experiences, both challenging and rewarding, that have shaped me as a scholar. With the support of so many, I look forward to the future with optimism and a continued passion for discovery.

Nomenclature

a, b, c	Learnable parameters in Parametric Activation Function (PAF)
$\mathbf{A}_c, \mathbf{A}_s$	Channel attention and Spatial attention maps in CBAM
α	Attention weight / Fusion weight / Learning rate
$\alpha(t)$	Dynamic balancing weight for loss function
c	Wave speed
c_g	Group velocity of a wave packet
c_p, c_s	Pressure (P-wave) and Shear (S-wave) velocities
C	Number of channels in a convolutional layer
d	Half the thickness of the plate
D	Damage location node in a graph
D_z	Dimension of the latent feature vector
$\mathbf{e}_{uv}$	Feature vector of an edge between nodes u and v
E	Excitation source node in a graph / Young's Modulus
$\mathcal{E}$	Set of edges in a graph

ϵ	Small constant for numerical stability / Bernoulli random variable
$f(x)$	Activation function (e.g., ReLU, PAF)
F	Non-linear mapping function in a neural network
$\mathscr{F}$	Fourier Transform operator
ϕ	Scalar potential in Helmholtz decomposition
$\Phi(y)$	Through-thickness distribution of the scalar potential
γ	Width of the Gaussian window in Morlet wavelet
$g(\cdot)$	Decoding function in the Wave Signal Decoder
G	Graph, consisting of nodes and edges, $G = (V, \mathscr{E})$
$\mathcal{G}(x)$	Gaussian positional encoding field
$\mathbf{h}_v^{(l)}$	Feature vector of node v at layer l
H	Set of parameters in a neural network / Input embedding matrix
$I(\cdot)$	Indicator function
k	Wavenumber

K	Kernel tensor in CNN
K	Number of attention heads
λ, μ	Lamé constants of an elastic material
λ_1, λ_2	Weighting factors for physics-informed loss terms
$\mathscr{L}$	Loss function
L	Length of a time-series signal / Path distance
M	Kernel size in a convolutional layer
μ_{ToF} , σ^2_{ToF}	Predictive mean and variance of Time-of-Flight
$N(v)$	Set of neighbors of node v
ω	Angular frequency
ω_0	Center angular frequency in Morlet wavelet
Ω	Geometric domain of an obstacle
$\mathbf{p}_i$	Physical coordinate vector of a sensor/node
P	Pooling window size in CNN
$\mathcal{P}$	Set of valid wave paths

ψ	Vector potential in Helmholtz decomposition
$\psi(t)$	Mother wavelet function
$\Psi(y)$	Through-thickness distribution of the vector potential
Q, **K**, **V**	Query, Key, and Value matrices in attention mechanism
ρ	Material density
$\mathbf{R}_b$	Reflection matrix for a boundary
$\mathscr{R}$	Residual of a Partial Differential Equation
$s(t)$	Time-domain signal (raw or wavelet-transformed)
S	Stride in a pooling layer
σ	Non-linear activation function / Standard deviation
t	Time
t_{ToF}	Time-of-Flight
T	Stress tensor
$\mathcal{T}_b(\mathbf{p})$	Affine reflection operator for a boundary b
τ	Bias term (scalar) / Temperature in softmax / Quantile level

θ	Model parameters in a neural network
$\mathbf{u}$	Displacement vector
v	Target node in a graph
V	Set of nodes (vertices) in a graph
$\mathbf{w}, \mathbf{W}$	Learnable weight vector/matrix in a neural network
$\mathbf{x}$	Input vector
x, y, z	Cartesian coordinates
y	Output label (e.g., true ToF)
$\hat{\mathbf{y}}$	Predicted output vector from the model
$\mathbf{Z}$	Latent feature vector
ξ	Wavenumber

Abbreviations

ABFM	Attention-based fusion module
BN	Batch Normalization
CBAM	Convolutional Block Attention Module

CNN	Convolutional Neural Network
DNN	Deep Neural Network
GAT	Graph Attention Network
GCN	Graph Convolutional Network
GIN	Graph Isomorphism Network
GNN	Graph Neural Network
GRU	Gated Recurrent Unit
IPCA	Incremental Principal Component Analysis
KDE	Kernel Density Estimation
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MC Dropout	Monte Carlo Dropout
MCM	Multi-scale Convolutional Module
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error

MSFE	Multi-Scale Feature Extraction Module
MSST	Multi-Scale Spatio-Temporal Fusion Network
NTK	Neural Tangent Kernel
PAF	Parametric Activation Function
PCA	Principal Component Analysis
PCAMQ-Loss	Physics-Constrained Adaptive MSE-Quantile Loss
PIML	Physics-Informed Machine Learning
PINN	Physics-Informed Neural Network
RBF	Radial Basis Function
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
SciML	Scientific Machine Learning
SENET	Squeeze-and-Excitation Network
SGD	Stochastic Gradient Descent
SHM	Structural Health Monitoring

SNN	Spiking Neural Network
SSI	Signal Spatial Information
STI	Signal Temporal Information
STFT	Short-Time Fourier Transform
SVR	Support Vector Regression
ToF	Time-of-Flight
UQ	Uncertainty Quantification

Table of content

List of figures

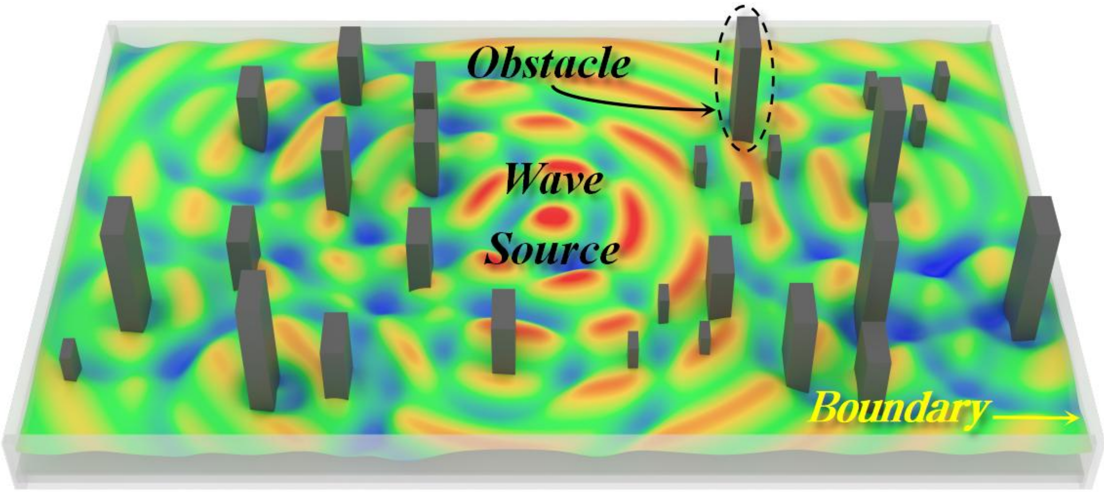
Chapter 1 Introduction

1.1 Research Background

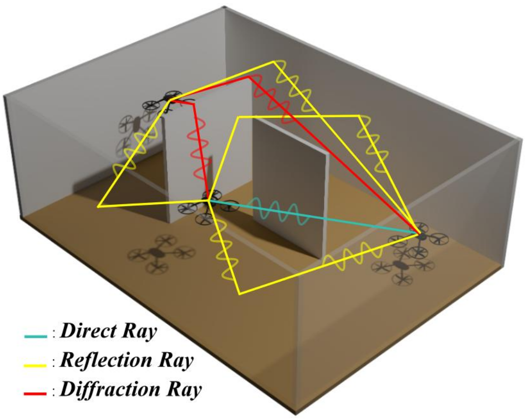
1.1.1 Time-of-Flight Extraction in Wave-Based Sensing: Significance and Universal Challenges

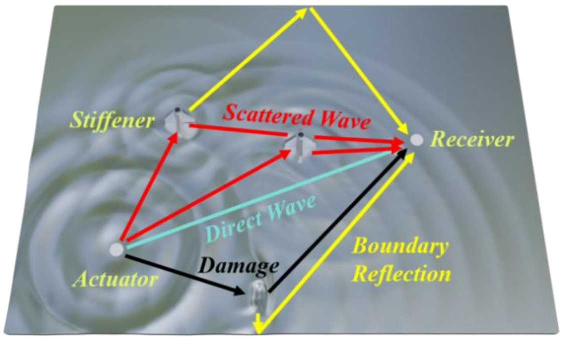
Wave propagation is a fundamental physical phenomenon that underpins our ability to sense, measure, and engage with the world around us. As a primary method of transferring energy and information through various media, waves in their acoustic, electromagnetic, and elastic forms support many modern technologies. In engineering and science, using waves to investigate a medium is a common practice. For example, ultrasonic waves are widely employed for non-destructive testing of materials and medical imaging [1-4], while electromagnetic waves are vital for radar, wireless communication, and geophysical exploration [5-7]. Similarly, optical waves in fiber sensors allow for long-distance monitoring of temperature and strain [8-10]. In all these instances, the main goal is to interpret how a wave changes as it travels through and interacts with a medium. The information of interest is contained in the wave's properties, such as amplitude, phase, and frequency. Among these, one of the most critical parameters is the wave's travel time.

The principle of measuring this travel time, known as Time of Flight (ToF), is a key aspect of modern sensing. This concept measures how long it takes for a signal to travel from a source to a target and often back to a receiver. This simple yet effective measurement provides vital data for estimating distance, mapping environments, and locating objects without physical contact. ToF is highly valuable because it translates time measurements into spatial information. In robotics, for instance, ToF data from LiDAR and sonar systems are crucial for simultaneous localization and mapping (SLAM) [11-14], helping autonomous agents navigate their surroundings. In autonomous vehicles, this enables the essential ability to perceive and respond to a changing environment, forming the foundation of collision avoidance and path planning [15-17]. ToF also influences consumer electronics by supporting features like facial recognition and augmented reality on smartphones [18, 19]. Even in fundamental science, ToF measurements are critical, from particle accelerators that identify subatomic particles based on their travel time to astronomical observations that measure cosmic distances [20, 21]. Across these fields, the common factor is an accurate measurement of arrival time to establish all subsequent spatial calculations. The accuracy of the entire system, whether the environmental model of an autonomous vehicle [22, 23] or the diagnostic images used by medical professionals [24, 25], fundamentally depends on the reliability of the ToF measurement.



(a)





(b) (c)

Figure 1.1 Challenges in time of arrival extraction of waves in complex environments: (a) A general complex structure with propagation wave; (b) Collaborative Localization of UAV via electromagnetic Waves; (c) Time of arrival extraction for structural health monitoring using guided waves.

Despite its critical importance, accurately extracting ToF remains a complex task, especially in real-world, intricate environments. The quality of the measured signal is often affected by various factors, including ambient noise and signal attenuation [26, 27]. However, a more fundamental and widespread challenge comes from multipath effects [28, 29]. The signal does not travel solely along a direct line-of-sight path in realistic settings with boundaries or obstacles. Instead, it reflects and scatters off nearby surfaces, producing multiple secondary wave arrivals at the receiver. These multipath components overlay the direct signal, often obscuring or distorting the initial arrival. This superposition makes it harder to correctly identify the wave packet associated with the event of interest, creating a significant obstacle to achieving reliable and accurate ToF estimation. This universal challenge is visually conceptualized in Figure 1.1, which illustrates how wave propagation in environments with obstacles and boundaries inevitably leads to complex multipath scenarios across different applications.

This multipath challenge is a common phenomenon, manifesting in various forms across different technological fields. Optical systems, such as LiDAR, can cause "ghost" points in point clouds, leading to inaccurate environment mapping [30]. In wireless localization and GPS, it is a primary cause of positioning errors [31, 32], while in acoustics and sonar, it creates reverberation that can obscure faint target echoes [33, 34]. The main issue is that these secondary arrivals are coherent, structured signals, not random noise. Traditional ToF estimation methods, such as leading-edge detection or cross-correlation [35, 36], often struggle in such scenarios because the signal's bandwidth limits their ability to distinguish closely arriving wave packets [37].

Therefore, developing robust methods to disentangle these complex, overlapping wavefields is a vital research area. Although this is a common challenge, it becomes particularly severe in structural health monitoring (SHM) [38-44], where the multipath effect, mainly caused by reflections from structural boundaries, directly interferes with detecting subtle damage signals. To illustrate and analyze the broader issue of ToF extraction, this thesis uses guided waves in SHM, specifically Lamb waves, as a primary example throughout the study. The upcoming sections will focus on this specific aspect of the ToF challenge, exploring its implications and possible solutions.

1.1.2 Guided Wave-based Structural Health Monitoring

While the challenge of ToF extraction is universal, it becomes especially critical and complex in the field of SHM. SHM marks a shift from traditional, periodic non-destructive evaluation (NDE) [45] to continuous, in-situ assessment of a structure's integrity throughout its operational life. SHM aims to deliver real-time, actionable data about a structure's condition, enabling early damage detection, predicting remaining useful life, and facilitating a transition from schedule-based to condition-based maintenance [46]. This is particularly important for high-value and safety-critical assets such as aircraft fuselages, pipelines, pressure vessels, and large-scale civil infrastructure, where unexpected failures can have catastrophic consequences.

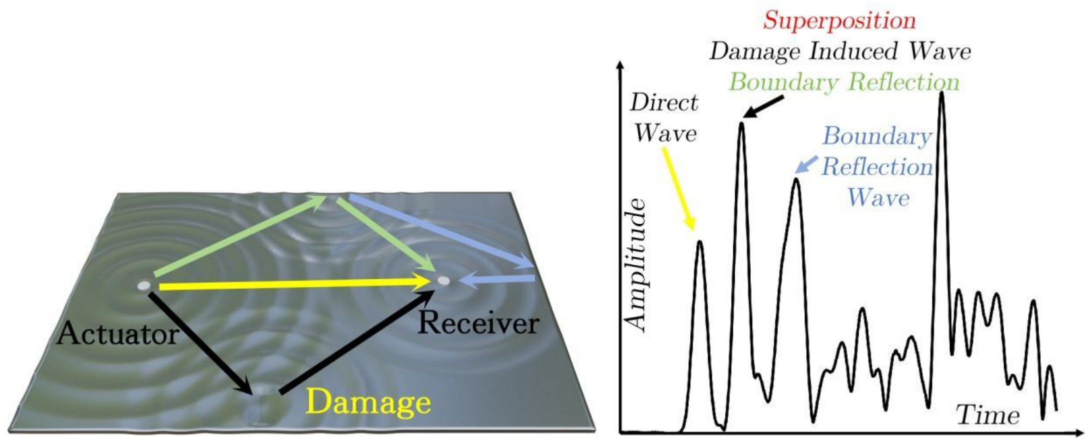
Among the various sensing methods used in SHM, techniques based on ultrasonic guided waves have become a promising approach for inspecting large, plate-like, and shell-like structures [47, 48]. Guided waves, such as Lamb waves, are elastic waves that travel within a structure's boundaries, allowing them to cover long distances with minimal energy loss. This makes them highly effective for examining large areas using just a few permanently attached transducers, typically piezoelectric wafer active sensors (PWAS) [49, 50]. Additionally, guided waves are susceptible to local changes in a structure's mechanical properties, enabling them to detect different types of damage, including cracks, corrosion, and delaminations, often before these problems become critical [41, 51-53].

The diagnostic process in guided wave-based SHM mainly depends on interpreting signals from this sensor network. When a guided wave encounters a defect, it scatters in multiple directions. Part of this scattered energy moves toward the sensors, carrying a distinctive signature of the damage. The main task for many damage localization algorithms is to analyze these signals and determine the ToF of the wave component that travels from an actuator, bounces off the damage, and reaches a sensor [54, 55]. Using ToF information from multiple actuator-sensor paths, the damage location can be pinpointed using imaging techniques like the delay-and-sum method [56-59]. Therefore, the accuracy of the whole SHM system relies heavily on precisely measuring the ToF of these often faint, damage-scattered waves. This highlights the universal challenge of multipath interference, previously mentioned, within the specific context of SHM.

1.1.3 Signal Superposition in Complex Scenarios

In guided-wave-based SHM, the common issue of multipath interference presents a complex challenge. Fundamentally, for any structure of finite size, guided waves inevitably interact with geometric boundaries, producing strong, coherent reflections that return to the inspection area [60, 61]. This is the primary and most understood source of signal superposition in plate-like structures.

This fundamental challenge is demonstrated in Figure 1.2. A signal captured by any sensor in the network is not a simple representation of a single wave event but rather a complex mixture of wave packets arriving at different times. The detected signal is a combination of the direct wave, boundary reflections, and the damage-induced scattered wave, all of which combine to produce a complex signal that is difficult to interpret.



(a) (b)

Figure 1.2 Wave propagation and superposition in Lamb wave-based SHM: (a) sketch of wave propagation paths in a plate with damage; (b) typical wavelet coefficients of a time-domain signal after wavelet transformation

However, the complexity of realistic engineering structures far exceeds these simple geometric boundaries. The main issue with boundary reflections is often worsened by many additional waves scattered from other non-damage-related structural features. Components typically include integral stiffeners, rivets, weld lines, or attached brackets as part of their design. These features, although not signs of damage, also act as strong scatterers, generating their own sets of wave components that further fill and complicate the wavefield [62, 63].

Therefore, the overall signal captured at a sensor usually includes: (1) the direct wave; (2) the desired but often low-amplitude wave scattered from actual damage; (3)

high-amplitude waves reflected from the structure's main boundaries; and (4) many waves scattered from other structural features. The temporal overlap of these different wave components limits the practical application of ToF-based methods. Reflections from boundaries and scattering from structural features can easily overwhelme or distort the subtle signature of the damage-scattered wave, especially when the damage is minor or near these features [64, 65]. This interference significantly decreases the accuracy of ToF-based localization algorithms and restricts the effective inspection area. Disentangling this complex wavefield to isolate the true ToF of the damage-scattered wave remains a persistent and critical challenge in the field, serving as a significant bottleneck for widespread industrial use of SHM technologies [66].

1.1.4 Evolution of Solutions and the Research Gap

In response to the significant challenge of signal superposition, the research community has explored various solutions that have evolved. Historically, efforts to address this problem primarily followed two approaches. The first involved direct physical methods, such as applying absorbing materials to the edges of a structure to reduce boundary reflections [67]. Although simple, this approach is often impractical in real-world settings because achieving perfect absorption is difficult, and the additional weight or materials may be undesirable, especially in aerospace applications. The second, more common approach uses advanced signal processing techniques. Methods such as cross-correlation [68], matched filtering [68], and various time-frequency analysis techniques, including wavelet transforms and the Short-Time Fourier Transform (STFT), have been employed to enhance the signal-to-noise ratio and attempt to separate overlapping wave packets [69, 70]. However, these methods have inherent limitations when wave packets significantly overlap in the time domain, a scenario that is frequently encountered in complex geometric structures. Such approaches often rely on physics-based models developed by experts and struggle to adapt to changing operational conditions or different types of damage.

With recent advances in computational power and data-driven methods, intense learning has made SHM a strong alternative [71, 72]. Different neural network architectures have successfully learned the complex, non-linear mapping from raw sensor signals directly to damage location or its ToF signature. For example, A Sattarifar *et al.* have used Convolutional Neural Networks (CNNs) to link signal features with damage location directly [73]. M Schaller *et al.* employed a noval convolution (ModeConv) to train on cross-correlations between simulation and experimental data [74]. These models often surpass traditional methods in accuracy. However, despite their strong predictive performance, these typical machine learning models have two main limitations that hinder their use in safety-critical engineering applications. First, they often operate as "black boxes," offering little insight into their decision-making process. This lack of physical interpretability makes it difficult to trust their predictions, especially in new scenarios not encountered during training [75]. Second, their dependence on end-to-end learning can be a double-edged sword. While it automates feature extraction, it overlooks essential intermediate physics, such as wave propagation and superposition, resulting in models that learn spurious correlations rather than true physical principles. This may result in poor generalization across different geometries, materials, or types of damage.

The limitations of both traditional and purely data-driven methods highlight a clear research gap: the requirement for answers that are not only exceptionally precise but also coherent in a physical sense and understandable. Consequently, a promising new approach has emerged: Physics-Informed Machine Learning (PIML) [76-79]. PIML aims to merge the predictive capabilities of deep learning with the strength and adaptability of first-principles physics. Rather than treating the model as an opaque entity, PIML incorporates domain knowledge straight into the learning framework. This can be achieved by designing the model's architecture, choosing physically meaningful input features, or more often, by adding physics-based constraints into the loss function [80, 81]. Ensuring that the predictions of the model adhere to established physical principles, like the wave equation, significantly lessens the requirement for extensive labeled datasets, improves generalization, and yields results that are more dependable and easier to interpret, as evidenced by recent research focused on assessing cracks and identifying damage in composite materials [82, 83]. This emerging method provides an effective way to address the ongoing challenges in ToF extraction for SHM. Framing the research within this innovative framework sets the academic frontiers of this thesis, which aims to create new PIML techniques that are not only effective but also deeply rooted in the physics of wave propagation.

1.2 Research Objectives

This doctoral research aims to create and validate innovative PIML frameworks for precise and dependable ToF extraction in guided-wave-based SHM. The main focus is to tackle the persistent challenge of signal superposition caused by boundary reflections and scattering from structural features in complex waveguides.

This thesis seeks to fulfill the overarching aim by targeting specific research objectives that align with the gradual progression of three related but separate frameworks.

1. **Develop a physics-guided neural network capable of eliminating boundary reflections for accurate ToF extraction.** The main goal is to address the core issue of signal aliasing in bounded structures. This involves designing a neural network architecture (MSST) that can learn to distinguish between damage-scattered waves and high-amplitude boundary reflections,

thereby increasing the effective inspection area and demonstrating the viability of a physics-guided, data-driven approach.

2. **To develop a physically interpretable, graph-based model that explicitly simulates wave propagation and superposition.** The aim is to create a model (PhysRayGNN) that accurately reflects the underlying physics. This involves using a physics-informed ray tracing method to build a high-fidelity graph of wave paths and employing a Graph Neural Network (GNN) to simulate how waves travel, attenuate, and superimpose. This approach aims to enhance both prediction accuracy and the crucial aspect of model interpretability.
3. **To create a robust and generalizable framework for ToF extraction in complex environments with multiple scatterers.** This objective is to address the challenges of real-world engineering structures, highlighting two key innovations:
    - The development of an innovative network architecture, termed WaveSenseNet, along with a parametric activation function, addresses the inherent low-frequency bias found in traditional neural networks. This allows the model to effectively capture the crucial high-frequency characteristics of scattered wave packets.

- Creating a **Gaussian positional encoding map** to effectively incorporate the physical geometry and scattering characteristics of complex features like stiffeners and bolts into the learning process, helping the network to make physically consistent predictions in highly cluttered wavefields.

Collectively, these goals create a defined research direction aimed at enhancing cutting-edge technologies through the development of structural health monitoring solutions that are not only reliable and accurate but also fundamentally rooted in physical principles.

1.3 Overview of the Thesis

This thesis consists of six chapters, ensuring a coherent transition from fundamental theory to more sophisticated solutions for intricate SHM situations. Chapter 1 offers a concise overview of the focus and contents of each chapter.

Chapter 2 establishes the theoretical foundations needed for the following chapters. It outlines the fundamentals of Lamb wave transmission in plate structures, the science behind wave modeling through ray tracing, and the essential elements of the deep learning frameworks employed.

Chapter 3 presents the first proposed framework, a Multi-Scale Spatio-Temporal fusion network, designed to explicitly eliminate the influence of boundary reflections and address the first research objective.

Chapter 4 presents a more advanced framework, PhysRayGNN, which pioneers using a Graph Neural Network to explicitly model the physical processes of wave propagation, attenuation, and superposition, achieving the second research objective.

Chapter 5 introduces WaveSenseNet, the third and most comprehensive framework, created for reliable ToF extraction in highly complex environments. This work features major architectural innovations to meet the third research goal.

Chapter 6 concludes the thesis by highlighting the key contributions, examining the limitations of the suggested methods, and suggesting encouraging avenues for future exploration in physics-informed SHM.

Chapter 2 Theoretical background

2.1 Introduction

The development of robust and physically-grounded frameworks for SHM, as outlined in this thesis, is based on integrating principles from wave mechanics, computational modeling, and machine learning. A solid understanding of these core concepts is crucial before exploring the innovative methods introduced in the following chapters. This chapter, therefore, functions as a standalone theoretical toolkit, offering the essential background in three key areas that underpin this research.

First, the chapter will begin by explaining the basic theory of Lamb wave propagation in thin-walled, isotropic plates. From the general equations of motion in an elastic medium, we will derive the key Rayleigh-Lamb equations. This analysis will clarify the dispersive and multi-modal nature of Lamb waves, which are essential for understanding both their potential and the challenges of using them for SHM applications.

Second, we will examine the principles of physical wave modeling through ray tracing. This section describes the fundamental assumptions of geometric acoustics and explains the image source method, an important technique for predicting wave paths in enclosed spaces. While using ray tracing to generate graphs is a new contribution of this thesis discussed in Chapter 4, this section will focus on the core theory of ray tracing itself, providing the necessary background for its later application.

Finally, the chapter will give a systematic overview of advanced deep learning architectures used for wave signal analysis. This includes the fundamental principles of CNN [84] for extracting features from time-series data, the message-passing approach of GNN [85], and the attention mechanism [86]. Moreover, we will examine essential physics-related elements of deep learning, including the intrinsic frequency bias present in neural networks and the foundational idea behind the Physics-Informed Neural Network (PINN) framework.

Through the development of these fundamental principles, this chapter seeks to provide the reader with the critical insights necessary to comprehensively grasp the reasoning, structure, and innovations of the specialized frameworks detailed in the remainder of this thesis.

2.2 Theory of Lamb Wave Propagation in Plate Structures

Lamb waves are a type of elastic guided wave that propagate through solid plates with free edges. Their ability to travel long distances, coupled with a significant sensitivity to structural damage, makes them important tools in SHM. A clear understanding of their propagation characteristics, especially their multi-modal and dispersive properties, is crucial for designing and interpreting any Lamb wave-based

SHM method. This section systematically derives the governing equations for Lamb waves from the fundamental principles of elastodynamics and explains their key physical features.

2.2.1 Governing Equations of Elastodynamics

Consider an infinitely large, homogeneous, and isotropic elastic plate of thickness $2d$. A Cartesian coordinate system is established such that the $x-z$ plane lies in the neutral plane of the plate, with the y-axis normal to the plate surface. The plate surfaces are located at $y=\pm d$.

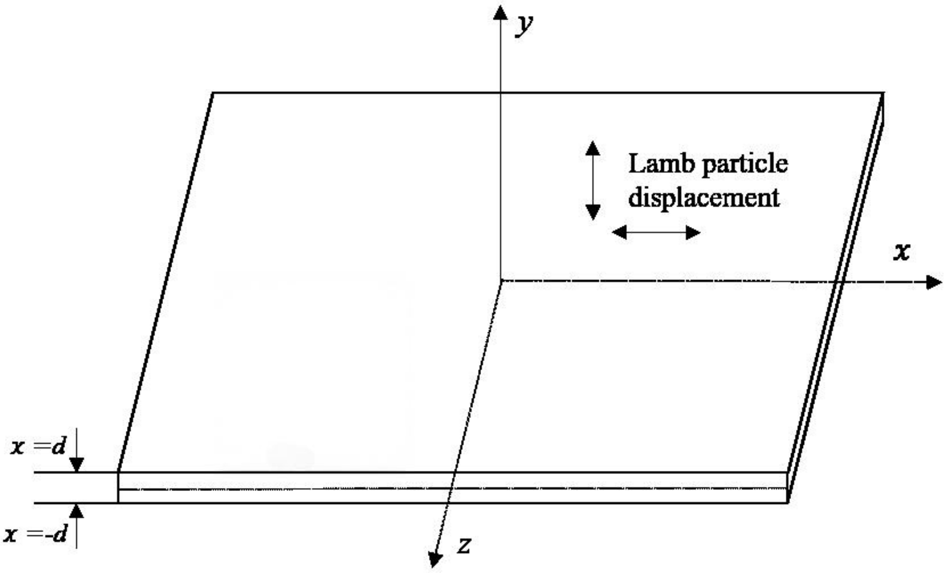


Figure 2.1 A diagram representing the plate configuration within a Cartesian coordinate framework, depicting the thickness of the plate and the displacement components of particles associated with Lamb waves.

In the absence of body forces, the equation of motion for any point within the elastic medium is given by the Navier-Cauchy equation as

$$\mu\nabla^2\mathbf{u} + (\lambda + \mu)\nabla(\nabla \cdot \mathbf{u}) = \rho\frac{\partial^2\mathbf{u}}{\partial t^2} \tag{2.1}$$

where $\mathbf{u}$ is the displacement vector, ρ is the material density, and λ and μ are the Lame constants of the material, which define its elastic properties. The governing equation that forms the basis of linear elastodynamics is obtained by integrating the motion equation with the strain-displacement relations along with the constitutive relationship of stress and strain for a linear isotropic material, typically referred to as Hooke's Law.

To solve this vector wave equation, it is convenient to employ the Helmholtz decomposition, which resolves the displacement vector field u into the sum of the gradient of a scalar potential ϕ and the curl of a vector potential ψ :

$$\mathbf{u} = \nabla\phi + \nabla \times \boldsymbol{\psi} \tag{2.2}$$

Substituting Eq. (2.2) into Eq. (2.1) allows the complex vector equation to be decoupled into two independent, simpler wave equations for the scalar and vector potentials:

$$\nabla^2 \phi = \frac{1}{c_p^2} \frac{\partial^2 \phi}{\partial t^2} \tag{2.3}$$

$$\nabla^2 \boldsymbol{\psi} = \frac{1}{c_s^2} \frac{\partial^2 \psi}{\partial t^2} \tag{2.4}$$

Here, $c_p = \sqrt{(\lambda + 2\mu)/\rho}$ is the velocity of the pressure (P-wave), and $c_s = \sqrt{\mu/\rho}$ is the velocity of the shear (S-wave). The scalar potential ϕ describes the dilatational component of the wavefield, corresponding to longitudinal wave propagation, while the vector potential ψ describes the rotational component, corresponding to shear wave propagation.

For a two-dimensional plane strain problem where the wave propagates in the $x - y$ plane (all field variables are independent of z), the vector potential ψ has only one non-zero component, ψ_z. The displacement components can thus be expressed as:

$$u_x = \frac{\partial \phi}{\partial x} - \frac{\partial \psi_z}{\partial y} \tag{2.5}$$

$$u_y = \frac{\partial \phi}{\partial y} + \frac{\partial \psi_z}{\partial x} \tag{2.6}$$

2.2.2 The Rayleigh-Lamb Equations

To identify the guided wave solutions that may exist within the plate, we consider harmonic waves traveling in the x-direction, represented in the following manner:.

$$\phi(x, y, t) = \Phi(y)e^{i(\xi x - \omega t)} \tag{2.7}$$

$$\psi_z(x, y, t) = \Psi(y)e^{i(\xi x - \omega t)} \tag{2.8}$$

where $\Phi(y)$ and $\Psi(y)$ are functions describing the through-thickness distribution of the potentials, ξ is the wavenumber, and ω is the angular frequency. Substituting these assumed solutions into the wave equations (2.3) and (2.4) yields ordinary differential equations for $\Phi(y)$ and $\Psi(y)$, whose general solutions are linear combinations of sine and cosine functions.

A valid Lamb wave solution must satisfy the boundary conditions of the plate, which are that the top and bottom surfaces ($y = \pm d$) are traction-free. This requires the normal stress T_{yy} and shear stress T_{xy} to vanish at these surfaces:

$$T_{yy} = \lambda\left(\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y}\right) + 2\mu\frac{\partial u_y}{\partial y} = 0 \text{ at } y = \pm d \tag{2.9}$$

$$T_{xy} = \mu\left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}\right) = 0 \text{ at } y = \pm d \tag{2.10}$$

The application of these boundary conditions results in the formulation of a system of linear equations that concerns the amplitudes of the potential functions involved. In order for a non-trivial solution to be feasible within this mathematical framework, it is essential that the determinant of the coefficient matrix associated with this system is equal to zero. This particular condition is critical as it leads to the derivation of the characteristic equations that govern the behavior of Lamb waves, which are typically known as the Rayleigh-Lamb equations. The specific solutions of these equations can be categorized into two distinct families based on the symmetry of their particle motion relative to the mid-plane $(y = 0)$:

1. **Symmetric Modes**: The in-plane displacement u_x is symmetric, and the out-of-plane displacement u_y is antisymmetric with respect to the mid-plane. The characteristic equation is:

$$\frac{\tan(qd)}{\tan(pd)} = -\frac{4\xi^2 qp}{(\xi^2 - q^2)^2} \tag{2.11}$$

2. **Antisymmetric Modes:** The in-plane displacement u_x is antisymmetric, and the out-of-plane displacement u_y is symmetric with respect to the mid-plane. The characteristic equation is:

$$\frac{\tan(pd)}{\tan(qd)} = -\frac{4\xi^2 qp}{(\xi^2 - q^2)^2} \tag{2.12}$$

where $p^2 = \left(\omega/c_p\right)^2 - \xi^2$ and $q^2 = (\omega/c_s)^2 - \xi^2$.

2.2.3 Dispersion Curves and Their Physical Significance

The Rayleigh-Lamb equations (2.11) and (2.12) are transcendental and do not have closed-form solutions. They implicitly define the relationship between the angular frequency ω and the wavenumber ξ. For a given frequency, there typically exist multiple solutions for the wavenumber, and vice versa. This relationship is visualized through dispersion curves, which are crucial for understanding and applying Lamb waves. Figure 2.2 shows a typical set of dispersion curves for an aluminum plate. These curves are generated by numerically solving the Rayleigh-Lamb equations using common material properties for aluminum. The parameters typically used are a Young's modulus of approximately 70 GPa, a Poisson's ratio of 0.33, and a density of 2700 kg/m^3.

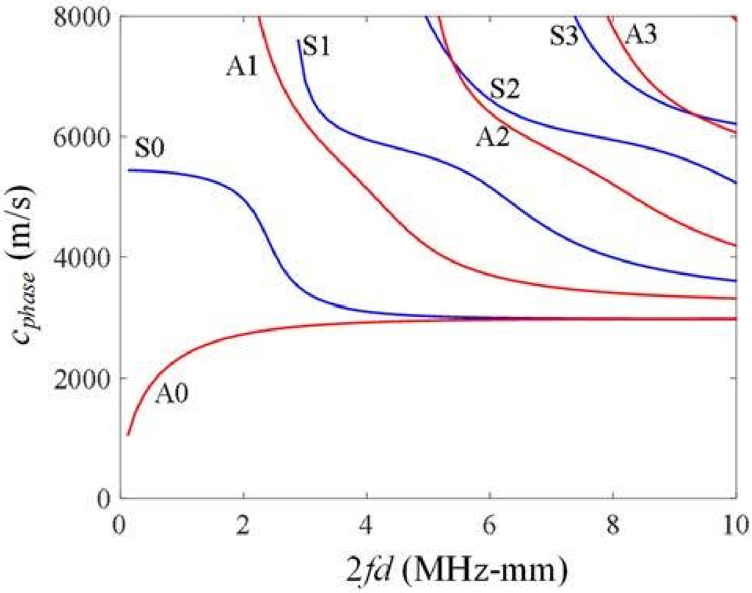


Figure 2.2 Dispersion characteristics of Lamb waves propagating in an aluminum plate.

The dispersion curves reveal two fundamental properties of Lamb waves:

- **Multi-modality**: At any given frequency (except at very low frequencies), multiple Lamb wave modes can exist simultaneously (e.g., S0, A0, S1, A1, etc.). Each mode represents a unique wave structure, corresponding to a specific pattern of particle vibration and energy distribution through the plate's thickness. As the frequency rises, the amount of potential propagating modes also grows. Various modes exhibit distinct sensitivities to different forms of damage, offering valuable diagnostic insights while simultaneously complicating the interpretation of signals.

- **Dispersion**: Dispersion refers to the occurrence in which the speed of a wave's propagation varies with its frequency. Generally, dispersion curves illustrate two kinds of velocity:

  1. **Phase Velocity** ($c_{ph} = \omega/\xi$): The speed at which the phase of a single-frequency harmonic wave propagates.
  2. **Group Velocity** ($c_g = d\omega/d\xi$): The speed at which the energy of a wave packet, which is composed of a group of frequencies, propagates. In SHM applications, the arrival time of a wave packet is the quantity of interest, making the group velocity the more critical physical parameter.

The dispersive characteristics of Lamb waves lead to the gradual spreading of a broadband wave packet as it moves, due to the fact that various frequency components propagate at differing velocities. This alteration in the shape of the wave needs to be considered during signal analysis for precise damage localization.

Among all the possible modes, the two fundamental ones—S0 (symmetric) and A0 (antisymmetric)-are especially important because they exist across the entire frequency spectrum, starting from zero frequency and having no cutoff. At lower frequencies, the S0 mode functions similarly to a tensile wave within the plate, whereas the A0 mode is akin to a flexural or bending wave. In practical SHM applications, a specific frequency range and mode are often chosen to maximize sensitivity to particular types of damage while reducing the complexity of the received signals.

2.3 Physical Modeling of Wave Propagation: Ray Tracing

While the wave equation presented in Section 2.2 offers a detailed physical explanation of Lamb wave propagation, solving it numerically for complex geometries can be computationally demanding. In many SHM applications, a full wavefield simulation is often unnecessary. Instead, the primary focus is on the ToF and paths of critical wave packets between sources, scatterers, and receivers. In this context, ray tracing provides a computationally efficient and physically intuitive alternative. This section explains the fundamental principles of ray tracing related to guided wave modeling in plate structures.

2.3.1 Geometric Acoustics Approximation

Ray tracing is based on the geometric acoustics approximation, which is valid when the wavelength λ of the wave is significantly smaller than the characteristic dimensions of the environment, such as the size of the plate or the distance between sensors. Under this condition, the complex wavefield can be simplified and represented as a collection of discrete "rays." Each ray represents a trajectory along which wave energy propagates.

This approximation enables us to transition from addressing complex partial differential equations to a more manageable problem of geometric pathfinding. The propagation of each ray is assumed to follow a straight line within a homogeneous medium, with its interaction with boundaries governed by straightforward geometric principles.

2.3.2 The Law of Reflection

When a ray encounters a boundary, its behavior is governed by the law of reflection. For a flat boundary, such as the edge of a plate, the law states that the angle of incidence θ_i is equal to the angle of reflection θ_r. Both angles are measured with respect to the normal of the boundary surface. The incident ray, the reflected ray, and the surface normal are all situated in a single plane.

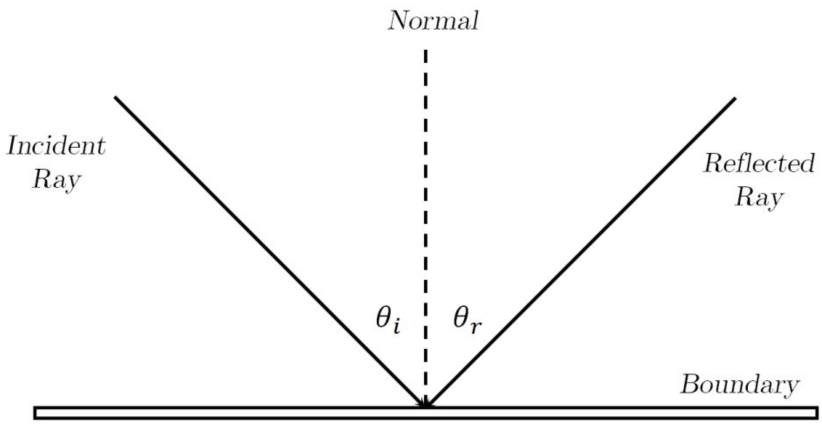


Figure 2.3 Illustration of the law of reflection at a planar boundary.

This principle is the foundation of tracking wave paths in enclosed structures. By repeatedly using this law, it is possible to trace the path of a wave as it undergoes multiple reflections from the plate.

2.3.3 The Image Source Method

While one could trace rays step-by-step, a much more elegant and computationally efficient way to find reflection paths in rectangular geometries is the Image Source Method. This approach provides a non-iterative solution to determine the exact path of a reflected wave by turning the reflection problem into a simple line-of-sight problem.

The core concept is to create "virtual" or "image" sources by reflecting the true source across the boundaries of the plate. The path of a reflected wave is then geometrically equivalent to the straight-line path from an image source to the receiver. This can be formalized through a set of affine reflection operators. For a rectangular plate defined by boundaries at $x_{\min}, x_{\max}, y_{\min}, y_{\max}$, we define a reflection operator $\mathcal{T}_b(\mathbf{p})$ for each boundary $b \in \mathcal{B} = \{\text{left, right, top, bottom}\}$:

$$\mathcal{T}_b(\mathbf{p}) = \mathbf{R}_b\mathbf{p} + \mathbf{t}_b \tag{2.13}$$

where $\mathbf{R}_b$ is a reflection matrix and $\mathbf{t}_b$ is a translation vector. For example, for the right boundary at $x = x_{max}$, the operator is defined by:

$$\mathbf{R}_{\text{right}} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{t}_{\text{right}} = \begin{bmatrix} 2x_{\text{max}} \\ 0 \end{bmatrix} \tag{2.14}$$

Using these operators, we can systematically identify wave paths of different orders:

1. Direct Path (0-order reflection): This is the direct line-of-sight path. The position of the 0-order image source is simply the source itself, $\mathbf{p}_s^{(0)} = \mathbf{p}_s$.
2. Single-Reflection Paths (1st-order reflection): The position of a 1st-order image source, $\mathbf{p}_s^{(1)}$, is found by applying a single reflection operator to the true source position:

$$\mathbf{p}_s^{(1)}(b_1) = \mathcal{T}_{b_1}(\mathbf{p}_s), \forall b_1 \in \mathcal{B} \tag{2.15}$$

3. Double-Reflection Paths (2nd-order reflection): The position of a 2 nd-order image source, $\mathbf{p}_s^{(2)}$, is generated by applying a second, different reflection operator to the position of a 1st-order image source:

$$\mathbf{p}_s^{(2)}(b_1, b_2) = \left(\mathcal{T}_{b_2} \circ \mathcal{T}_{b_1}\right)(\mathbf{p}_s) = \mathcal{T}_{b_2}\left(\mathcal{T}_{b_1}(\mathbf{p}_s)\right), \forall b_1, b_2 \in \mathcal{B}, b_1 \neq b_2$$

(2.16)

For each potential path of order $k \in \{0,1,2\}$, the total path distance $L^{(k)}$ is the Euclidean distance from the corresponding k-th order image source to the target receiver at position $\mathbf{p}_r$:

$$L^{(k)} = \left\| \mathbf{p}_r - \mathbf{p}_s^{(k)} \right\|_2 \tag{2.17}$$

The geometric principle of this single-reflection method is shown in Figure 2.4. The true, curved trajectory of reflection from the source to the receiver can be determined by extending a straight line from the virtual image source, which serves as a mirror representation of the actual source situated on the opposite side of the boundary.

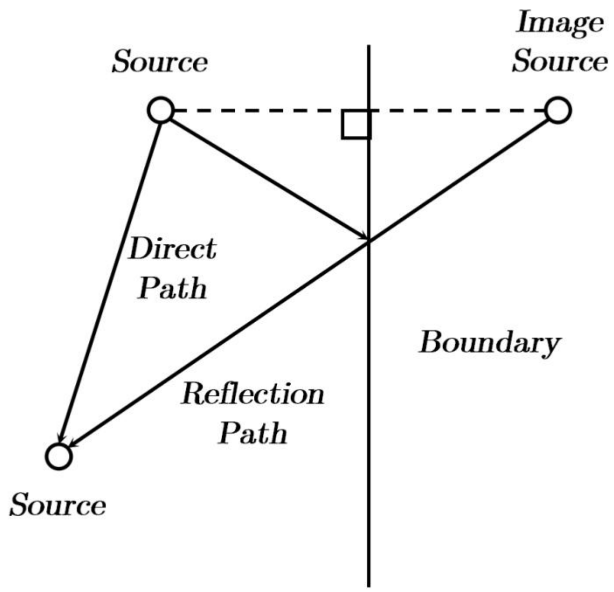


Figure 2.4 Illustration of the Image Source Method.

Using this Method, we can quickly and precisely find the paths and travel distances of all direct and reflected waves up to a certain reflection order. This offers a powerful, physics-based way to model the complex multipath environment in a plate, serving as an essential computational tool for the analyses in Chapter 4.

2.4 Deep Learning Architectures for Wave Signal Analysis

While physical models offer a basic understanding of wave propagation, deep learning architectures provide powerful, data-driven tools for interpreting the complex and often noisy signals collected by sensor networks. The models presented in this dissertation are grounded in various essential ideas related to deep learning. This part offers an in-depth examination of these core architectures, detailing how they function and their importance in the analysis of wave signals.

2.4.1 Convolutional Neural Networks

Convolutional Neural Networks, which gained prominence primarily for their application in image analysis, are remarkably effective at handling sequential data such as the time-domain signals found in SHM. For one-dimensional data, such as a wave signal $\mathbf{x} \in \mathbb{R}^{L \times C_{\text{in}}}$ with L time steps and C_{in} input channels, the core operation is the 1D convolution with a kernel tensor $\mathbf{K} \in \mathbb{R}^{M \times C_{\text{in}} \times C_{\text{out}}}$, where M is the kernel size and C_{out} is the number of output channels. The operation involves sliding each of the C_{out} kernels across the input signal. For the j-th output channel, the feature map $\mathbf{y}_j$ is computed as:

$$\mathbf{y}_j[n] = \sigma\left(b_j + \sum_{c=1}^{C_{in}} \sum_{m=1}^{M} \mathbf{x}_c[n+m-1] \cdot \mathbf{K}_{m,c,j}\right) \tag{2.18}$$

where σ is activation function, b_j is a learnable bias term for the j-th kernel, and $\mathbf{K}_{m,c,j}$ is the weight of the j-th kernel at position m for input channel c. Each kernel learns to identify a particular local pattern. In the context of wave signals, a kernel might detect the characteristic shape of a wave packet's leading edge or a specific oscillatory pattern.

CNN architectures generally incorporate pooling layers, like max-pooling, to diminish the spatial dimensions of the feature maps. For an input feature map y, the output z after max-pooling with a window of size P and stride S is:

$$\mathbf{z}[n] = \max_{i=0,\ldots,P-1}(\mathbf{y}[n \cdot S + i]) \tag{2.19}$$

This operation makes the feature representation more robust to small shifts in the input signal, helping to create a more abstract summary of the information present.

2.4.2 Graph Neural Networks

While CNNs are well-suited for regularly structured data, many SHM problems involve irregular spatial relationships, such as the sensor network on a plate. GNNs are designed for such scenarios. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$.

The fundamental operational paradigm of most GNNs is message passing. Within this framework, each node repeatedly adjusts its feature vector by consolidating information obtained from its neighboring nodes. The update process for a node $\mathbf{v}$ at layer l can be described by a general formula:

$$\mathbf{h}_{\mathbf{v}}^{(l+1)} = \mathrm{UPDATE}^{(l)}\left(\mathbf{h}_{\mathbf{v}}^{(l)}, \mathrm{AGGREGATE}^{(l)}\left(\left\{\mathbf{m}_{\mathbf{u}\to\mathbf{v}}^{(l)} : \mathbf{u} \in \mathcal{N}(\mathbf{v})\right\}\right)\right) \tag{2.20}$$

Where $\mathbf{m}_{\mathbf{u}\to\mathbf{v}}^{(l)}$ represents the "message" from neighbor $\mathbf{u}$ to node $\mathbf{v}$. For a GCN, a popular GNN variant, the message is typically a linear transformation of the neighbor's feature vector, and the update rule simplifies to:

$$\mathbf{h}_{\mathbf{v}}^{(l+1)} = \sigma\left(\sum\nolimits_{\mathbf{u}\in\mathcal{N}(\mathbf{v})\cup\{\mathbf{v}\}} \frac{1}{\sqrt{\deg(\mathbf{v})\deg(\mathbf{u})}}\mathbf{W}^{(l)}\mathbf{h}_{\mathbf{u}}^{(l)}\right) \tag{2.21}$$

Here, $\mathbf{h}_{\mathbf{v}}^{(l)}$ denotes the feature vector of node $\mathbf{v}$ at layer $l, \mathcal{N}(\mathbf{v})$ represents the set of its neighbors, $\deg(\cdot)$ is the degree of a node, $\mathbf{W}^{(l)}$ is a learnable weight matrix, and σ is an activation function. By stacking multiple layers, a node can integrate information from progressively farther away in the graph, which is analogous to simulating wave propagation across the sensor network.

2.4.3 The Attention Mechanism

The attention mechanism improves neural networks by enabling them to assign different levels of importance to various pieces of information dynamically. The most common form is the scaled dot-product attention. Given an input embedding matrix $\mathbf{H} \in \mathbb{R}^{n\times d_m}$, it is first projected into three matrices: queries $\mathbf{Q} = \mathbf{H}\mathbf{W}_Q$, keys $\mathbf{K} = \mathbf{H}\mathbf{W}_K$, and values $\mathbf{V} = \mathbf{H}\mathbf{W}_V$, where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$ are learnable weight matrices. The attention output is then computed as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{QK}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{2.22}$$

The dot product $\mathbf{QK}^T$ computes a similarity score between each query and every key. The scaling factor $\sqrt{d_k}$ ($d_k$denotes the dimension of keys) ensures stable gradients. The softmax function converts these scores into weights that sum to one.

In order to allow the model to concentrate on various facets of the information at the same time, multi-head attention is often employed. The input is segmented into different "heads," and attention is computed for each of these heads at once. Subsequently, the outcomes are combined and subjected to a linear projection.

$$\begin{aligned} &\text{MultiHead }(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{ head }_1, \ldots, \text{ head }_h)\mathbf{W}_O \\ &\text{where head}_i = \text{ Attention }\left(\mathbf{HW}_Q^{(i)}, \mathbf{HW}_K^{(i)}, \mathbf{HW}_V^{(i)}\right) \end{aligned} \tag{2.23}$$

For wave propagation modeling, an attention mechanism can learn that direct paths are usually more critical than multi-reflection paths, reflecting the physics of wave attenuation.

2.4.4 Frequency Principle of Deep Neural Networks

A key property of standard deep neural networks is their inherent spectral bias, often referred to as the Frequency Principle[87]. This principle indicates that throughout the training process, deep neural networks are inclined to learn the low-frequency aspects of a target function more rapidly than the high-frequency aspects. This phenomenon can be understood from two complementary perspectives: an intuitive analysis of a simple model and a more rigorous derivation based on the Neural Tangent Kernel (NTK) theory [88].

An Intuitive View from a Simple Model

We can first build intuition by analyzing a simple two-layer network fitting a 1D function $f^*(x)$:

$$h(x) = \sum_{j=1}^{m} a_j \sigma(w_j x + b_j) \tag{2.24}$$

The Fourier transform of a single activated neuron, using $\sigma(x) = tanh\,(x)$ for its analytical tractability, is:

$$\hat{\sigma}(w_j x + b_j)(k) = \frac{2\pi i}{|w_j|} \exp\left(\frac{i b_j k}{w_j}\right) \frac{1}{\exp\left(-\frac{\pi k}{2w_j}\right) - \exp\left(\frac{\pi k}{2w_j}\right)} \tag{2.25}$$

The key insight is the exponential decay with respect to the frequency k, which arises because smooth activation functions like $tanh\,(x)$ have rapidly decaying Fourier transforms. The Fourier transform of the full network output is a linear combination of these terms:

$$\hat{h}(k) = \sum_{j=1}^{m} \frac{2\pi i a_j}{|w_j|} \exp\left(\frac{i b_j k}{w_j}\right) \frac{1}{\exp\left(-\frac{\pi k}{2w_j}\right) - \exp\left(\frac{\pi k}{2w_j}\right)} \tag{2.26}$$

In the course of training using gradient descent, we aim to reduce a loss function, which is often the Mean Squared Error. Following Parseval's theorem, the loss can be equivalently calculated in either the spatial or the frequency domain:

$$L = \int \frac{1}{2}|f^*(x) - h(x)|^2 dx = \int \frac{1}{2}\left|\hat{f}^*(k) - \hat{h}(k)\right|^2 dk = \int L(k) dk \tag{2.27}$$

The gradient of the frequency-domain loss $L(k)$ with respect to a network parameter θ (e.g., w_j) can be approximated as:

$$\left|\frac{\partial L(k)}{\partial \theta}\right| \approx \left|\hat{h}(k) - \hat{f}^*(k)\right| \exp\left(-\frac{\pi|k|}{2|w_j|}\right) G(\boldsymbol{\theta}, k) \tag{2.28}$$

where $G(\boldsymbol{\theta}, k)$ is an $O(1)$ function. This equation is the mathematical cornerstone of the Frequency Principle. It shows that the magnitude of the gradient is exponentially suppressed for higher frequencies (larger $|k|$), especially at the beginning of training when the weights w_j are small. As a result, the error's low-frequency components are rectified at a significantly quicker pace compared to the high-frequency components.

A Rigorous Analysis via the Neural Tangent Kernel (NTK)

The Neural Tangent Kernel (NTK) provides a more formal framework for analyzing this observed behavior. The evolution of the network's parameters θ is described by the gradient flow $d\theta/dt = -\nabla_\theta R_S(\theta)$, where R_S is the empirical risk. This induces a change in the function's output $f(x, \theta(t))$:

$$\frac{df(x,\theta(t))}{dt} = \nabla_\theta f(x, \theta(t)) \cdot \frac{d\theta}{dt} \tag{2.29}$$

For sufficiently wide networks, the training dynamics can be linearized. In this NTK regime, the kernel $K(x, x^{'}) = \nabla_\theta f(x, \theta) \cdot \nabla_\theta f(x^{'}, \theta)$ remains nearly constant throughout training, $K(t) \approx K(0)$.

Let $u(x, t) = f(x, \theta(t)) - f^*(x)$ be the error. Its dynamics can be approximated by:

$$\frac{du(x,t)}{dt} \approx -\sum_{i=1}^{n} K(x, x_i, 0) u(x_i, t) \tag{2.30}$$

Applying a Fourier transform to this linearized equation results in the Linear Frequency Principle (LFP) model [89], which characterizes how the error evolves in the frequency domain.

$$\partial_t \mathcal{F}[u](\boldsymbol{\xi}, t) \approx -(\gamma(\boldsymbol{\xi}))^2 \mathcal{F}[u_\rho](\boldsymbol{\xi}) \tag{2.31}$$

Here, $\mathcal{F}[u]$ represents the Fourier transform of the error, ρ is the data sampling density, and $(\gamma(\boldsymbol{\xi}))^2$ is a damping factor that depends on the frequency vector $\boldsymbol{\xi}$. For a two-layer ReLU network, this damping factor has the form:

$$(\gamma(\xi))^2 \propto \frac{C_1}{||\xi||^{d+3}} + \frac{C_2}{||\xi||^{d+1}} \tag{2.32}$$

Where d is the input dimension and C_1, C_2 are constants. This equation shows that the damping factor is much larger for low frequencies (small $||\boldsymbol{\xi}||$). Consequently, low-frequency components of the error decay faster, confirming that the network learns low-frequency functions first.

In summary, both the intuitive model and the rigorous NTK analysis show that standard DNNs, when trained with gradient-based methods, have a strong spectral bias. They focus on learning low-frequency functions before fitting high-frequency details.

This principle holds important consequences for applications such as wave signal analysis, in which high-frequency onsets frequently signify the precise arrival time of a wave. A network biased toward low frequencies may overlook these important details, leading to a smoothed wave representation and less precise estimates. This theoretical insight drives the development of specialized architectures designed to overcome this inherent bias.

2.5 Traditional Methods for Time-of-Flight Extraction

Accurate ToF extraction is critical for guided wave-based SHM, as it directly influences the precision of damage localization. Traditional signal processing techniques for ToF estimation can be broadly categorized into threshold-based methods [90], envelope-based methods [91], and time-frequency analysis methods [92].

2.5.1 Threshold-based methods

The simplest approach involves defining a threshold amplitude; the arrival time is registered when the signal exceeds this level. However, this method is highly sensitive to noise and amplitude attenuation. To address this, statistical approaches such as the Akaike Information Criterion (AIC) have been widely adopted. The AIC method, originally adapted for seismic wave onset detection by Maeda, treats the arrival time picking as a change-point detection problem in an autoregressive process. While AIC improves robustness against noise compared to simple thresholding, its accuracy degrades significantly when dealing with highly dispersive waves or low signal-to-noise ratio (SNR) environments.

2.5.2 Envelope-based Methods (Hilbert Transform)

To mitigate the oscillatory nature of guided waves, the Hilbert Transform (HT) is commonly employed to extract the signal envelope. The peak of the envelope is then identified as the arrival time of the wave packet. This method is computationally efficient and effective for non-dispersive signals. However, in the context of Lamb waves, significant dispersion often causes wave packets to broaden and overlap. In such cases, the Hilbert Transform may fail to separate closely spaced modes (e.g., S_0 and A_0 modes), leading to merged peaks and inaccurate ToF estimation.

2.5.3 Time-Frequency and Decomposition Methods

To handle the dispersive and multi-modal nature of Lamb waves, time-frequency analysis techniques such as the Continuous Wavelet Transform (CWT) have been introduced. CWT maps the time-domain signal into the time-scale domain, allowing for the isolation of specific frequency components and better handling of dispersion. Similarly, the Matching Pursuit (MP) algorithm decomposes the signal into a dictionary of atoms to resolve overlapping echoes. Although these methods offer high resolution, they are often computationally expensive and require careful manual selection of parameters to match the specific wave propagation characteristics.

2.6 Summary

This chapter has established the crucial theoretical foundation necessary for comprehending the new frameworks introduced in this thesis. We initiated our study by extracting the core principles governing Lamb wave propagation within plate-like structures through elastodynamics equations. This examination emphasized the significant characteristics of multi-modality and dispersion, which influence the effectiveness and intricacy of employing these waves for SHM.

Next, we established the theoretical basis for a computationally efficient modeling method, ray tracing. By explaining the geometric acoustics approximation and the image source technique, we provided the physical basis for modeling complex wave reflection paths. This tool will be used in later chapters to develop our intelligent models.

Finally, we reviewed the core deep learning architectures that form the foundation of our proposed solutions. We explained how CNN extract features from time-series data, how GNN model relational structures, and how the attention mechanism enables adaptive feature weighting. The overall paradigm of PINN was presented as the conceptual link that integrates these different elements.

Chapter 3 Physics-Guided Neural Network for Structural Health Monitoring with Lamb Waves through Boundary Reflection Elimination

3.1 Introduction

The preceding chapters have established the fundamental challenges in guided wave-based SHM, identifying the superposition of wave signals as the primary obstacle to accurate ToF estimation. As detailed in Section 1.3, this superposition arises from multiple sources, with reflections from structural boundaries being one of the most dominant and persistent contributors to signal complexity. These high-amplitude boundary reflections can easily mask or distort the faint, damage-scattered waves, thereby compromising the reliability of damage localization algorithms and severely limiting the effective inspection area of a structure.

To address this foundational issue, this chapter presents the first of three novel frameworks developed in this thesis. The primary objective here is to tackle the boundary reflection problem head-on by developing a physics-guided neural network capable of eliminating their influence from the received signals. While traditional approaches have attempted to solve this through physical damping materials or conventional signal processing, they often fall short due to practical limitations or an inability to handle severely overlapped wave packets [93]. Recent advances in machine learning, particularly through hybrid approaches like physics-informed neural networks (PINNs) [94-97], which offer an innovative alternative in various applications. PINNs uniquely combine physical laws represented by partial differential equations (PDEs) with neural networks, enhancing model accuracy and reducing reliance on extensive datasets. In particular, the work by Shukla *et al.* [98] demonstrates the practical application of PINNs for the nondestructive quantification of surface-breaking cracks in metal plates. Their approach effectively integrates ultrasonic imaging data with physics-based models, significantly improving detection accuracy while minimizing the need for large datasets. This highlights the transformative potential of PIML in the field of SHM. More generally, deep learning enables extraction and processing of multiple signal features from the inputs, allowing nonlinear mapping between input time-domain signals and damage locations [99, 100]. For example, Zhang *et al.* [101] used a one-dimensional CNN to correlate the time-varying damage index (DI) directly with the damage location, which allows accurate localization of damage in plates with a few transducers. Additionally, a kernel regression method which incorporates deep metric learning was used to locate damage in metallic plates based on linear guided waves [102]. Wang *et al.* [103] proposed a real-time guided wave imaging method using a CNN for quantitative corrosion evaluation, alongside the establishment of the relationship between the wave signals and the velocity map. Rautela *et al.* [104] assessed some typical machine learning frameworks, including Dense Neural Networks, 1D-CNNs, recurrent neural networks, and long short-term memory (LSTM). The study assessed the robustness of the models with noisy datasets and their potential for real-time applications. Moreover, Sbarufatti *et al.* [105] utilized a multi-layer perceptron (MLP) to localize a crack in an aluminum plate. The model was trained using cross-correlations between the simulation data and in-field measurements. By artificially incorporating noise into the training data, the model exhibited strong generalization to experimental data. For composite structures, Lin *et al.* [106] also applied a CNN model to detect and locate defects in a composite wing. Heesch *et al.* [107] presented a generative adversarial network architecture to generate guided wave signals with the open guided waves database. Liu *et al.* [108] presented a novel approach using a structural digital twin for damage detection in carbon fiber reinforced plastics (CFRP) composites, leveraging a meta transfer learning framework. This method addresses the challenges of limited training data by generating virtual data under various damage conditions, significantly enhancing the accuracy of damage localization and classification. Xu *et al.* [82] introduced a physics-guided convolutional neural network (PGCNN), which combines physical models and data-driven techniques to achieve over 90% accuracy in detecting damage in CFRP composites across varying structures and conditions, further underscoring the effectiveness of hybrid models in SHM. This research highlights the transformative potential of machine learning for SHM, improving both the methods and outcomes of damage detection and localization. However, as discussed in Section 1.4, many of these models function as "black boxes" lacking the physical interpretability required for safety-critical applications.

This chapter presents an MSST fusion network, a specially designed architecture aimed at understanding the physics of wave reflection and scattering. By analyzing both the temporal data from wave signals (using wavelet transforms) and the spatial layout of the sensor network, the MSST network learns to differentiate between wave packets caused by damage and those caused by boundaries. The purpose is not just to classify signals but to perform a virtual "subtraction" of boundary reflections, effectively isolating the damage-related wave and allowing for accurate extraction of its ToF.

The development and validation of this framework represent the thesis's first major contribution. By demonstrating that a neural network can be guided by physical principles to solve the boundary reflection problem, this chapter establishes a foundation for more advanced models in later sections, which will address even more complex superposition scenarios. The effectiveness of the proposed MSST network will be shown with experimental data from a guided wave sensor array on an aluminum plate, emphasizing its ability to significantly expand the reliable detection area for SHM systems.

3.2 The multi-scale spatio-temporal fusion network

Prior to the construction of the multi-scale spatio-temporal fusion model, methods for signal generation, acquisition, and pre-processing are briefly introduced. Lamb wave signals are generated and captured using PZTs on a plate, used as an illustrative example to exemplify a thin-walled structure. As the pre-processing of the captured time-domain signals, the wavelet transform is used to extract the arrival time of different wave packets [109], which is termed as the Signal Temporal Information (STI). To extract the envelope of the response signal, the complex Morlet transform is utilized. The coefficients of the wavelet transform are given by:

$$CWT_f(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(t)\psi^*\left(\frac{t-b}{a}\right) \mathrm{d}t \tag{3.1}$$

where the mother wavelet $\psi(t)$ is formulated as:

$$CWT_f(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(t)\psi^*\left(\frac{t-b}{a}\right) \mathrm{d}t \tag{3.2}$$

Here, a and b are the scale and shift parameters, respectively. $\psi^*(t)$ represents the complex conjugate of the Morlet wavelet function. Here, ω_0 is the angular frequency, and γ defines the width of the Gaussian window. This analysis emphasizes the modulus at a constant scale to align with the central frequency of the excitation signal. The narrowband nature of the complex Morlet wavelet can effectively reduce noise interference.

Simultaneously, the structural dimensions (specifically, the plate length and width) and the spatial coordinates of the PZTs are incorporated as input features, collectively referred to hereafter as Signal Spatial Information (SSI). The STI and SSI are further synthesized in a fully connected module in the designed network. In general, with the input of the extracted temporal wavelet coefficients, the proposed network aims at eliminating the influence of boundary reflections to precisely extract the ToF of the scattered waves induced by the damage. This allows for the precise ToF extraction of waves scattered by structural flaws, a process outlined in Figure 3.1.

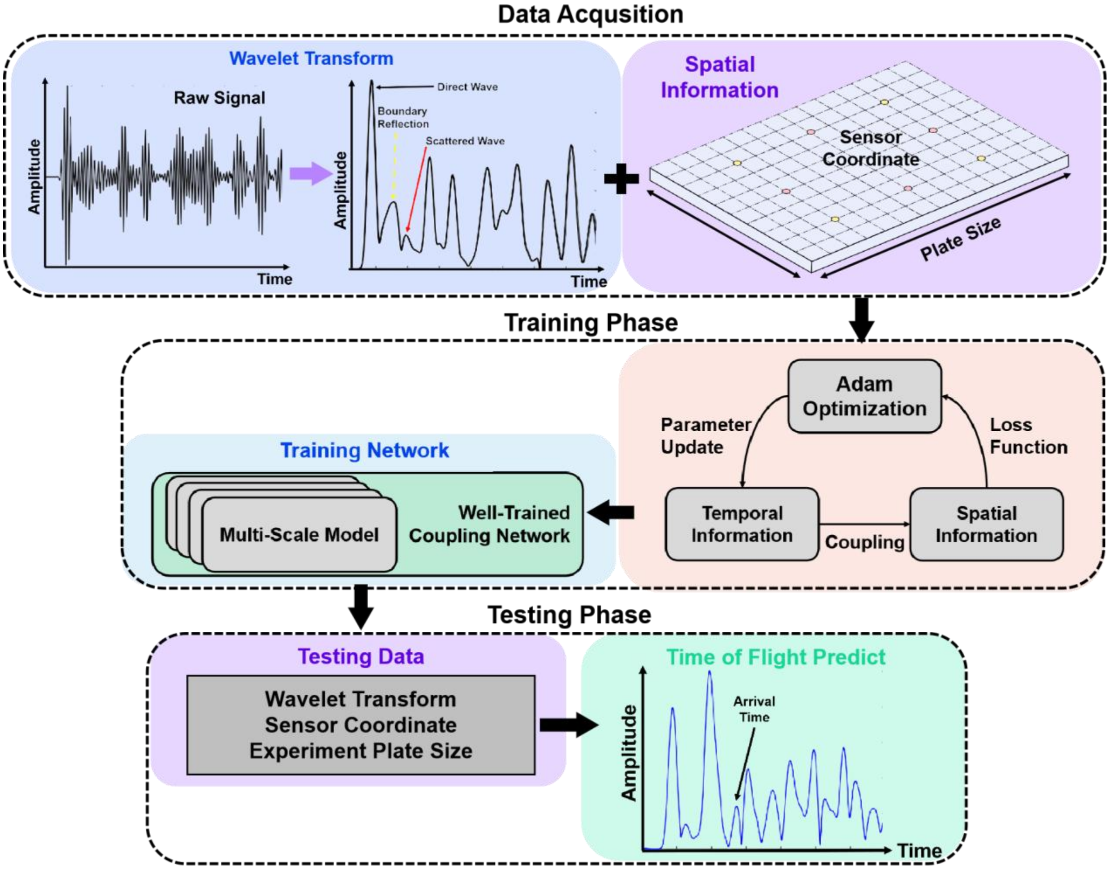


Figure 3.1 Flowchart of the proposed method.

3.2.1 Network Architecture

Assuming there is a total of m actuator-sensor paths, the corresponding time-domain signals are processed with the wavelet transform to extract their STI. Simultaneously, the coordinates of the actuators and sensors, along with the dimensions of the structure, constitute the SSI. Both STI and SSI form an input feature set $\mathbf{x}^{(i)} \in \mathbb{R}^{1\times m}$. Correspondingly, the ToF of scattered waves from damage is used as the output, which is labelled as $\mathbf{y}^{(i)} \in \mathbb{R}^{1\times 1}$. In the context of machine learning, the model essentially functions as a parametric regression model, trained to approximate a set of parameters representing the non-linear mapping relationship $F: \mathbf{x}^{(i)} \to \mathbf{y}^{(i)}$. The training process involves optimizing a mean squared error (MSE) loss function through parameter updates using stochastic gradient descent (SGD) method [110]. The specific expression is given as follows:

$$L(\mathbf{H}) = \frac{1}{M}\sum_{i=1}^{M} \|\mathbf{y}^{(i)} - F\big(\mathbf{x}^{(i)}; \mathbf{H}\big)\|_2^2 \tag{3.3}$$

where **H** represents the set of parameters to be optimized.

The overall network architecture consists of two main components: a multi-scale module and a fully connected module, as illustrated in Figure 3.2. In the first part, the temporal signals undergo a holistic convolutional layer with 256 filters, followed by a normalization and max-pooling layer. Subsequently, four parallel convolutional layers with different kernel sizes, comprising 32, 64, 64, and 128 filters respectively, are applied.

The features extracted from the first part are then flattened and concatenated with spatial information in the second part. This concatenated tensor is fed into two fully connected layers. The first layer consists of 64 neurons with ReLU activation function and a dropout rate of 0.01. The second layer serves as the output layer with a single neuron and linear activation function. The entire network is optimized using the Adam optimization algorithm [111] with a learning rate of 2×10^{-6}.

One-dimensional convolutional layers, commonly employed in signal processing and time-series data analysis, are also adopted in the proposed network. These layers apply convolution operations along a singular dimension, adept at capturing temporal dependencies and patterns while concurrently extracting temporal features [112]. The one-dimensional convolution operation is mathematically expressed as follows:

$$(\mathbf{X} * \boldsymbol{\varphi})_i = \sum_{k=1}^{K} \mathbf{X}_{i+k-1} \boldsymbol{\varphi}_k + \tau_i \tag{3.4}$$

In the equation, $\mathbf{X}$ represents the input sequence, φ denotes the one-dimensional convolutional filter, τ is the bias term, and $\sum$ signifies the summation over the convolutional filter size K.

Usually, a single-scale convolutional network faces challenges in capturing diverse and multi-scale temporal characteristics. Therefore, within the proposed network architecture, a multi-scale module has been meticulously designed specifically for extracting features from time-domain signals after the wavelet transformation, as depicted in Figure 3.2. This module intends to enhance the network's ability to discern subtle patterns in the input data. By employing convolutional layers with kernel sizes corresponding to scales 2, 3, 4, and 10, the Multi-scale Convolutional Module (MCM) facilitates feature extraction at different resolutions. For a given input set **X**, the MCM processes **X** through a set of convolutional layers, each associated with a specific scale. By denoting the convolutional operation at the i-th scale as Conv_i and applying the ReLU as nonlinear activation function [113], the module's output O is expressed as:

$$\mathbf{O} = \mathrm{ReLU}\left(\mathrm{Conv}_2\,(\mathbf{X})\right) \oplus \mathrm{ReLU}\left(\mathrm{Conv}_3\,(\mathbf{X})\right) \oplus \mathrm{ReLU}\left(\mathrm{Conv}_4\,(\mathbf{X})\right) \oplus \mathrm{ReLU}\left(\mathrm{Conv}_{10}\,(\mathbf{X})\right) \tag{3.5}$$

The $\oplus$ operator concatenates the outputs along the channel dimension. These convolutional operations capture features at various scales, enabling the network to distinguish intricate details.

In the lead-up to the multi-scale convolutional network, we introduce an initial global convolution operation, succeeded by Batch Normalization (BN) [114] and Max Pooling layers [115]. This combination aims at refining the feature extraction and overall performance. BN normalizes convolutional layer outputs to mitigate the vanishing gradient for improved stability and faster convergence as:

$$\mathrm{BN}(\mathbf{x}) = \gamma\left(\frac{\mathbf{x}_i-\mu_B}{\sqrt{\sigma_B^2+\epsilon}}\right)+\beta \tag{3.6}$$

where $\mathbf{x}_i$ represents the input, μ_B and σ_B are the mean and standard deviation, γ and β are learnable parameters, with ϵ preventing division by zero. BN accelerates training, augments stability, and improves generalization.

Max Pooling reduces spatial dimensions by selecting maximum values within each input window, aiming at decreasing complexity and enhancing model robustness to translations. Each window in Max Pooling selects only one maximum value to focus on crucial information. This integrated design not only elevates network performance but also addresses gradient challenges, enhances generalization, and accelerates training, which provides a robust foundation for diverse tasks and datasets.

Following the multi-scale convolutional module, the output undergoes flattening and concatenation with spatial information to merge temporal and spatial features. This concatenated information is then processed through a Multi-Layer Perceptron (MLP) [116] for precise ToF estimation. The flattening operation transforms temporal features into a one-dimensional vector, enhancing the network's capacity for feature fusion. Algebraically, with flattened temporal features $\mathbf{X}_f$ and spatial features $\mathbf{X}_s$, the merged feature is expressed as:

$$\mathbf{X}_{\text{merged}} = \mathbf{X}_f \oplus \mathbf{X}_s \tag{3.7}$$

The subsequent Multi-Layer Perceptron (MLP) performs a nonlinear transformation:

$$\text{MLP}\left(\mathbf{X}_{\text{merged}}\right) = \sigma\left(\mathbf{W}_2\sigma\left(\mathbf{W}_1\mathbf{X}_{\text{merged}} + \mathbf{b}_1\right) + \mathbf{b}_2\right) \tag{3.8}$$

Here, σ represents the ReLU function, $\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1$ and $\mathbf{b}_2$ denote the weight matrices and bias vectors, respectively. This design leverages flattening and concatenation for effective feature fusion to enhance the efficiency of the designed network.

After constructing the architecture and initializing its parameters randomly, we employ SGD to iteratively update the parameters. Following each gradient descent step, the Adam optimizer, known for adaptive learning based on gradients, is applied to update all parameters to achieve swift convergence of results. The updating rule for each parameter is expressed as follows:

$$\theta \leftarrow \theta - \alpha \frac{\hat{\mathbf{m}}_t}{\sqrt{\hat{\mathbf{v}}_t} + \epsilon} \tag{3.9}$$

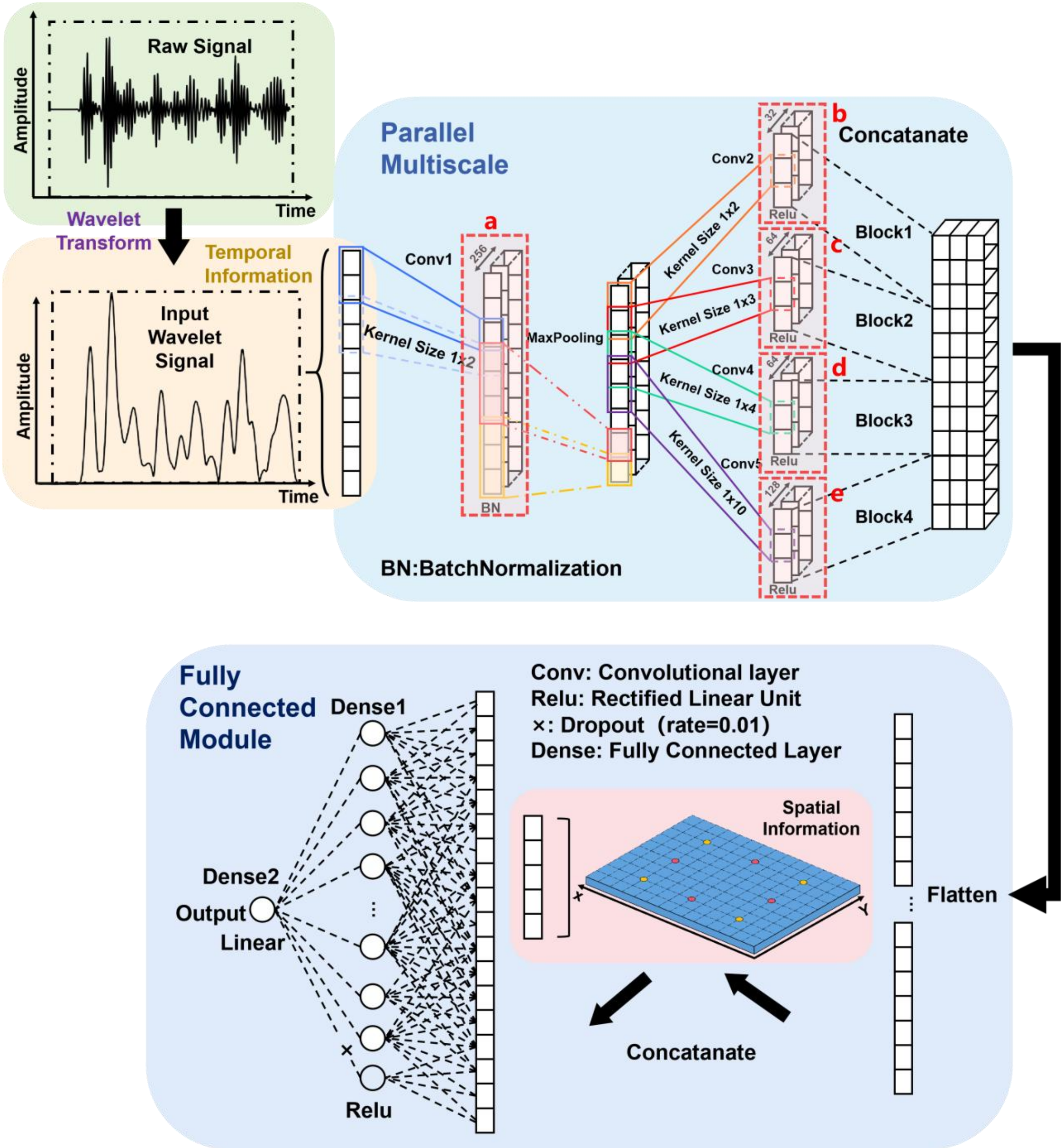


Figure 3.2 Architecture of the proposed spatio-temporal fusion with multi-scale parallel neural network.

Where α denotes the learning rate, and ϵ is a small constant introduced for numerical stability. Adam maintains two moving averages for each parameter θ:

$$\mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1)\mathbf{g}_t \tag{3.10}$$

$$\mathbf{v}_t = \beta_2 \cdot \mathbf{v}_{t-1} + (1 - \beta_2)\mathbf{g}_t^2 \tag{3.11}$$

where $\mathbf{m}_t$ is the first moment estimate, $\mathbf{v}_t$ is the second raw moment estimate, $\mathbf{g}_t$ is the gradient at time *t*, β_1 and β_2 are the exponential decay rates for the moment estimates. The bias-corrected estimates are calculated as:

$$\hat{\mathbf{m}}_t = \frac{\mathbf{m}_t}{1-\beta_1^t} \tag{3.12}$$

$$\hat{\mathbf{v}}_t = \frac{\mathbf{v}_t}{1-\beta_2^t} \tag{3.13}$$

Here, β_1^t and β_2^t represent the *t*-th power of β_1 and β_2, respectively. The adaptive learning rate in the Adam optimizer enables efficient adjustment of step sizes for various parameters, striking a balance between adaptability and stability throughout the optimization process.

In the training process, the parameters in the designed network undergo updates for multiple epochs until convergence. After that, the efficacy of the network is assessed through practical testing. This entails evaluating the network's performance on a dataset that was not part of the training data.

3.3 Experiment setup and analysis

3.3.1 Experiment setup

In this work, guided wave data are experimentally measured from a typical aluminum plate with a dimension of 500 mm × 700 mm × 2 mm. Eight PZT transducers were mounted on the surface of the plate by epoxy resin to actuate and receive Lamb waves. For ease of presentation, the plate is divided into 140 uniform regions, as shown in Figure 3.3. The positions for placing scatterers are chosen at the centers of every 4-unit region. Aiming at diversifying the datasets, 4 PZTs marked by the yellow points in Figure 3.4 are solely used for sensing, while those at the pink points serving as actuators and sensors as needed.

A pair of circular magnets is attached to the plate as the scatterer to simulate the damage. For the same scatterer, different excitation channels produce different boundary reflection paths as a way to increase the diversity of the datasets. Four excitation channels are selected in the present case, resulting in four data sets to assess the proposed method. Each signal in the data sets includes both direct waves and boundary reflected waves. The collected dataset, comprising a total of 924 signals, was partitioned into training and testing subsets, following an 80/20 ratio.

The experimental system works as follows, as shown in Figure 3.4: a KEYSIGHT 33500B waveform generator is used to output the excitation signal. The

PZT actuator is driven by an excitation signal that is first conditioned by a power amplifier. The sensor captures and records the propagating Lamb waves by the NI PXIe-5105 data acquisition module. The excitation signal is a Hann-windowed sinusoidal tone burst for all sets. The central frequency is selected with the following considerations: 1) the amplitude of response signals should be as high as possible; 2) the wavelength should be comparable to the size of the damage so that the damage can serve as a strong scatter; 3) the waves should be as clean as possible, i.e., the waves with fewer modes are preferable. Guided by these principles, the excitation frequency is selected as 160kHz. Examining the waves in the response signals, the A0 mode Lamb waves dominate the responses.

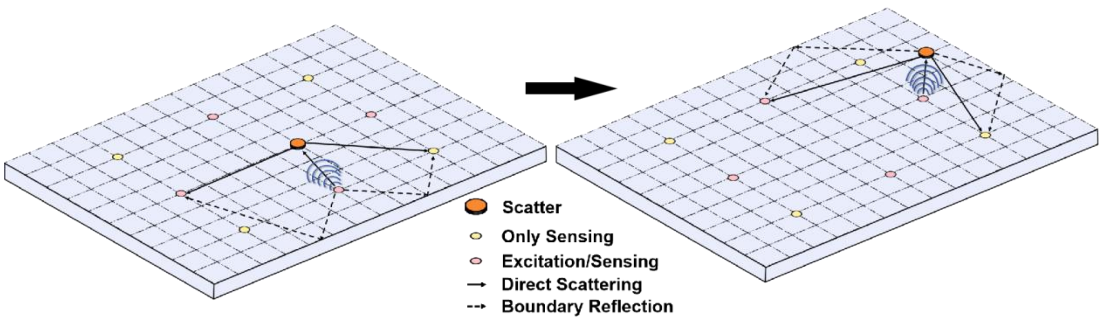


Figure 3.3 Schematic illustration of the data collection method.

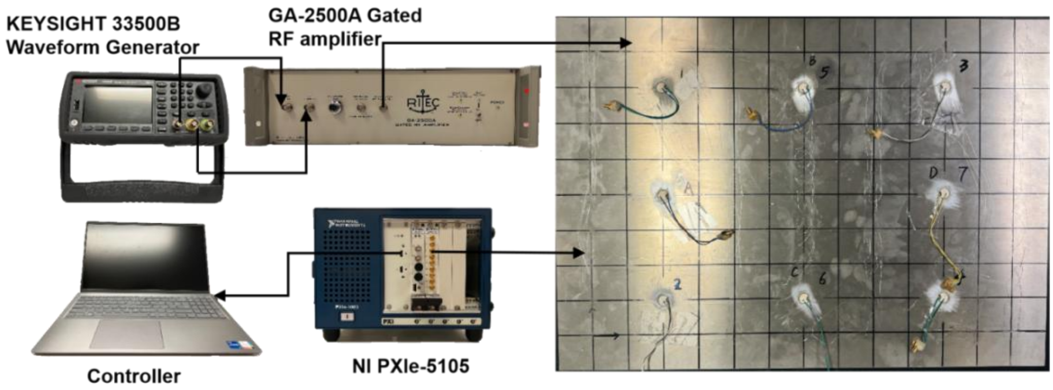


Figure 3.4 Experimental setup.

3.3.2 Performance evaluation

The Mean Squared Error (MSE) and the Mean Absolute Error (MAE) between the predicted ToF and the actual ToF associated with the scattered waves from damage are used to evaluate the performance of the designed network. The two indices are mathematically expressed as:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|\mathbf{y}^{(i)} - \hat{\mathbf{y}}^{(i)}\right| \tag{3.14}$$

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{y}^{(i)} - \hat{\mathbf{y}}^{(i)}\right)^2 \tag{3.15}$$

where N is the total number of samples, $\mathbf{y}^{(i)}$ is the actual arrival time, and $\hat{\mathbf{y}}^{(i)}$ is the predicted arrival time. Given that the predicted arrival time is measured in seconds (s), the units for MAE and MSE are seconds (s) and seconds squared (s^2), respectively.

Figure 3.5 illustrates the training and validation loss curves of the proposed model over 300 epochs. The training loss (blue line) decreases steadily, demonstrating that the model is learning effectively throughout the training process. The validation loss (red line) follows a similar trend, indicating that the model generalizes well to unseen data. Following an initial sharp decrease in losses, they start to stabilize around the 50th epoch, with the training loss consistently lower than the validation loss. This consistent behavior further validates the robustness and accuracy of our model.

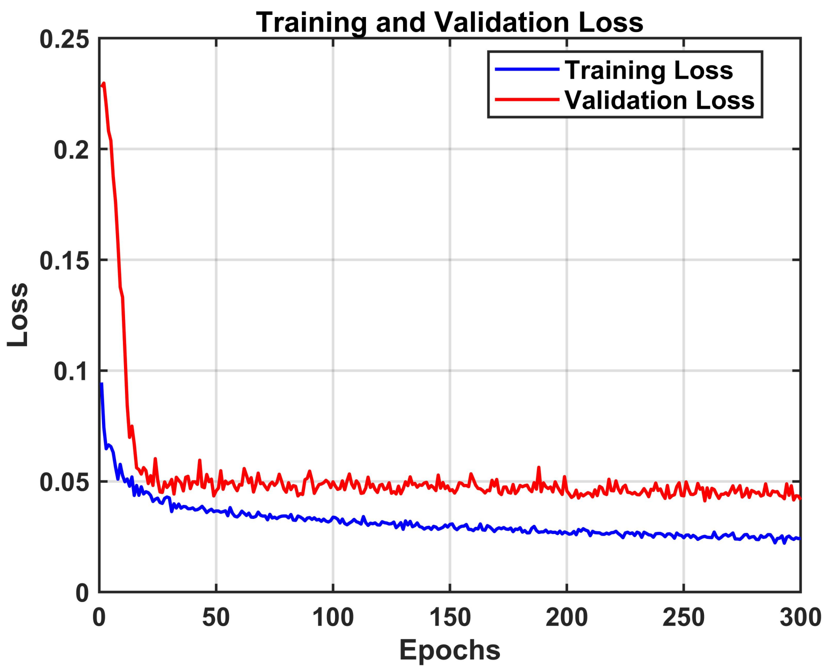


Figure 3.5 Training and validation loss curves.

Table 3.1 presents a comparative analysis of the performance of our proposed method with traditional machine learning and deep learning approaches. The comparison includes machine learning models such as Linear Regression [117], Ridge Regression [118], Lasso Regression [119], Decision Tree [120], Principal Component Analysis (PCA) [121], Support Vector Regression (SVR) [122], and Non-negative Matrix Factorization (NMF) [123], as well as several deep learning models LSTM [124], RNN [125], GRU [126], DNN [127], Transformer [128], DenseNet [129], ResNet18 [130], and AlexNet [131].

Our dual-input network processes both temporal and spatial information separately, while in the traditional machine learning models, temporal and spatial data are concatenated into a single input. In the deep learning comparisons, only the temporal part of our network was replaced, keeping the spatial component unchanged to better validate our network's performance.

The experimental results demonstrate that our proposed method achieves superior performance compared to both existing approaches. Specifically, our method yields an MAE that is 4-orders of magnitude lower and an MSE that is 7-orders of magnitude lower than those of traditional machine learning models, while also outperforming the deep learning models included in the comparison. This exceptional performance in both MAE and MSE metrics highlights the effectiveness of our proposed network in integrating temporal and spatial information, underscoring its significant accuracy advantage over existing methods.

Table 3.1 Comparison between the performance of the proposed method versus traditional machine learning approaches.

	Model	MAE	MSE
	Our Model	2.91981249932E-05	1.28721169000E-09
Machine Learning Methods	**LinearRegression**	4.32372880772E-01	5.06988352482E-01
	Ridge	1.53132763600E-01	3.72314677860E-02
	Lasso	1.57350022302E-01	3.73620651276E-02
	Decision Tree	1.63530463634E-01	4.77629257822E-02
	PCA+LinearRegression	1.56491938842E-01	3.72886161208E-02
	PCA+Ridge	1.56491938581E-01	3.72886159445E-02
	PCA+SVR	1.60217224315E-01	3.92246366963E-02
	NMF+LinearRegression	1.60135578253E-01	3.88855283405E-02
	NMF+Ridge	1.58633513046E-01	3.80224650692E-02
	NMF+SVR	1.58911557352E-01	3.86085520174E-02
Deep Learning Methods	**LSTM**	3.16880453326E-05	1.57238322832E-09
	RNN	3.14389894018E-05	1.52228323027E-09
	GRU	3.24486383443E-05	1.70872061698E-09
	DNN	3.13540027655E-05	1.45930177241E-09
	Transformer	3.27024178897E-05	1.67044803484E-09
	Densenet	3.18908274611E-05	1.54580940390E-09
	Resnet 18	3.18590071187E-05	1.58138232759E-09
	Alexnet	3.08384310074E-05	1.52713839646E-09

3.3.3 Evaluation of network input information

To validate the relevance of integrating STI and SSI into the network, we utilized t-distributed stochastic neighbor embedding (t-SNE) [132] to visualize high-dimensional data. T-SNE transforms data point similarities into joint probabilities, minimizing the Kullback-Leibler divergence between the joint probabilities of low-dimensional embedding and high-dimensional data. The definition of t-SNE is expressed as follows:

$$\sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{ij} log \left(\frac{p_{ij}}{q_{ij}}\right) \tag{3.16}$$

where P_i represents pairwise similarities in the original high-dimensional space, and Q_i denotes similarities in the lower-dimensional space. p_{ij} is the conditional probability that point i would pick point j as its neighbor in the original space, and q_{ij} is the conditional probability in the low-dimensional space. The KL divergence in t-SNE measures the variation in the similarity of data distributions in the low-dimensional space after dimensionality reduction with different input signal encoding forms. A more minor KL divergence indicates that data points are closer in the low-dimensional space, suggesting a more apparent similarity in the high-dimensional space.

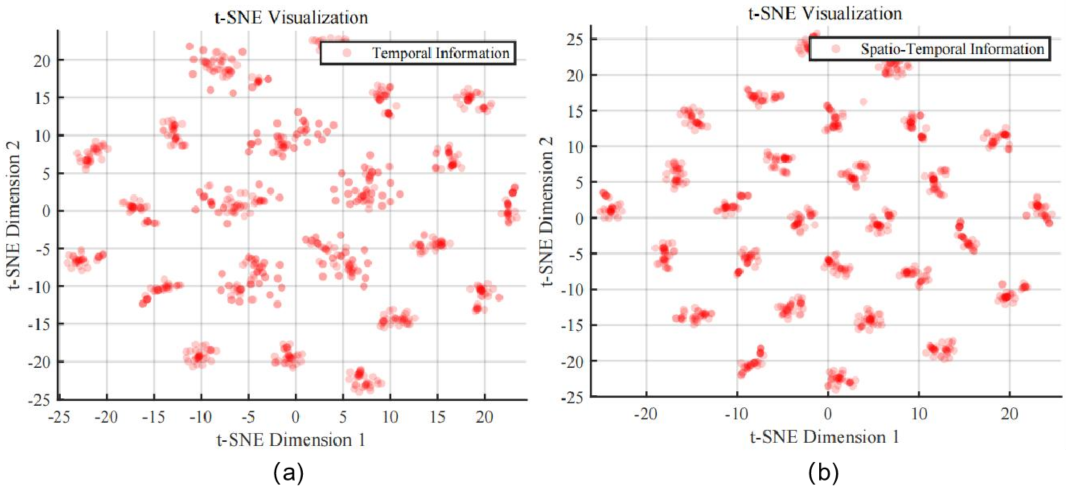
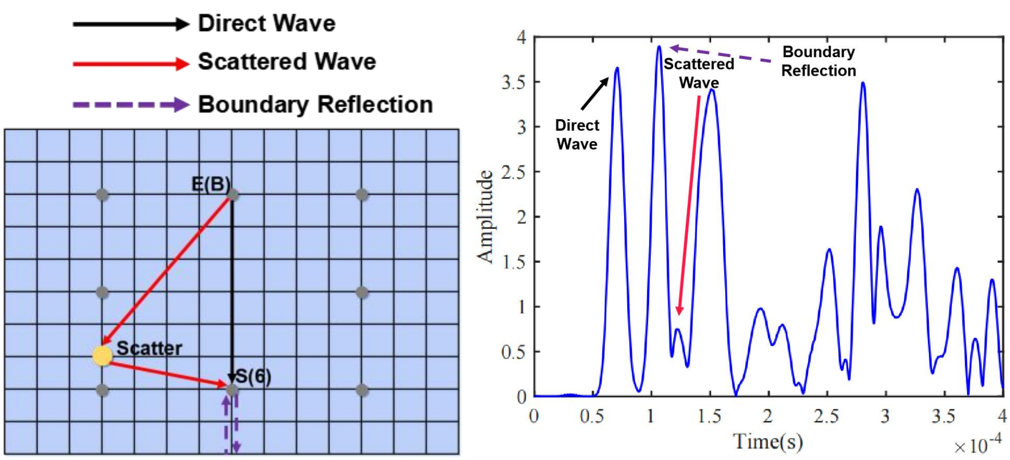


Figure 3.6 Illustration of the results obtained from the dimensionality reduction algorithm with (a) wavelet coefficients only and (b) both wavelet coefficients and sensor coordinates.

As shown in Figure 3.6, after reducing dimensionality, the outcomes derived solely from the STI display a distributed pattern. By contrast, when SSI is integrated, the dimensionality reduced results show a more concise envelope where temporal information surrounds spatial details. This indicates that this encoding approach effectively captures the reflection path of guided waves.

3.3.4 Interpretability analysis

To explain the reason for the high accuracy achieved by the proposed model, we select a typical input (in Figure 3.7) and track the signal flow through the model. The actual ToF of the scattered wave from the damage corresponds to the third peak, which would be very difficult to identify by any conventional means. The first peak attributes to the direct wave, and remaining peaks associate with boundary reflections. Through deconvolution [133] calculations on the input, we manage to illustrate the knowledge acquired by each network layer.



(a) (b)

Figure 3.7 A representative case to showcase the effect of the proposed method: (a) the system configuration and (b) the identified peaks in the wavelet coefficients.

Five cases, labeled as (a), (b), (c), (d), and (e) in Figure 3.8 correspond to the output of the convolution operations at different locations depicted in Figure 3.2. Here, we investigate four distinct patterns observed while applying a neural network model to eliminate boundary reflections in guided wave signals. These patterns signify varied outcomes: Effective Elimination: demonstrating successful removal of boundary reflections while retaining desired signal components; Unprocessed Signal: indicating no suppression of boundary reflections and an unaltered signal; Partial Suppression: illustrating partial mitigation of boundary reflections while preserving desired signal integrity; Misdirected Suppression: highlighting accidental suppression of desired signal components rather than boundary reflections. Understanding these modes is imperative for optimizing the model's performance in guided wave signal processing applications.

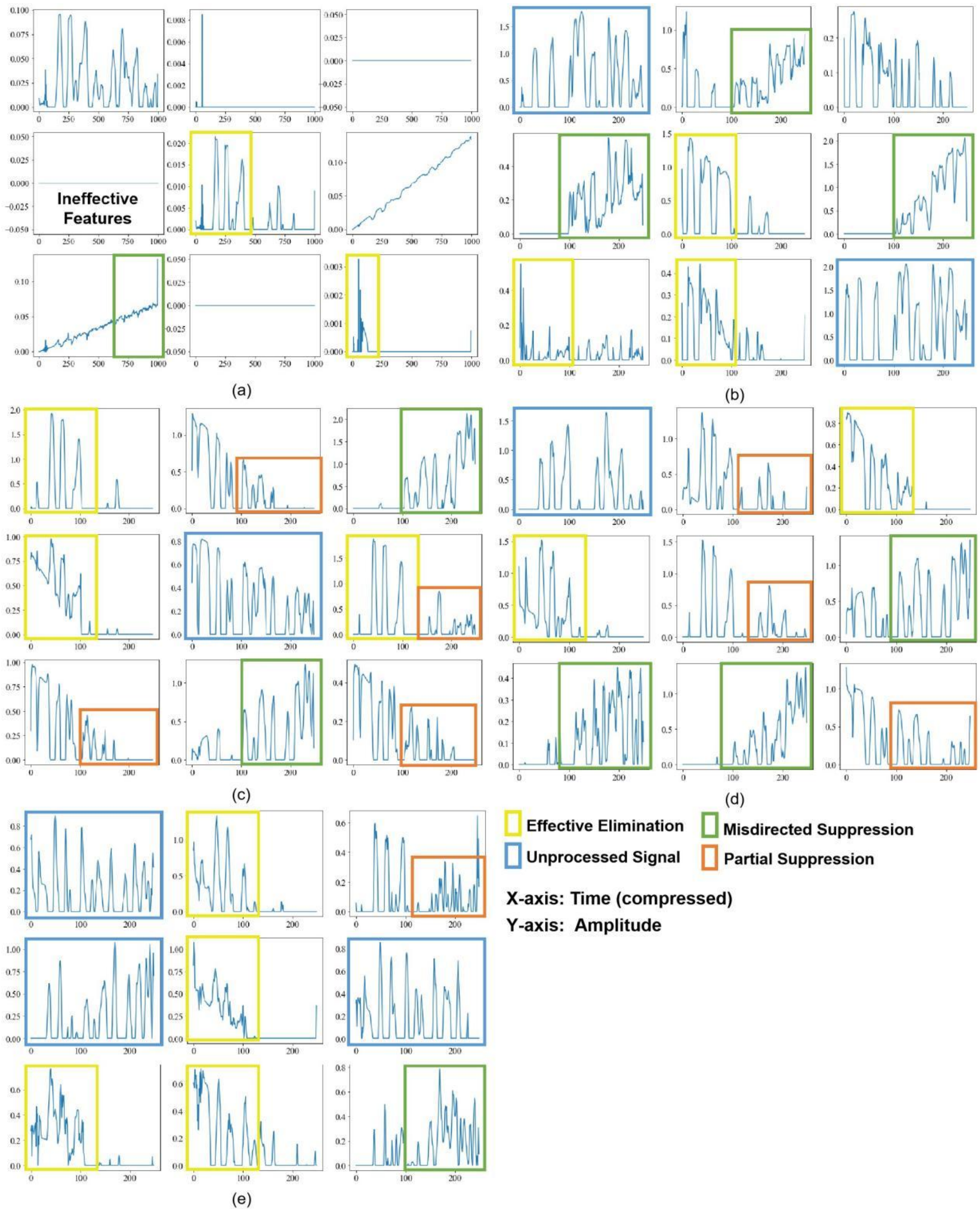


Figure 3.8 Results of the deconvolution operations at different locations in the proposed network to illustrate the effect of the multi-scale scheme for boundary reflection elimination: (a) after deconvolution following global convolution of the

input signal; (b) after deconvolution with a 2-point kernel; (c) after deconvolution with a 3-point kernel; (d) after deconvolution with a 4-point kernel; (e) after deconvolution with a 10-point kernel.

The visualization results in Figure 3.8 (a) reveal three notable observations: successful removal of boundary reflections, elimination of unsuccessful instances, and occurrences of ineffective learning. These visualizations underscore the importance of maintaining diversity in the network's learning content during the initial phase. Specifically, to validate the network's ability to eliminate boundary reflections, we performed a deconvolution on the results shown in Figure 3.8 (b). In this figure, the third wave peak represents the desired time of arrival, while the subsequent peaks are attributed to boundary reflections. Our primary focus is to determine whether the network effectively removes these unwanted peaks.

In Figure 3.8, the x-limits represents time, since the input signal comes from the wavelet transform processing in the time domain. However, the length of the time axis is shortened due to the compression effect of the convolution layers in the multiscale module, the limits of which are determined by the downsampling effect of these convolution operations. After deconvolution is performed to reconstruct the signal and reduce boundary reflections, the x-limits still represents time, although it is compressed compared to the original input.

In the multi-scale convolution module, the outcomes with a kernel size of 2 (illustrated in Figure 3.8 (b)) closely resemble those in Figure 3.8 (a). This similarity primarily arises from the small receptive field of the network when using a kernel size of 2, making it susceptible to local minima and excessive focus on detailed information. Upon increasing the kernel size to 3, as depicted in Figure 3.8 (c), the network notably improves its capability to suppress boundary reflections. Compared to the first two scenarios, there is a remarkable enhancement in accurately learning the ToF. This improvement primarily stems from the network's increasing focus on macroscopic signal information with the larger kernel size. Furthermore, when the kernel size is set to 4, the output of network closely resembles that with a size of 3, albeit with a slight increase in failure to eliminate reflections as shown in Figure 3.8 (d). This slight increase is mainly due to both scenarios maintaining a similar-sized receptive field. Subsequently, with a kernel size of 10, a significant improvement is observed in successfully eliminating boundary reflections, as shown in Figure 3.8 (e). This suggests the network prioritizes macroscopic waveform information with a larger kernel size, closely tied to the provided plate size and PZTs spatial coordinates.

In summary, multi-scale convolution substantially enhances the learning of signal features, thereby improving the method's accuracy in eliminating boundary reflections.

More essentially, the multi-scale module considers the mathematical model of neurons and the frequency-domain decay characteristics of activation functions, as shown in Figure 3.9. For a one-hidden layer ReLU-DNN fitting a 1-dimensional function *f*, the mathematical representation is:

$$h(\mathbf{x}) = \mathbf{a}_j \sigma(\mathbf{w}_j \mathbf{x} + \mathbf{b}_j) \tag{3.17}$$

where $\mathbf{w}_j$ and $\mathbf{a}_j$ denote the weights, $\mathbf{x}$ is the input, and $\mathbf{b}_j$ represents the bias, σ is the ReLU activation. The Fourier transform of the output $h(x)$ can be expressed as:

$$F(k) = \int_{-\infty}^{\infty} h(\mathbf{x}) e^{-ikx} dx \tag{3.18}$$

The frequency domain representation of a neural network mathematical model is mainly determined by the activation function. Specifically, a Fourier transform is applied to the $h(x)$ function:

$$F(k) =- \frac{\mathbf{a}_j \mathbf{w}_j}{k^2} - i \frac{\mathbf{b}_j}{k} \tag{3.19}$$

From the above equation, it is evident that as the frequency approaches infinity, the amplitude converges towards 0. Consequently, when influenced by the activation function, neurons exhibit a bias towards low frequencies, suggesting that the network prioritizes the acquisition of low-frequency information, which is referred to as the frequency principle [134].

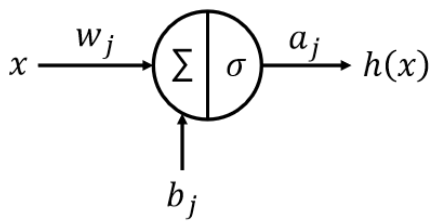


Figure 3.9 Illustration of the mathematical model of neurons.

As shown in Figure 3.10, the commonly used ReLU activation function becomes less effective in capturing high-frequency information due to exponential decay in the frequency domain during the learning process. This limitation restricts the network's ability to represent detailed features. In the frequency domain, ReLU attenuates high-frequency components through nonlinear clipping, which complicates the network’s capacity to learn higher-frequency details of the signal.

To address this limitation, multi-scale architectures have been introduced, incorporating layers with different receptive field sizes. This design enables the network to capture information across multiple scales, preserving high-frequency components while integrating them with low-frequency information. Our multi-scale convolutional network specifically employs convolutional kernels of sizes 2, 3, 4, and 10, each tuned to capture distinct frequency bands within the input signal. Kernels of sizes 2 and 3 are optimized for capturing high-frequency elements, such as rapid transitions and sharp variations. A kernel size of 4 achieves a balance between high- and mid-frequency features, capturing both fine details and moderate signal changes, while a kernel size of 10 targets low-frequency components, capturing broader trends and smoother variations in the signal.

The architecture achieves comprehensive learning by processing these features in parallel across various kernel sizes, followed by a gradual fusion across layers. This design maintains the integrity of low-frequency information while preventing the total suppression of high-frequency signals due to ReLU's attenuation. Moreover, by dynamically adjusting the significance of different frequency components during training, the multi-scale architecture further enhances the network's capacity to accurately represent complex signals.

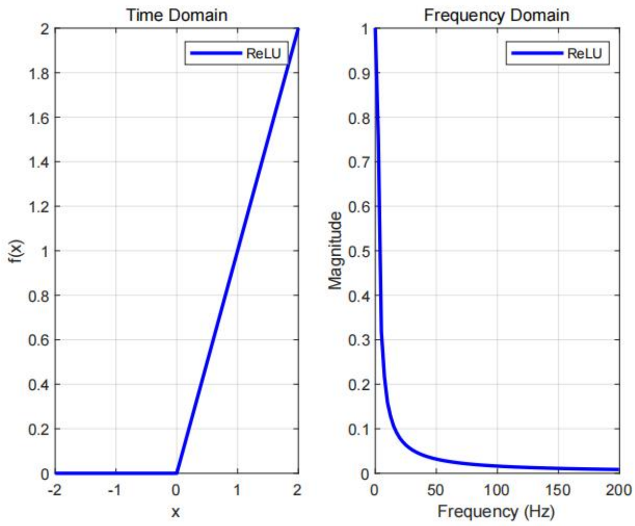


Figure 3.10 Form of the ReLU activation function in time and frequency domains.

3.3.5 Ablation experiment

To evaluate the effectiveness of our proposed multi-scale network, we conducted a series of ablation experiments summarized in Table 3.2. These experiments systematically compare different network variants at three scales (small, medium, and large), with and without spatio-temporal fusion.In the "Ablation 2x in Parallel Multiscale" experiment, we used convolutions with a kernel size of 2 across all scales to focus on small-scale features. Similarly, "Ablation 3x in Parallel Multiscale" and "Ablation 10x in Parallel Multiscale" employed kernel sizes of 3 and 10 for medium and large scales, respectively, allowing us to assess the impact of scale selection on model performance.We also conducted "Ablation w/o Multiscale" experiments to explore the effects of removing the multi-scale structure. For instance, "Ablation w/o Multiscale (Single 2x Conv)" utilized a single convolution layer with a kernel size of 2, while "Ablation w/o Spatial Info" assessed performance without integrating spatial information.By comparing these results to our multi-scale architecture, we can evaluate the contribution of multi-scale fusion for feature extraction and its effectiveness in processing complex data.

This detailed experimental approach allows for a thorough analysis of the contributions of each configuration, as well as a quantified assessment of the impact of spatio-temporal fusion on overall performance. The results of these controlled experiments unequivocally demonstrate the superior effectiveness of our multi-scale network. These findings highlight the crucial role of multi-scale fusion, which improves the network's ability to process information across various dimensions and underscores the importance of multi-scale integration in addressing complex tasks.

Table 3.2 Ablation study results.

Method	MAE	MSE
Ablation 2x in Parallel Multiscale	3.03963360E-05	1.35731420E-09
Ablation 3x in Parallel Multiscale	3.01363561E-05	1.36770098E-09
Ablation 4x in Parallel Multiscale	3.04813807E-05	1.46242259E-09
Ablation 10x in Parallel Multiscale	2.92590390E-05	1.30759970E-09
Ablation w/o Spatial Info	2.95890107E-05	1.32726064E-09
Ablation w/o Multiscale (Single 2x Conv)	3.08380499E-05	1.40681359E-09
Ablation w/o Multiscale (Single 3x	3.09936803E-05	1.45320109E-09

Conv)		
Ablation w/o Multiscale (Single 4x Conv)	3.07001596E-05	1.40353542E-09
Ablation w/o Multiscale (Single 10x Conv)	3.06420031E-05	1.39146826E-09
Our Model	2.91981250E-05	1.28721169E-09

3.4 Applications

3.4.1 Signal analysis and processing

To confirm the efficiency of our proposed network in eliminating boundary reflections and extracting the ToF of the scattered waves from damage, we place a scatterer near the boundary as a representative case, as shown in Figure 3.11. It is worth noting that the position of the scatterer has not been used in the previous training process. We collect and test the data from excitation E to receiver S. The black line indicates the direct wave, the red line associates with the path of the scattered wave, and the dashed lines illustrate the boundary reflections.

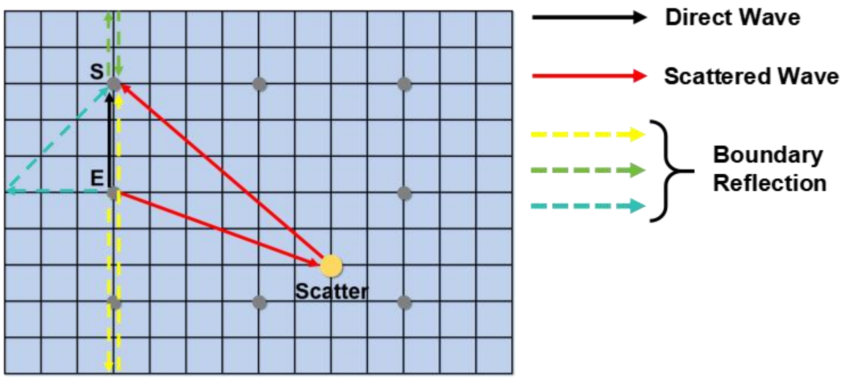
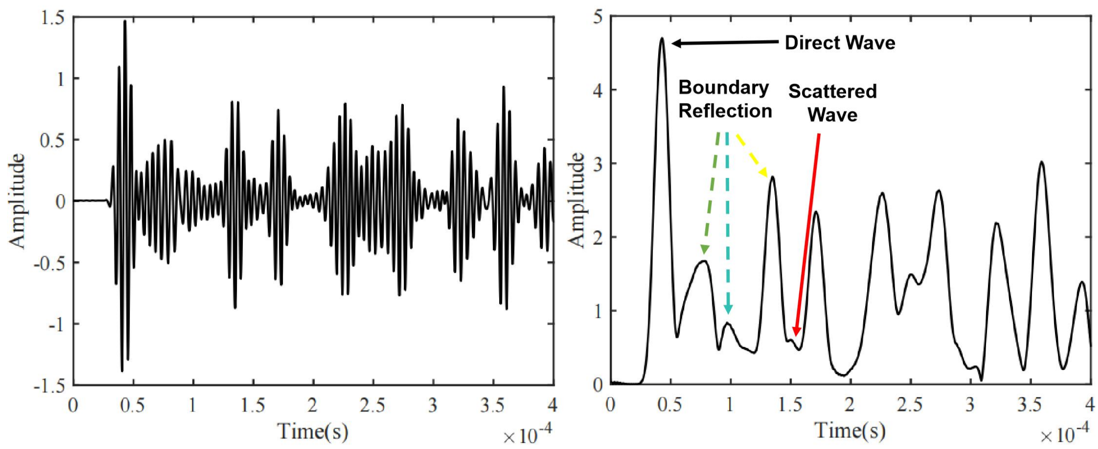


Figure 3.11 Wave path diagram.

The time-domain signal from the specific path is shown in Figure 3.12(a), followed by the corresponding wavelet coefficients in Figure 3.12(b). The first peak in wavelet coefficients is presumed to be the direct wave. In this specific case, the second peak corresponds to the boundary reflection rather than the scattered wave. Instead, the fifth peak corresponds to the scattered wave from damage. As to the output of the proposed method, the predicted ToF is 1.5e-04, which agrees well with the ground truth. This confirms that the designed network adeptly recognizes and accurately identifies the ToF of scatter waves despite their relatively low amplitude.

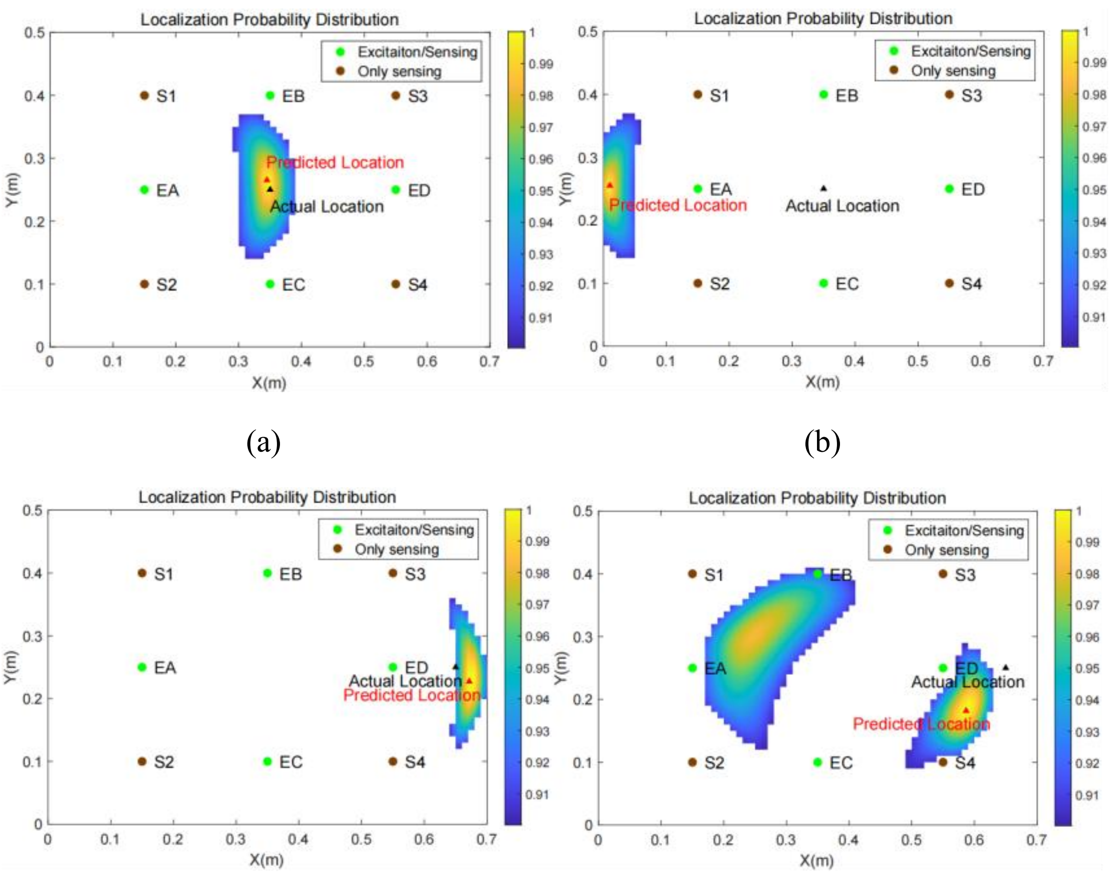


(a) (b)

Figure 3.12 (a) the raw signal (b) the identified peaks in the wavelet coefficients.

3.4.2 Localization results

With the extracted ToFs, damage localization can be achieved by the ellipse positioning method [135], resulting in the corresponding damage image. Two representative cases are selected to illustrate the efficacy of the method, as shown in Figure 14. In the first case, when damage is located near the center of the plate, the damage image is reconstructed with the ToFs extracted by the proposed method, as depicted in Figure 3.13 (a). It can be seen the accuracy of damage localization is very high with a localization error of merely 1.31 cm. By comparison, a traditional strategy is adopted by using the ToF of the second peaks in the wavelet coefficients. This should be fine when the structure is sufficiently large. However, in the present case, influenced by the boundary reflections, the accuracy suffers as shown in Figure 3.13 (b) with a localization error of 34.29 cm. In the second case, the scatter is placed very close to the structural boundary. The damage images reconstructed with the present method and traditional method are obtained in Figure 3.13 (c) and (d), respectively. It can be seen that the proposed method induces a localization error of 3.14 cm, whereas the traditional method incurs an error of 9.29 cm. These outcomes again underscore the efficacy of the proposed method in advancing damage localization.



(a) (b)

(c) (d)

Figure 3.13 Comparison of location results: (a), (c)-eliminating boundary reflection through the network; (b), (d)-second peak as the default arrival time.

3.5 Summary

In this chapter, we propose a multi-scale CNN with spatio-temporal fusion to accurately extract the ToF of scattered Lamb waves from damage while effectively mitigating the influence of boundary reflections. The effectiveness of our proposed network architecture is validated through experiments, in which we compare its performance with traditional machine learning algorithms. Additionally, we analyze the encoded representations of different network inputs using the t-distributed stochastic neighbor embedding (t-SNE) algorithm. Interpretability analyses are then conducted by tracing data flows through deconvolution operations and analyzing them from a frequency-domain perspective. Moreover, ablation experiments are executed to assess the impact of different scales on the results. Finally, testing is performed to evaluate the performance of our network in eliminating boundary reflections compared with manually identified arrival time localization results.

The proposed network achieves significantly lower MAE and MSE than traditional methods. Specifically, the MAE of our proposed method is four orders of magnitude lower than that of other existing methods, and the MSE is seven orders of magnitude lower, unequivocally demonstrating the superior accuracy of our proposed approach. Furthermore, the t-SNE results indicate that coupling temporal and spatial information as inputs to the network better assists in capturing the reflection paths of guided waves. The effectiveness of our network in eliminating unnecessary boundary reflections during intermediate processing stages is confirmed by deconvolution operations. Additionally, the designed multi-scale network avoids the neural network's bias towards low-frequency features, allowing for comprehensive learning of features at different scales. Ablation experiments demonstrate that combining four scales (2, 3, 4, 10) and the spatio-temporal fusion form currently yields the most effective results. Finally, by comparing the network-identified arrival times with manually identified ToF for damage localization, we find that the localization errors after boundary reflection elimination by the network are only 1.31 cm and 3.14 cm for two typical scenarios, as opposed to the respective errors of 34.29 cm and 9.29 cm after manual identification.

The developed CNN with spatio-temporal fusion and multi-scale properties leads to significant improvements in the accuracy of the ToF estimation of the scattered Lamb wave for damage localization. It can be used as an enabler to enhance the detection and localization capabilities of existing SHM methods based on ToF. Meanwhile, the study paves the way for understanding and explaining the internal working mechanism of deep learning models in such tasks.

Chapter 4 PhysRayGNN: Physics-Informed Ray Tracing Augmented Graph Neural Network for Structural Health Monitoring

4.1 Introduction

As outlined in Chapter 1, deriving ToF from wave signals presents a major challenge in sensing tasks, mainly because multipath disturbances hinder auurate signal decoding in real-world scenarios. Here, we focus on Lamb wave applications in SHM, where echoes from edges often mask faint damage signals, leading to unreliable evaluations of structural health status. Consequently, standard post-processing approaches—such as delay-and-sum imaging [136], time-reversal focusing [137, 138], and spatial wavenumber filtering [139, 140]—tend to struggle, as they leavearge inaccurate ToF as imput information.

The chapter 3 introduced a framework designed to eliminate the influence of boundary reflections, which are a primary source of interference in guided-wave-based SHM. While this approach is effective in isolating distinct wave packets, its efficacy diminishes when damage-scattered waves and boundary reflections overlap, arriving at the receiver as a single, composite wave packet. In such instances, the clean elimination of the boundary reflection component becomes challenging, as any attempt to do so risks distorting the target damage signal embedded within the mixed wave. This limitation necessitates a paradigm shift: rather than treating reflections solely as interference to be removed, we must adopt a higher-level perspective by explicitly modeling the entire physical process of wave propagation, interaction with boundaries and damage, and ultimately their superposition at the sensor.

This approach, focused on physics-based modeling, is naturally appropriate for strong implementation in GNNs. In the realm of SHM, it is possible to effectively depict a sensor network as a graph, with sensors acting as nodes and the diverse paths of wave propagation connecting them representing the edges. Unlike traditional neural networks, such as CNNs or RNNs, which are limited to processing data in Euclidean spaces (e.g., grids or sequences), GNNs are specifically designed to operate on irregular, non-Euclidean structures. Through their intrinsic message-passing mechanism, GNNs can directly simulate the dynamics of wave propagation, attenuation, and superposition across the sensor network. This capability enables the model's internal operations to reflect the underlying physics, facilitating a transition from "black-box" prediction to a more interpretable and physically consistent solution.

This effort parallels the recent emergence of Scientific Machine Learning (SciML), in which incorporating physical priors is recognized as an essential approach to addressing the constraints associated with exclusively data-driven techniques. As indicated in the preceding chapter, methods like PINNs have shown significant potential in SHM by incorporating the governing partial differential equations (PDEs) directly into the learning framework. [141-143]. For example, Pinello *et al*. [144] successfully employed PINNs to model guided wave propagation in aluminum plates, thereby enhancing damage diagnosis by integrating displacement data with PDE constraints. Similarly, Rathod *et al*. [145] introduced a PINN framework to identify material properties from wave measurements, effectively showcasing the potential of combining physical laws with sparse data. These advancements affirm that the incorporation of physical knowledge is a promising avenue for developing more robust and reliable SHM solutions.

This chapter introduces PhysRayGNN, a unified system that combines ray tracing with GNNs for precise ToF retrieval in Lamb wave SHM. It creates a physics-enhanced graph using simulated rays, considering attenuation, overlaps, and bounces, allowing the GNN to distinguish damage reflections from boundary noise. Notably, it does not require prior baselines and offers clear insights—with attention scores reflecting physical decay and feature changes resembling merges—the method performs well on tests with aluminum plates fitted with piezoelectric sensors, achieving localization errors below 0.25 cm despite difficulty in peak identification, outperforming conventional methods.

In Section 4.2, we describe the system's structure, encompassing ray enhancements and GNN elements. Section 4.3 covers the testing protocol and outcomes, and Section 4.4 examines constraints and potential advancements.

4.2 The physics-informed ray tracing-augmented graph neural network

The signal acquisition and preprocessing in PhysRayGNN follow the same process as outlined in Section 3.2 for the multi-scale spatio-temporal fusion network. In summary, PZTs are utilized in a pitch-catch setup to capture Lamb wave signals, while the Continuous Wavelet Transform (CWT) employs a complex Morlet mother wavelet to convert the raw data, allowing for the extraction of time-frequency characteristics that enhance damage signatures and minimize noise. For detailed equations and parameter adjustments, refer to Section 3.2. These processed features then serve as inputs to the ray-augmented graph, ensuring smooth integration with the physics-informed components.

The PhysRayGNN framework combines wavelet-transformed signal analysis with physics-informed ray tracing, providing a solid base for modeling wave dynamics. Inside the GNN, a multi-head attention mechanism highlights the most important wave propagation features across different paths, enabling accurate ToF extraction in complex structural environments. This mechanism adjusts weights dynamically to reflect wave attenuation patterns, emphasizing direct wave paths while also accounting for multi-path reflections. The multi-layer GNN further enhances these attention-weighted features, resulting in accurate predictions of the ToF for waves generated by damage.

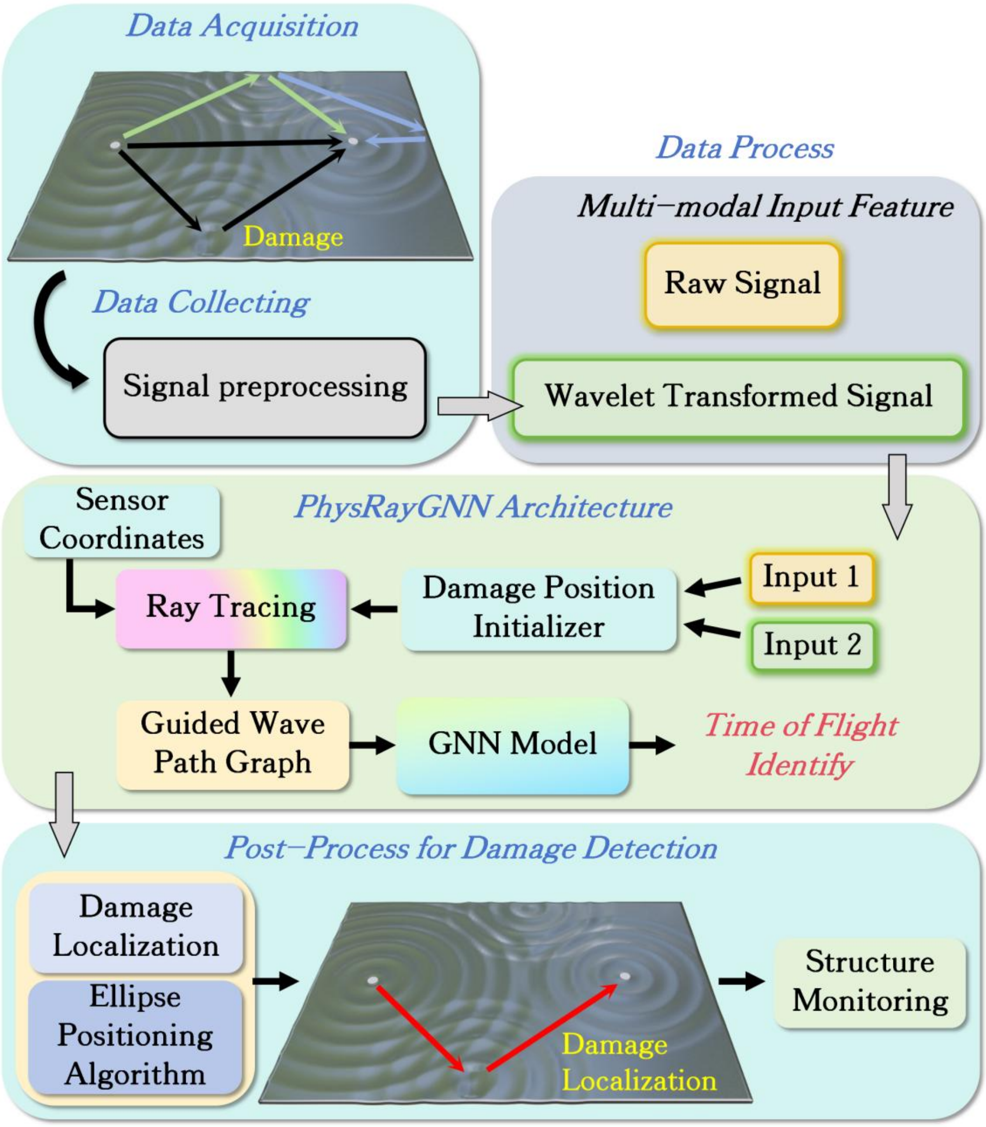


Figure 4.1 Flowchart of the proposed method.

Essentially, integrating wavelet-derived temporal features with ray tracing-enhanced graph representations allows the PhysRayGNN framework to effectively handle complexities in wave propagation, such as boundary reflections and scattering. This combination enhances the accuracy of ToF estimation for detecting damage in structures. The methodology's flowchart is shown in Figure 4.1.

Figure 4.1 illustrates the overall workflow of the proposed PhysRayGNN framework. The process begins with Data Acquisition and Pre-processing, where raw time-domain signals (Input 1) and their wavelet-transformed counterparts (Input 2) are prepared as multi-modal inputs. These features are initially fed into a Damage Position Initializer to estimate an initial damage location. Guided by this estimation and the known sensor coordinates, a physics-based Ray Tracing module constructs a Guided Wave Path Graph that encodes all physically plausible wave paths. This graph is subsequently processed by the core GNN Model, which simulates wave propagation and superposition to accurately identify the ToF. Finally, during the Post-Process stage, the predicted ToF is utilized in an Ellipse Positioning Algorithm to achieve final damage localization.

4.2.1 Overview of PhysRayGNN Architecture

The PhysRayGNN framework integrates physics-informed modeling and data-driven learning within a multi-module architecture tailored for SHM, enabling accurate prediction of ToF for damage-induced Lamb waves. Figure 4.2 illustrates that the process begins with the Wave Temporal Extractor. This extractor preprocesses both raw and wavelet-transformed signals through multi-scale convolution and adaptive fusion, resulting in compact features that are compatible with physical principles. These features are implicitly supervised by the Wave Signal Decoder, ensuring consistency with wave physics, although the high-dimensional reconstructed signals are not used in downstream tasks.

The extracted features are then sent to the Damage Position Initializer, which uses a multi-layer perceptron to generate a preliminary estimate of the damage location by combining temporal features with the positional data of actuators and receivers. This initial estimate guides the Ray Tracing Module, which creates a detailed graph structure by tracing wave propagation paths, allowing for up to two reflections. It is worth noting that the explicit dimensions of the plate are not directly used as inputs into the PhysRayGNN. Instead, the structural boundary information is implicitly captured and embedded within the ray tracing results, as the generated reflection paths are inherently determined by the plate's physical boundaries. The resulting graph, made up of nodes and edges, is processed through a three-layer GNN architecture to improve wave propagation features and predict the final ToF.

The associated training process, as depicted in Figure 4.2, adopts a physics-guided three-stage strategy. Stage 1 focuses on the supervised pre-training of the Event Position Initializer. Its objective is to minimize the discrepancy between the estimated preliminary event location and the actual ground truth location. Stage 2 involves physics-guided pre-training of the GNN using an "ideal topology." In this phase, the graph structures are constructed based on the true damage locations. The GNN is then optimized using a composite physics-informed loss function that incorporates not only the primary task loss but also auxiliary physical constraints, such as wave equation residuals. Finally, Stage 3 is the end-to-end fine-tuning using a "predicted topology." Here, the graph structure is built dynamically based on the locations predicted by the pre-trained initializer. The entire network is then optimized jointly to minimize the ultimate ToF prediction error. This method, explained further in Appendix A.1, uses ray tracing to enhance graph construction, providing a solid foundation for the module-specific designs discussed in the following subsections.

The training of PhysRayGNN is orchestrated through a hybrid loss function that balances data-driven and physics-informed objectives. The total loss is formulated as:

$$L_{total}=L_{data}+\lambda_1 L_{wave}+\lambda_2 L_{wavelet} \tag{4.1}$$

where L_{data} is the MSE between predicted and tactual ToF values, L_{wave} and

$L_{wavelet}$ are physics-informed losses, and λ_1 and λ_2 are weighting factors to balance these constraints. The Adam optimizer is employed, utilizing 80% of the dataset for training and 20% for validation, with loss weights fine-tuned across actuator configurations to ensure robust convergence.

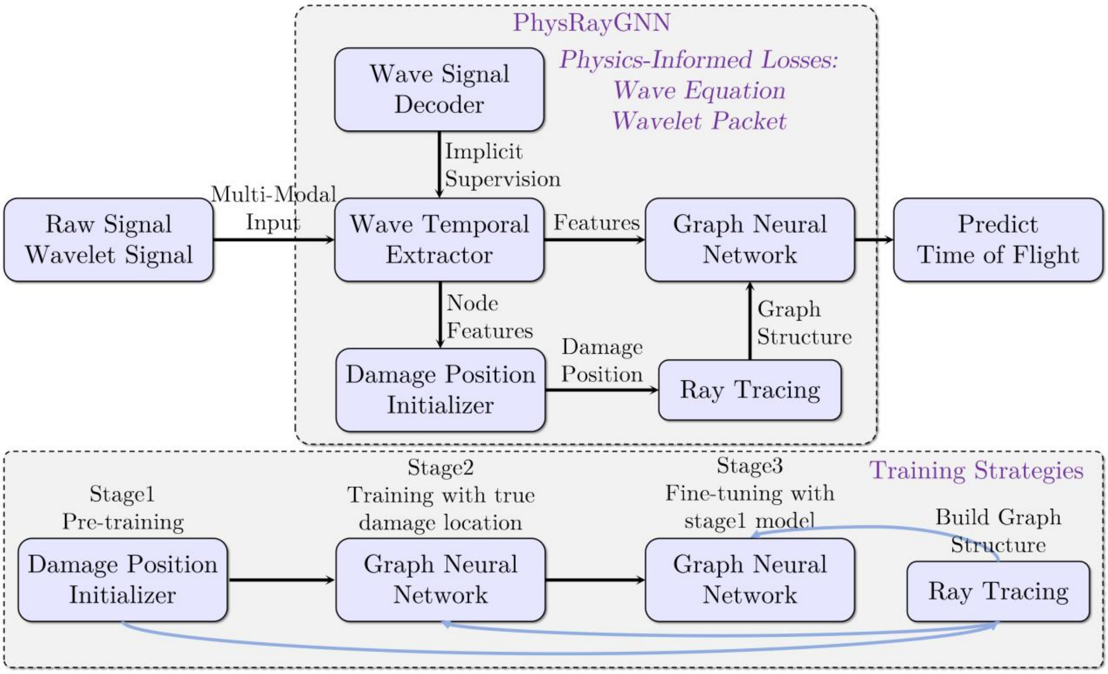


Figure 4.2 Flowchart of the proposed PhysRayGNN and training strategies.

4.2.2 Architecture of Wave Temporal Extractor

The Wave Temporal Extractor is a crucial initial module in the PhysRayGNN framework, engineered to distill salient temporal characteristics from multi-modal input signals. As illustrated in Figure 4.3, its architecture is founded on a multi-scale convolutional design. Let the raw timedomain signal be $\mathbf{s}_{raw} \in \mathrm{R}^{L}$ and the wavelet-transformed signal be $\mathbf{s}_{wavelet} \in \mathrm{R}^{L}$, where L is the number of time steps. These are concatenated to form the input tensor $\mathbf{X}_{in} \in \mathrm{R}^{2\times L}$.

To effectively capture wave dynamics across varying durations, the module employs three parallel branches, each characterized by a 1D convolutional kernel of a specific size. For each branch $k \in \{\text{short}, \text{mid}, \text{long}\}$, with a corresponding kernel size $k_{size} \in \{3,5,7\}$, the feature extraction is a two-stage process. First, an intermediate feature map $\mathbf{H}_k \in \mathbb{R}^{C\times L}$ is generated by applying a convolution followed by a ReLU activation [146]:

$$\mathbf{H}_k = \mathrm{ReLU}\big(\mathrm{Conv1D}_{\,k}(\mathbf{X}_{in})\big) \tag{4.2}$$

Here, C represents the number of hidden channels, and a channel-wise gating mechanism adaptivelt modulates these features by generating a gating signal $\mathbf{G}_k \in \mathrm{R}^{C\times L}$ through a 1D convolution with a kernel size of 1, followed by a sigmoed function σ:

$$\mathbf{G}_k = \sigma\big(\mathrm{Conv1D}_{gate,k}(\mathbf{H}_\mathrm{k})\big) \tag{4.3}$$

The final output of the branch, $\mathbf{F}_k \in \mathbb{R}^{C\times L}$, is obtained by performing an element-wise multiplication between the feature map and its corresponding gate. This step allows the model to selectively suppress or emphasize features at each time step and channel:

$$\mathbf{F}_\mathrm{k} = \mathbf{H}_\mathrm{k} \odot \mathbf{G}_\mathrm{k} \tag{4.4}$$

where $\odot$ represents the Hadamard product. The outputs from the multi-scale branches are then integrated through an adaptive fusion mechanism. This process computes a weighted sum of the feature maps, where the weights are dynamically learned during training. The fused feature map, $\mathbf{F}_{fused}$, is calculated as:

$$\mathbf{F}_{fused} = \sum_{k\in\{3,5,7\}} \alpha_k \cdot \mathbf{F}_k \tag{4.5}$$

where the fusion weights α_k are determined by applying a softmax function to a set of learnable scalar parameters w_k :

$$\alpha_k = \frac{e^{w_k}}{\sum_{j\in\{3,5,7\}} e^{w_j}} \tag{4.6}$$

This flexible weighting enables the model to focus on the most pertinent temporal scales related to a specific input signal, thereby improving the effectiveness of feature extraction.

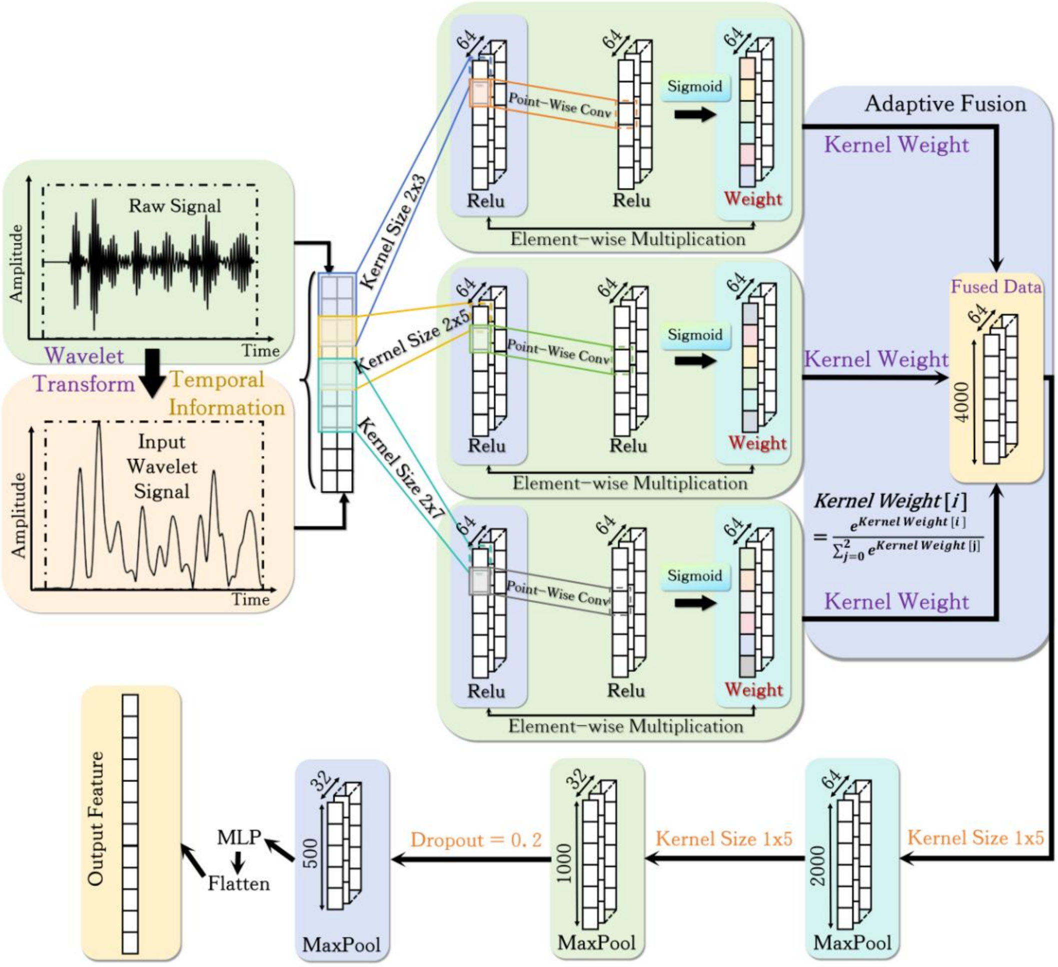


Figure 4.3 Architecture of the Wave Temporal Extractor.

Following the fusion, the feature map obtained goes through multiple convolutional and max-pooling layers, incorporating dropout to ensure regularization. This layered approach progressively diminishes the temporal dimension and expands the receptive field, ultimately converting the high-dimensional time-series data into a concise, low-dimensional feature vector. This final output, a dense 32-dimensional vector, captures the essential temporal information of wave propagation and functions as the initial node feature for the next Damage Position Initializer and GNN modules.

4.2.3 Architecture of Wave Signal Decoder

After the Wave Temporal Extractor produces a low-dimensional feature vector, the Wave Signal Decoder assumes a critical role in offering implicit, physics-informed supervision. Instead of directly influencing the final ToF prediction, the decoder's main role is to act as a regularizer, ensuring that the latent features learned by the extractor are physically meaningful and retain enough information to recreate the original wave dynamics. This auxiliary task encourages the extractor to produce representations that are not just correlational but are fundamentally based on wave propagation principles.

As depicted in Figure 4.4, the decoder accepts a dense 32-dimensional feature vector, denoted as $\mathbf{z} \in \mathbb{R}^{D_z}$, which is produced by the Wave Temporal Extractor. The architecture then bifurcates into two parallel, independent branches, each tasked with reconstructing one of the original input signal modalities from this compressed representation.

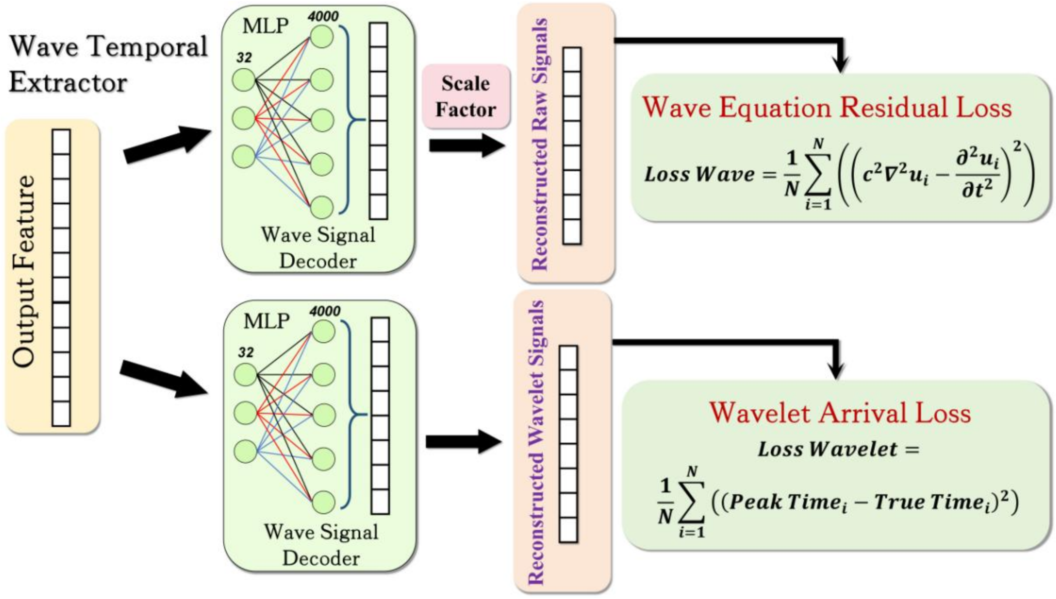


Figure 4.4 Architecture of the Wave Signal Decoder.

The first branch reconstructs the raw time-domain signal. This process is governed by a decoding function, g_{raw}, which is parameterized as a linear layer. The reconstructed raw signal, $\hat{\mathbf{s}}_{\text{raw}} \in \mathbb{R}^{L}$, is formulated as:

$$\hat{\mathbf{s}}_{\text{raw}} = \beta \cdot g_{raw}(\mathbf{z}) = \beta \cdot (\mathbf{W}_{raw}\mathbf{z} + \mathbf{b}_{raw}) \tag{4.7}$$

where $\mathbf{W}_{raw} \in \mathbb{R}^{L\times D_z}$ and $\mathbf{b}_{raw} \in \mathbb{R}^{L}$ represent the learnable weight matrix and bias vector of the raw signal decoder, respectively. An additional learnable scalar parameter, β, is introduced as a scale factor to provide adaptive control over the amplitude of the reconstructed signal, enhancing the model's flexibility.

The second branch is responsible for reconstructing the wavelet-transformed signal, $\hat{\mathbf{s}}_{wavelet} \in \mathbb{R}^L$. This is achieved through a similar decoding function, $g_{wavelet}$, with its own set of unique parameters:

$$\hat{\mathbf{s}}_{wavelet} = g_{wavelet}(\mathbf{z}) = \mathbf{W}_{wavelet}\mathbf{z} + \mathbf{b}_{wavelet} \tag{4.8}$$

where $\mathbf{W}_{wavelet} \in \mathbb{R}^{L \times D_z}$ and $\mathbf{b}_{wavelet} \in \mathbb{R}^L$ are the dedicated weight and bias parameters for the wavelet signal decoder.

Crucially, these reconstructed signals, $\hat{\mathbf{S}}_{raw}$ and $\hat{\mathbf{s}}_{wavelet}$ are not propagated to downstream modules like the GNN. Instead, they are channeled directly into two physics-informed loss functions.

The Wave Equation Residual Loss (L_{wave}) penalizes any deviation from the governing partial differential equations of wave motion. For a set of N receivers, the loss is defined as the mean squared residual of the 2D wave equation:

$$L_{wave} = \frac{1}{N}\sum_{i=1}^{N} \left\| c^2 \nabla^2 \hat{u}_i(t) - \frac{\partial^2 \hat{u}_i(t)}{\partial t^2} \right\|_2^2 \tag{4.9}$$

where $\hat{u}_i(t)$ is the reconstructed raw signal at the i-th receiver, and c is the wave speed. The temporal second derivative is approximated using the central finite difference scheme: $\frac{\partial^2 \hat{u}_i(t)}{\partial t^2} \approx \frac{\hat{u}_i(t+\Delta t) - 2\hat{u}_i(t) + \hat{u}_i(t-\Delta t)}{(\Delta t)^2}$. The spatial Laplacian $\nabla^2 \hat{u}_i(t)$ is approximated by summing the contributions from neighboring sensors $j: \nabla^2 \hat{u}_i(t) \approx \sum_{j \neq i} \frac{\hat{u}_j(t) - \hat{u}_i(t)}{\|p_j - p_i\|^2}$, where p_i and p_j are the spatial coordinates of the respective sensors. The central finite difference scheme employed in this study provides second-order accuracy, with a truncation error in the order of $O((\Delta t)^2)$. Given the sufficiently high sampling rate used in our data acquisition, the time step Δt is small enough to ensure that the approximation error remains negligible for the purpose of the physics-informed loss calculation.

Concurrently, the Wavelet Arrival Loss ($L_{wavelet}$) assess the accuracy of the primary peak's timing in the reconstructed wavelet signal. This loss is formulated as the MSE between the detected peak arrival time and the true ToF:

$$L_{wavelet} = \frac{1}{N} \sum_{i=1}^{N} \left(t_{peak,i} - t_{true,i}\right)^2 \tag{4.10}$$

where $t_{true,i}$ is the ground-truth ToF for the i-th receiver. The peak arrival time, $t_{peak,i}$, is determined by identifying the first time index t where the amplitude of the reconstructed wavelet signal $\hat{\mathbf{s}}_{wavelet,i}$ exceeds a dynamic threshold, typically defined as three times the signal's standard deviation:

$$t_{peak,i} = \min\left\{t \cdot \Delta t \middle| \hat{\mathbf{s}}_{wavelet,i}(t) > 3 \cdot \mathrm{std}\left(\hat{\mathbf{s}}_{wavelet,i}\right)\right\} \tag{4.11}$$

By optimizing these two physics-informed loss terms, the framework implicitly guides the Wave Temporal Extractor to produce a feature vector $\mathbf{z}$ that is compact and richly encodes the physical properties and critical temporal markers of the original high-dimensional signals.

4.2.4 Architecture of Damage Position Initializer

A reliable initial estimate of the damage location is imperative for the subsequent Ray Tracing module. The Damage Position Initializer does this by combining temporal features from each receiver node with their spatial geometry to predict the approximate coordinates.

As depicted in Figure 4.5, the module takes multi-source inputs. For a sensor network comprising N receivers, each receiver node i provides two types of information: (1) a low-dimensional temporal feature vector $\mathbf{z}_i \in \mathbb{R}^{D_z}$ extracted by the Wave Temporal Extractor, and (2) the physical coordinates of the receiver within the structure, $\mathbf{p}_i \in \mathbb{R}^2$.

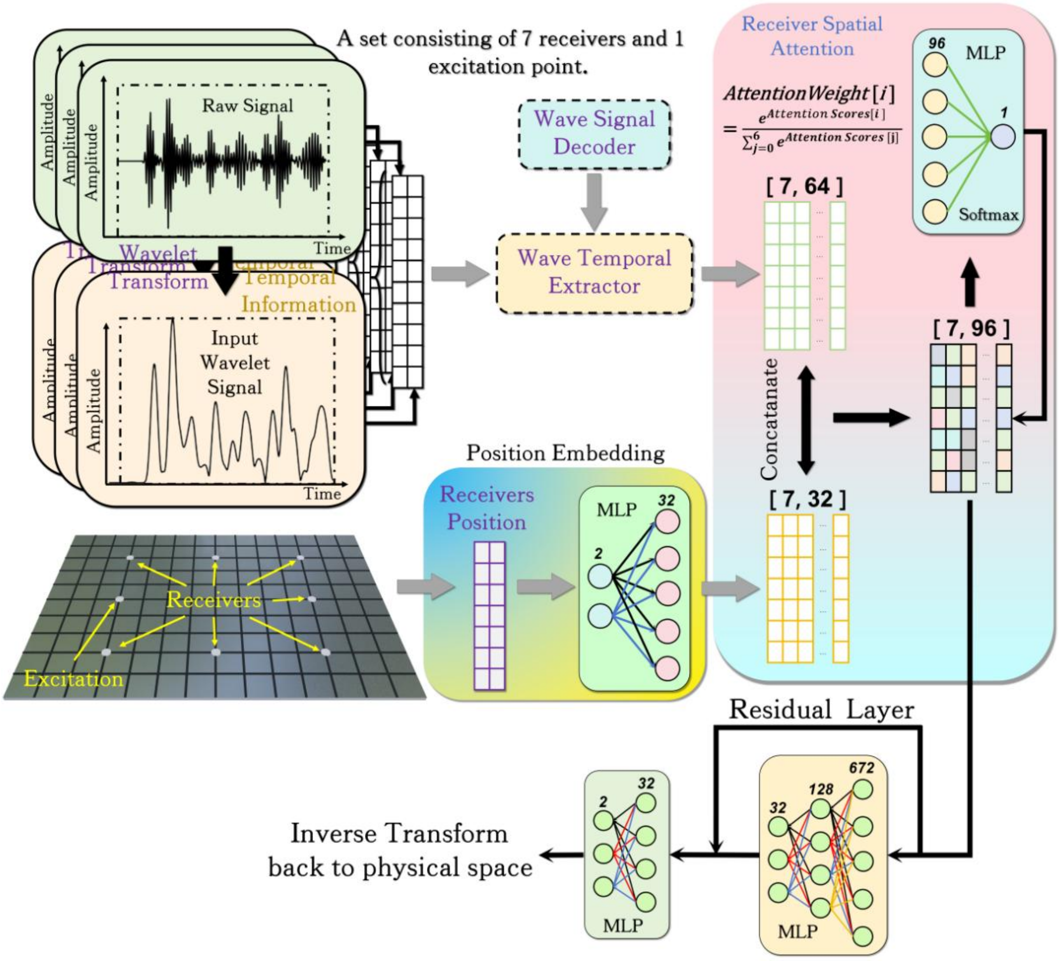


Figure 4.5 Architecture of the Damage Position Initializer.

To effectively integrate these two heterogeneous sources of information, the module first encodes and enhances them independently. The temporal feature vector $\mathbf{z}_i$ is processed through parallel multi-scale 1D convolutional layers to capture richer contextual information, yielding an enhanced temporal feature $\mathbf{z}_i^{'} \in \mathbb{R}^{D_{z'}}$. Concurrently, the 2D physical coordinates $\mathbf{p}_i$ are fed into a Multi-Layer Perceptron (MLP) for position embedding, mapping them into a higher-dimensional feature space to obtain a position embedding vector $\mathbf{e}_i \in \mathbb{R}^{D_e}$:

$$\mathbf{e}_i = \mathrm{MLP}_{pos}(\mathbf{p}_i) = \mathrm{ReLU}(\mathbf{W}_{pos}\mathbf{p}_i + \mathbf{b}_{pos}) \tag{4.12}$$

where $\mathbf{W}_{pos} \in \mathbb{R}^{D_e \times 2}$ and $\mathbf{b}_{pos} \in \mathbb{R}^{D_e}$ are the learnable parameters of the position embedding network.

Subsequently, a spatial attention mechanism is introduced to dynamically assess the contribution of each receiver's information to the final damage localization. The enhanced temporal feature $\mathbf{z}_i^{'}$ and the position embedding vector $\mathbf{e}_i$ are concatenated along the feature dimension to form a combined feature vector $\mathbf{c}_i = [\mathbf{z}_i^{'}; \mathbf{e}_i] \in \mathbb{R}^{D_{z}^{'}+D_e}$. This combined feature is then used to compute a scalar attention score s_i :

$$s_i = \mathbf{w}_{att}^T \tanh(\mathbf{W}_{att}\mathbf{c}_i) \tag{4.13}$$

where $\mathbf{W}_{att}$ and $\mathbf{w}_{att}$ are the learnable parameters of the attention network. The final attention weights α_i, which indicate the significance of the i-th receiver, are obtained by normalizing these scores through a Softmax function:

$$\alpha_i = \frac{\exp{(s_i)}}{\sum_{j=1}^{N} \exp{(s_j)}} \tag{4.14}$$

The information from all receivers is aggregated through attention-weighting. The combined feature vector c_i from each receiver is multiplied by its corresponding attention weight α_i, and all weighted feature vectors are then concatenated to form a global feature vector, $\mathbf{C}_{global} = [\alpha_1 c_1; \alpha_2 c_2; \dots; \alpha_N c_N]$.

Finally, this information-dense global vector is passed through a deep MLP network incorporating residual connections to predict the normalized damage coordinates $\hat{\mathbf{p}}_{damage} \in \mathbb{R}^2$. The use of residual architecture aids in alleviating the vanishing gradient issue encountered while training deep networks, thus improving their performance. This procedure can be expressed as:

$$\mathbf{h} = \mathrm{Dropout}\left(\mathrm{ReLU}\left(\mathrm{LN}\left(\mathrm{MLP}_1\left(\mathbf{C}_{global}\right)\right)\right)\right) \tag{4.15}$$

$$\hat{\mathbf{p}}_{damage} = \mathrm{MLP}_{final}\left(\mathbf{h} + \mathrm{MLP}_{res}\left(\mathbf{C}_{global}\right)\right) \tag{4.16}$$

where LN denotes Layer Normalization [147] and $\mathrm{MLP}_{\mathrm{res}}$ is a linear mapping in the residual connection. The predicted normalized coordinates $\hat{\mathbf{p}}_{damage}$ are ultimately mapped back to the physical coordinate system of the structure via an inverse scaling transformation, yielding the initial damage position $\mathbf{p}_{damage}$. These coordinates serve as the input for the Ray Tracing algorithm to construct a more physically accurate wave propagation path graph.

4.2.5 Architecture of ray tracing

The Ray Tracing module is the crucial link between the initial damage estimation and the sophisticated wave propagation modeling performed by the GNN. Its fundamental purpose is to translate the spatial configuration of sensors and the estimated damage location into a physically meaningful graph structure, $G = (V, \mathcal{E})$, which serves as the computational domain for the GNN. This process enriches the model with strong inductive biases derived from the principles of geometric acoustics.

The construction of the graph begins by defining the set of nodes, $\mathcal{V}$. This set comprises the excitation source (E), the estimated damage location ($\mathbf{p}_{damage}$) provided by the Damage Position Initializer, and the complete set of N receiver locations. These nodes represent all key points in the wave interaction field.

The core of the module is the generation of the edge set, $\mathcal{E}$, by systematically tracing physically plausible wave paths between any source node u at position $\mathbf{p}_u = [x_u, y_u]^{\mathrm{T}}$ and target node v at position $\mathbf{p}_v$. We employ the image source method to efficiently compute these paths, the detailed theoretical basis and mathematical derivation of which have been elaborated in Section 2.3.3.

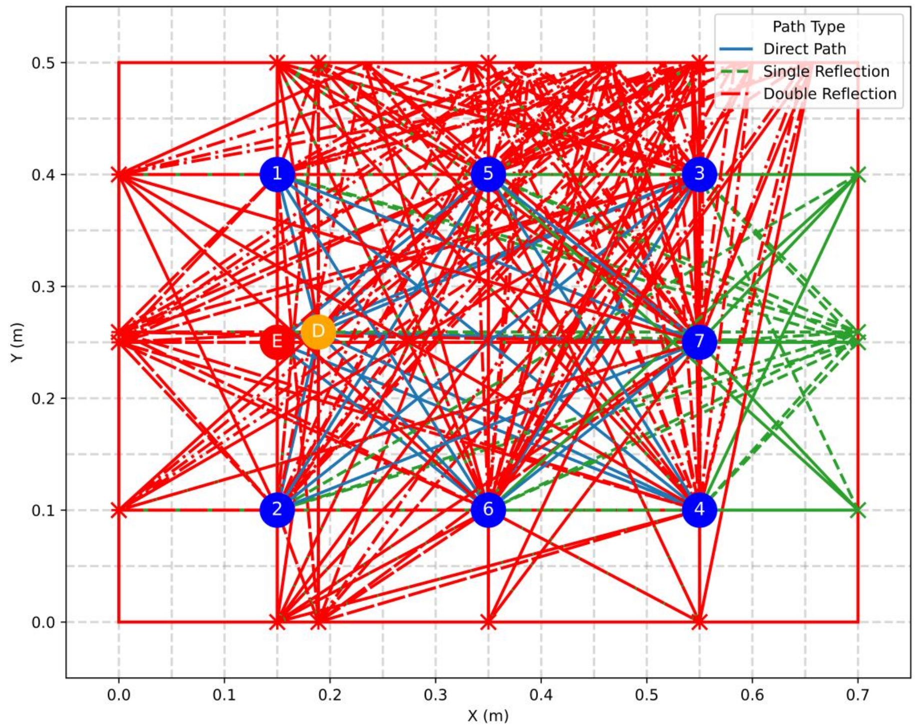


Figure 4.6 Graph structure based on ray tracing.

Based on this theory, we systematically identify several orders of wave propagation paths:

1. **Direct Path (0-order reflection)**: This path corresponds to the direct line-of-sight propagation between the source and the target.

2. **Single-Reflection Paths (1st-order reflection)**: The path is determined by mirroring the source node across a single plate boundary.

3. **Double-Reflection Paths (2nd-order reflection)**: The path is generated by mirroring a 1st-order image source across another, different boundary.

For each potential path of order $k \in \{0,1,2\}$, defined by a sequence of reflections

$\mathbf{B}_k = (b_1, \ldots, b_k)$, the total path distance $d_{uv}^{(k)}$ is the Euclidean distance from the corresponding image source to the target node.

A critical, physics-informed constraint is applied to prune the paths. A path is considered physically plausible and included in the final edge set only if its theoretical propagation time, $t_{uv}^{(k)} = d_{uv}^{(k)}/c$, falls within a predefined temporal window, T_{window}. The complete edge set $\mathcal{E}$ is thus the union of all plausible paths up to the second order:

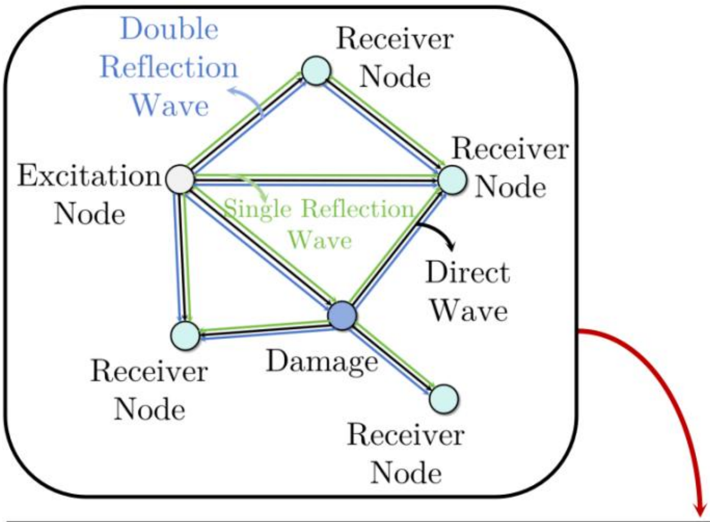
$$\mathcal{E} = \bigcup_{k=0}^{2} \left\{ (u, v) | \exists B_k, \frac{\left\| \mathbf{p}_v - \mathbf{p}_u^{(k)}(B_k) \right\|_2}{c} \leq T_{window} \right\} \tag{4.17}$$

The detailed implementation of this multi-reflection algorithm is given in Appendix *A.2*. The final output of this module is a dense, multi-relational graph, whose topology is determined by physical laws. As shown in Figure 4.6, this process produces a complex graph structure featuring many direct, single-reflection, and double-reflection paths, which represent the network of potential wave interactions.

For each edge $(u, v) \in \mathcal{E}$ corresponding to a specific path, a feature vector $\mathbf{e}_{uv} \in \mathbb{R}^2$ is created, encoding its physical properties:

$$\mathbf{e}_{uv} = \left[d_{uv}^{(k)}(B_k), t_{uv}^{(k)}(B_k) \right]^T \tag{4.18}$$

This graph, complete with node and edge features, acts as the direct input to the GNN. Figure 4.7 shows a specific example of this input format, listing the features for some edges, including their type, physical properties, and the initial features of their connected nodes. This organized, physically grounded foundation is vital for the later simulation of wave superposition and the final ToF prediction.



Edge Index Source→Target	Edge Type	Distance (m)	Propagation Time(s)	Source Node Features	Target Node Features
E→D	Direct	0.24	4.68e-5	[-0.01,…,0.01]	[0.00,…,0.00]
E→1	Direct	0.15	2.91e-5	[-0.01,…,0.01]	[0.02,…,0.00]
D→1	Direct	0.31	6.07e-5	[0.00,…,0.00]	[0.02,…,0.01]
D→1	Single Reflection	0.47	9.12e-5	[0.00,…,0.00]	[0.02,…,0.01]
D→1	Double Reflection	1.23	2.39e-4	[0.00,…,0.00]	[0.02,…,0.01]
E→D	Single Reflection	0.60	1.17e-4	[-0.01,…,0.01]	[0.00,…,0.00]

Figure 4.7 graph neural network input example.

4.2.6 Architecture of Graph Neural Network

The GNN, the core component of the PhysRayGNN framework, models the complex physics of wave propagation and superposition. It operates on the graph created by the physics-informed Ray Tracing module, using message-passing to simulate how wave features change across different paths.

The GNN takes as input a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the node set $\mathcal{V}$ includes the excitation source (E), the estimated damage location (D), and the N receivers. The edge set $\mathcal{E}$ represents all physically plausible wave propagation paths identified by the ray-tracing algorithm. Each node $v \in V$ is initialized with a feature vector $h_v^{(0)} \in \mathbb{R}^{D_z}$ from the Wave Temporal Extractor, while each edge $(u, v) \in \mathcal{E}$ is associated with a feature vector $\mathbf{e}_{uv} \in \mathbb{R}^2$ containing its physical distance and theoretical propagation time.

As illustrated in Figure 4.8, our GNN architecture is composed of a specialized three-layer stack, where each layer serves a distinct physical interpretation: a WaveProp Layer, a DirectWaveAttn Layer, and a WaveFusion Layer.

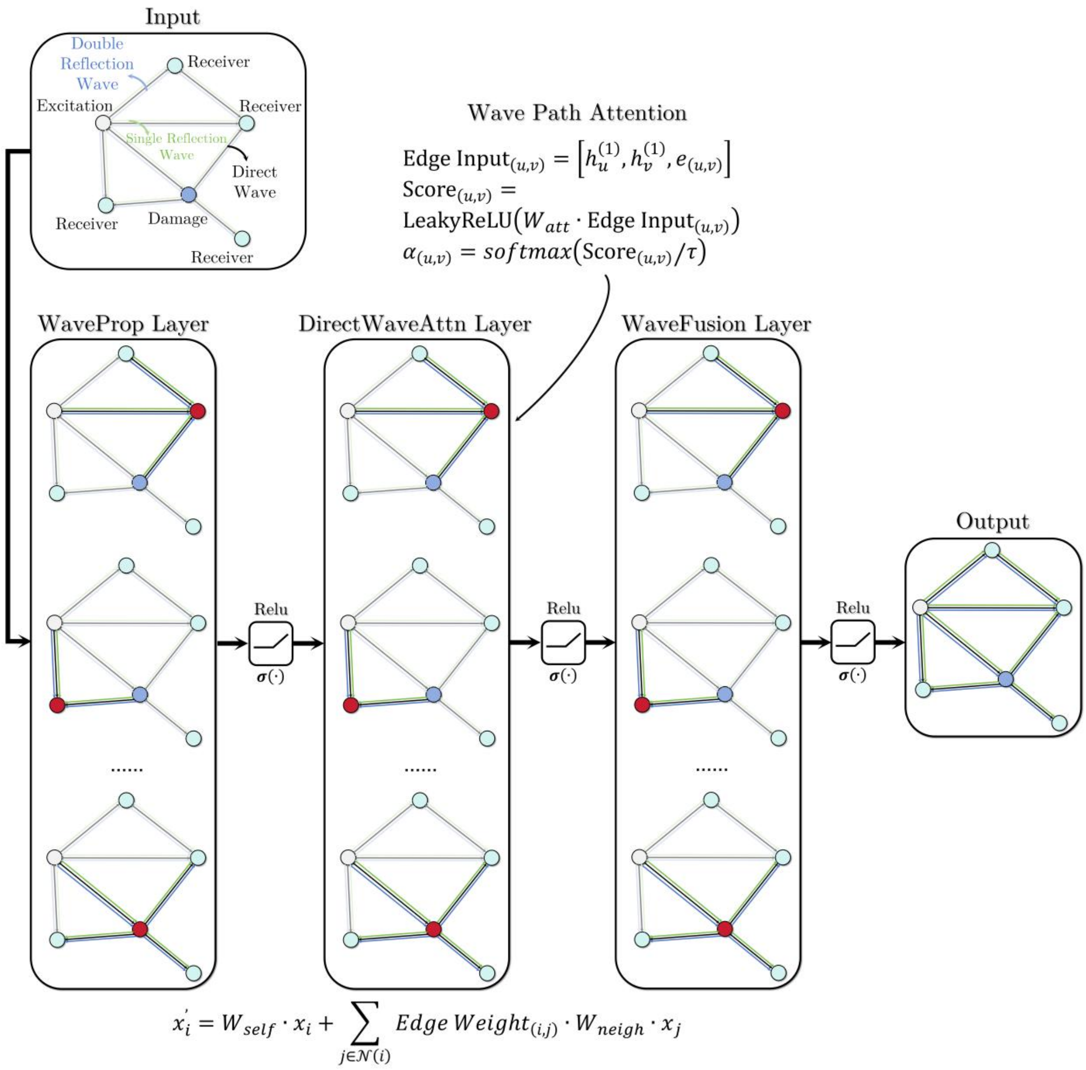


Figure 4.8 The structure of the proposed Graph Neural Network.

1. WaveProp Layer: The initial layer functions as a feature propagator, allowing initial node features to be shared among immediate neighbors. This step enriches the local information at each node without imposing any attentional bias, simulating the initial, omnidirectional dispersion of wave energy. The node representations are then updated as follows:

$$\mathbf{h}_v^{(1)}=\mathrm{ReLU}\left(\mathbf{W}_1^{(0)}\mathbf{h}_v^{(0)}+\sum\nolimits_{\mathrm{u}\in\mathrm{N}(\mathrm{v})}\mathbf{W}_2^{(0)}\mathbf{h}_u^{(0)}\right) \tag{4.19}$$

where $\mathbf{h}_v^{(1)}$ is the updated feature vector for node v, $N(v)$ is the set of neighbors of node v, and $\mathbf{W}_1^{(0)}$ and $\mathbf{W}_2^{(0)}$ is the learnable weight matrix of the first graph convolution layer.

2. DirectWaveAttn Layer: The second layer is the most important, introducing a multi-head attention mechanism to model wave attenuation and path significance. This layer dynamically calculates attention weights for each edge, enabling the network to focus more on influential wave paths rather than less significant ones.

For each edge (u,v), an input vector is formed by concatenating the features of the source node $\mathbf{h}_u^{(1)}$, the target node $\mathbf{h}_v^{(1)}$, and the edge itself $\mathbf{e}_{uv}$, which includes physical attributes such as path distance and reflection type to incorporate wave propagation characteristics. For each of the K attention heads, a scalar attention score $s_{uv}^{(k)}$ is computed:

$$s_{uv}^{(k)}=\mathrm{LeakyReLU}\left(\mathbf{a}^{(k)T}\left[\mathbf{W}_{att}\mathbf{h}_u^{(1)};\mathbf{W}_{att}\mathbf{h}_v^{(1)};\mathbf{W}_{edge}\,\mathbf{e}_{uv}\right]\right) \tag{4.20}$$

where $\mathbf{W}_{att}$ and $\mathbf{W}_{edge}$ represent adjustable weight matrices for both node and edge characteristics, enabling the model to grasp different facets of wave states and path attributes, and $\mathbf{a}^{(k)}$ is the weight vector for the k-th attention head. This multi-head design enables the mechanism to focus on diverse wave propagation facets, such as direct paths versus reflections, enhancing robustness in modeling complex superposition effects in Lamb wave dynamics. These scores are then normalized across all edges in the graph using a temperature-scaled softmax function to obtain the attention weights:

$$\alpha_{uv}^{(k)} = \frac{\exp\left(s_{uv}^{(k)}/\tau\right)}{\sum_{(i,j)\in\mathcal{E}} \exp\left(s_{ij}^{(k)}/\tau\right)} \tag{4.21}$$

where τ is a temperature hyperparameter that controls the sharpness of the distribution, promoting selective emphasis on physically significant paths while mitigating noise from less relevant reflections. The final attention weight α_{uv} for the edge is the average of the weights from all heads:

$$\alpha_{uv} = \frac{1}{K}\sum_{k=1}^{K} \alpha_{uv}^{(k)} \tag{4.22}$$

This averaging fuses multi-perspective insights, simulating the cumulative interference of waves from multiple propagation routes. These attention weights are then used as dynamic edge weights in the subsequent graph convolution layers to update the node features, effectively modulating the flow of information based on path importance and aligning the model's message passing with physical principles of wave attenuation, reflection, and superposition:

$$\mathbf{h}_v^{(2)} = \mathrm{ReLU}\left(\mathbf{W}_1^{(1)}\mathbf{h}_v^{(1)} + \sum\nolimits_{u\in\mathcal{N}(v)} \alpha_{uv}\cdot\mathbf{W}_2^{(1)}\mathbf{h}_u^{(1)}\right) \tag{4.23}$$

3. WaveFusion Layer: The final layer combines the attention-refined features to generate the ultimate node representations. It reuses the attention weights α_{uv} computed in the previous layer, applying a final graph convolution. This enables the model to consolidate information from the most important paths and complete the simulation of wave superposition at each node.

$$\mathbf{h}_v^{(3)} = \mathrm{ReLU}\left(\mathbf{W}_1^{(2)}\mathbf{h}_v^{(2)} + \sum\nolimits_{u\in\mathcal{N}(v)} \alpha_{uv}\cdot\mathbf{W}_2^{(2)}\mathbf{h}_u^{(2)}\right) \tag{4.24}$$

Finally, for each receiver node $v \in \{\text{ Receivers }\}$, the refined feature vector $\mathbf{h}_v^{(3)}$ is passed through a linear output layer to predict the final ToF value, $\hat{t}_v$:

$$\hat{t}_v = \mathbf{w}_{\mathrm{out}}^T\mathbf{h}_v^{(3)} + b_{\mathrm{out}} \tag{4.25}$$

This multi-stage, attention-driven architecture allows the GNN to develop a detailed model of wave physics, resulting in reliable and precise ToF predictions even when faced with complex boundary reflections and signal superposition.

4.3 Experiment setup and analysis

4.3.1 Experiment setup

The experimental setup for data acquisition in this study is identical to that described in chapter 3. The experiments were performed on a 500 mm × 700 mm × 2 mm aluminum plate, on which eight PZTs were mounted using epoxy resin to actuate and sense Lamb waves. To simulate structural defects, a pair of circular magnets was attached to the plate surface at 35 different locations, which were selected from the centers of predefined grid regions as shown in Figure 4.9. Ultimately, four datasets were collected, each comprising 231 samples, resulting in a total of 924 samples.

The data acquisition system, illustrated in Figure 4.10, employs the same hardware configuration described in Chapter 3. In this chapter's experiments, the excitation signal was altered to a sinusoidal tone burst featuring a five-peak Hann window, centered at a frequency of 200 kHz. This frequency was intentionally chosen to primarily activate the S0 Lamb wave mode, thus providing a relatively clear signal appropriate for further analysis.

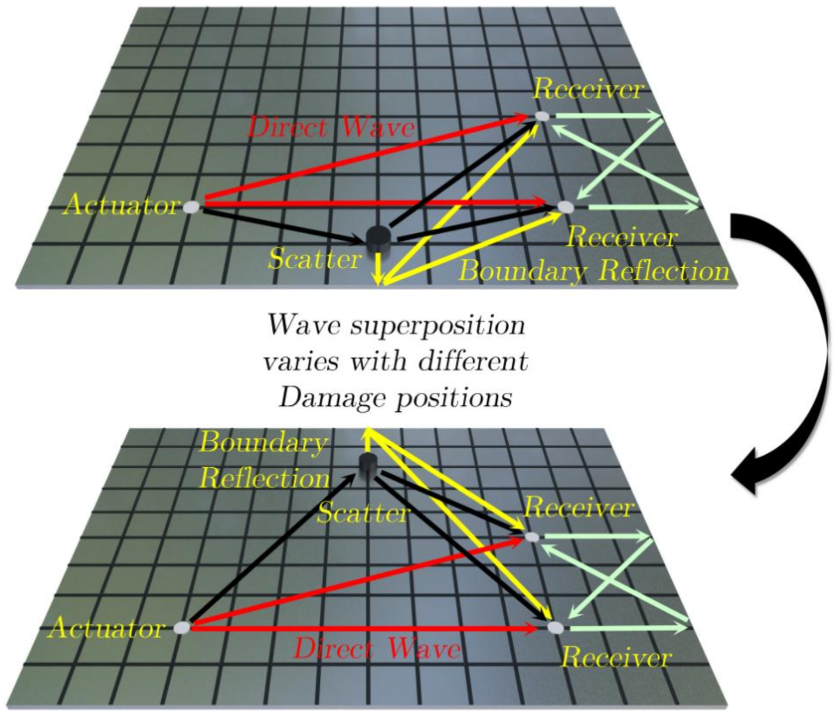


Figure 4.9 Schematic illustration of the data collection method.

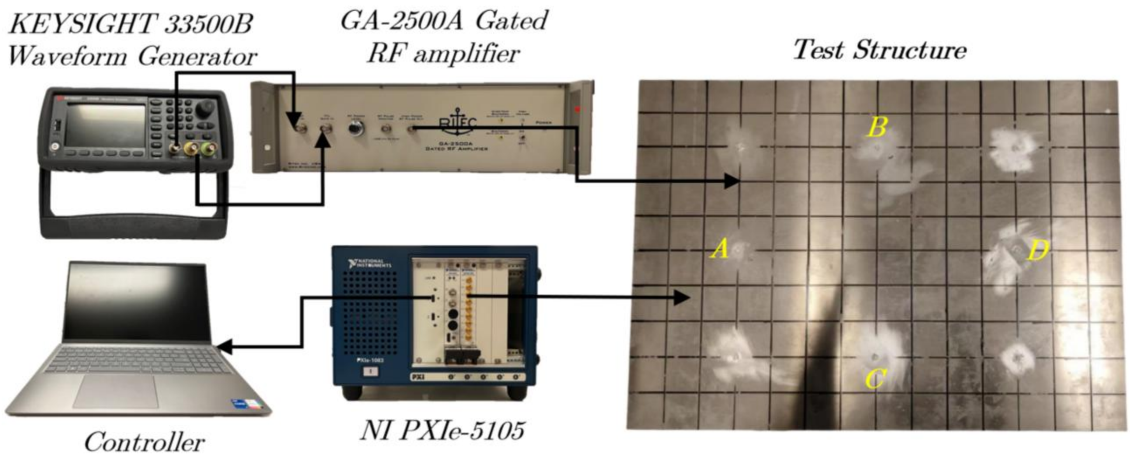


Figure 4.10 Experimental setup.

A key distinction in this study is how the dataset is organized to support the

GNN-based method. Although the total volume of data remains consistent with Chapter 3, it is partitioned into four independent datasets. Each dataset corresponds to a single PZT actuator (labeled A, B, C, and D), with the other seven PZTs acting as receivers. This single-actuator-per-dataset approach was strategically chosen for three reasons. First, it reduces the complexity of the graph structures generated by the ray-tracing algorithm, thereby improving the interpretability of the GNN's message-passing process. Second, training separate models on these smaller subsets serves to demonstrate the data efficiency of the proposed method, proving that high-performance reconstruction can be achieved even with a limited number of samples. Finally, this independent partitioning facilitates a more rigorous evaluation of the model's generalization capabilities across different excitation channels. For each dataset, the signals from the 35 damage locations are randomly split into a training set (80%) and a testing set (20%) to assess performance.

4.3.2 Performance evaluation

To assess the performance of the proposed PhysRayGNN framework in quantitative terms, the main metrics employed are the MSE and the MAE that compare the predicted values of ToF with the actual values. These metrics are defined mathematically as follows:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} \left|\hat{t}_i - t_i\right| \tag{4.26}$$

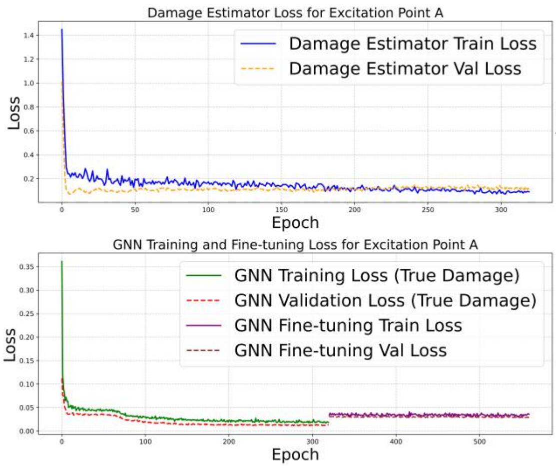
$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N} (\hat{t}_i - t_i)^2 \tag{4.27}$$

where N is the total number of test samples, $\hat{t}_i$ is the predicted ToF, and t_i is the ground-truth ToF for the i-th sample. Given that the predicted arrival time is measured in seconds (s), the units for MAE and MSE are seconds (s) and seconds squared (s^2), respectively.

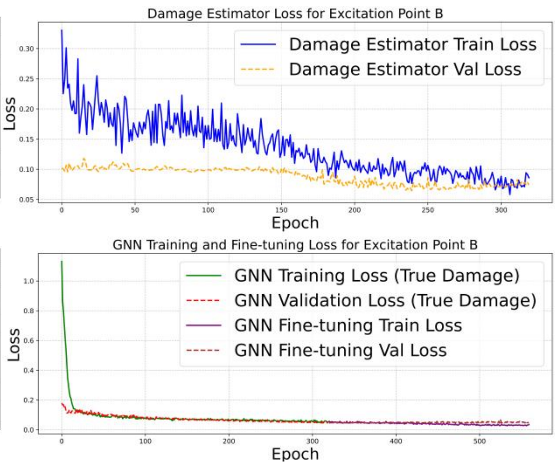
Figure 4.11 illustrates the training and validation loss curves for the PhysRayGNN model, which was trained separately on the four datasets associated with excitation points A, B, C, and D. For each dataset, the model undergoes a two-phase training process: an initial 320 epochs of pre-training where the GNN uses graphs built from true damage locations, followed by 240 epochs of fine-tuning where the graphs are constructed using the locations predicted by the Damage Position Initializer. The loss curves across all four datasets display a consistent and desirable trend: a rapid initial decrease followed by stable convergence. This suggests that the model is capable of learning efficiently and generalizing effectively to new data without succumbing to overfitting, irrespective of the position of the actuator. The steady performance across various excitation conditions reinforces the strength and dependability of the training strategy and model framework we have proposed.

The phenomenon where the validation loss is lower than the training loss is primarily attributed to the use of regularization techniques, specifically Dropout, within the network architecture. During the training phase, a fraction of neurons is randomly deactivated to prevent overfitting, which artificially reduces the model's capacity and introduces noise, leading to a higher training loss. In contrast, during the validation phase, the Dropout mechanism is disabled, and the full network capacity is utilized for inference. Consequently, the model acts as a more robust predictor during validation, resulting in a lower loss compared to the training phase where stochastic noise is present.

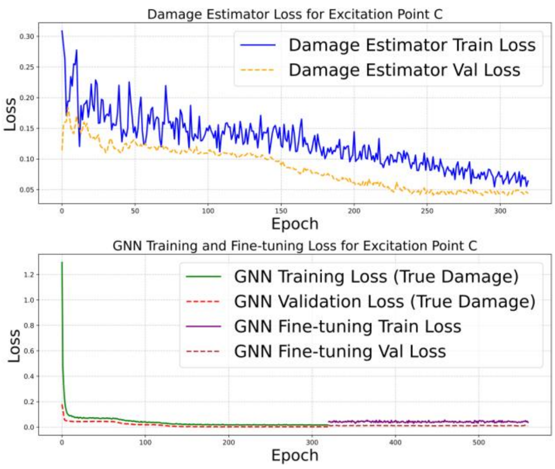
To assess the effectiveness of PhysRayGNN more thoroughly, a detailed comparative analysis was conducted involving various well-established deep learning architectures. For this comparison, the dataset for actuator A was used as a representative case. The baseline models include standard architectures such as MultiLayer Perceptron (MLP) [148], Convolutional Neural Networks (CNN) [84], Recurrent Neural Networks (RNN) [125], Long Short-Term Memory (LSTM) [124], and Gated Recurrent Units (GRU) [126]; advanced models like the Transformer [149]; and prominent vision models including AlexNet [131], ResNet-18 [130], ResNet-50 [150], VGG-16 [151], EfficientNet variants [152], DenseNet-121 [129], and SENET [153]. Additionally, we compare against the Multi-Scale Spatio-Temporal (MSST) network from our previous work [154] and other GNN variants, specifically Graph Attention Network (GAT) [155], GraphSAGE [156], and Graph Isomorphism Network (GIN) [157].



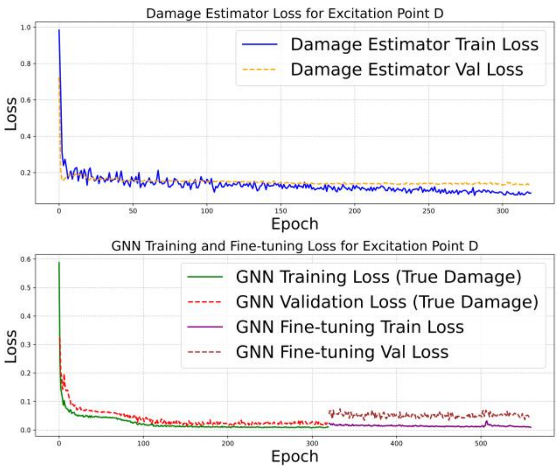
(a)



(b)



(c)



(d)

Figure 4.11 Training and validation loss curves for different group datasets: (a) Loss curves for the Actuator A dataset; (b) Loss curves for the Actuator B dataset; (c) Loss curves for the Actuator C dataset; (d) Loss curves for the Actuator D dataset.

Table 4.1 Comparison between the performance of the proposed method and other deep learning approaches.

Model	MAE	MSE
MLP	3.02735e-05	1.48724e-09
CNN	3.11769e-05	1.40534e-09
RNN	8.38325e-05	8.44980e-09
LSTM	3.36578e-05	2.01512e-09
GRU	8.38325e-05	8.44980e-09
Transformer	3.23843e-05	1.85367e-09
Alexnet	3.27833e-05	1.64637e-09
ResNet 18	3.98662e-05	2.54137e-09
ResNet 50	3.70094e-05	2.29130e-09
VGG-16	3.03568e-05	1.42207e-09
EfficientNetB0	3.70108e-05	1.93124e-09
EfficientNetB1	3.30990e-05	1.67093e-09
EfficientNetB3	3.14421e-05	1.60657e-09
DenseNet-121	6.16296e-05	6.80215e-09
SENET	3.05399e-05	1.33582e-09
MSST	3.46841e-05	1.90008e-09
GAT	3.41960e-05	1.58261e-09
GraphSAGE	2.65268e-05	1.01314e-09
GIN	3.89535e-05	2.03581e-09
Our Model (Actuator A)	1.98797e-05	5.96621E-10
Our Model (Actuator B)	2.28243E-05	7.94933E-10
Our Model (Actuator C)	1.51140E-05	3.82528E-10
Our Model (Actuator D)	2.47743E-05	9.93623E-10

As shown in Table 4.1, our proposed model significantly outperforms all other methods. Notably, PhysRayGNN achieves the lowest MAE and MSE across all four actuator-specific datasets. This superior performance highlights the effectiveness of our physics-informed approach, where integrating ray tracing to build a physically meaningful graph, combined with the specialized multi-layer GNN, offers a clear advantage in accurately modeling the complex wave propagation phenomena and predicting the ToF of damage-scattered waves.

4.3.3 Analysis of wave superposition processes in networks

Aside from attaining elevated prediction accuracy, a primary goal of the PhysRayGNN framework is to guarantee that its internal processes are understandable and consistent with the principles of wave propagation. To provide a clear and interpretable analysis, we focus on a representative test case from the dataset involving actuator A, with the scatterer located at (0.15 m, 0.35 m). The geometric setup of this specific case is shown in Figure 4.12. The analysis below examines the internal workings of the GNN for this scenario to demonstrate that the network learns a physically consistent model of wave interactions.

To visualize how information is aggregated and fused across the network, we first analyze the node feature representations and their intercorrelations at each of the three GNN layers. Figure 4.13 displays heatmaps of the normalized node features, alongside their corresponding Pearson correlation matrices. The correlation coefficient r between the feature vectors of two nodes, $\mathrm{h_i}$ and $\mathrm{h_j}$, is calculated as:

$$r(\mathbf{h}_i, \mathbf{h}_j) = \frac{\sum_{k=1}^{D_z} (h_{ik} - \bar{h}_i)(h_{jk} - \bar{h}_j)}{\sqrt{\sum_{k=1}^{D_z} (h_{ik} - \bar{h}_i)^2} \sqrt{\sum_{k=1}^{D_z} (h_{jk} - \bar{h}_j)^2}} \tag{4.28}$$

where h_{ik} is the k-th component of the feature vector $\mathbf{h}_i$, and $\bar{h}_i$ is its mean. The feature representations evolve distinctly through the layers: in Layer 1, features are sparse and correlations are weak; by Layer 3, the features are highly activated, and the correlation matrix reveals a strong positive correlation among all receiver nodes. This high correlation signifies the successful simulation of wave superposition, where the final feature representation of each receiver is a rich aggregation of information propagated from all other relevant nodes.

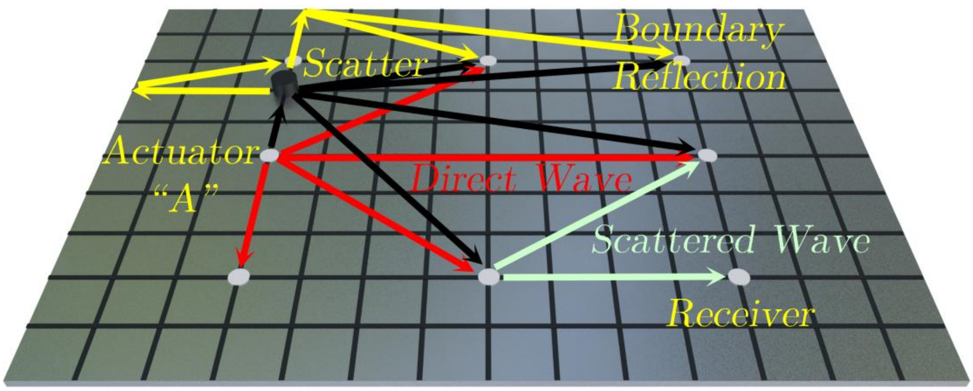
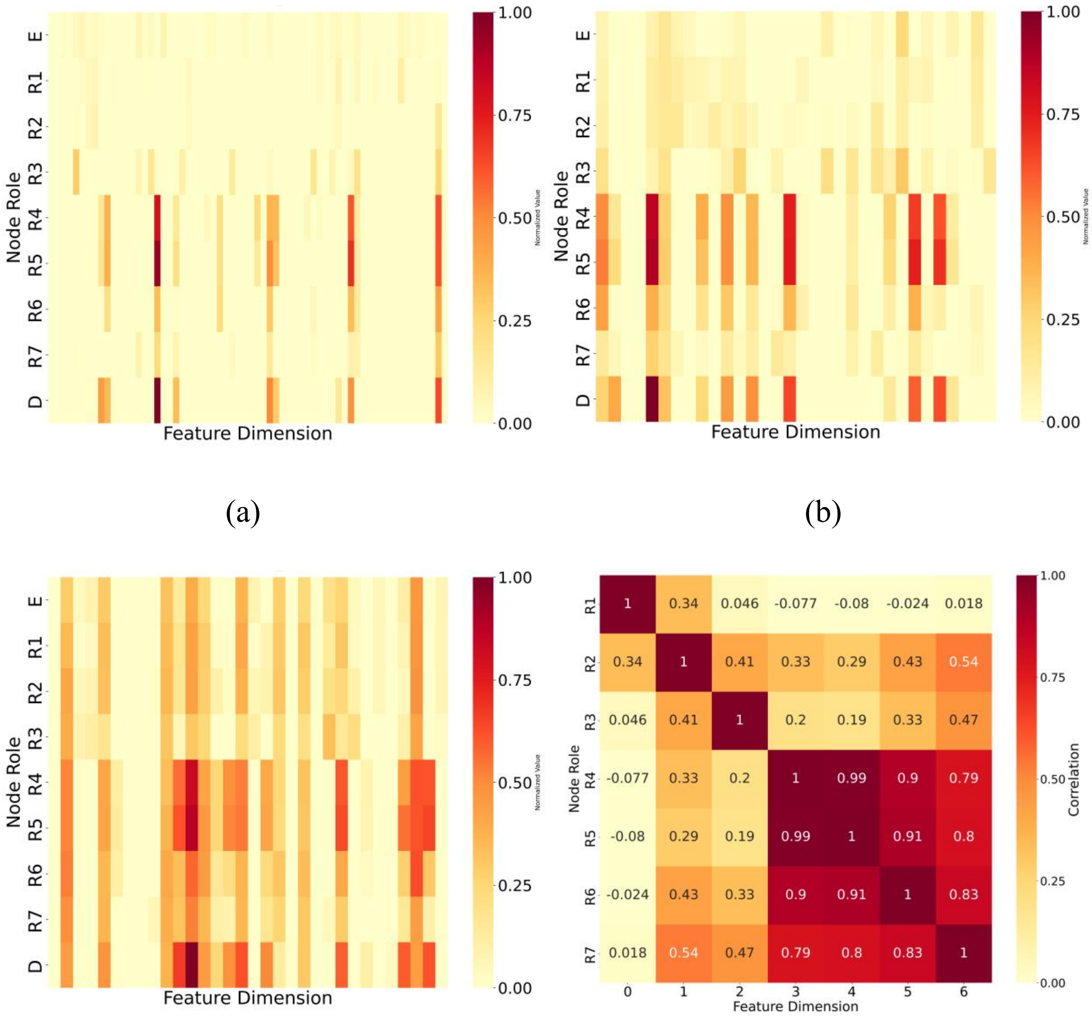


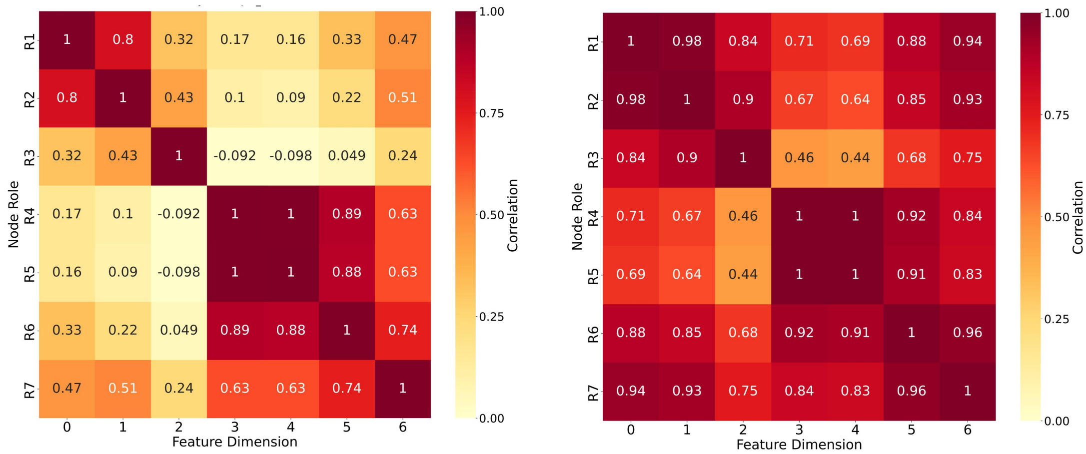
Figure 4.12 Geometric layout of the sensor network and scatter location for the case

study used in the GNN interpretability analysis.



(a) (b)

(c) (d)



(e) (f)

Figure 4.13 Node feature and correlation for each layer in GNN:(a) Node feature heatmap for Layer 1; (b) Node feature heatmap for Layer 2; (c) Node feature heatmap for Layer 3; (d) Receiver node correlation matrix for Layer 1; (e) Receiver node correlation matrix for Layer 2; (f) Receiver node correlation matrix for Layer 3.

This effective fusion is guided by the attention mechanism, which assigns importance to different wave paths. As illustrated in Figure 4.14, the GNN consistently assigns the highest attention weights (α_{uv} from Eq. 29) to direct paths, followed by single-reflection paths, with double reflection paths receiving the lowest weights. This behavior directly mirrors the physical phenomenon of wave attenuation, where energy dissipates over longer, more complex paths. The GNN has consequently acquired the ability to prioritize pathways according to their physical importance, efficiently utilizing attention weights to represent wave energy or signal-to-noise ratio.

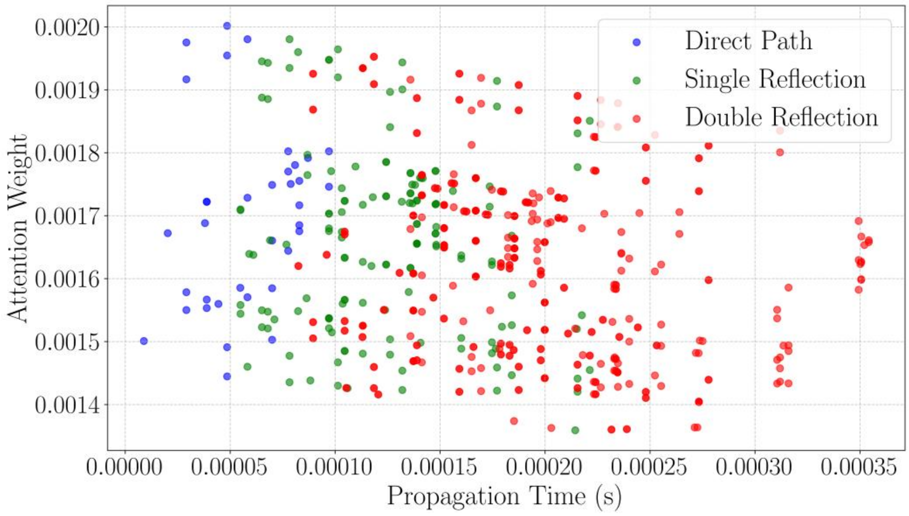


Figure 4.14 Attention weight for different path types.

This physically-grounded weighting directly influences the feature space dynamics. Figure 4.15 tracks the Euclidean distance between the feature vectors of node pairs, defined as $d(\mathbf{h}_i, \mathbf{h}_j) = \|\mathbf{h}_i - \mathbf{h}_j\|_2$, across the GNN layers. A clear trend emerges: the feature distances between most node pairs decrease from Layer 1 to Layer 3. This convergence indicates that as message-passing proceeds, the node representations are collectively updated to reflect a globally consistent state, analogous to how a physical wavefield stabilizes after all superpositions have occurred. The evolution of a feature vector for a single receiver (R1), shown in Figure 4.16, becomes progressively denser and more complex. This can be interpreted as a hierarchical feature construction process: the sparse activations in Layer 1 may encode the primary direct wave arrival. In contrast, the richer representations in subsequent layers incorporate the more complex signatures of reflected and scattered waves, culminating in a final feature that represents the fully superposed signal.

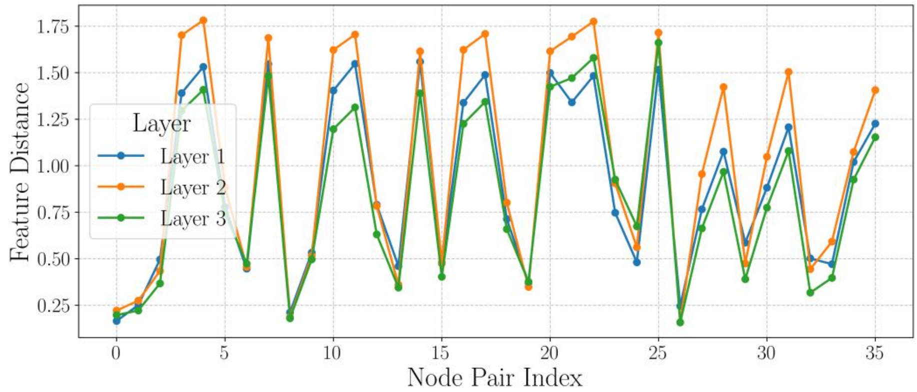


Figure 4.15 Feature distance propagation in GNN.

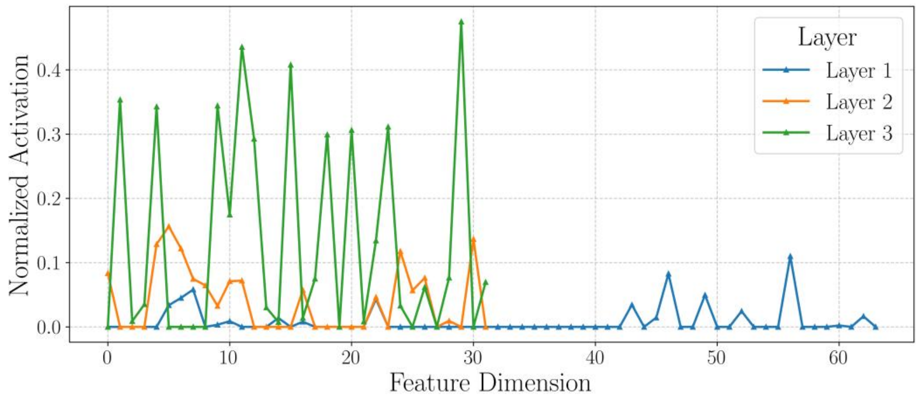


Figure 4.16 Feature evolution for receiver 1 in GNN.

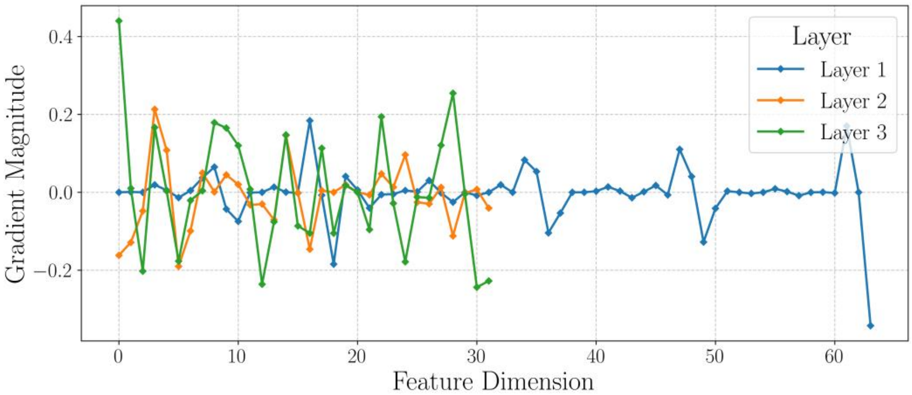


Figure 4.17 Feature gradient in GNN.

Finally, Figure 4.17 examines the gradient of the feature vectors along their dimensions, which represents the rate of change of features within the latent space. The discrete gradient for a feature vector $\mathbf{h}_v$ at dimension k is approximated using the central difference:

$$\nabla \mathbf{h}_v[k] \approx \frac{\mathbf{h}_v[k+1] - \mathbf{h}_v[k-1]}{2} \tag{4.29}$$

The gradient magnitude increases significantly across the layers, reaching its peak in Layer 3. This shows that the fusion process adds major non-linear transformations that enhance feature representations. More than just combining information, the GNN learns to "sculpt" the feature space, creating high-gradient regions in specific areas. These sharp changes are important because they probably encode the exact timing information needed for accurate ToF prediction, similar to how wave interference produces localized peaks in a physical signal.

4.3.4 Performance and Physical Interpretability of GNN Attention Mechanisms

The attention mechanism's normalization strategy is critical, as it defines the competition among wave paths and directly impacts the model's performance and physical interpretability. To ensure a rigorous comparison, all experiments in this section were conducted with a global random seed set to 12. This analysis compares three distinct strategies: a global Softmax approach, a local Softmax approach, and a local Softmax with physical prior variant.

The analysis begins with the global softmax strategy, where attention scores are normalized across all edges in the entire graph. A sensitivity analysis on the softmax temperature hyperparameter (τ) was conducted to find the optimal setting. The results, summarized in Table 4.4, indicate that the model's performance on the Actuator A dataset improves as the temperature increases, achieving optimal results at $\tau = 0.7$.

Table 4.2 Performance of global softmax with different temperatures (Dataset A)

Softmax Temp (τ)	**MAE**	**MSE**
0.1	2.12641e-05	6.21226e-10
0.3	2.06912e-05	6.16492e-10

0.5	2.03441e-05	5.78729e-10
0.7	1.89562e-05	5.61221e-10

To assess the model's ability to generalize, we performed an extrapolation test by training the model exclusively on the Actuator A dataset and then evaluating it directly on the unseen datasets for actuators B, C, and D. The results, shown in Table 4.3, demonstrate remarkable consistency and robustness, with only a minor degradation in performance. This indicates that the global attention mechanism learns a generalized physical model of wave propagation that is not overfitted to a specific actuator location.

Table 4.3 Generalization test results for global softmax (Trained on A)

Test Dataset	**MAE**	**MSE**
Actuator B	2.22406e-5	1.01663e-9
Actuator C	2.22427e-5	1.06508e-9
Actuator D	2.16500e-5	1.07829e-9

The attention weights generated by this method, as depicted in Figure 4.14, were analyzed by fitting them to an equivalent attenuation model to quantify their physical behavior:

$$w \approx C \cdot \exp(-B \cdot t) \cdot r^{m} \tag{4.30}$$

In this model, w represents the attention weight, t is the propagation time, and m is the number of reflections for a given path. The regression yielded a statistically significant time decay coefficient of $B = 285.14\ s^{-1}(\ p < 0.001)$, confirming that the network learns a foundational attenuation principle.

In contrast, the local softmax strategy restricts normalization to groups of edges sharing the same target node. As shown in Table 4.4, this approach achieved its best performance at $\tau = 0.5$. It also showed robust generalization capabilities comparable to the global method, as detailed in Table 4.5.

Table 4.4 Performance of local softmax with different temperatures (Dataset A)

Softmax Temp (τ)	**MAE**	**MSE**
0.1	2.23001e-05	7.30094e-10
0.3	2.18769e-05	6.88563e-10
0.5	2.06861e-05	6.25550e-10
0.7	2.07003e-05	6.45888e-10

Table 4.5 Generalization test results for local softmax (Trained on A)

Test Dataset	**MAE**	**MSE**
Actuator B	2.17344e-5	1.06119e-9
Actuator C	2.22195e-5	1.07726e-9
Actuator D	2.13403e-5	1.14253e-9

The primary distinction of the local softmax method lies in its physical interpretability. Fitting the same decay model to these weights yielded a substantially larger time decay coefficient of $B = 371.79\ s^{-1}(p = 0.003)$. It is important to note an apparent paradox when visually comparing the scatter plots of attention weights from the local and global softmax models. While the fitted decay coefficient for the local model is significantly larger, indicating a faster relative decay, its corresponding plot in Figure 19 does not appear steeper than the plot for the global model shown in Figure 4.14.

This discrepancy arises because the decay coefficient B represents a relative decay rate, whereas visual steepness corresponds to an absolute decay amount. The global softmax weights (Figure 4.14) have high initial values, leading to a large absolute drop that appears visually steep despite a lower relative decay rate. Conversely, the local softmax weights (Figure 4.18) are concentrated in a much lower range; thus, even with a higher relative decay rate, their minuscule absolute change appears visually flat.

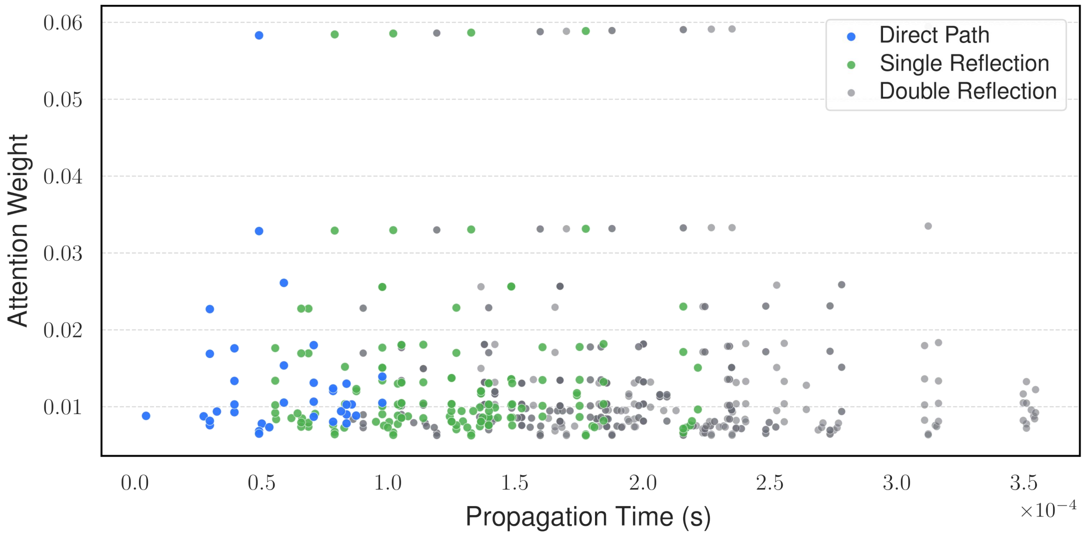
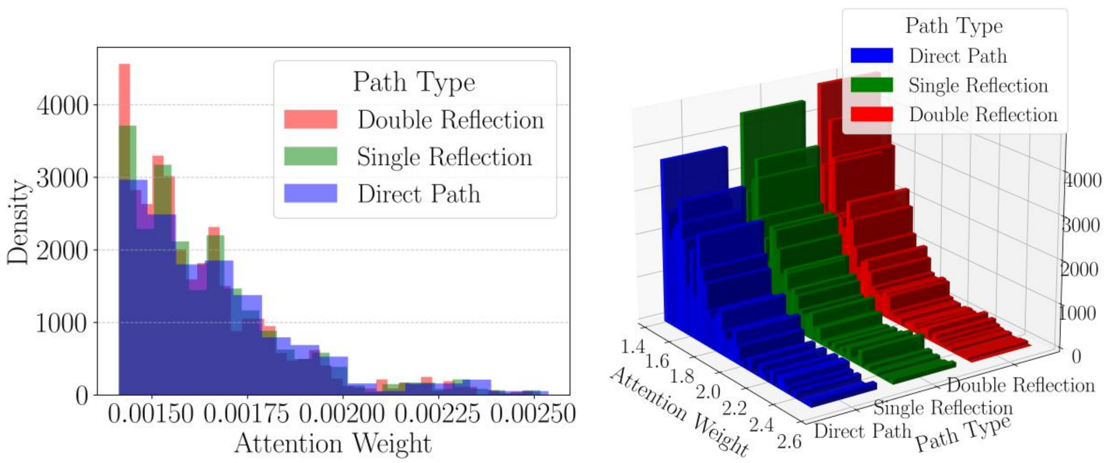


Figure 4.18. Attention weight for different path types from local softmax.

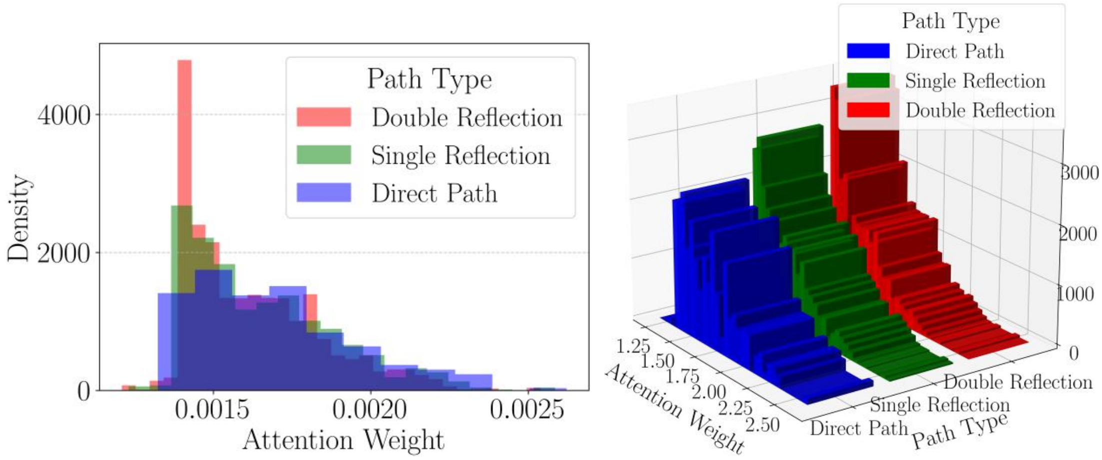
To conclude, we validate the physical reasonableness of the learned decay coefficients by treating the fitted model as an equivalent attenuation model. For a 200 kHz S0 Lamb wave in a 2 mm aluminum plate, the theoretical time decay coefficient from material damping alone ($B_{material} = \alpha \cdot v_g$) is approximately 119 s^{-1}. Our fitted coefficients (Local: 371.79 s^{-1}; Global: 285.14 s^{-1}) are substantially larger than this theoretical lower bound. This is physically consistent, as the GNN's effective decay coefficient B naturally incorporates not only material damping but also the more dominant energy loss from geometric spreading. This comparison demonstrates that the network has not simply memorized data; it has learned an effective physical principle. The fact that the GNN autonomously learns a significant decay trend-which becomes more pronounced with the more discerning local attention mechanism-provides strong evidence that the framework's internal mechanisms are learning a meaningful and physically consistent analogue of wave attenuation.

4.3.5 Analysis of time window size on different path types

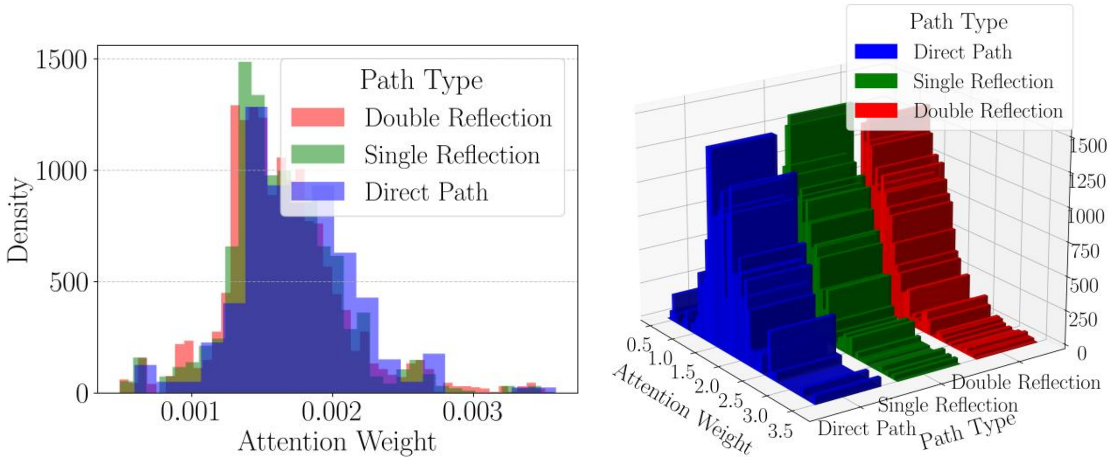
The time window, T_{window}, used in the Ray Tracing module, is a key hyperparameter that directly affects the complexity of the generated graph by eliminating physically unlikely long paths. This section explores the relationship between this temporal limit, the physics-informed loss functions, and the GNN's learned attention mechanism. We show that selecting an appropriate time window, along with the physics-informed losses, is crucial for guiding the network to learn a physically consistent distribution of attention weights across various path types.



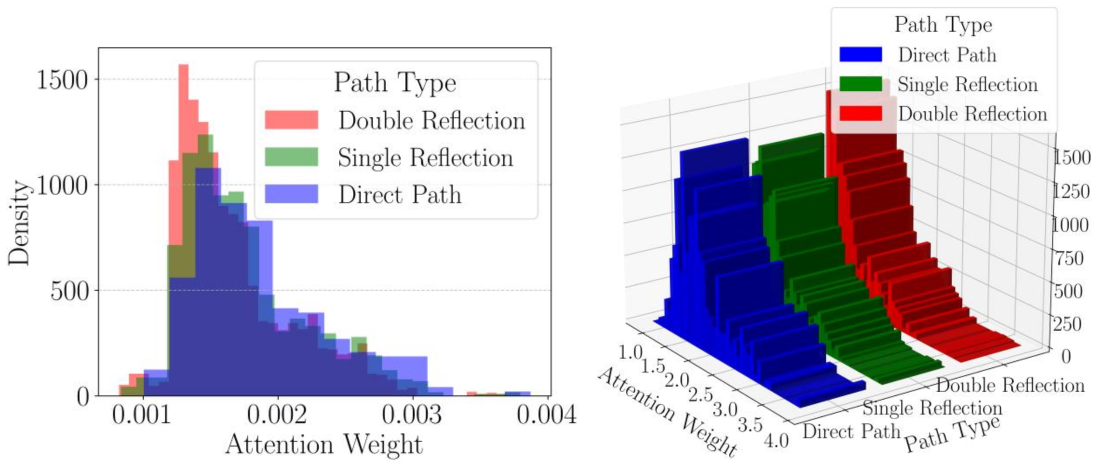
(a)


Path Type
Double Reflection
Single Reflection
Direct Path
Density
0
2000
4000
0.0015
0.0020
0.0025
Attention Weight
Path Type
Direct Path
Single Reflection
Double Reflection
0
1000
2000
3000
1.25
1.50
1.75
2.00
2.25
2.50
Attention Weight
Direct Path
Single Reflection
Double Reflection
Path Type

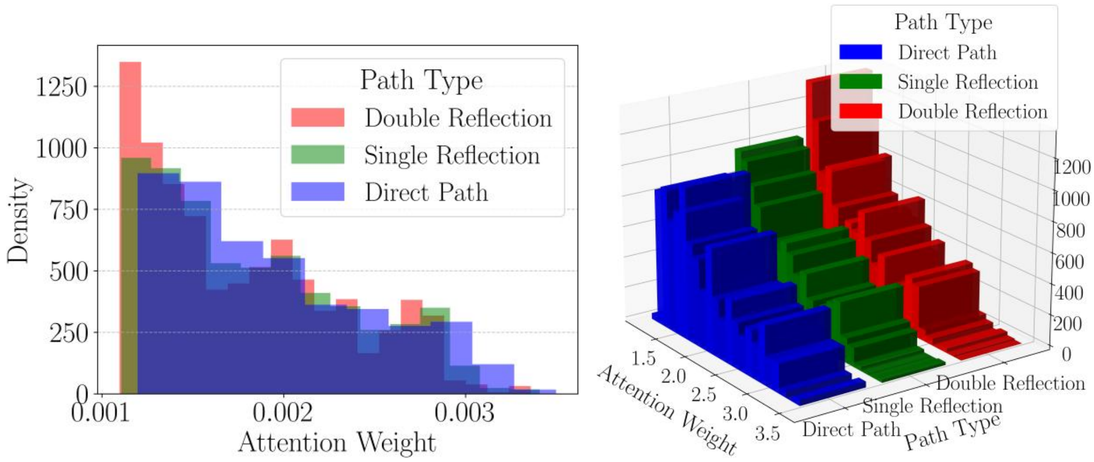
(b)


Path Type
Double Reflection
Single Reflection
Direct Path
Density
0
500
1000
1500
0.001
0.002
0.003
Attention Weight
Path Type
Direct Path
Single Reflection
Double Reflection
0
250
500
750
1000
1250
1500
0.5
1.0
1.5
2.0
2.5
3.0
3.5
Attention Weight
Direct Path
Single Reflection
Double Reflection
Path Type

(c)


Path Type
Double Reflection
Single Reflection
Direct Path
Density
0
500
1000
1500
0.001
0.002
0.003
0.004
Attention Weight
Path Type
Direct Path
Single Reflection
Double Reflection
0
250
500
750
1000
1250
1500
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Attention Weight
Direct Path
Single Reflection
Double Reflection
Path Type

(d)



(d)

Figure 4.19 Histogram of attention weights under different time windows: (a) No physics-informed constraints (T_{window} = 4000 μs); (b) With physics-informed constraints (T_{window} = 4000 μs); (c) With physics-informed constraints (T_{window} = 3500 μs); (d) With physics-informed constraints (T_{window} = 3000 μs); (e) With physics-informed constraints (T_{window} = 2500 μs).

The effect of these components is visualized in Figure 4.19, which shows the probability density function $p(\alpha|\text{ path type }, T_{window})$ of the attention weights (α_{uv}) for direct, single-reflection, and double-reflection paths. Figure 4.19(a) presents a baseline scenario where the model is trained with a large time window ($4000\mu s$) but without the physics-informed losses (L_{wave} and $L_{wavelet}$). In this case, the distributions for all three path types are largely overlapping and indistinguishable, indicating that the data-driven model alone fails to learn the physical hierarchy of path importance.

In contrast, Figure 4.19(b) shows the result for the same time window (4000μs), but with the physics-informed constraints. A clear and physically meaningful separation immediately appears: the distribution for direct paths shifts toward higher weights, while the distributions for reflected paths are concentrated at lower values. This demonstrates that the physics-informed losses are crucial for guiding the model. The analysis of Figures 4.19(c) through 4.19(e) reveals how the time window refines this process. As the time window is moderately shortened, the separation between the distributions is maintained and becomes even more distinct. This occurs because reducing the graph complexity by pruning the longest, least relevant paths allows the GNN to focus more decisively on the remaining, more physically plausible paths. However, there is a key trade-off, as an overly restrictive window could mistakenly exclude critical reflection paths that carry vital information about the damage. Therefore, the goal is to select an optimal window (4000μs in our case) that is large enough to include all significant wave interactions but small enough to exclude unlikely, high-attenuation paths.

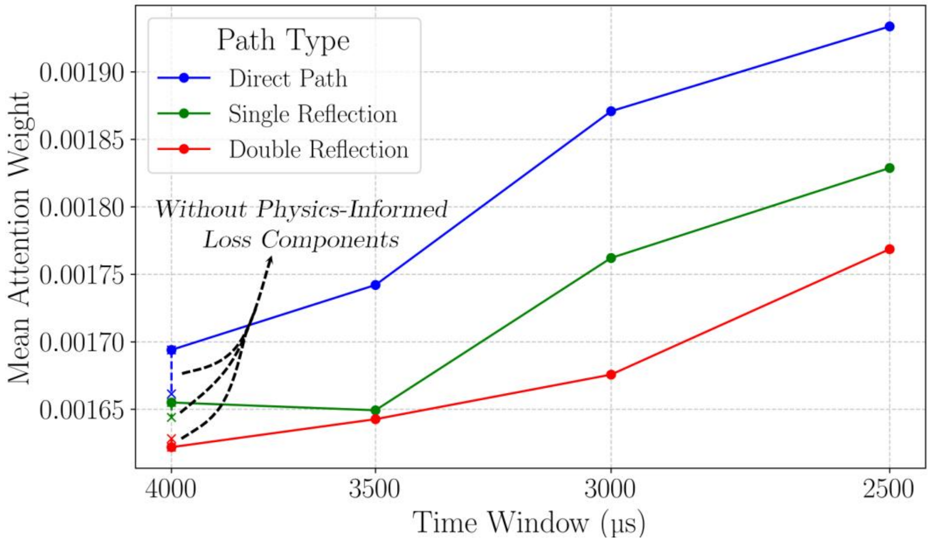


Figure 4.20 Mean attention weight versus time windows for different path types.

This relationship is shown in Figure 4.20, which displays the average attention weight for each path type as a function of the time window. For the physics-informed model, the physical hierarchy (Direct > Single Reflection > Double Reflection) remains consistently across all tested time windows. The dashed lines, representing the model without physics-informed losses, emphasize their crucial role; without them, the average weights cluster together and lack any physical ordering.

Table 4.6 Statistical tests of the distribution of attention.

Path Type	KS Statistic	KS P-Value	MWU Statistic	MWU P-Value
Direct Path	0.1230158730	0.0440505628	33377	0.3203591159
Single Reflection	0.1220238095	5.8846210e-07	500332	0.5557480724
Double	0.1640211640	6.5311638e-	4257386	3.5159540725

Reflection		36		

To formally validate the significance of these distributional changes, we conducted two non-parametric statistical tests comparing the attention distributions of the physics-informed (A_{phys}) and non-physics-informed (A_{no_phys}) models at the operational $T_{window} = 4000\mu s$. The results are summarized in Table 4.6. We employed the two-sample Kolmogorov-Smirnov (KS) test [158] to evaluate the greatest disparity between the empirical cumulative distribution functions (CDFs) of two samples, $F_{phys}(x)$ and $F_{no\_phys}(x)$. The KS statistic D is defined as:

$$D = \sup_x \left| F_{\text{phys}}(x) - F_{\text{no\_phys}}(x) \right| \tag{4.31}$$

Additionally, we used the Mann-Whitney U(MWU) test [159], which assesses whether one distribution is stochastically greater than the other by comparing the sum of ranks. The U statistic for the first sample is given by:

$$U_1 = n_1 n_2 + \frac{n_1(n_1+1)}{2} - R_1 \tag{4.32}$$

where n_1 and n_2 are the sample sizes, and R_1 is the sum of ranks for the first sample in the combined, sorted data. In both tests, the assumption is that the two samples originate from the same distribution. The results show statistically significant differences between the distributions for all path types (KS p-value < 0.05 for all). The extremely low p-values for the double-reflection paths (KS p-value $\approx 6.53 \times 10^{-36}$) confirm that the physics-informed constraints are most impactful in helping the model correctly interpret complex, multireflection paths.

4.3.5 Analysis of the effect of physical constraints on different reflection paths

The two physics-informed loss terms, the Wave Equation Residual Loss (L_{wave}) and the Wavelet Arrival Loss ($L_{wavelet}$), are central to guiding the PhysRayGNN framework towards physically plausible solutions. The extent of their influence is governed by their respective weighting factors in the total loss function, as defined in Eq. (4.1). This section presents a sensitivity analysis to investigate how these weights impact both the model's prediction accuracy and its internal attention mechanism. For this study, we set the weights equal, $\lambda_1 = \lambda_2 = W$, and systematically vary the value of W to identify the optimal balance between data-driven learning and physics-based regularization.

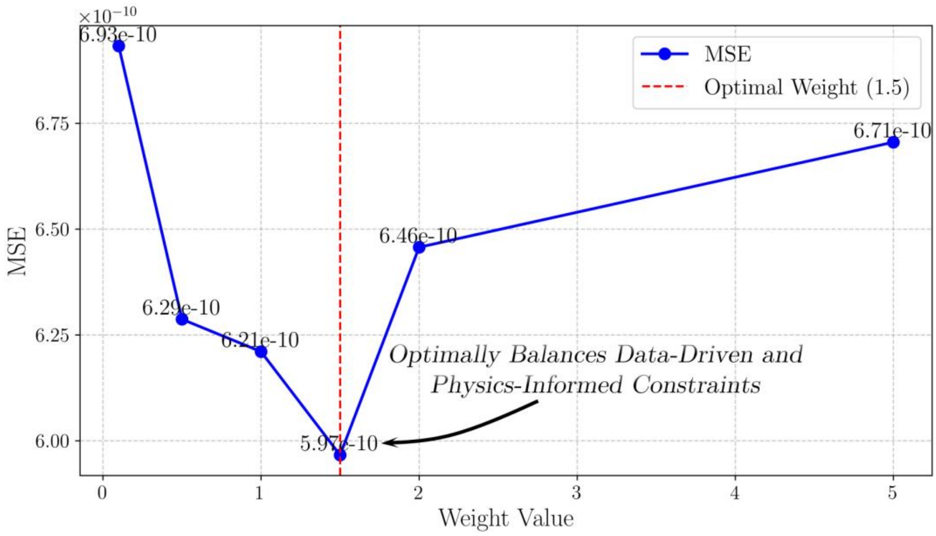


Figure 4.21 MSE results corresponding to different physical constraint weights.

The relationship between the physics constraint weight W and the model's final prediction accuracy is presented in Figure 4.21. The graph demonstrates a clear U-shaped curve for the MSE, showing that the performance of the model is greatly affected by this hyperparameter. The MSE reaches its minimum value when $W = 1.5$. This result is highly significant, as it demonstrates the existence of an optimal trade-off. When the weight is too low, the physics-informed losses are insufficient to regularize the model, allowing it to converge to a solution that may fit the training data but violates underlying physical principles. Conversely, when the weight is too high, the model is over-constrained, forcing it to adhere too rigidly to the idealized physics, which may not perfectly capture the nuances of the real experimental data, thereby increasing the prediction error.

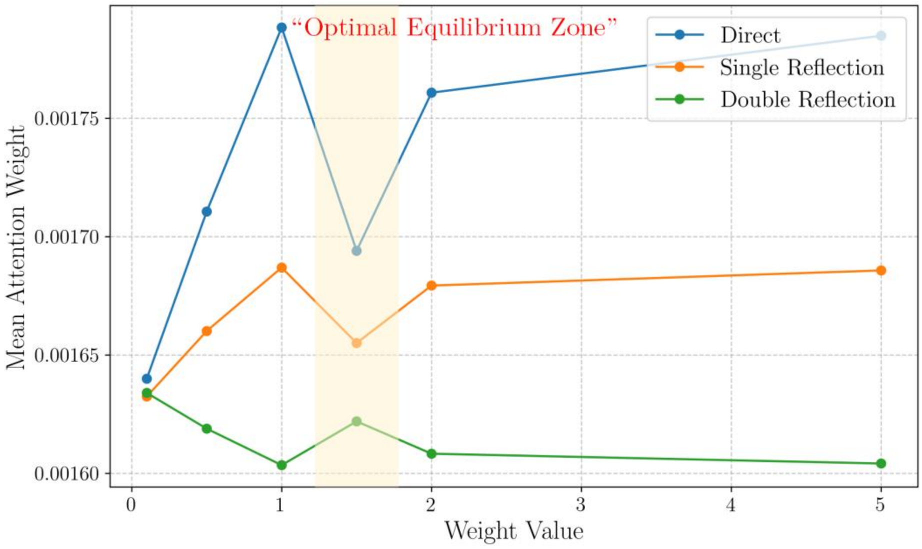


Figure 4.22 Mean attention weights corresponding to different physical constraint weights.

To understand how this optimal performance at $W = 1.5$ manifests within the GNN, we analyzed the mean attention weights assigned to different path types, as shown in Figure 4.22. The results reveal that as the weight W increases from 0.1 to 1.5, the mean attention weights for the three path types diverge, establishing the physically correct hierarchy (Direct > Single Reflection > Double Reflection). Critically, at and around $W = 1.5$, this hierarchy stabilizes, and the weights appear to converge. We term this region the "optimal equilibrium zone". Within this zone, the physics-informed losses have provided sufficient guidance for the GNN to learn a physically coherent attention mechanism. In contrast, the data-driven loss ensures the model remains faithful to the experimental observations. This equilibrium represents the ideal synergy between physics-based guidance and data-driven learning.

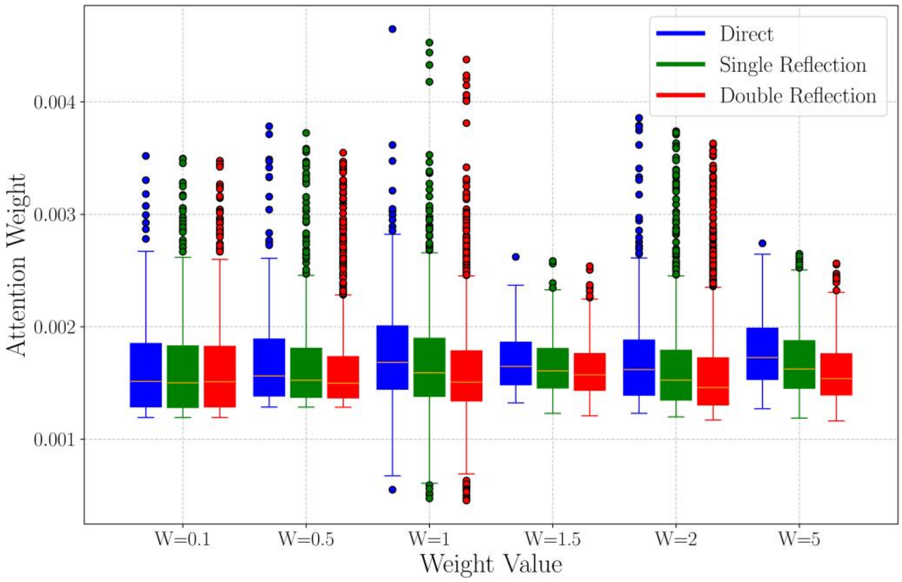


Figure 4.23 Boxplot attention weights.

A more detailed, distributional perspective is provided by the boxplots in Figure 4.23. This visualization details the statistical distribution of attention weights for each path type across different values of W. Each boxplot illustrates the median (orange line), the interquartile range (IQR) [160], and outliers. The IQR is a measure of statistical dispersion, defined as:

$$\mathrm{IQR} = Q_3 - Q_1 \tag{4.33}$$

where Q_3 and Q_1 are the $75th$ and $25th$ percentiles of the data, respectively. The boxplots clearly show that at $W = 1.5$, the distributions for the three path types are distinctly separated with minimal overlap, and the variance within each distribution is well-contained. In contrast, at lower weights ($W = 1.0$), the distributions, particularly for direct and single-reflection paths, exhibit a larger number of outliers and greater variance, suggesting a less stable or less confident assignment of attention. This confirms that the weight of 1.5 not only minimizes the overall error but also induces a stable and physically interpretable attention mechanism within the GNN, validating the critical role of precisely tuned physical constraints in our framework.

4.3.6 Analysis of the number of reflections setting in ray tracing

The maximum number of reflections, k_{max}, considered by the Ray Tracing module is a fundamental hyperparameter that dictates the tradeoff between the physical fidelity and the computational complexity of the generated graph. A higher k_{max} allows for the modeling of more complex wave phenomena but also risks introducing noisy, low-energy paths that could degrade the GNN's performance. This section presents an ablation study to determine the optimal setting for k_{max} by evaluating its impact on both prediction accuracy and the GNN's internal attention mechanism.

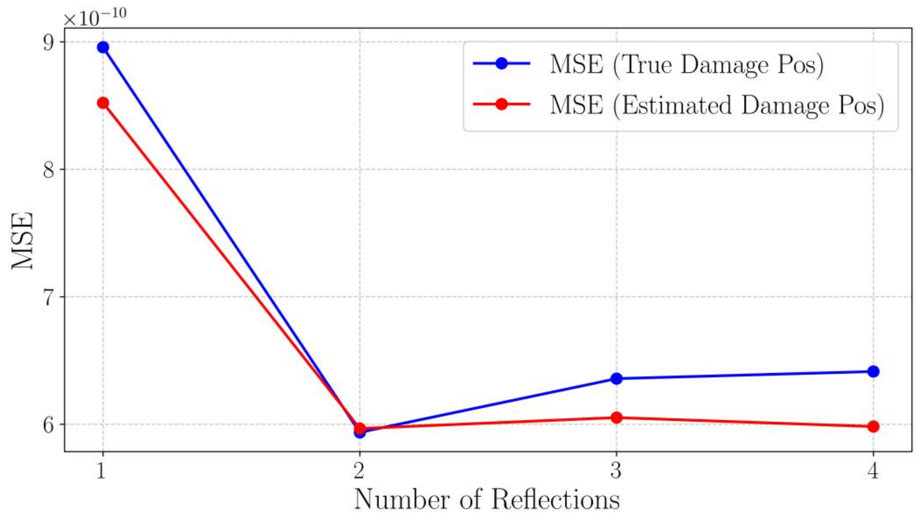


Figure 4.24 Comparison of GNN for true and estimated damage location training with different numbers of reflections.

First, we examine the effect of k_{max} on the final prediction error. We trained the PhysRayGNN model with graphs generated using different maximum reflection counts, ranging from one to four. The analysis was performed under two separate scenarios: (1) an "oracle" setting where graphs were constructed using the ground-truth damage locations, representing the upper performance limit; and (2) a realistic setting where graphs were built using locations provided by the pre-trained Damage Position Initializer. The MSE for both cases is plotted against the number of reflections in Figure 4.24.

The results reveal a clear and consistent trend for both training methodologies. The model's performance is suboptimal when only one reflection is considered, improves significantly when increased to two reflections, and then slightly degrades as the number is further increased to three and four. The optimal performance for both scenarios is achieved when $k_{max} = 2$. This U -shaped performance curve can be physically interpreted: at $k_{max} = 1$, the graph is too sparse and fails to capture crucial double-reflection paths that often carry significant information about the damage. Conversely, at $k_{max} = 3$ and $k_{max} = 4$, the graph becomes overly dense with high-order reflection paths that are physically attenuated and contribute more noise than signal, thus slightly impairing the GNN's learning process. The close tracking of the two curves also validates the robustness of our Damage Position Initializer, as the model's behavior remains consistent even when using its estimated locations.

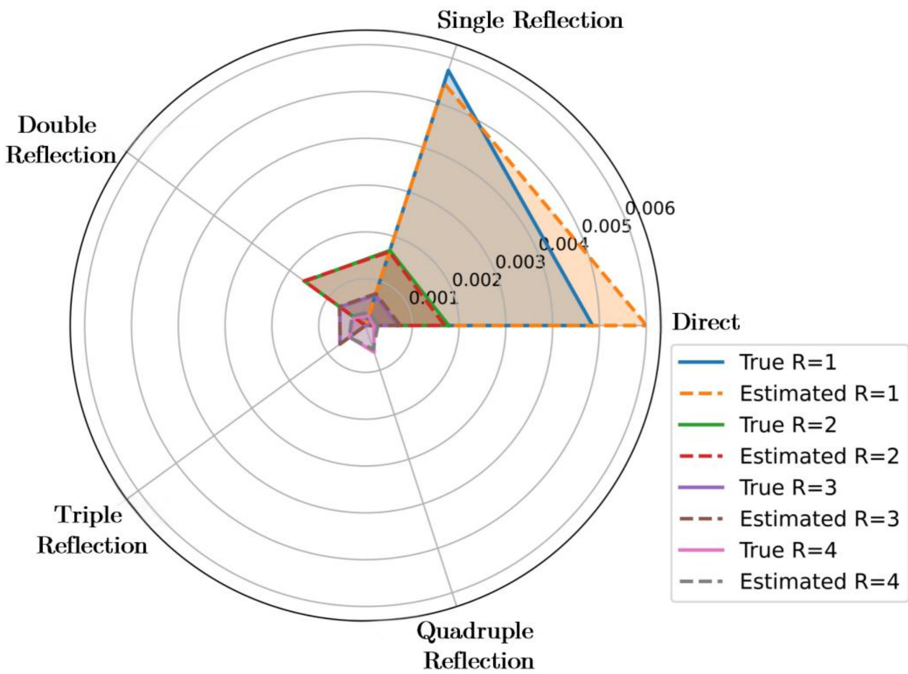


Figure 4.25 Path attention weight contribution versus number of reflections.

To better understand how the GNN allocates its focus with varying graph complexity, we analyzed the contribution of different path types to the total attention learned by the network. For a given path type p, its total attention contribution, C_p, is defined as the sum of attention weights of all edges of that type, normalized by the total attention sum over all edges in the graph $\mathcal{E}$:

$$C_p = \frac{\sum_{(u,v)\in\mathcal{E}_p} \alpha_{uv}}{\sum_{(i,j)\in\mathcal{E}} \alpha_{ij}} \tag{4.34}$$

where $\mathcal{E}_\mathrm{p}$ is the subset of edges corresponding to path type p. Figure 4.25 presents a radar chart visualizing these contributions for different settings of k_{max}. The results are unequivocal: across all settings, the vast majority of the network's attention is concentrated on direct and single-reflection paths. The contribution of double-reflection paths is notable but substantially smaller, while the contributions from triple and quadruple reflections are negligible.

This outcome provides strong evidence that the GNN's attention mechanism learns to align with the physical principle of wave attenuation. The model intrinsically identifies that higher-order reflection paths are less reliable due to energy dissipation and correctly allocates minimal attention to them. The significant attention given to double-reflection paths when $k_{max} \geq 2$, combined with the minimal contribution of higher-order paths, reinforces the conclusion from the MSE analysis. Therefore, setting the maximum number of reflections to two provides the optimal balance, ensuring the graph is sufficiently rich to capture all physically significant wave interactions without being burdened by the noise of inconsequential high-order paths. This analysis validates our design choice of $k_{max} = 2$ for all other experiments in this study.

The parameter k_{max} was selected based on a combination of hyperparameter tuning and physical constraints. Empirically, we chose the value that minimized the validation loss. Physically, this choice is validated by the wave propagation limit, where higher-order reflections exceeding the sampling time window (T)—given the wave speed (c)—provide no meaningful information to the model.

4.3.7 Ablation experiment

To thoroughly assess the individual contribution of each key component within the PhysRayGNN framework, an extensive ablation study was carried out. In this study, we systematically removed or replaced specific modules of our proposed model and evaluated the resulting impact on performance. The results of these experiments, benchmarked against the full PhysRayGNN model using the dataset for actuator A, are summarized in Table 4.7.

The study first confirms the key architectural decisions of our framework. Replacing the entire GNN module with a standard Multi-Layer Perceptron (without GNN, with MLP) or removing the GNN's internal attention mechanism (without GNN Attention Mechanism) both lead to a significant increase in prediction error. This highlights the importance of a graph-based representation and the attention mechanism for effectively modeling complex wave interactions. Similarly, replacing our physics-informed ray tracing with a simple, fully-connected graph structure (without Ray Tracing, with Simple Graph Structure) also causes a notable drop in performance, confirming that the physical inductive biases provided by the ray-traced paths are essential for guiding the model.

Next, we evaluated the effect of the physics-informed loss functions. Removing either the Wave Equation Loss or the Wavelet Transform Loss alone led to a significant drop in performance. Importantly, when both physics-informed constraints are excluded (w/o Wave Equation & Wavelet Transform Loss), the model's error rises sharply. This clearly shows that these physics-based regularizers are crucial for guiding the feature extraction process and ensuring the learned representations are rooted in wave physics.

Table 4.7 Ablation study results.

Method	**MAE**	**MSE**

w/o AdvancedTCN, w/ MLP	2.70180E-05	1.17116E-09
w/o Wave Equation Loss	2.88505E-05	1.17287E-09
w/o Wavelet Transform Loss	2.30815E-05	7.67739E-10
w/o Wave Equation & Wavelet Transform Loss	2.98112E-05	1.22408E-09
w/o GNN Attention Mechanism	2.84149E-05	1.31155E-09
w/ 3x in Wave Temporal Extractor	2.21473E-05	6.99052E-10
w/ 5x in Wave Temporal Extractor	2.11364E-05	6.50190E-10
w/ 7x in Wave Temporal Extractor	2.39058E-05	8.29518E-10
w/o GNN, w/ MLP	2.88731E-05	1.34757E-09
w/o Damage Position Initializer, w/ MLP	2.68784E-05	1.01763E-09
w/o Ray Tracing, w/ Simple Graph Structure	2.47876E-05	8.62771E-10
w/o Damage Position Initializer & Ray Tracing, w/ Simple Graph Structure	2.49351E-05	9.77746E-10
Our Model (Actuator A)	1.98797e-05	5.96621E-10

Finally, the contributions of the upstream modules were assessed. Replacing the specialized AdvancedTCN in the Wave Temporal Extractor with a simpler MLP (w/o AdvancedTCN, w/ MLP) increased the error, confirming its effectiveness. Furthermore, removing the Damage Position Initializer and Ray Tracing modules entirely (w/o Damage Position Initializer & Ray Tracing, w/ Simple Graph Structure) also caused a significant performance decline. This emphasizes the synergistic effect of the entire proposed pipeline, where each component plays a crucial role in the model's overall accuracy.

4.4 Analysis of Data Efficiency and Model Robustness

To further investigate the performance boundaries of the proposed method and assess its capability under data-scarce conditions, a rigorous comparative experiment was conducted focusing on data efficiency. In this analysis, the training dataset size was progressively reduced from 100% down to 20%, while the testing set remained fixed to ensure a consistent evaluation standard. GraphSAGE, which demonstrated the most competitive performance among the baseline models in the full-data scenario, was selected as the benchmark for this comparison.

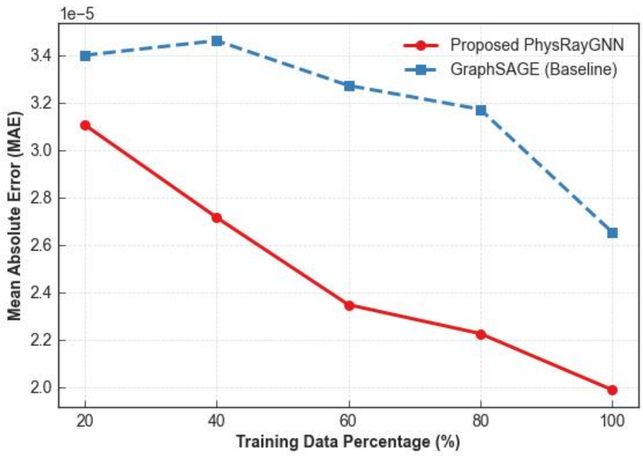


Figure 4.26 Comparison of Mean Absolute Error (MAE) showing the superior data efficiency of PhysRayGNN, particularly in low-data regimes.

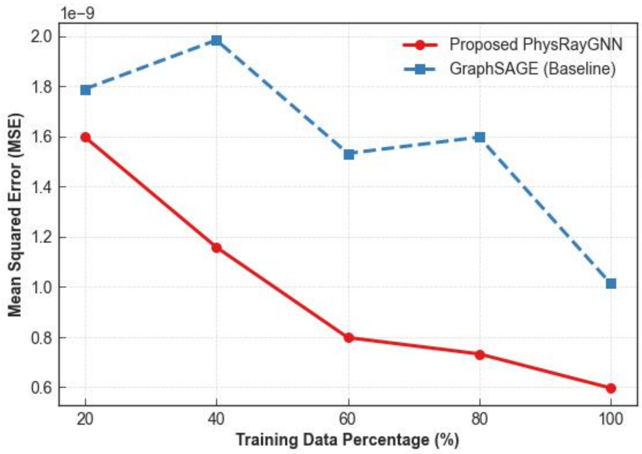


Figure 4.27 Comparison of Mean Squared Error (MSE) between PhysRayGNN and the best baseline (GraphSAGE) under varying training data percentages.

The comparative results in terms of Mean Squared Error (MSE) and Mean Absolute Error (MAE) are presented in Figure Figure 4.27 and Figure Figure 4.26, respectively.

As illustrated in the figures, PhysRayGNN consistently outperforms GraphSAGE across all data ratios, demonstrating superior robustness. A critical observation is the performance divergence as data availability decreases:

Performance at the Limit (20% Data): When the training data is reduced to only 20%, the performance of the purely data-driven GraphSAGE degrades significantly, with a sharp increase in both MSE and MAE. In contrast, PhysRayGNN maintains a high level of accuracy, exhibiting an error rate at 20% data that is comparable to, or even better than, GraphSAGE's performance at 60-80% data availability.

Discussion on the Mechanism: This superior data efficiency can be attributed to the hybrid nature of PhysRayGNN, which effectively combines data-driven learning with physics-informed constraints. Purely data-driven models like GraphSAGE rely heavily on large-scale datasets to learn the underlying statistical distributions; when data is scarce, they are prone to overfitting and fail to capture the complex wave propagation patterns. Conversely, PhysRayGNN leverages the Ray Tracing Module to construct a physics-consistent graph topology and utilizes Physics-Informed Loss to guide the optimization. These physical priors act as a strong regularization, compensating for the lack of data by ensuring the model's predictions adhere to the fundamental laws of wave propagation. Consequently, PhysRayGNN requires significantly fewer samples to converge to a generalized solution, making it highly suitable for practical SHM applications where labeled data is often limited.

4.5 Applications

4.5.1 Signal analysis and processing

To validate the efficacy of the proposed PhysRayGNN framework in accurately detecting the ToF of damage-scattered waves amid complex superpositions, this section examines two representative test cases. These scenarios are intentionally selected to challenge the model, featuring complex wave interactions where the damage-induced signal is not a standalone wave packet but is instead combined with boundary reflections and waves scattered by other PZTs. It is important to note that the specific damage locations in these cases were not included in the training dataset, making this a true test of the model's ability to generalize.

The first case, illustrated in Figure 4.28, involves actuator C. As shown in the wave path diagram (Fig. 4.28(a)), the wave scattered from the damage arrives at the receiver concurrently with both a boundary reflection and waves scattered from a neighboring PZT. This results in a complex superposition, which is evident in the wavelet coefficients of the received signal (Fig. 4.28(c)). The wave packet corresponding to the damage-scattered wave is clearly a composite of at least two interfering signals. Manually identifying the precise ToF from such a signal would be extremely challenging. Despite this complexity, the PhysRayGNN model accurately predicts a ToF of 9.64e-05 s, a value that correctly falls within the arrival window of this superposed wave packet. This result demonstrates the model's remarkable ability to deconvolve the mixed signal and isolate the timing of the damage-induced component.

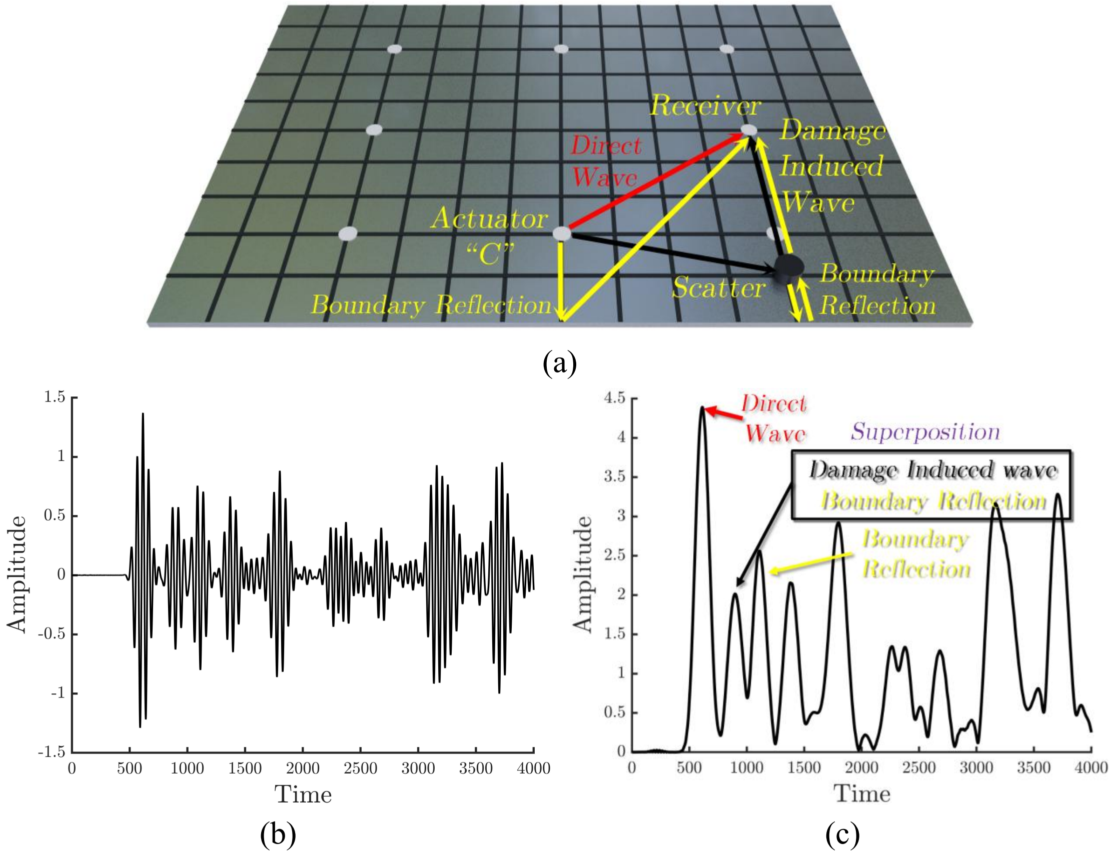


Figure 4.28 A representative case to showcase the effect of the proposed method: (a) Wave path diagram and (b) the raw signal (c) the identified peaks in the wavelet coefficients.

The second case, shown in Figure 4.29, uses actuator A and presents a different geometric challenge. Here again, the wave scattered by the damage travels to the receiver along a path that causes it to superimpose with boundary reflections and scattering from another PZT (Fig. 4.29(a)). The resulting wavelet coefficients (Fig. 4.29(c)) clearly show this superposition, where the damage-induced wave is part of a broader, more complex wave packet. In this scenario, the PhysRayGNN framework predicts a ToF of 1.37e-04 s. This prediction is again highly accurate, correctly identifying the arrival time within the complex signal.

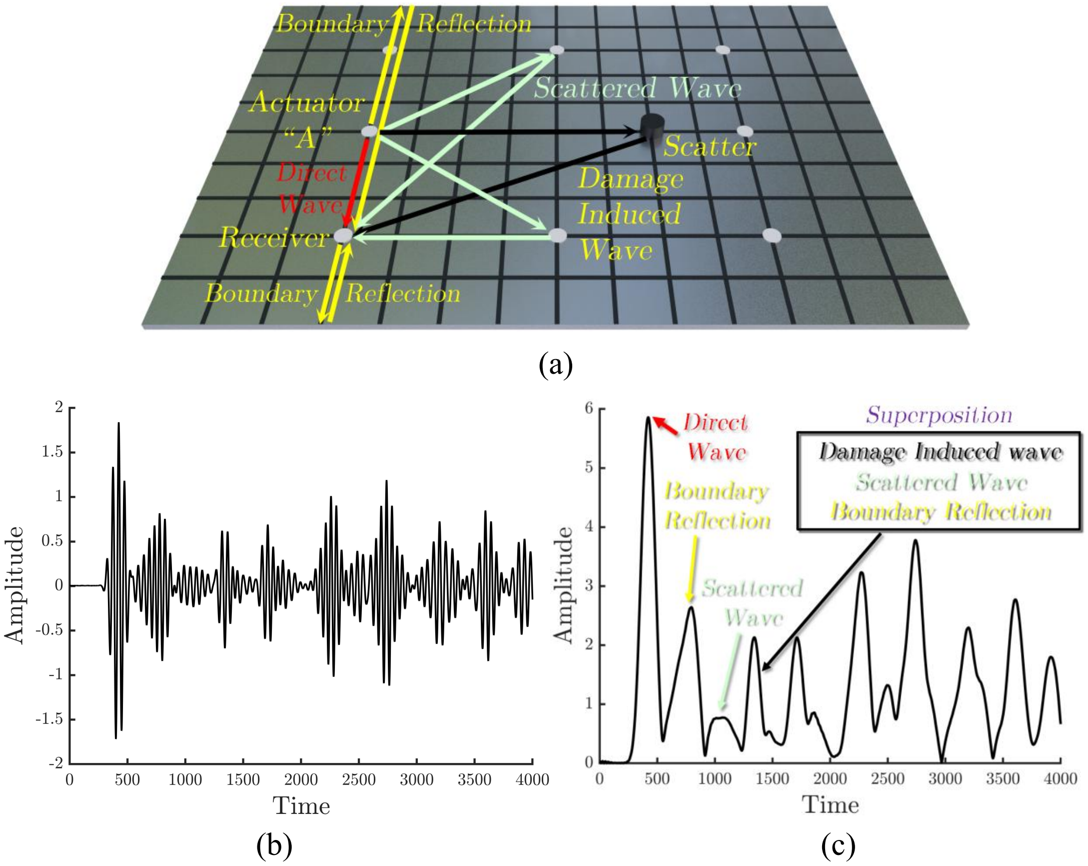


Figure 4.29 A representative case to showcase the effect of the proposed method: (a) Wave path diagram, (b) the raw signal, and (c) the identified peaks in the wavelet coefficients.

The successful ToF extraction in these two challenging and diverse cases strongly validates the PhysRayGNN framework. The results show that by modeling wave propagation and superposition on a physics-informed graph structure, the network can effectively learn to separate complex wavefields and identify the arrival time of damage-scattered waves, even when they are hidden by other high-energy wave events. This ability is crucial to the framework's effectiveness in later damage localization tasks.

4.5.2 Localization results

The ultimate measure of the framework's success is its ability to accurately translate predicted ToF values into precise damage localization. One significant benefit of our method lies in its operation within a no-baseline framework; it does not depend on a clear "healthy" signal for comparison, enhancing its relevance for practical structural health monitoring situations. To demonstrate the practical impact of our method, the ToF values extracted by PhysRayGNN are used as inputs to a standard ellipse positioning algorithm [161] to generate damage probability maps.

To establish a clear benchmark, we compare our results with a common manual identification method. In this foundational method, the time of flight (ToF) is identified by choosing the arrival time of the second prominent wave packet within the wavelet-transformed signal, a technique often employed when sophisticated processing options are unavailable. Figure 4.30 illustrates the localization outcomes for two exemplary test scenarios.

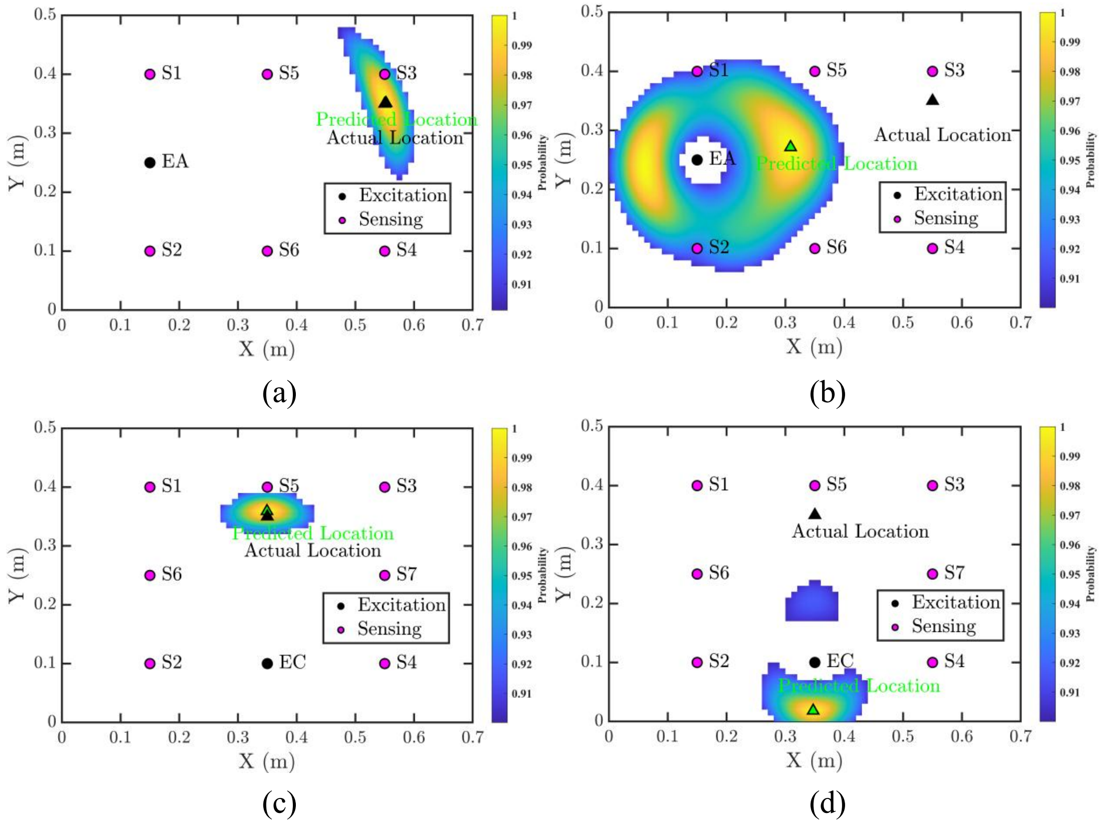


Figure 4.30 Comparison of location results: (a)-(c) Identify default arrival time through the network; (b)-(d) Second peak as the default arrival time.

In the first scenario, using actuator A, the localization result derived from the ToF values predicted by our PhysRayGNN is shown in Figure 4.30(a). The damage image is highly focused, and the predicted location is extremely accurate, with a localization error of just 0.25 cm. In contrast, when manually identified ToF values are used, the localization result, shown in Figure 4.30(b), is diffuse and highly inaccurate, with an error of 25.37 cm. The baseline method fails because it cannot distinguish the true damage-scattered wave from earlier, higher-energy boundary reflections, a challenge that our framework successfully addresses.

The second case, using actuator C, further confirms these results. As shown in Figure 4.30(c), the ToF values from our network again yield a precise localization, with an error of only 0.91 cm. In contrast, the baseline method, which again misidentifies the second wave packet as the damage signal, results in a complete failure in localization, with an error of 33.91 cm (Figure 4.30(d)).

These comparative results clearly show the significant impact of accurate ToF extraction on the final localization accuracy. The PhysRayGNN framework, by learning to deconvolve complex, overlapping wave signals, provides ToF data of a quality that enables highly precise damage localization. In contrast, traditional conventional signal peak selection method are shown to be unreliable when boundary reflections and signal superposition occur, resulting in localization errors that make them impractical for real-world SHM. This clear performance difference highlights the practical value and superiority of our proposed physics-informed, no-baseline approach.

4.6 Summary

In this section, we presented PhysRayGNN, an innovative framework that combines physics-informed principles with ray tracing to enhance graph neural networks. This framework aims to tackle the significant challenge posed by wave superposition in SHM systems utilizing Lamb waves. Our research marks the inaugural use of a graph neural network to directly model the physical phenomena associated with wave propagation and superposition for ToF estimation in SHM applications. By synergistically combining a data-driven GNN with the physical principles of wave propagation, our framework achieves highly accurate and interpretable ToF predictions, even in complex scenarios involving boundary reflections and signal interference.

The contributions of this work are multifaceted. We introduced a novel multi-module architecture where a physics-informed ray tracing algorithm dynamically creates a high-fidelity graph representation of the wavefield. The GNN operates on this physically meaningful graph, utilizing a specialized three-layer, attention-based message-passing scheme to effectively simulate wave propagation and attenuation. The success of this architecture is supported by our physics-informed learning strategy; combining a Wave Equation Residual Loss with a Wavelet Arrival Loss was crucial for guiding the model to learn latent representations that are fundamentally rooted in wave physics. Additionally, extensive analysis of the GNN's internal mechanisms showed that the model learns physically consistent behaviors, with the attention mechanism acting as a proxy for wave attenuation and the node feature evolution successfully mimicking the physical process of wave superposition.

The practical effectiveness of the PhysRayGNN framework was demonstrated through its outstanding performance in damage localization. By providing highly accurate ToF estimations, our no-baseline approach allowed a standard ellipse positioning algorithm to achieve remarkable accuracy. In difficult test cases where damage-scattered waves were hidden by other signals, our method produced localization errors as low as 0.25 cm and 0.91 cm. This is in stark contrast to the traditional manual peak-selection method, which, under the same conditions, resulted in major errors of 25.37 cm and 33.91 cm, respectively. This significant improvement highlights the practical value and superiority of our proposed approach.

In summary, the PhysRayGNN framework represents a notable progression in the creation of deep learning models for structural health monitoring (SHM) that are not only more accurate but also robust and interpretable. Our approach successfully connects the flexibility of data-driven methodologies with the precision of physics-based principles by explicitly integrating physical priors into both the graph architecture and the learning procedure. This work not only introduces a powerful new tool for SHM but also provides a clear methodological blueprint for integrating physical principles into GNN across various scientific and engineering challenges.

Chapter 5 WaveSenseNet: A Multimodal Multi-Scale Attention Fusion Network for Accurate Time of Flight Extraction in Complex Environments

5.1 Introduction

The preceding chapters have progressively developed frameworks to address the challenges associated with signal superposition in guided-wave-based Structural Health Monitoring (SHM). Chapter 3 introduced the Multi-Scale Signal Transformation (MSST) network, which successfully demonstrated the feasibility of employing a physics-guided deep learning model to mitigate the primary source of interference: boundary reflections. Building upon this foundation, Chapter 4 presented the PhysRayGNN model, which transitioned the focus from signal processing to explicit modeling of physical processes. By utilizing a Graph Neural Network to simulate wave propagation and superposition, PhysRayGNN significantly improved the physical interpretability of the model's predictions.

However, the models mentioned earlier mainly address signal aliasing caused by single or regular boundaries. The reality of engineering structures is far more complex. These structures often include many functional features like stiffeners, rivets, and weld lines that do not cause damage. These features are strong scatterers themselves, generating many interference signals that can be as strong as or stronger than boundary reflections. This creates a highly congested and disorganized wavefield. In such conditions, the target signal—such as a faint scattered wave from minor damage—is hidden within a complex noise field produced by numerous scatterers, resulting in a very low signal-to-noise ratio. Therefore, accurately separating and identifying the ToF of the target signal from this complex wavefield presents a more advanced and practical challenge for current methods.

Addressing such complexity often pushes existing methods to their limits. Traditional signal processing approaches have leveraged multipath information to image damage in geometrically complex structures. A notable example is the work of Hall and Michaels, who developed a multipath imaging algorithm that utilizes measured wavefield data as a dictionary to deconvolve signals from a stiffened aluminum plate with fasteners and through-holes [162]. Although innovative in its use of echoes and reverberations, this method fundamentally relies on a baseline subtraction process, rendering it susceptible to variations in environmental and operational conditions. In the realm of deep learning, research has also begun to address these intricate scenarios. For instance, Zi Zhang *et al*. employed a CNN to diagnose welding defects in metallic pipes, where the weldment itself serves as a significant scattering feature [163]. Their approach successfully classified various defect types but depended on a large dataset generated from numerical simulations to train the model for recognizing complex wave interactions. Collectively, these studies underscore a critical gap: the necessity for a robust, no-baseline method that can explicitly model and disentangle the complex wavefield in structures with multiple, diverse scatterers without relying on extensive simulated datasets.

To address these challenges, this chapter introduces a new neural network architecture called WaveSenseNet, a multimodal, multi-scale attention fusion network designed to more effectively identify ToF in complex environments. Additionally, the signal feature extraction component features a multi-scale fusion mechanism that combines multi-scale feature extraction with attention fusion strategies. Inspired by the frequency principle of neural networks, we propose a new parametric activation function to enhance the network's ability to process signals with varying frequencies. For better modeling of wave scattering characteristics, we define a Gaussian positional encoding map that helps the network learn spatial-temporal relationships more effectively. We improve the model by using Monte Carlo (MC) Dropout [164] for uncertainty estimation, allowing the calculation of variance in ToF predictions for post-processing. The hybrid loss combines quantile regression for uncertainty modeling and physics-constrained MSE to ensure wave-consistent predictions, balancing data-driven learning with domain knowledge. A dynamic optimization strategy further improves the training process to achieve optimal results.

This chapter will offer a detailed explanation of the WaveSenseNet architecture, its main modules, and the physical principles that support them. Then, it will present a series of experiments on a complex structural plate with multiple scatterers to demonstrate the framework's performance benefits over current leading methods. It will also thoroughly analyze its model interpretability and uncertainty quantification features.

5.2 The multimodal framework of multi-scale attention fusion network

The signal acquisition process in WaveSenseNet is the same as that in Chapter 3 and Chapter 4 and the response signals are still pre-processed using wavelet transform [165] to extract temporal features. In addition, a gaussian positional encoding map is designed to represent the environment by modeling obstacles, boundaries, and scattering regions. This encoding provides spatial context, which combined with the temporal features, serves as multimodal input to the network to facilitate accurate ToF extraction.

The fusion of wavelet-transformed signals and positional encodings forms the core of the network. A multi-scale attention mechanism selectively focuses on the most informative features across both temporal and spatial domains. This mechanism assigns different weights to signal regions at various scales, enabling the network to capture critical features for accurate ToF extraction in complex environments. The attention-modulated features are then processed through fully connected layers to refine the representation and extract the ToF.

In summary, by combining the wavelet-transformed temporal features with the spatial context, the network is able to address signal propagation issues, such as scattering from obstacles, boundary reflections and focus on accurately estimating the ToF of scattered waves caused by environmental factors. The flowchart of the method is depicted in Figure 5.1.

The overall workflow of the proposed method is illustrated in Figure 5.1. The framework initiates with a multimodal data input stage, comprising two parallel streams: first, temporal features are extracted from the raw signals using a Wavelet Transform; second, a Gaussian Positional Encoding Map is generated to represent the complex physical environment, including any scatterers. These two modalities of information are subsequently integrated into the WaveSenseNet architecture. Within the network, temporal features processed by a Multi-Scale Feature Extraction Module are combined with spatial features from the positional map through an Attention-based Fusion Module. Ultimately, the framework outputs not only a precise ToF prediction but also its associated uncertainty, which is utilized to achieve high-accuracy damage localization.

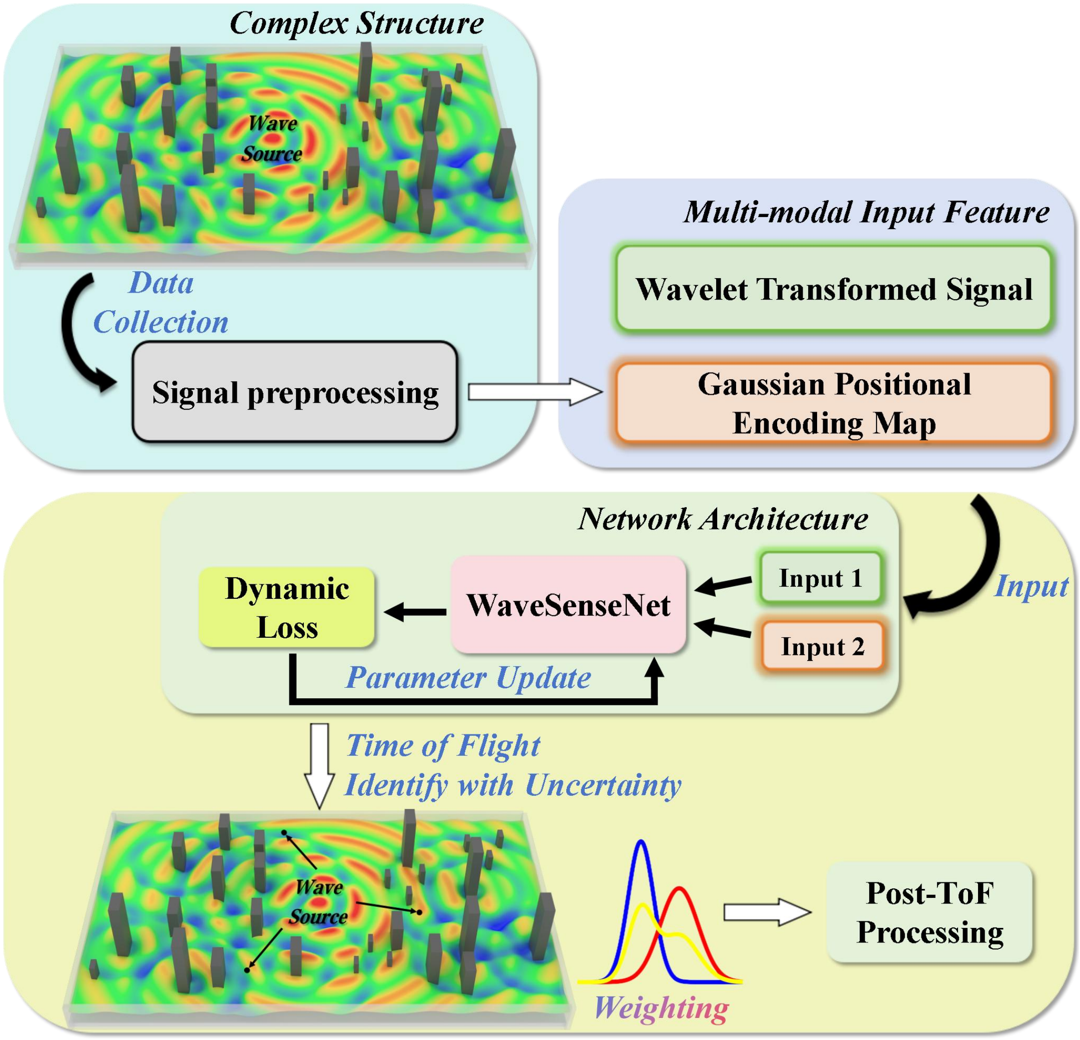


Figure 5.1 Flowchart of the proposed method.

5.2.1 Network Architecture and Preprocessing Framework

The WaveSenseNet architecture incorporates a multimodal framework that is designed to robustly extract ToF measurements in complex environments. This is achieved through a combination of adaptive feature learning, attention mechanisms, and physics-aware optimization. The network's hybrid loss function unifies quantile loss and a MSE term, which is regularized by physical constraints. This ensures probabilistic uncertainty modeling and guarantees wave-dynamic consistency. The total loss, denoted as $\mathcal{L}_{\text{total}}$, is formulated as a dynamic weighted sum:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_Q + \alpha(t)\mathcal{L}_{\text{MSE}} \tag{5.1}$$

where $\alpha(t)$ balances the contributions of the quantile loss $\mathcal{L}_Q$ and the physics-informed MSE loss $\mathcal{L}_{\text{MSE}}$. The quantile loss captures prediction intervals across multiple confidence levels $\tau \in \mathcal{T}$, defined as:

$$\mathcal{L}_Q = \frac{1}{N}\sum_{i=1}^{N} \sum_{\tau\in\mathcal{T}} \left[\tau \cdot \max\left(y_i - \hat{y}_i^{(\tau)}, 0\right) + (1-\tau) \cdot \max\left(\hat{y}_i^{(\tau)} - y_i, 0\right)\right] \tag{5.2}$$

where $\hat{y}_i^{(\tau)}$ denotes the τ-th quantile prediction. The MSE term penalizes deviations from physical plausibility, incorporating a projection operator $\text{Proj}_{\mathcal{P}}$ to enforce bounds $\mathcal{P}$:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N} (y_i - \text{Proj}_{\mathcal{P}}(\hat{y}_i))^2 \tag{5.3}$$

The loss function applies a projection operator $\mathrm{Proj}_{\mathcal{P}}$ to y_i, mapping it into a physically feasible space $\mathcal{P}$. The final MSE is computed between the projected predictions $\mathrm{Proj}_{\mathcal{P}}$ and the ground truth y_i, intrinsically ensuring alignment with physical laws without requiring explicit regularization terms.

The overall architecture of WaveSenseNet is designed to process multimodal inputs through a segmented pipeline, as illustrated in Figure 5.2. The preprocessing of the input wavelet-transformed temporal signal is conducted using a dual-branch architecture. This involves a sequential refinement pipeline consisting of three key components: the residual module, 1D deformable convolution, and the convolutional block attention module (CBAM) [166]. In the first branch, the residual module, constructed by stacking ReLU layers [167], is combined with the CBAM. This branch extracts hierarchical temporal features while maintaining gradient stability through skip connections. The intermediate features are further refined by a CBAM to prioritize discriminative temporal patterns. Channel attention weights $\mathbf{A}_c \in \mathbb{R}^{C\times 1}$ are computed by aggregating global temporal context through average pooling and max pooling operations, followed by a shared multi-layer perceptron (MLP) [168]:

$$\mathbf{A}_c = \sigma\left(\mathrm{MLP}\left(\mathbf{F}_{\mathrm{avg}}\right) + \mathrm{MLP}\left(\mathbf{F}_{\max}\right)\right) \tag{5.4}$$

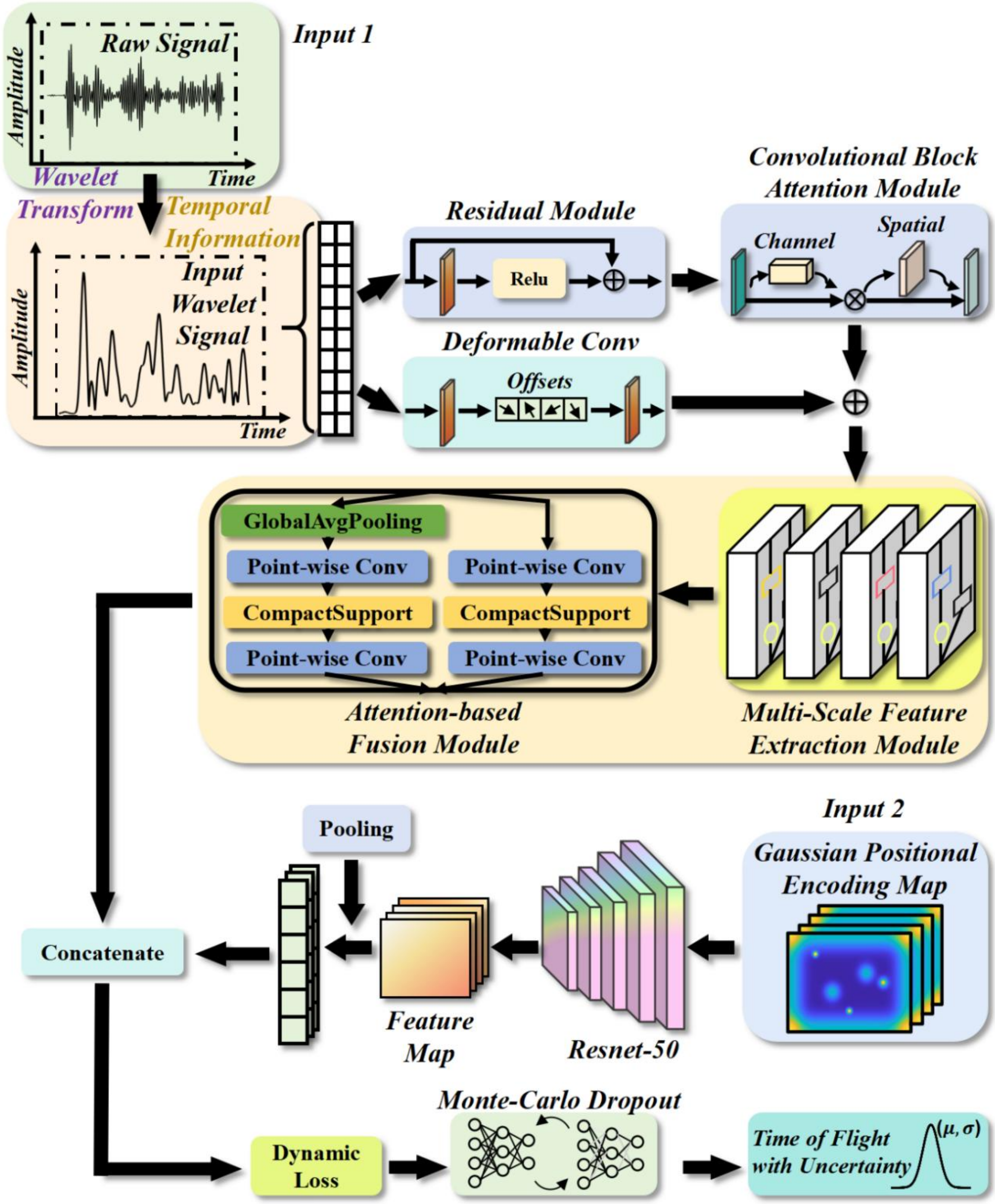


Figure 5.2 Architecture of the proposed multimodal neural network (WaveSenseNet).

where $\mathbf{F}_{\text{avg}}, \mathbf{F}_{\text{max}} \in \mathbb{R}^{C\times1}$ represent pooled features, and σ denotes the sigmoid function. Spatial attention $\mathbf{A}_s \in \mathbb{R}^{1\times T}$ is then derived by concatenating channel pooled features and convolving with a 1D kernel:

$$\mathbf{A}_s = \sigma\left(f^{1\times k}\left(\left[\!\left[\mathbf{F}_{\text{avg}}^{\text{sp}}, \mathbf{F}_{\text{max}}^{\text{sp}}\right]\!\right]\right)\right) \tag{5.5}$$

where k is the kernel size. The refined output of this branch, $\mathbf{F}_{\text{res}} = \mathbf{A}_s \otimes (\mathbf{A}_c \otimes \mathbf{F}_{\text{res-in}})$, suppresses noise while emphasizing critical temporal regions.

In parallel, the second branch employs a 1D deformable convolution layer customized for temporal signal processing. This operation adaptively adjusts the receptive field by learning spatially-varying offset vectors $\Delta p_n \in \mathbb{R}^1$ for each kernel position p_n along the temporal axis. The modulated feature $\hat{x}(t)$ at time step t is computed as:

$$\hat{x}(t) = \sum\nolimits_{p_n \in \mathcal{R}} w(p_n) \cdot x(t + p_n + \Delta p_n) + b \tag{5.6}$$

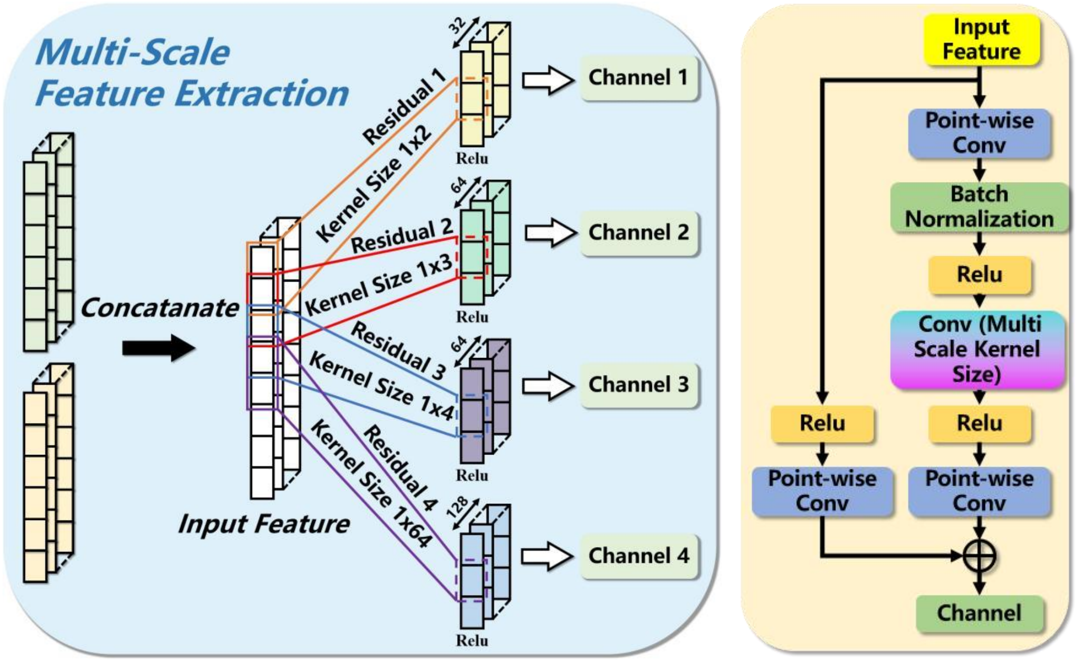
where $\mathcal{R}$ represents the 1D kernel's temporal receptive field, w denotes learnable weights, and b is the bias term. This mechanism dynamically adapts the kernel with non-uniform temporal patterns, such as capturing long-term dependencies in the time series. Finally, the outputs of both branches are fused through element-wise summation:

$$\mathbf{F}_{\text{preprocessed}} = \mathbf{F}_{\text{res}} \oplus \mathbf{F}_{\text{def-conv}} \tag{5.7}$$

This integration combines the robustness of residual attention mechanisms with the adaptive sampling capabilities of deformable convolutions, ensuring high-fidelity feature representation for downstream processing. By jointly addressing noise suppression, temporal alignment, and feature enhancement, the preprocessing pipeline lays a critical foundation for accurate ToF extraction in complex environments.

5.2.2 Architecture of multi-scale feature extraction module

The multi-scale feature extraction module (MSFE) is a critical component of WaveSenseNet, designed to process preprocessed temporal signals through a stacked framework that captures both local and global patterns across multiple temporal scales. As depicted in Figure 5.3(a), the MSFE employs a parallel architecture comprising four branches, each specialized to extract features at a specific scale through convolutional operations with kernel sizes of 2, 3, 4, and 64, respectively. Each branch within the MSFE operates independently to process the input signal at its designated scale. The smaller kernel sizes, such as 2, 3, and 4, focus on local temporal variations, enhancing sensitivity to high-frequency components and abrupt signal transitions. Conversely, the large kernel with a size of 64 captures long-range dependencies and global contextual patterns, which are essential for reducing interference from distributed scatterers.



(a) (b)

Figure 5.3 Architecture of the proposed multi-scale feature extraction module (MSFE): (a) multi-scale parallel branch architecture; (b) scale-specific processing unit.

A detailed view of an individual scale branch is illustrated in Figure 5.3(b). Each branch incorporates a residual processing unit optimized for stability and efficiency. The input features first processed through point-wise convolution with a kernel size of 1 to reduce channel dimensionality, followed by BN and ReLU activation. This compressed representation is then processed by a convolutional layer with a branch-specific kernel size, enabling scale-specific feature extraction. To mitigate gradient degradation and enhance feature reuse, a residual connection adds the original input to the output of the convolutional layer [169], as formulated below:

$$\mathcal{B}_s(\mathbf{F}_{\text{out}}) = \underbrace{f^{1\times1} \circ \text{ReLU}}_{\text{Residual Path}}(\mathbf{F}_{\text{in}}) + \underbrace{f^{1\times1} \circ \text{ReLU} \circ f^{k_s\times1} \circ \text{ReLU} \circ \text{BN} \circ f^{1\times1}}_{\text{Scale-specific Conv Block}}(\mathbf{F}_{\text{in}}) \tag{5.8}$$

Where $\mathcal{B}_s(\cdot)$ represents the processing of the $s-$th scale branch. $f^{k_s\times1}$ denotes the convolution operation with kernel size k_s for scale s. BN stands for batch normalization [114] and ReLU represents rectified linear unit activation. $\circ$ signifies the function composition operator. By synergizing residual learning and adaptive gating with multi-scale feature extraction, the framework lays the foundation for robust ToF estimation in subsequent modules through hierarchical signal decomposition.

5.2.3 Architecture of attention-based fusion module

The attention-based fusion module (ABFM) in WaveSenseNet adaptively integrates multi-scale temporal features from the preceding MSFE stages, emphasizing critical signal components while suppressing noise and redundant information. As depicted in Figure 5.4, the ABFM processes input features through a Structured architecture that combines multi-scale channel attention, spatially adaptive weighting, and a novel parametric activation function (PAF). This innovative framework addresses the spectral limitations of conventional attentional fusion models, such as AFF [170], by capturing high-frequency wave packets essential for precise ToF extraction in cluttered environments.

Given four multi-scale input features $[\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3, \mathbf{F}_4] \in \mathbb{R}^{C_i \times T \times S}$ generated by MSFE, with C_i channels and feature map of size $T \times S$. The fusion begins with pairwise interactions between maximally divergent scales ($\mathbf{F}_1$ and $\mathbf{F}_4$) and minimally divergent scales ($\mathbf{F}_2$ and $\mathbf{F}_3$). For the $\mathbf{F_1} - \mathbf{F_4}$ fusion pair, initial convolutional operations are applied:

$$\mathbf{F}_1^{'} = f^{1\times1}(\mathbf{F}_1), \mathbf{W}_1 \in \mathbb{R}^{64\times C_{in}\times 1} \tag{5.9}$$

$$\mathbf{F}_4^{'} = f^{1\times1}(\mathbf{F}_4), \mathbf{W}_4 \in \mathbb{R}^{64\times C_{in}\times 1} \tag{5.10}$$

where $f^{1\times1}$ denotes the 1D convolution operator for kernel size with 1×1, and $\mathbf{W}_i$ represents the parameters of the $k-\text{th}$ projection kernel applied to feature $\mathbf{F}_i$, reducing its channel dimension from C_{in} to 64. This ensures $\mathbf{F}_1^{'}, \mathbf{F}_4^{'} \in \mathbb{R}^{64\times T\times S}$, enabling subsequent operations. Subsequently, the mixed representation is obtained by element-wise summation of the two aligned features:

$$\mathbf{X}_{\text{mix}} = \mathbf{F}_1^{'} \oplus \mathbf{F}_4^{'} \tag{5.11}$$

Where ⊕ represents the element-wise summation.

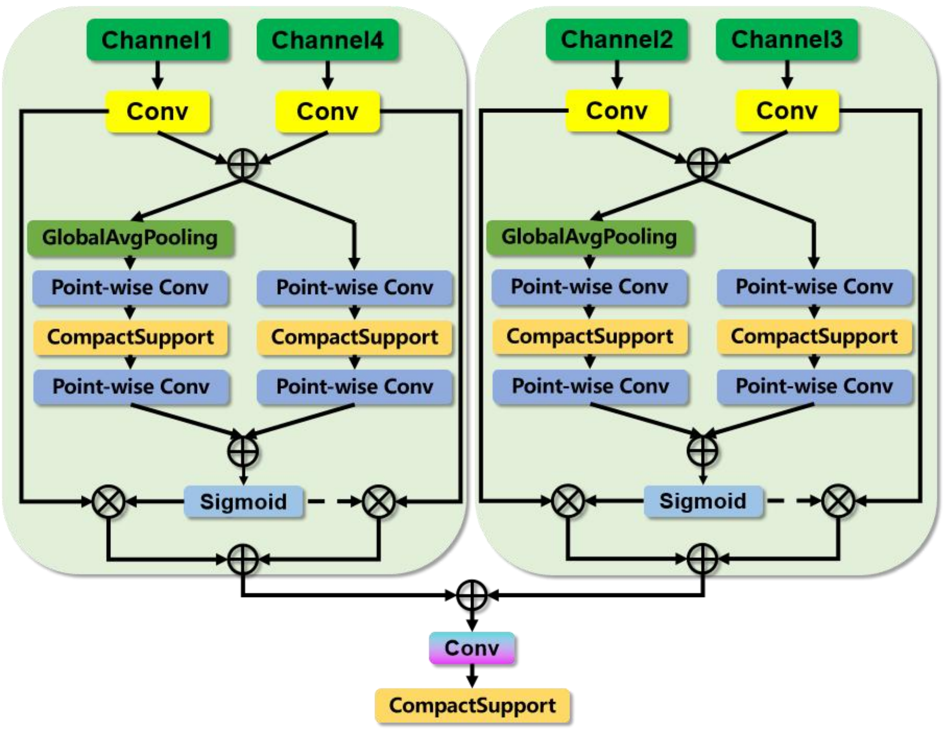


Figure 5.4 Architecture of the proposed attention-based fusion module (ABFM).

The hybrid tensor is then fed into two parallel pathways: a local context branch employing sequential convolutions and the parametric activation function (PAF) to capture fine-grained spatial dependencies, and a global context branch leveraging global average pooling to aggregate channel-wise statistics. The local branch refines features as:

$$\mathbf{X}_{\text{local}} = f^{1\times1} \circ \text{PAF} \circ f^{1\times1}(\mathbf{X}_{\text{mix}}) \tag{5.12}$$

where PAF denotes the tunable compact-support activation function, which is formulated as follows:

$$\text{PAF}\,(x) = \cos\,(cx) \cdot \mathrm{e}^{-ax} \cdot \max\left(0, 1 - \mathrm{e}^{-bx}\right) \tag{5.13}$$

with learnable parameters a, b, c controlling decay rate and phase modulation in frequency domain, which can effectively mitigate the issue of neural network struggling to learn high-frequency information. The global branch computes as:

$$\mathbf{X}_{\text{global}} = f^{1\times1} \circ \text{PAF} \circ f^{1\times1} \circ \text{GAP}\,(\mathbf{X}_{\text{mix}}) \tag{5.14}$$

where GAP(·) compresses spatial dimensions into a global descriptor. The outputs of both branches are combined through sigmoid-activated tensor summation to generate dynamic attention weights $W_{\text{att}} \in [0,1]$:

$$W_{\text{att}} = \sigma\left(\mathbf{X}_{\text{local}} \oplus \mathbf{X}_{\text{global}}\right) \tag{5.15}$$

where σ is the sigmoid function, $\oplus$ denotes element-wise summation. Finally, the fused output $\mathbf{F}_{\text{fused}}$ is computed as a weighted combination of the aligned features:

$$\mathbf{F}_{\text{fused}} = W_{\text{att}} \odot \mathbf{F}_1^{'} + (1 - W_{\text{att}}) \odot \mathbf{F}_4^{'} \tag{5.16}$$

Where $\odot$ is element-wise multiplication.

To rigorously analyze the spectral properties of the PAF and its role in high-frequency feature enhancement, we provide formal mathematical derivations that underpin its design. For detailed derivations, please refer to Appendix *A.3*.

5.2.4 Gaussian positional encoding map

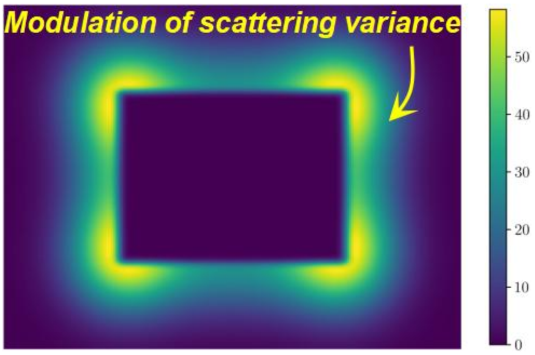
The gaussian positional encoding map in WaveSenseNet is a unified framework designed to model wave scattering effects induced by obstacles of varying geometries and scales. By spatially distributing gaussian kernels [171], this encoding translates geometric features into continuous intensity fields, capturing reflection, diffraction, and attenuation phenomena critical for guided wave propagation analysis. The core idea lies in adapting the density and variance of gaussian kernels to the physical scale of obstacles, ensuring both accuracy and computational efficiency.

For an obstacle occupying a geometric domain Ω, the encoding field $\mathcal{G}(x)$ is defined as a superposition of gaussian kernels centered at scattering sources $\{p_i\}$:

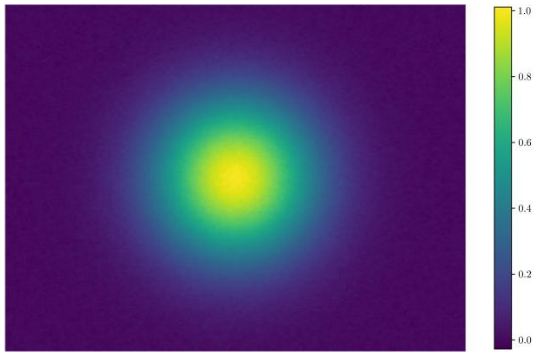
$$\mathcal{G}(x) = \sum_{p_i \in \mathcal{S}} e^{\left(-\frac{\|x-p_i\|^2}{2\sigma^2}\right)} \cdot \mathbb{I}(x \notin \Omega) \tag{5.17}$$

where $\mathcal{S}$ represents the set of scattering sources, σ denotes the Gaussian standard deviation controlling the spatial spread of scattering, and $\mathbb{I}(\cdot)$ is the indicator function nullifying the field inside Ω.

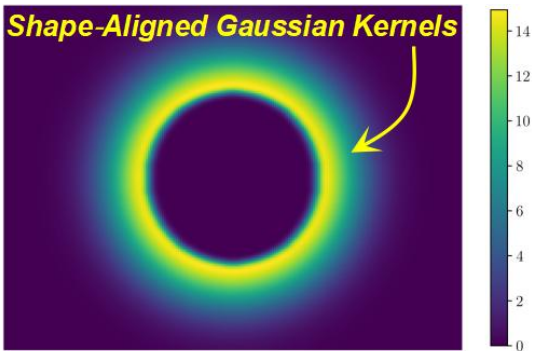
The gaussian positional encoding framework adapts to obstacles of different scales by employing two distinct encoding strategies based on the relationship between the obstacle's size (L) and the wavelength (λ) . For obstacles significantly larger than the wavelength ($L \gg \lambda$), the Boundary-Sampled Encoding method is utilized. This approach densely samples scattering sources along the obstacle's boundaries to resolve fine-grained details, capturing critical phenomena like edge diffraction and multi-path reflections. Conversely, for obstacles with dimensions comparable to the wavelength $(L \sim \lambda)$, the framework switches to centroid encoding, where a single Gaussian kernel centered at the obstacle's centroid suffices. This simplification leverages the localized energy concentration characteristic of wavelength-scale scatterers, where centralized reflections dominate over boundary effects.



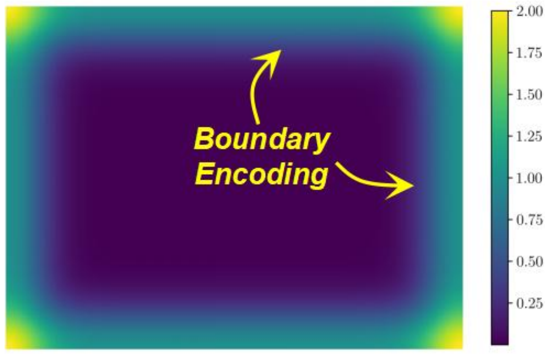
(a)



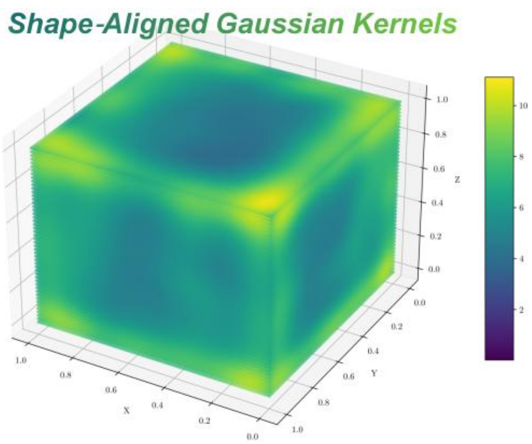
(b)



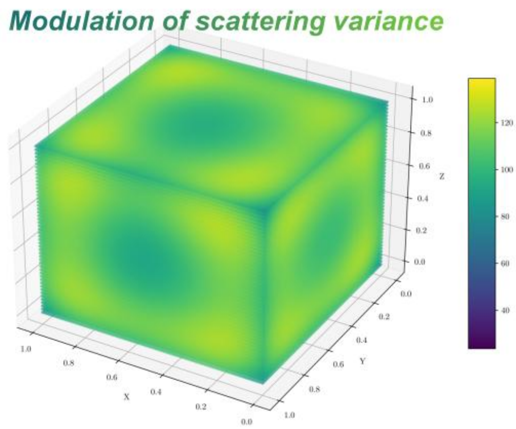
(c)



(d)



(e)



(f)

Figure 5.5 Visualizing scatter intensity using gaussian positional encoding in guided wave propagation: (a) Rectangular obstacle encoding ($\sigma = 0.03$); (b) Small-scale obstacle encoding ($\sigma = 0.01$); (c) Circular obstacle encoding ($\sigma = 0.03$); (d) Boundary scattering encoding ($\sigma = 0.03$); (e) 3D cuboid surface encoding ($\sigma = 0.1$); (f) 3D cuboid surface encoding ($\sigma = 0.2$).

Figure 5.5 demonstrates the framework's adaptability across diverse obstacle types and scales. As depicted in figure 5.5(a), a rectangular obstacle with length L and width W requires explicit modeling of edges and corners. Each edge is sampled with M points, and corners are augmented with additional Gaussian kernels and K points to mitigate sharp diffraction effects：

$$\mathcal{G}(x) = \sum_{edges} \sum_{m=1}^{M} e^{\left(-\frac{(x-x_m)^2+(y-y_m)^2}{2\sigma^2}\right)} + \sum_{corners} \sum_{k=1}^{K} e^{\left(-\frac{(x-x_k)^2+(y-y_k)^2}{2\sigma^2}\right)} \tag{5.18}$$

where (x_m, y_m) represents the edge sampling points and (x_k, y_k) denotes the corner points to smooth scattering at acute angles.

For obstacles with dimensions comparable to the wavelength, such as bolts and small fixtures, scattering energy concentrates near the centroid due to limited geometric complexity, as shown in figure 5.5(b). A single gaussian kernel suffices:

$$\mathcal{G}_{centroid}(x, y) = e^{\left(-\frac{(x-x_0)^2+(y-y_0)^2}{2\sigma_{centroid}^2}\right)} \tag{5.19}$$

where (x_0, y_0) represents the centroid coordinates of the obstacle and σ_{centroid} is the calibrated based on the obstacle's effective scattering range.

For a circular obstacle with radius r, N points are uniformly sampled along the circumference, as sketched in Figure 5.5(c):

$$x_i = x_c + r\cos\theta_i,\ y_i = y_c + r\sin\theta_i,\ \theta_i = \frac{2\pi i}{N} \tag{5.20}$$

The encoding field combines radial symmetry and boundary integrity:

$$\mathcal{G}_{circle}(x, y) = \sum\nolimits_{n=1}^{N} e^{\left(-\frac{(x-x_n)^2+(y-y_n)^2}{2\sigma^2}\right)} \cdot \mathbb{I}((x-x_c)^2 + (y-y_c)^2 > r^2) \tag{5.21}$$

The boundary scattering model adopts the same edge sampling strategy as the rectangular obstacle encoding but omits corner enhancements, as illustrated in Figure 5.5(d):

$$\mathcal{G}_{boundary}(x, y) = \sum\nolimits_{s=1}^{S} e^{\left(-\frac{(x-x_s)^2+(y-y_s)^2}{2\sigma^2}\right)} \tag{5.22}$$

where (x_s, y_s) represents the uniformly sampled points along the boundary.

As shown in figure 5.5(e) and 5.5(f), for cuboid obstacles, gaussian kernels are distributed across all six faces. Each face $\mathcal{F}$ contributes:

$$\mathcal{G}_{\mathcal{F}}(x,y,z) = \sum_{i,j=1}^{Q} e^{\left(-\frac{(x-x_{ij})^2+(y-y_{ij})^2+(z-z_{ij})^2}{2\sigma^2}\right)} \tag{5.23}$$

where x_{ij}, y_{ij}, z_{ij} are perturbed grid points on the face. The total 3D field aggregates contributions from all faces, enabling multi-directional scattering modeling.

5.2.5 Uncertainty estimation of time of flight through Monte Carlo dropout

In complex environments, the accuracy of ToF estimation is critical for reliable wave propagation analysis. However, neural network predictions intrinsically carry uncertainties due to input noise, environmental variability, or model limitations. To quantify this uncertainty and provide confidence intervals for ToF predictions, we integrate Monte Carlo dropout into WaveSenseNet, enabling probabilistic inference without modifying the network architecture.

In Bayesian deep learning, the goal is to infer the posterior distribution over network weights θ given training data $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$:

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} \tag{5.24}$$

where $p(\theta)$ is the prior distribution over weights, and $p(\mathcal{D}|\theta)$ is the likelihood. However, exact posterior inference is intractable for large models. Variational inference approximates the true posterior with a simpler distribution $q_\phi(\theta)$, parameterized by ϕ, by minimizing the Kullback-Leibler (KL) divergence:

$$\phi^* = \arg\min_\phi \mathrm{KL}\big(q_\phi(\theta) \parallel p(\theta|\mathcal{D})\big) \tag{5.25}$$

This is equivalent to maximizing the evidence lower bound (ELBO):

$$\mathcal{L}(\phi) = \mathbb{E}_{q_\phi(\theta)}[\log p(\mathcal{D}|\theta)] - KL\big(q_\phi(\theta) \parallel p(\theta)\big) \tag{5.26}$$

Dropout training in neural networks implicitly defines a variational distribution $q_\phi(\theta)$, where each weight matrix $\mathbf{W}$ is represented as:

$$\mathbf{W} = \mathbf{M} \cdot \mathrm{diag}\,(\epsilon),\ \epsilon_i \sim \mathrm{Bernoulli}\,(p) \tag{5.27}$$

For dropout probability p, mask matrix $\mathbf{M}$, and Bernoulli random variables ϵ. This corresponds to a Bernoulli variational distribution over weights.

During inference, MC dropout approximates Bayesian model averaging by performing T stochastic forward passes with dropout enabled. The predictive distribution for ToF y given input x is:

$$p(y|x,\mathcal{D}) \approx \frac{1}{T}\sum_{t=1}^{T} p(y|x,\theta_t),\ \theta_t \sim q_\phi(\theta) \tag{5.28}$$

where θ_t denotes weights sampled via dropout masks at pass t.

For ToF estimation, let $\hat{y}_t = f_\theta(x, \epsilon_t)$ denote the network's prediction in the t-th forward pass with dropout mask ϵ_t. The predictive mean μ_{ToF} and variance σ^2_{ToF} are computed as:

$$\mu_{\mathrm{ToF}} = \frac{1}{T}\sum_{t=1}^{T} \hat{y}_t \tag{5.29}$$

$$\sigma^2_{\mathrm{ToF}} = \frac{1}{T}\sum_{t=1}^{T} \hat{y}_t^2 - \left(\frac{1}{T}\sum_{t=1}^{T} \hat{y}_t\right)^2 \tag{5.30}$$

Here, σ^2_{ToF} captures epistemic uncertainty, reflecting the model's confidence in its predictions due to parameter uncertainty.

5.3 Case study and analysis

5.3.1 Experiment setup

The guided wave data in this study were experimentally measured on a 500 mm × 700 mm × 2 mm aluminum plate, with eight PZT transducers mounted on the surface using epoxy resin for Lamb wave actuation and sensing. As illustrated in Figure 5.6, black cylindrical components are positioned at manually selected locations across the plate to simulate damage. This study utilizes bolt-in columns with stiffeners, where bolts are anchored by radial stiffeners to enhance structural stability. These components introduce scattering effects, including boundary reflections, localized scattering from the bolt heads, and diffraction caused by the stiffener geometry.

Three PZTs are designated as dedicated sensors, while the others can dynamically function as either sensor, depending on the configuration of the excitation channels. By altering the excitation-receiver pairs among the PZTs, distinct wave propagation paths are generated, capturing interactions between direct waves, boundary reflected waves, and signals modulated by the bolt-stiffener assembly. This dynamic configuration yields three unique datasets comprising a total of 495 samples, each divided into an 80% training subset and a 20% testing subset, to rigorously evaluate the robustness of the proposed ToF extraction method under varying scattering conditions.

The experimental system works as follows, as shown in Figure 5.7: a KEYSIGHT 33500B waveform generator is used to output the excitation signal. A power amplifier is used to amplify the excitation voltage for the PZT actuator. The sensor captures and records the propagating Lamb waves by the NI PXIe-5105 data acquisition module. The excitation signal is a Hann-windowed sinusoidal tone burst for all sets. The central frequency is selected with the following considerations: 1) the amplitude of response signals should be as high as possible; 2) the wavelength should be comparable to the size of the damage so that the damage can serve as a strong scatter; 3) the waves should be as clean as possible, i.e., the waves with fewer modes are preferable. Guided by these principles, the excitation frequency is selected as 200kHz. Examining the waves in the response signals, the S0 mode Lamb waves dominate the responses.

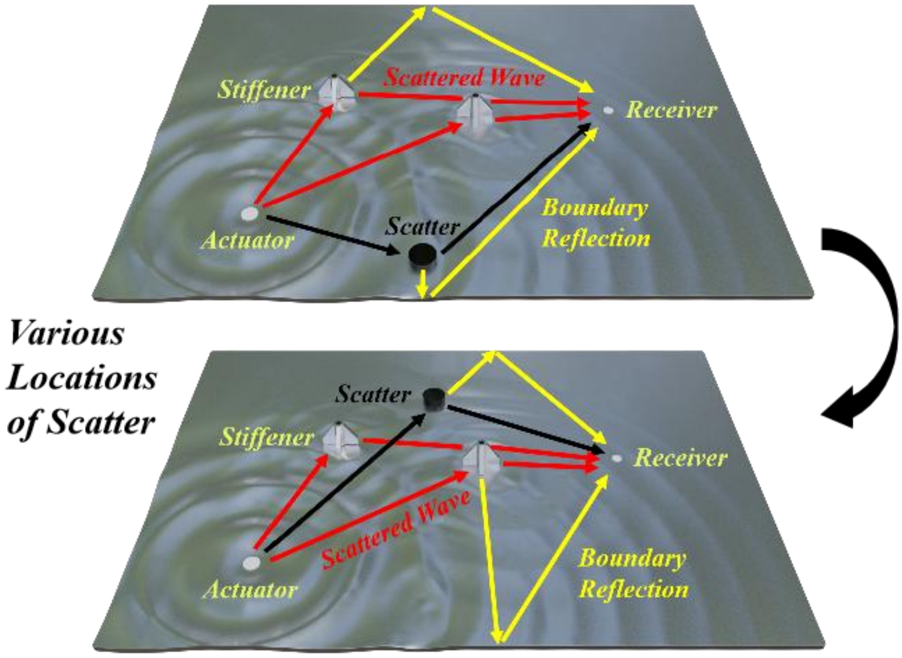


Figure 5.6 Schematic illustration of the data collection method.

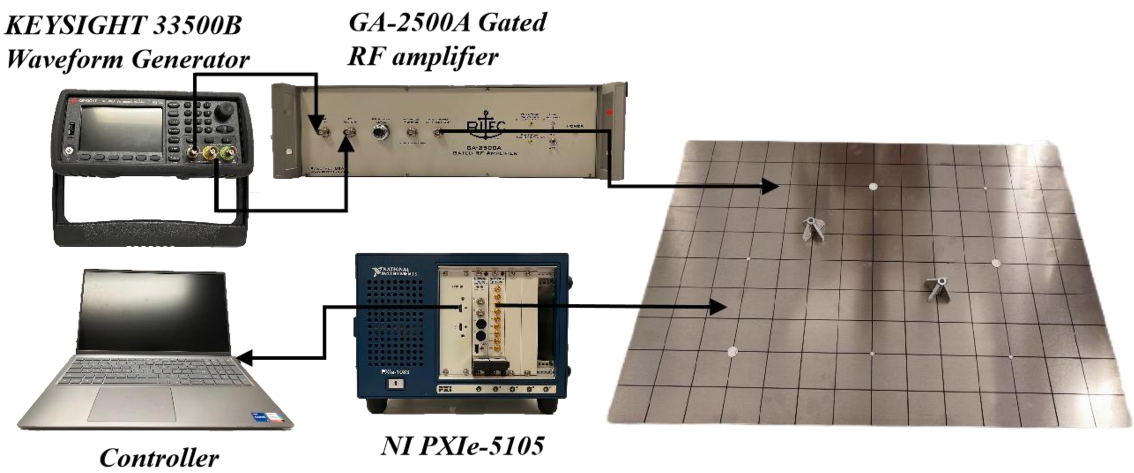


Figure 5.7 Experimental setup.

5.3.2 Customized Gaussian positional encoding map

In guided wave-based SHM, the scattering characteristics of PZTs and bolt-in columns require a single gaussian kernel modeling due to their dimensions being comparable to the wavelength. Building on the general gaussian positional encoding framework introduced in section 5.2.4, this study customizes the encoding scheme to model scattering characteristic in plate structures. The adaptation synthesizes contributions from actuators, scatterers, and boundaries, with the following key parameterizations:

1.Actuator PZTs

Three actuators are modeled using a single gaussian kernel, based on equation (5.19), centered at their coordinates (x_0, y_0) with a narrow variance $\sigma_{\text{piezo}} = 0.01$, reflecting their localized scattering effects.

2. bolt-in columns with stiffeners

Bolt-in columns are encoded with a broader gaussian kernel with $\sigma_{\text{scatterer}} = 0.04$ to approximate their limited geometric complexity and effective scattering range, as defined in Equation (5.19).

3. Boundary Reflections

Structural boundaries are modeled by uniformly sampling points along each edge and superimposing gaussian kernels at these locations, as illustrated in Equation

(5.22). Corner enhancements are omitted for computational simplicity, and the variance is set to $\sigma_{\text{boundary}} = 0.05$ to govern the spatial decay of reflections.

4. Excitation-Specific Enhancement

When a PZT as as an actuator, an additional gaussian term with enhanced variance $\sigma_{\text{relative}} \in (0.001, 0.001, 0.02)$ is superimposed at its location to amplify the excitation strength in the proximal region.

The final encoding map $I(x, y)$ integrates all components:

$$I(x,y) = \sum_{i=1}^{N_{\text{piezo}}} G_{\text{piezo},i}(x,y) + \sum_{j=1}^{N_{\text{scatterer}}} G_{\text{scatterer},j}(x,y) + \sum_{k=1}^{4} G_{\text{boundary},k}(x,y) + \sum_{m=1}^{N_{\text{active}}} G_{\text{relative},m}(x,y) \tag{5.31}$$

Receiver PZTs, being smaller and acting as passive sensors, are marked as red dots without gaussian encoding, as depicted in Figure 10. The spatial resolution of the map is set to 200×200 grid points over a $0.7\ \text{m} \times 0.5\ \text{m}$ plate. Figure 5.8 illustrates the intensity distribution, where brighter regions correspond to stronger scattering effects from actuators, bolt-stiffener, and boundaries, while receiver positions are explicitly highlighted. Regarding the Gaussian parameterization, the variance σ is tuned based on the ratio between the dominant wavelength of the guided waves and the characteristic dimensions of the scatterers. This strategy ensures that the spatial encoding reflects the effective scattering area, providing a physically consistent and computationally efficient representation of guided wave interactions in SHM.

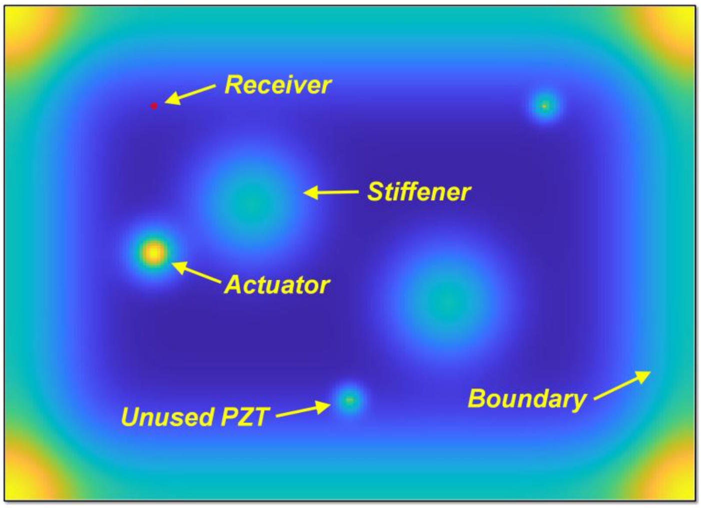


Figure 5.8 Visualizing scatter intensity using gaussian positional encoding in guided wave.

5.3.3 Physics-Constrained Adaptive MSE-Quantile Loss

The Physics-Constrained Adaptive MSE-Quantile Loss (PCAMQ-Loss) extends the general hybrid loss framework, as per equation (3), for guided wave-based ToF prediction. It integrates quantile regression to model uncertainty and incorporates a dynamically weighted physics-informed term. For this case study, the physics constraints are explicitly embedded using pre-calibrated wave speeds of three PZT actuators, $[5.229, 5.3276, 5.3011]\times10^3$ m/s, which were determined through experimental calibration. The predicted and ground-truth ToF values $(\hat{y}_i, y_i)$ are converted to flight distances via $d = vt$, ensuring adherence to wave propagation principles:

$$d_{\text{pred}}^{(i)} = \hat{y}_i \cdot v^{(i)},\ d_{\text{true}}^{(i)} = y_i \cdot v^{(i)} \tag{5.32}$$

where $v^{(i)}$ is dynamically selected for each sample. The physics-constrained MSE term is then computed as:

$$\mathcal{C}_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}\left(d_{\text{pred}}^{(i)} - d_{\text{true}}^{(i)}\right)^2 \tag{5.33}$$

The dynamic weight $\alpha(t)$, initialized at 0.5, adaptively balances the quantile loss $(\tau = 0.45)$, as shown in equation (3), and (4). It is updated based on their relative magnitudes:

$$\Delta = \mathcal{C}_{\text{MSE}} - \mathcal{C}_{\text{Q}}, A = [1 + 0.1 \cdot \tanh(\Delta)] \tag{5.34}$$

$$\text{If } \Delta > 0,\ \omega = 0.995 \cdot [1 + 0.1 \cdot \tanh(\Delta)] \tag{5.35}$$

$$\text{If } \Delta < 0,\ \omega = 1.002/[1 + 0.1 \cdot \tanh(\Delta)] \tag{5.36}$$

$$\alpha(t) = \text{clip}\,(\alpha(t-1) \cdot \omega, 0.01, 0.8) \tag{5.37}$$

where tanh ensures smooth transitions, and clipping prevents dominance of either term. Implementation details, including wave speed indexing and weight monitoring by a custom callback, are outlined in Appendix *A.4*. This approach maintains the flexibility of the general framework while enabling robust ToF extraction in scattering environments through case-specific adaptations, effectively guiding the network to converge faster.

5.3.4 Performance evaluation

The MSE and the MAE between the predicted ToF and the actual ToF associated with the scattered waves from damage are used to evaluate the performance of the designed network. The two indices are mathematically expressed as:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|y^{(i)} - \hat{y}^{(i)}\right| \tag{5.38}$$

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - \hat{y}^{(i)}\right)^2 \tag{5.39}$$

where N is the total number of samples, $y^{(i)}$ is the actual arrival time, and $\hat{y}^{(i)}$ is the predicted arrival time. Given that the predicted arrival time is measured in seconds (s), the units for MAE and MSE are seconds (s) and seconds squared (s^2), respectively.

The training dynamics of the PCAMQ-Loss are illustrated in Figure 5.9, which depicts the evolution of loss components and dynamic weight across 200 epochs, with per-batch resolution to highlight fine-grained convergence behavior. The total loss, physics-constrained MSE loss, quantile loss, and validation loss exhibit rapid convergence, stabilizing within approximately 700 batches. Notably, the total loss decreased rapidly during the initial phase (first 200 batches) and entered a plateau thereafter, while the validation loss followed a parallel trajectory, indicating robust generalization without overfitting.

The dynamic weight $\alpha(t)$, which governs the balance between physical consistency and uncertainty modeling, rapidly reached its upper bound of 0.8 within 100 batch index. Despite fluctuations caused by shifts in data distribution, $\alpha(t)$ consistently reverted to 0.8, highlighting the resilience of the adjustment mechanism. This behavior is attributed to the integration of a tanh facilitated smooth transition and strict clipping, which collectively suppress oscillations while maintaining adaptive responsiveness. The stability of $\alpha(t)$ further corroborates the efficacy of the proposed strategy in harmonizing competing objectives.

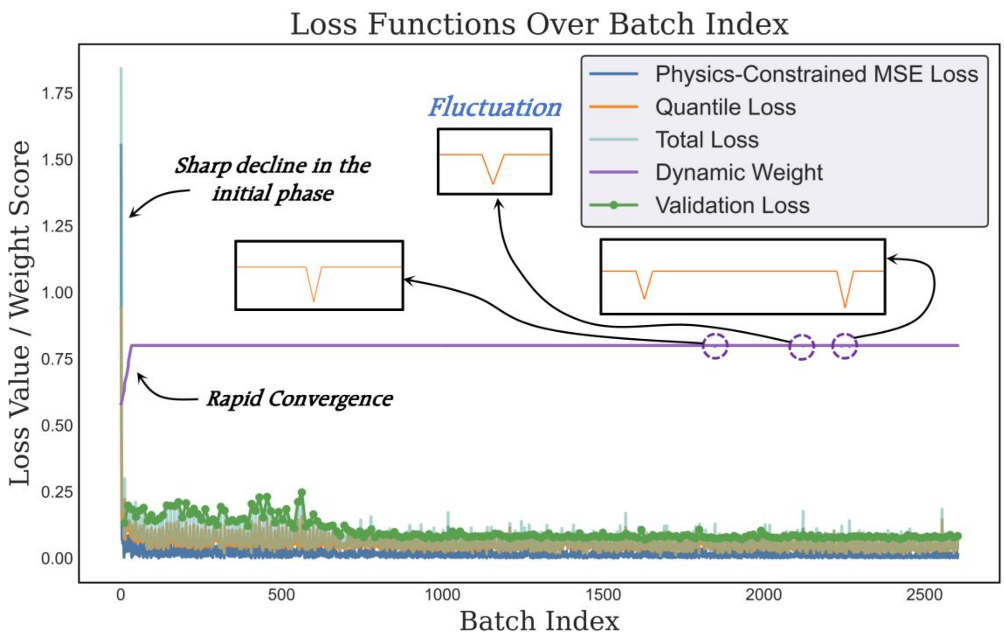


Figure 5.9 Impact of dynamic weight adjustment on loss function across batches.

By the 700th batch, all loss components had converged, demonstrating the efficiency of PCAMQ-Loss in balancing physical constraints with probabilistic modeling. The balance between quantile regression and dynamically weighted physics-aware optimization not only accelerates convergence but also ensures reliable ToF predictions in complex scattering environments.

Table 5.1 presents a comparative analysis of the performance of our proposed method with deep learning approaches, such as Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Recurrent Neural Network(RNN) [125], Long Short-Term Memory(LSTM) [172], Gated Recurrent Unit(GRU) [173], Autoencoder [174], Transformer [149], AlexNet [131], ResNet 18 [130], ResNet 50, VGG-16 [151], EfficientNetB0 [152], EfficientNetB1, EfficientNetB3, InceptionV3 [175], SENET [153], MSST [154].

In the deep learning comparisons, only the temporal part of our network was replaced, keeping the gaussian encoding map processing component unchanged to better validate our network's performance.

The experimental results demonstrate that our proposed method achieves superior performance compared to both existing approaches. Specifically, our model outperforming state-of-the-art deep learning architectures included in the comparison. The results highlight a 1 orders of magnitude improvement in MAE and MSE over conventional methods. This enhanced performance is a result of the network's capability to integrate multimodal information, combining wavelet signal data with gaussian positional encoding maps, to explicitly model wave scattering physics. By embedding physical priors within the encoding maps and adaptively balancing feature extraction, our framework effectively captures spatiotemporal relationships and scattering dynamics, enabling precise ToF prediction in complex environments.

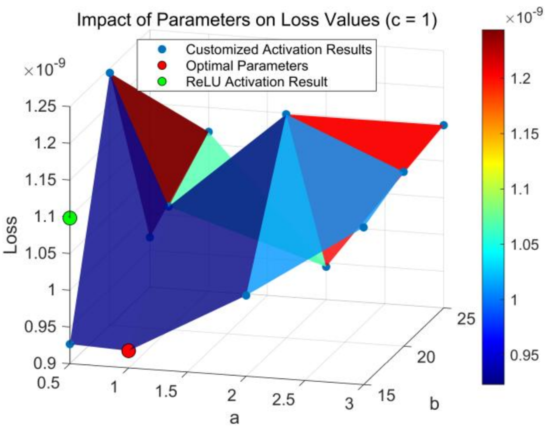
Table 5.1 Comparison between the performance of the proposed method and other deep learning approaches.

Model	**MAE**	**MSE**
MLP	3.53454357922E-05	1.95031079379E-09
CNN	3.35166029796E-05	1.65606449719E-09

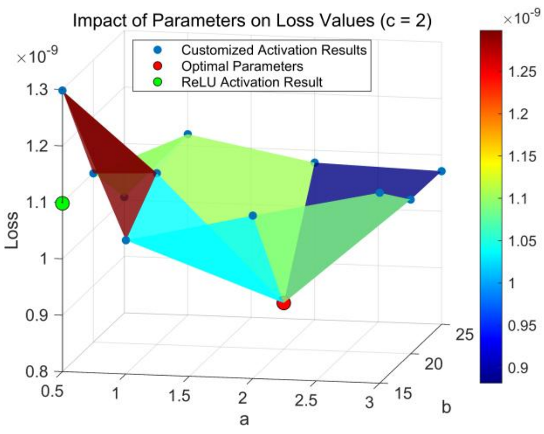
RNN	3.42191877903E-05	1.80528505715E-09
LSTM	3.43224901002E-05	1.90683916167E-09
GRU	3.66988607251E-05	2.09853111786E-09
Autoencoder	6.98401437508E-05	6.65241183528E-09
Transformer	3.52119441152E-05	2.07352861909E-09
Alexnet	3.70707499233E-05	2.17368114930E-09
ResNet 18	3.57920393623E-05	2.01982647915E-09
ResNet 50	3.37908063162E-05	1.76401850357E-09
VGG-16	3.60155133213E-05	1.96644053264E-09
EfficientNetB0	5.18015302141E-05	3.83060745878E-09
EfficientNetB1	3.48821682984E-05	1.86869121891E-09
EfficientNetB3	3.45374259372E-05	1.85806844756E-09
InceptionV3	4.92186197545E-05	3.66064251223E-09
DenseNet-121	2.72133221612E-05	1.10999577220E-09
SENET	3.67692119063E-05	2.16199453319E-09
MSST	2.85966783237E-05	1.18684643454E-09
Our Model	2.40286013326E-05	8.81687960652E-10

5.3.5 Performance analysis of parametric activation function

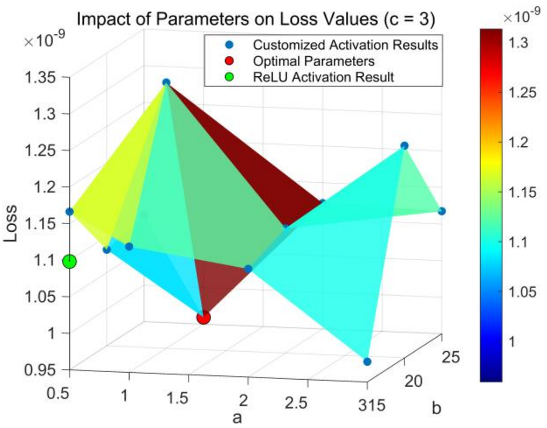
The efficacy of the proposed parametric activation function is validated through two complementary analyses: parameter sensitivity evaluation and frequency-domain feature characterization. Systematic experiments are conducted with $a \in \{0.5,1,2,3\}$, $b \in \{15,20,25\}$, and $c \in \{1,2,3\}$ to assess the impact of parameter configurations on model performance. As illustrated in Figure 5.10, the MSE surfaces under different c values show that the parametric activation function generally performs better than ReLU across most tested parameter combinations. For example, at $c = 1$, the optimal parameter set ($a = 1, b = 15$) achieves a lower MSE compared to ReLU's best result. Similar trends are observed for $c = 2$ and $c = 3$, where the proposed function exhibits significant MSE reductions relative to ReLU, as evidenced by the relative positions of minima in the error landscapes.



(a)



(b)



(c)

Figure 5.10 Impact of parameters a, b, and c on loss values in parametric activation function: (a) Impact of parameters a and b on loss values ($c = 1$); (b) Impact of parameters a and b on loss values ($c = 2$); (c) Impact of parameters a and b on loss values ($c = 3$).

To further investigate the mechanism behind this improvement, frequency-domain analysis is performed on features extracted after the ABFM. Figure 5.11 compares the relative log amplitude of Fourier-transformed feature maps between the proposed activation function and ReLU. The parametric activation function demonstrates superior retention of high-frequency components (normalized frequency $> \pi/2$), with notably slower amplitude decay compared to ReLU. Quantitative analysis reveals a 4.14 dB advantages in the high frequency regime, calculated as:

$$\Delta \mathrm{dB} = 20 \cdot \log_{10}\left(\frac{\text{Avg. Amplitude (Proposed)}}{\text{Avg. Amplitude (ReLU)}}\right) \tag{5.40}$$

In contrast, ReLU features exhibit rapid attenuation beyond $\pi/2$, failing to preserve critical high-frequency information. This capability is particularly vital for ToF prediction in complex scattering environments, where weak high-frequency wave packets-generated by multipath reflections carry essential timing information. The enhanced spectral response of the proposed activation function directly contributes to its improved accuracy in resolving subtle temporal shifts.

These results collectively underscore the dual advantages of the proposed activation function: (1) The adjustable parameters enable customized optimization for diverse ToF extraction tasks, dynamically adapting to scattering characteristics across varying environments and (2) overcomes the inherent low-frequency preference of conventional neural networks, effectively resolving high-frequency wave packets critical for precise ToF estimation in complex scattering scenarios.

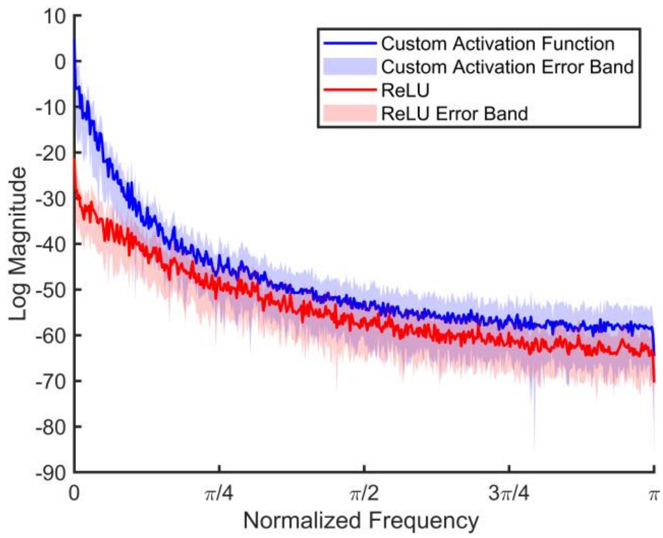
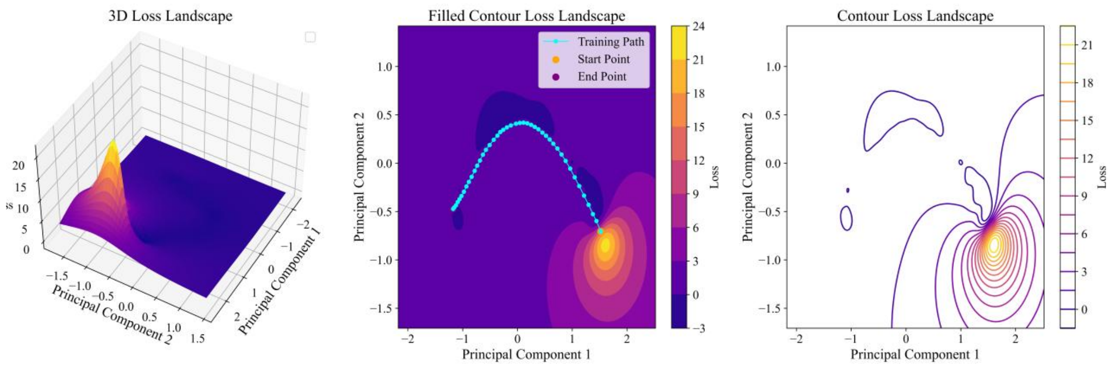


Figure 5.11 Relative log amplitude of fourier transformed feature maps.

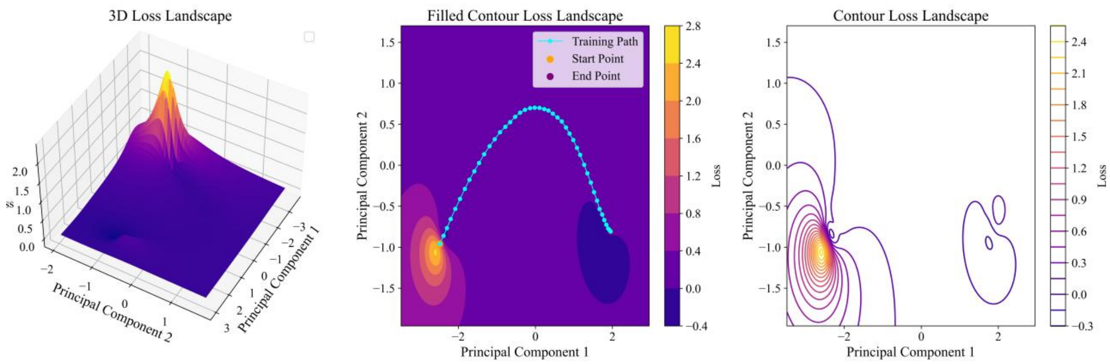
5.3.6 Performance analysis of parametric activation function

The analysis of loss landscapes provides critical insights into the optimization behavior of different loss functions, particularly for high-dimensional models with 43 million parameters in WaveSenseNet. Traditional methods for loss landscape visualization, which rely on exhaustive sampling of the parameter space, are computationally expensive and impractical for large-scale networks. To tackle this challenge, we propose a computationally efficient framework that combines incremental dimensionality reduction, adaptive interpolation, and selective Hessian analysis. Key steps of the methodology are summarized below, with implementation details provided in Appendix *A.5*.

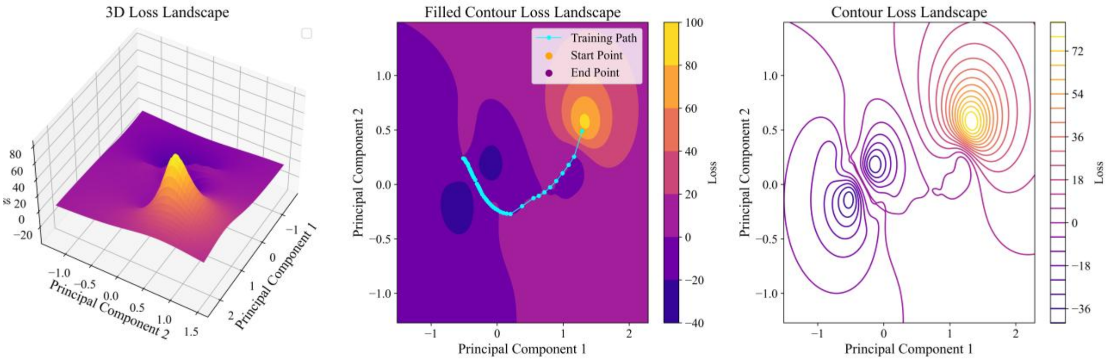
Firstly, given the high dimensionality of the parameter space ($\mathbf{w} \in \mathbb{R}^{43\mathrm{M}}$), we employ incremental PCA (IPCA) [176] to project weights onto two principal directions ($\mathbf{v}_1, \mathbf{v}_2$) that capture the maximum variance in the optimization trajectory.



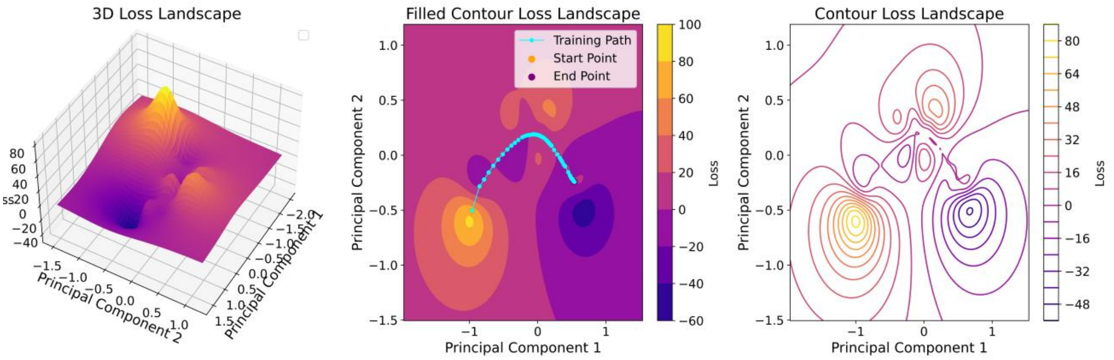
(a)



(b)



(c)



(d)

Figure 5.12 Loss landscapes of different loss functions: (a) Loss landscape of Physics-Constrained Adaptive MSE-Quantile Loss ; (b) Loss landscape of only quantile loss; (c) Loss landscape of standard MSE loss with Physics-Constrained Adaptive MSE loss; (d) Loss landscape of only standard MSE loss.

For a sequence of weight snapshots $\{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_N\}$, the incremental update rule for PCA avoids storing the full covariance matrix, reducing memory overhead:

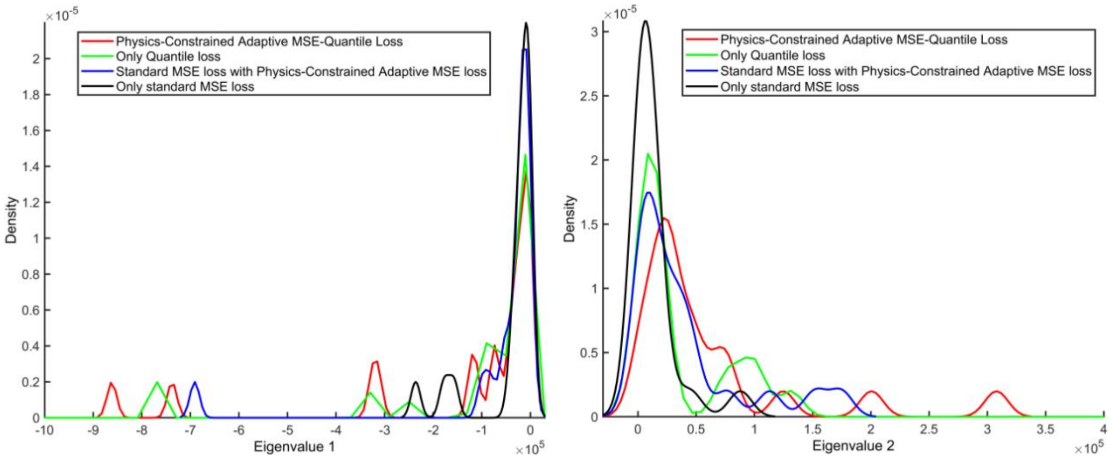
$$\mathbf{C}^{(t)} = \frac{t-1}{t}\mathbf{C}^{(t-1)} + \frac{1}{t}\mathbf{w}_t\mathbf{w}_t^T \tag{5.41}$$

where $\mathbf{C}^{(t)}$ is the covariance estimate at step t. The principal components $\mathbf{v}_1, \mathbf{v}_2$ are iteratively refined, enabling real-time dimensionality reduction during training.

Secondly, to reconstruct the loss surface without exhaustive sampling, we interpolate sparse observations using a hybrid radial basis function (RBF) and linear spline approach. For a grid point (Δ_1, Δ_2) in the 2D PCA subspace, the loss is estimated as:

$$\mathcal{L}(\Delta_1, \Delta_2) = \sum_{i=1}^{N} \alpha_i \, \phi\left(\sqrt{\left(\Delta_1 - \Delta_{1,i}\right)^2 + \left(\Delta_2 - \Delta_{2,i}\right)^2}\right) + \beta_1 \Delta_1 + \beta_2 \Delta_2 \quad (5.42)$$

where $\phi(r) = e^{-\epsilon r^2}$ is the Gaussian kernel, and coefficients α_i, β_j are solved through least squares. This hybrid strategy balances smoothness and computational efficiency, critical for large-scale parameter spaces. This approach enables tractable visualization of the loss landscape in WaveSenseNet.



(a) (b)

Figure 5.13 Spectral analysis of hessian eigenvalues near the end of training phase: (a) Negative Hessian eigenvalue spectra analysis; (b) Positive Hessian eigenvalue spectra analysis.

The optimization dynamics of different loss functions are analyzed through loss landscape visualization and spectral analysis of Hessian eigenvalues, as shown in Figures 5.12 and 5.13. Four configurations are compared: (1) the proposed Physics-Constrained Adaptive MSE-Quantile Loss, (2) quantile loss only, (3) standard MSE with physics-constrained adaptive MSE, and (4) standard MSE loss without physical constraints.

The 3D loss landscapes (Figure 5.12) reveal distinct optimization characteristics. For the proposed PCAMQ-Loss (Figure 5.12 (a)), the terrain exhibits a smooth, convex-like basin centered around the minimum, indicating stable convergence and reduced sensitivity to parameter perturbations. In contrast, the standard MSE loss

(Figure 5.12 (d)) shows a rugged landscape with multiple local minima and sharp ridges, reflecting optimization challenges due to unconstrained parameter drift. Intermediate configurations (Figures 5.12 (b)-(c)) display partial smoothing, suggesting that both the quantile term and physics-aware constraints contribute to regularization. All configurations demonstrate similar geometric patterns in their optimization trajectories, attributed to modifications in the loss function while maintaining a consistent network architecture. Despite identical network architectures across configurations, the PCAMQ-Loss exhibiting the smoothest and most stable convergence toward the global basin.

The spectral analysis of Hessian eigenvalues corroborates the loss landscape observations (Figure 5.12). For each configuration, 20 random points near the optimization endpoint were analyzed. All points exhibited one positive and one negative eigenvalue, confirming saddle point dominance in high-dimensional non-convex optimization [177].

Kernel density estimation (KDE) [178] with a Gaussian kernel quantified eigenvalue distributions after Z-score normalization. As shown in Figures 5.13, the proposed PCAMQ-Loss (red curves) exhibits sharply peaked densities near zero for both eigenvalues, with significantly lower amplitude compared to other configurations. This minimal curvature, characterized by comparable magnitudes of positive and negative components, signifies a balanced basin geometry that stabilizes the loss surface against parameter perturbations. In stark contrast, the standard MSE loss (black curves) shows broad distributions with higher amplitude, particularly for negative eigenvalues, reflecting a rugged terrain dominated by steep curvatures and unstable minima. Intermediate configurations display partial regularization, consistent with their partially smoothed landscapes in Figure 5.12 (b)-(c).

The spectral results correspond precisely with the geometric patterns depicted in Figure 5.12. The low-amplitude, narrow eigenvalue profiles of PCAMQ-Loss (Figure 5.13 (a)-(b)) match its smooth, convex-like basin (Figure 5.12 (a)), facilitating stable convergence. In contrast, the high-amplitude, dispersed eigenvalues of the standard MSE (Figure 5.13 (a)-(b)) reflect its chaotic, jagged landscape (Figure 5.12 (d)), which increases sensitivity to parameter perturbations. By integrating quantile-based error weighting with physics-aware constraints, PCAMQ-Loss effectively suppresses sharp curvatures, promoting flatter solutions that enhance generalization while maintaining physical fidelity requirements.

5.3.7 Interpretative analysis of Grad-CAM visualizations on gaussian positional encoding map

To validate the interpretability of the proposed multimodal framework, which integrates wavelet signals with Gaussian positional encoding maps, we employ Grad-

CAM [179] to visualize the network's attention to structural scattering information encoded in the maps. Two experimental setups are designed: (1) physically consistent encoding, as depicted in figure 5.14, where the Gaussian maps accurately reflect the scattering environment, and (2) unreasonable encoding as shown in figure 5.15, where conflicting actuator are artificially introduced.

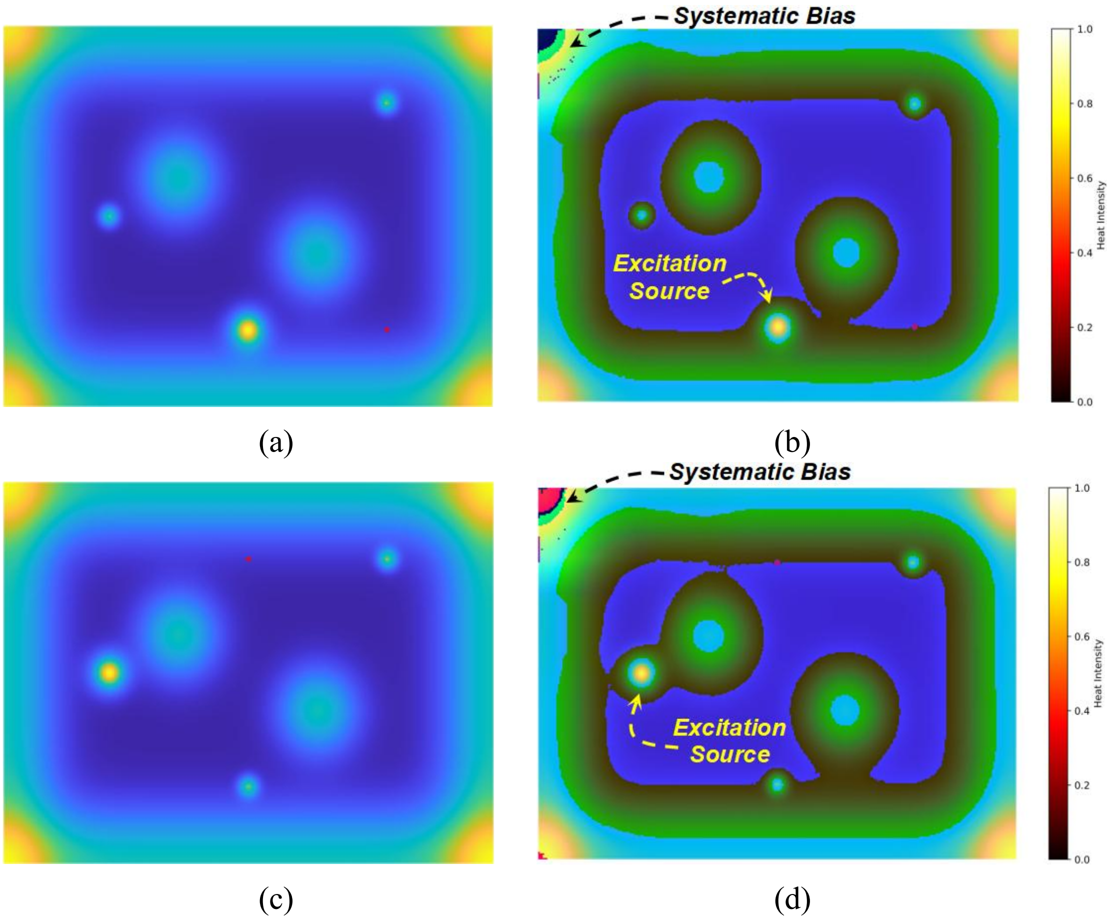
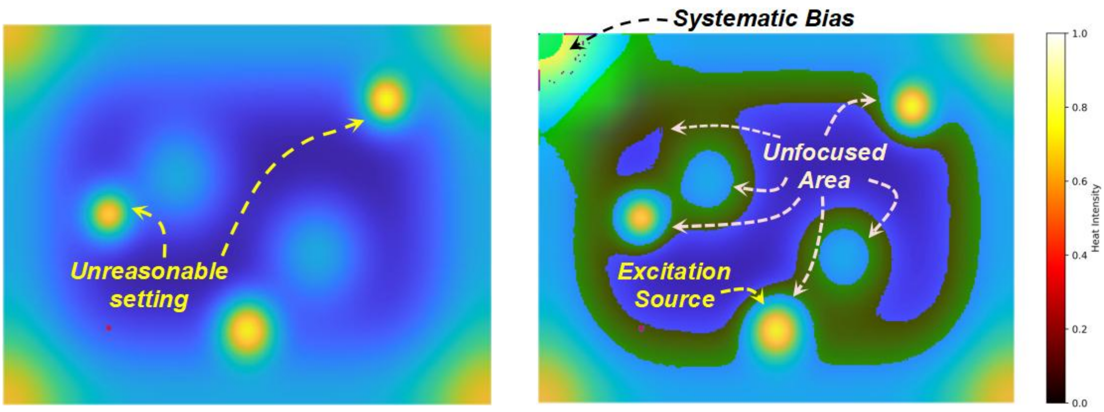
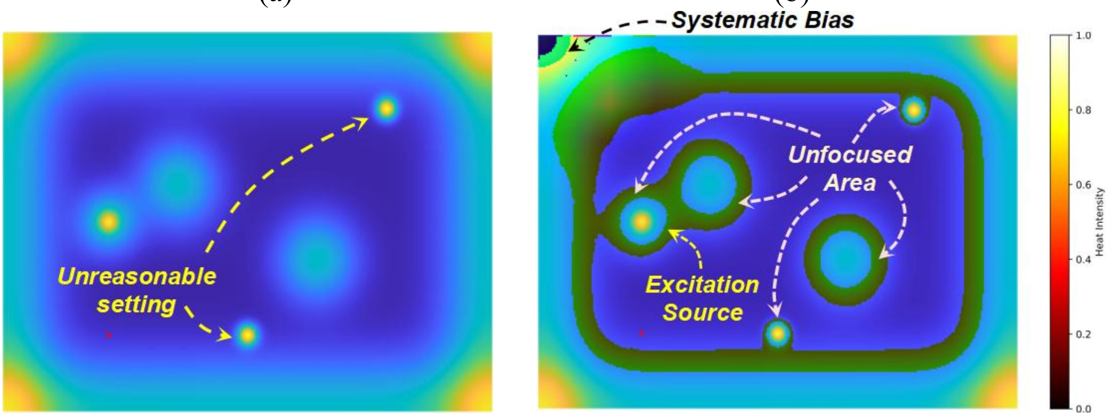


Figure 5.14 Grad-CAM visualizations of heatmaps highlighting key regions in input image: (a) Original input image with bottom actuator; (b) Grad-CAM result for (a); (c) Original input image with left actuator; (d) Grad-CAM result for (c).

The Grad-CAM analysis reveals how the proposed framework utilizes Gaussian positional encoding maps to interpret wave scattering dynamics by focusing on regions critical for signal scattering interactions. For physically valid encoding scenarios (Figures 5.14 (a), (c)), the heatmaps (Figures 5.14 (b), (d)) demonstrate that the network concentrates attention on scattering regions aligned with wave propagation paths encoded in the maps. For example, when the bottom PZT is activated (Figure 5.14(a)), the Grad-CAM results (Figure 5.14(b)) reveal that the network's attention is predominantly localized to scattering regions encoded by the Gaussian positional map. This alignment validates that the network prioritizes encoded scattering information to resolve multipath interference, effectively correlating obstacle geometries with signal reflections. Similarly, activation of the left PZT (Figure 5.14(c)) directs attention to scattering paths near structural features (Figure 5.14(d)), confirming the network's ability to learn spatial relationships between encoded scattering dynamics and received signals for precise ToF extraction.



(a) (b)



(c) (d)

Figure 5.15 Grad-CAM visualizations of heatmaps highlighting key regions in input image: (a) Original input image with bottom actuator with unreasonable setting; (b) Grad-CAM result for (a); (c) Original input image with left actuator with unreasonable setting; (d) Grad-CAM result for (c).

The Grad-CAM results for unreasonable encoding scenarios, such as simultaneous activation of three PZTs (Figures 5.15 (a), (c)), demonstrate the network's adaptive suppression of attention to conflicting scattering regions (blue areas in Figures 5.15 (b), (d)), despite their prominence in the input maps. This behavior highlights the framework's ability to filter inconsistencies between incomplete prior knowledge and actual signal patterns, maintaining robust ToF prediction accuracy even under physically implausible conditions. By implicitly downweighting invalid regions, the network enforces conformance to wave propagation principles, for instance, rejecting multi-source excitation conflicts while preserving focus on coherent signal-encoding interactions. These findings emphasize the dual utility of Gaussian encoding maps: beyond guiding the network to learn scattering-aware features, they act as diagnostic tools to evaluate the physical plausibility of input assumptions.

The results validate that Gaussian positional encoding maps serve as critical guides rather than mere auxiliary inputs, enabling the network to resolve complex scattering phenomena by localizing scattering dynamics through attention mechanisms that reflect spatial correlations between encoded obstacles and wave propagation paths. By interpreting Grad-CAM-derived attention patterns, we can audit the network's compliance to domain-specific physics. The framework's ability to suppress attention to conflicting regions in invalid encoding scenarios demonstrates robustness to imperfect prior knowledge and suggests potential for real-time fault detection in sensor deployment. This analysis confirms that the proposed framework complementarily integrates signal data with physics-informed encoding, bridging data-driven learning and domain knowledge.

5.3.8 Ablation experiment

The ablation study evaluates the contributions of key components in WaveSenseNet by systematically removing or modifying architectural modules and analyzing their impact on ToF prediction. Results in Table 5.2 demonstrate that omitting critical components, such as the Gaussian positional encoding map or the ABFM, significantly degrades performance, underscoring their essential roles in capturing scattering dynamics and integrating multi-scale features. In the "Ablation 2x in MSFE" and "Ablation 3x in MSFE" experiment, we used convolutions with a kernel size of 2 and 3 across all scales to focus on small-scale features. Similarly, "Ablation 4x in MSFE" and "Ablation 64x in MSFE" employed kernel sizes of 4 and 64 for medium and large scales, respectively, allowing us to assess the impact of scale selection on model performance. The CBAM and Deformable Convolution layers also prove vital for refining spatial feature adaptation, as their removal leads to measurable performance declines.

Table 5.2 Ablation study results.

Method	MAE	MSE
Ablation 2x in MSFE	2.76483067193E-05	1.18581759094E-09
Ablation 3x in MSFE	2.64791030590E-05	1.10186750347E-09
Ablation 4x in MSFE	2.72868211767E-05	1.13155339234E-09
Ablation 64x in MSFE	3.09203728625E-05	1.52571495023E-09
Ablation w/o Encoding Map	2.95534649968E-05	1.26467813960E-09
Ablation w/o CBAM	2.89058836632E-05	1.29523319458E-09
Ablation w/o Deformable Conv	2.83992612663E-05	1.25922370290E-09
Ablation w/o ABFM	3.76231712559E-05	2.13301290135E-09
Ablation w/ ReLU in ABFM	2.68930902671E-05	1.09824644921E-09
Ablation w/o MSAFN	3.10227226265E-05	1.49211513082E-09
Our Model	2.40286013326E-05	8.81687960652E-10

When fixing the first three scales to smaller kernel sizes (2x, 3x, and 4x) for capturing localized signal reflections and structural boundaries, adjusting the fourth scale demonstrates a clear trade-off. When the fourth scale is too small, it cannot be well distinguished from the first three scales. When it is too large, although it captures the long-term sequential relationship, it masks the results captured by the small scales. As shown in Figure 5.16, the relationship between the fourth scale parameter and prediction error follows an inverse-U trend, with the 64x configuration achieving the lowest MSE. This optimal balance enables the network to hierarchically integrate fine-scale details (from smaller kernels) with long-term dependencies (from the 64x kernel), effectively resolving ambiguities caused by multipath scattering.

The full WaveSenseNet architecture achieves optimal performance, outperforming all ablated variants. This confirms the collaborative interaction of physics-aware encoding, adaptive attention mechanisms, and multi-scale fusion in resolving complex wave extraction. The study not only validates the necessity of each module but also provides insights into parameter tuning for robust SHM applications. By isolating component contributions, the analysis reinforces the framework’s ability to balance physical interpretability with data-driven precision, ensuring reliable ToF prediction in noisy, scattering-dominated environments.

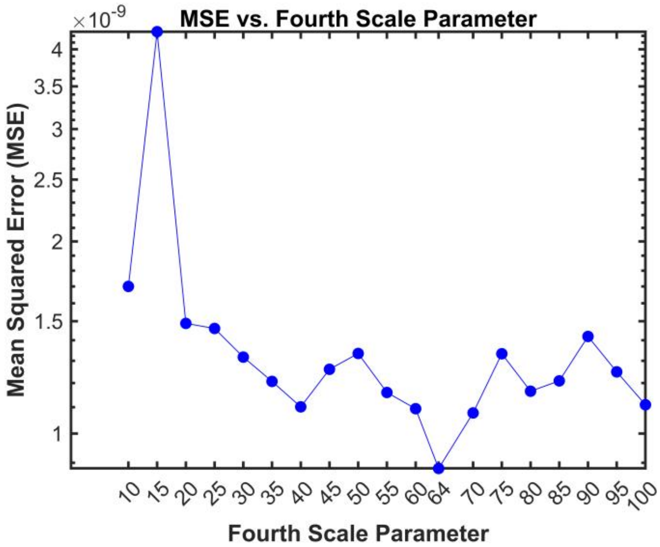


Figure 5.16 Relationship between the fourth scale parameter and mean squared error.

5.4 Robustness Analysis and Uncertainty Quantification

To rigorously examine the reliability of the proposed dual-modal network and its sensitivity to different information streams, we conducted a systematic noise-injection experiment focused on the waveform branch. While the geometric maps provide a spatial prior, the high-precision ToF estimation fundamentally relies on the transient characteristics of the wave signals.

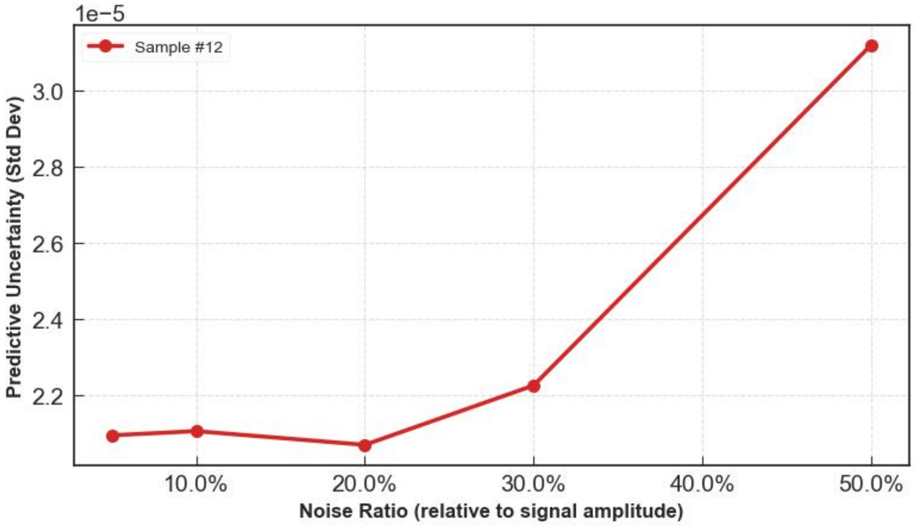


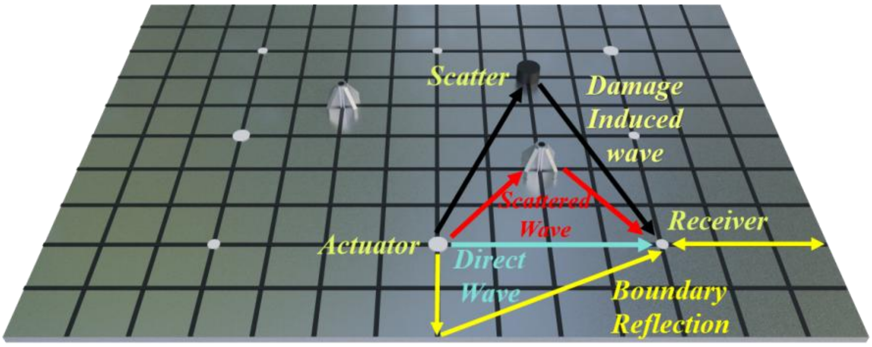
Figure 5.17 Sensitivity of predictive uncertainty to varied noise ratios in waveform signals.

As quantified in the results, the model exhibits remarkable robustness under moderate noise conditions (noise ratios ≤ 30%), where the predictive uncertainty remains within a stable band around 2.1×10^{-5}. This stability is attributed to the novel activation functions employed in our architecture, which effectively filter high-frequency background noise while preserving the essential wave packets. However, as the noise ratio escalates to 50%, the predictive uncertainty experiences a sharp incline to 3.199×10^{-5}. This strategic increase in variance serves as a self-diagnostic indicator: it signifies that the network has detected a significant degradation in the quality of the physical signal. Even though the model can still produce a regularized mean prediction based on the spatial priors from Input 1, the elevated uncertainty accurately reflects the loss of confidence in the specific arrival-time features of the corrupted waveform. Such behavior confirms that the proposed model is not merely performing a ‘shortcut mapping’ but is actively integrating multi-modal information in a physics-aware manner, providing a crucial safety margin for real-world structural health monitoring applications.

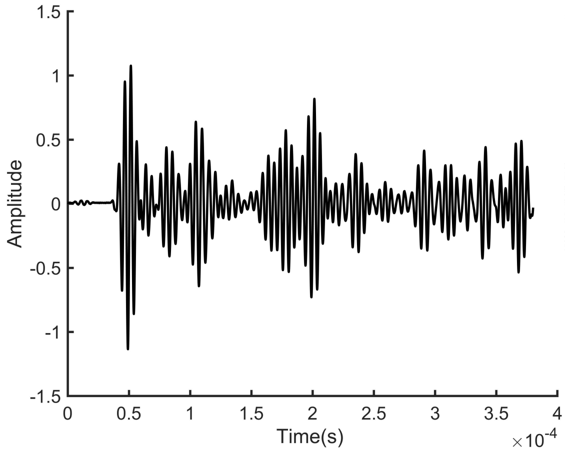
5.5 Application

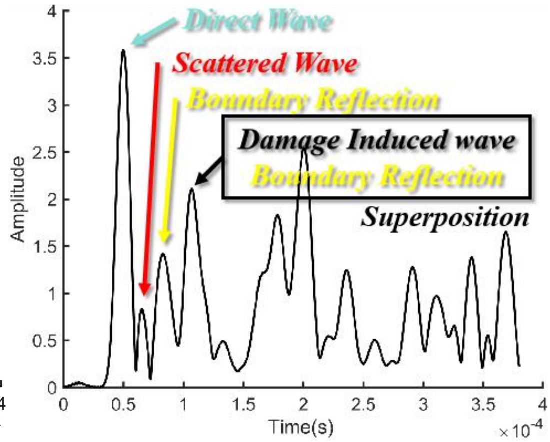
5.5.1 Signal analysis and processing

To validate the robustness of WaveSenseNet in resolving complex scattering phenomena, we employ a circle scatter as simulated damage and analyze its ToF estimation performance under two challenging scenarios: (1) damage near structural stiffeners and (2) damage adjacent to boundaries. These cases test the network’s ability to decode overlapping wave paths and prioritize damage-induced scattering signals, even when overlapped by high-energy reflections from stiffeners or boundaries. It is worth noting that the position of the scatterer has not been used in the previous training process. We collect and test the data from actuator to receiver. The blue line indicates the direct wave, the red line associates with the path of the scattered wave, the yellow lines represent the boundary reflection, and the black lines illustrate the damage scattering.



(a)





(b) (c)

Figure 5.18 A representative case to showcase the effect of the proposed method: (a) Wave path diagram and (b) the raw signal (c) the identified peaks in the wavelet coefficients.

For the first scenario, the time-domain signal from the specific path is shown in Figure 5.18(a), followed by the corresponding wavelet coefficients in Figure 5.18(b). The first peak corresponds to the direct wave propagating along the shortest path, followed by a stiffener-induced scattering peak, a boundary reflection, and a fourth peak resulting from coupled scattering between the damage and boundary. Despite the interference from stiffener and boundary reflections, WaveSenseNet accurately identify the ToF of the damage-boundary coupled wave at 1.14e-04 seconds. This precision highlights the network's ability to detect weak scattering signatures overlapped by stronger reflections, a crucial capability for detecting early-stage damage in reinforced structures.

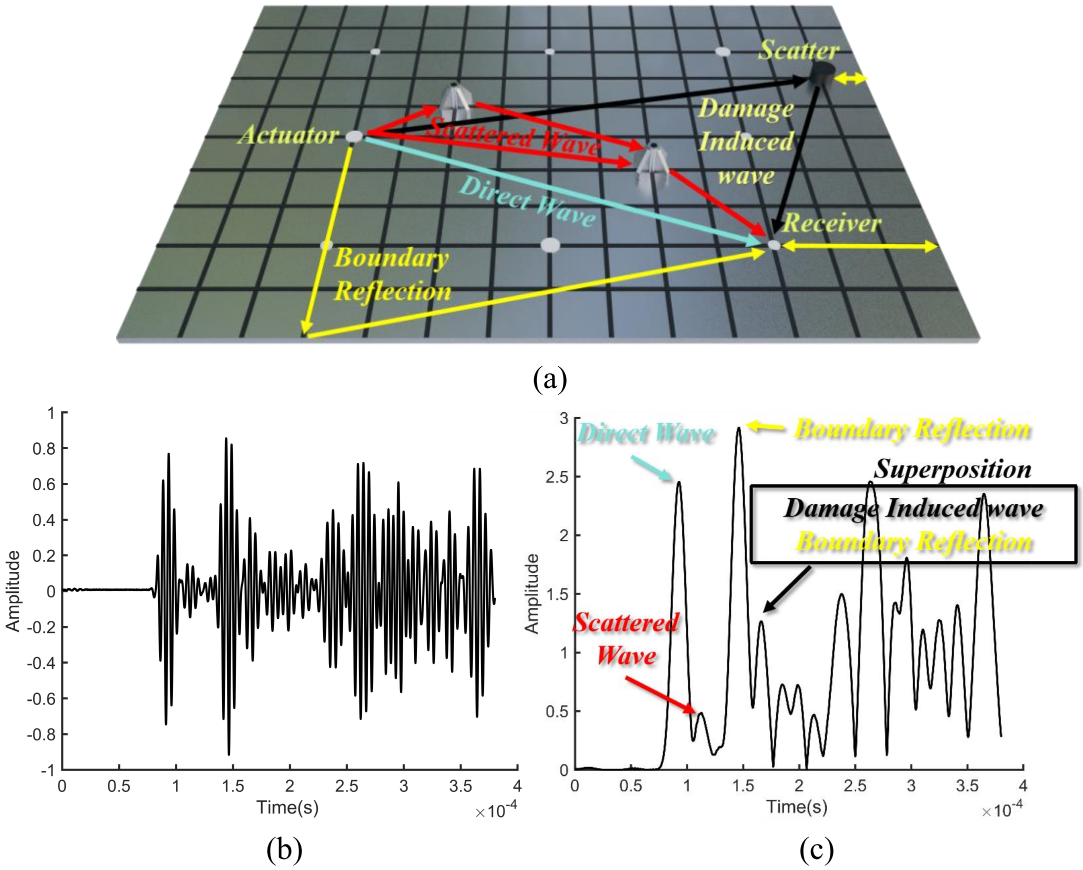


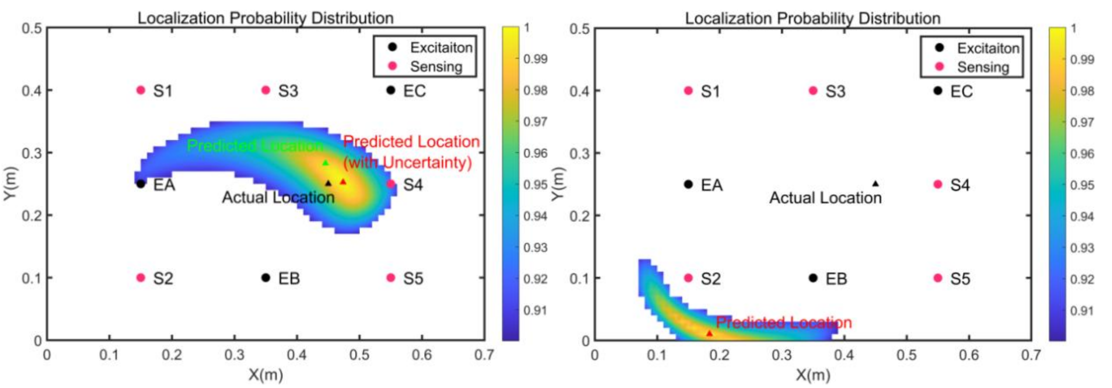
Figure 5.19 A representative case to showcase the effect of the proposed method: (a) Wave path diagram and (b) the raw signal (c) the identified peaks in the wavelet

coefficients.

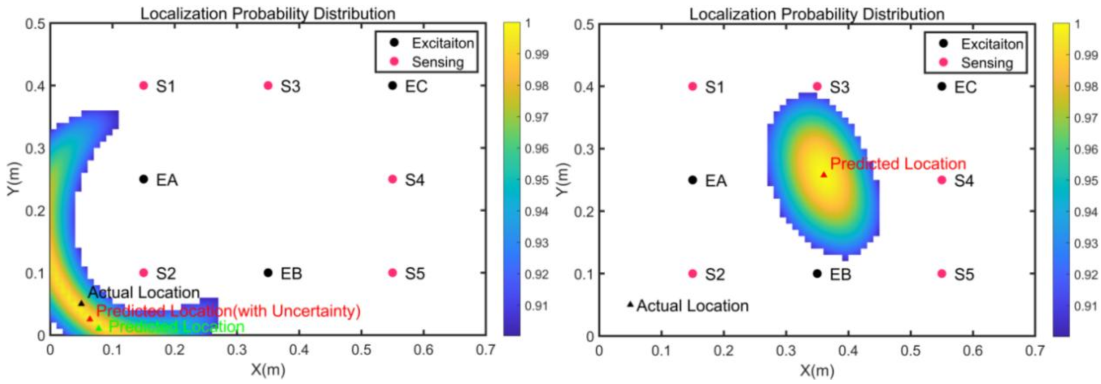
In the second scenario, as shown in figure 5.19, the damage is positioned near a structural boundary. The third peak in the wavelet coefficients corresponds to a high energy boundary reflection, while the fourth peak represents the damage induced scattering wave propagating along a path partially overlapping with the boundary reflection. Although the boundary reflection exhibits significantly higher amplitude, WaveSenseNet successfully identifies the damage-related ToF at 1.61e-04 seconds, demonstrating its ability to prioritize physically meaningful scattering paths over spurious high-energy noise. This capability stems from the framework's integration of Gaussian positional encoding maps, which encode spatial scattering dynamics, and stacked feature fusion, which separates multi-source interference.

5.5.2 Localization results

With the extracted ToFs, damage localization can be achieved by the ellipse positioning method [161], resulting in the corresponding damage image. Two representative cases are selected to illustrate the efficacy of the method, as shown in Figure 5.20. As illustrated in Figure 5.20, subfigures (a) and (c) present the localization results based on the ToF predicted by WaveSenseNet, where Monte Carlo Dropout was employed to quantify prediction uncertainty. These results were further weighted in the localization algorithm. In contrast, subfigures (b) and (d) depict comparative results derived from manually selecting the second signal peak as the ToF, a traditional approach susceptible to noise and multipath interference. Experimental data demonstrate a significant improvement in localization accuracy when uncertainty weighting is incorporated. For damage near stiffeners, the weighted localization error in Figure 5.20(a) is 2.3871 cm, while the manual ToF selection method (Figure 5.20(b)) yields a substantially higher error of 35.832 cm. Similarly, for boundary-located damage, the weighted result (Figure 5.20(c)) achieves an error of 2.8256 cm, compared to 37.358 cm for the manual ToF selection case (Figure 5.20(d)). This stark contrast underscores the critical role of uncertainty-aware weighting in mitigating environmental noise and signal distortion.



(a) (b)



(c) (d)

Figure 5.20 Comparison of location results: (a), (c)-Identify default arrival time through the network; (b), (d)-Second peak as the default arrival time.

To quantify the reliability of ToF predictions from different sensor signals, a normalized inverse standard deviation method was adopted to compute the weights:

$$w_i = \frac{1/\sigma_i}{\sum_j \ (1/\sigma_j)^2} \tag{5.43}$$

where σ_i represents the standard deviation of the i-th ToF prediction from Monte Carlo Dropout. This formulation assigns higher weights to predictions with lower uncertainty, thereby prioritizing more reliable signals. For instance, in the case of damage near stiffeners, the variances of ToF predictions from EA excitation-S4 receiver, EB excitation-S2 receiver, and EC excitation-S2 receiver were calculated as 6.34905e-06, 1.77674e-05, and 3.71239e-06, respectively. These values correspond to normalized weights of 0.6542, 0.2345, and 0.1113, effectively amplifying the influence of high-confidence signals while suppressing noisy inputs in the localization algorithm.

These results validate WaveSenseNet's capability as a generalizable framework for robust ToF extraction in complex environments, combining data-driven adaptability with physics-guided localization. Its effectiveness is demonstrated through SHM scenarios, showcasing precise damage detection amidst multipath scattering and boundary reflections.

5.6 Summary

This chapter introduces WaveSenseNet, a robust and generalizable deep learning framework designed to address the critical challenge of extracting accurate ToF measurements in complex environments characterized by noise, multipath interference, and heterogeneous scattering. The framework's key innovation lies in its multimodal architecture, which synergistically combines wavelet-transformed temporal signals with a physics-informed Gaussian positional encoding map that represents the structure's complex geometry. This approach allows the network to effectively learn the spatiotemporal relationships of wave propagation and scattering.

Key architectural designs, including a novel parametric activation function, enable the network to overcome the inherent spectral bias of standard models and retain critical high-frequency features essential for precise ToF estimation. To ensure the reliability of its predictions, WaveSenseNet also integrates Monte Carlo Dropout to provide uncertainty estimates. The training is guided by a novel physics-constrained loss function that balances data-driven learning with physical plausibility, ensuring stable convergence and robust performance.

The framework's universal applicability was validated through rigorous experiments on a guided wave-based SHM application involving a complex structure with stiffeners and other scatterers. WaveSenseNet demonstrated superior accuracy in ToF extraction, significantly outperforming other deep learning methods. This enhanced precision in ToF subsequently led to remarkable damage localization results, which were an order of magnitude better than those from conventional methods. These findings confirm that WaveSenseNet provides a powerful and practical solution for reliable, real-time sensing in complex, realistic environments.

WaveSenseNet establishes a universal paradigm for ToF extraction in applications where complex scattering and environmental variability degrade traditional methods. By harmonizing data-driven adaptability with domain-specific physics, the framework advances the frontier of reliable, real-time sensing in complex environments.

Chapter 6 Conclusion and suggestions for future work

This chapter systematically summarizes and distills the research work of this entire doctoral dissertation, and based on this foundation, to propose future directions for research. The main goal of this thesis has been to address the universal and fundamental challenge of accurately extracting ToF in wave-based sensing, a problem that becomes especially difficult when signal superposition from multipath interference occurs. To explore this challenge in a physically rich and practically relevant context, this thesis has focused on guided wave-based SHM as its primary application, developing a series of innovative machine learning frameworks closely aligned with the principles of physics.

6.1 Summary of thesis

The thesis starts with a broad introduction and literature review, establishing the research motivation by emphasizing the universal challenge of ToF extraction in wave-based sensing, especially within the context of SHM. It critically examines the literature on both traditional signal processing and purely data-driven approaches, pinpointing a key research gap: the need for solutions that are not only accurate but also physically consistent and interpretable. This justification leads to the formation of the research objectives, which focus on developing a series of PIML frameworks.

Chapter 2 then provides the necessary theoretical foundations for the subsequent research. It briefly reviews the theory of Lamb wave propagation in isotropic plates, the principles of physical wave modeling via ray tracing, and the fundamental concepts of the deep learning architectures used, such as CNN and GNN. This chapter acts as an essential theoretical toolkit, equipping the reader with the knowledge needed to understand the novel frameworks developed later in the thesis.

With this foundation, the thesis introduces three increasingly advanced deep learning frameworks. First, in Chapter 3, we introduce the MSST (Multi-Scale Spatio-Temporal) network. This framework addresses the core issue of boundary reflections. Its innovation lies in a multi-scale module that captures signal features at various temporal scales; this approach indirectly reduces the low-frequency bias typical of neural networks by enabling the model to focus on different spectral components. Using deconvolution-based visualization, we verify that the network effectively learns to eliminate boundary reflections, separating the target damage-scattered wave from high-energy reflected waves and thus expanding the effective monitoring area.

In Chapter 4, we developed the PhysRayGNN (Physics-Informed Ray-Tracing Graph Neural Network) model, improving the physical understandings of the wave propagation process. The core of this framework is the innovative integration of physics-based ray tracing, which efficiently creates a graph structure representing all major wave paths, and a GNN, which explicitly models the physical processes of wave propagation, attenuation, and superposition along these paths. To further anchor the model in physics, we included two distinct physics-informed loss functions: a wavelet transform loss and a wave equation loss, which provide implicit supervision on the extracted signals, ensuring they conform to known wave characteristics and dynamics.

In Chapter 5, we presented the WaveSenseNet framework, a powerful and adaptive solution designed for cluttered, real-world engineering settings. This framework combines multiple innovations to handle complex scattering effects. Architecturally, it features a new parametric activation function to capture essential high-frequency wave packet details, along with a Gaussian positional encoding map that enables the multimodal fusion of time-domain signals with spatial information about scatterers. For reliable prediction, it uses Monte Carlo Dropout to estimate uncertainty in the extracted ToF. The training process is guided by a custom Physics-Constrained Adaptive MSE-Quantile Loss, which balances data fitting and physical plausibility. Additionally, we developed a new analytical method that merges incremental PCA with a hybrid RBF and linear spline approach to visualize the high-dimensional loss landscape. This analysis shows that our loss function creates a smoother optimization surface, leading to better model generalization.

In summary, this dissertation's research work forms a complete and cohesive system. It starts by addressing the core issue of boundary reflections, moves on to the clear and interpretable modeling of wave physics, and ends with a strong, uncertainty-aware framework capable of managing real-world complexity.

6.2 Main contributions

The research findings of this thesis provide not only effective solutions to a specific engineering problem but also make several original contributions at the methodological, theoretical, and analytical levels.

- Established a progressive and interpretable research methodology, systematically advancing from simplified boundary-reflection problems in basic structures to complex multi-scatterer environments. This approach ensures that each developed network architecture is transparent and grounded in clear physical principles.
- Provided visual and verifiable evidence of a neural network's ability in physics-based signal separation. For the MSST network, we went beyond traditional performance metrics by using deconvolution to visualize the internal feature maps, confirming that the network effectively learned to isolate and eliminate boundary reflections. This provides strong proof of the model's physics-guided reasoning.

- Led the integration of Graph Neural Networks with Ray Tracing for interpretable wave process modeling. The PhysRayGNN framework is the first attempt to interperte wave propagation and superposition as a physically meaningful graph, enabled by ray tracing. This approach shifts the deep learning model from a 'black box' predictor to a 'white box' simulator, greatly improving explainability and trust in safety-critical applications.
- Incorporated multiple complementary physics-informed constraints within a single GNN framework. In PhysRayGNN, the graph's architectural constraint was enhanced with two separate implicit loss-based constraints (wavelet and wave equation). This showcases an advanced approach to PIML, where domain knowledge is integrated at various levels of the learning process to improve both accuracy and physical fidelity.
- Developed innovative architectural components in WaveSenseNet for advanced multimodal wave analysis. We introduced two key innovations: (1) a parametric activation function designed to reduce the low-frequency bias of neural networks, allowing it to capture high-frequency signal features, and (2) a Gaussian positional encoding map for a new multimodal fusion that combines temporal signals with the physical geometry of the environment.
- Proposed a robust and uncertainty-aware training and optimization strategy for

WaveSenseNet. To ensure prediction reliability, we incorporated two key components: (1) Monte Carlo Dropout, which offers essential uncertainty quantification for ToF estimations, and (2) a custom Physics-Constrained Adaptive Mean Squared Error-Quantile Loss, guiding the model toward physically consistent and stable solutions.

- Introduced a new and efficient method for analyzing loss landscapes in high-dimensional spaces. By combining incremental PCA with a hybrid RBF spline interpolation, we created a practical tool for visualizing the optimization terrain of complex neural networks. This analytical contribution demonstrated that our proposed physics-constrained loss function encourages smoother minima, which are directly linked to better generalization performance.

6.3 Suggestions for future work

Although this dissertation has made significant progress, vast research opportunities remain to be explored. Based on the current work, we propose the following promising directions for future research:

1. Extension to anisotropic and heterogeneous materials. The current research has mainly concentrated on isotropic and uniform structures. An important future step is to expand these frameworks to anisotropic media, such as composite materials. This expansion requires integrating direction-dependent wave speed models into

both the ray-tracing and GNN components, as well as developing advanced encoding schemes to accurately depict complex properties like fiber orientation and ply stacking. This improvement is likely to significantly enhance applicability in the aerospace industry.

2. Integration with nonlinear guided wave phenomena. The current frameworks excel at analyzing linear wave propagation. However, many types of early damage, such as micro-cracks and plasticity, show through nonlinear acoustic effects, including harmonic generation. A promising future goal is to develop PIML frameworks that can capture this nonlinearity. A major challenge is to separate the weak, damage-related nonlinear waves from stronger nonlinearities caused by the measurement system, like the adhesive layer of the transducer. A future PIML model could be designed to explicitly learn and distinguish these different sources of nonlinearity, enabling much earlier and more sensitive damage detection.
3. Onboard and embedded deployment for real-time monitoring. While the proposed models are powerful, they are also computationally intensive. A key future focus involves exploring model compression, quantization, and hardware acceleration techniques, such as utilizing FPGAs or dedicated AI chips. These advances are vital for deploying these algorithms on embedded systems, allowing for real-time,

online diagnosis of structural health status.

4. Deep integration of uncertainty quantification (UQ) methods. While WaveSenseNet has introduced a UQ component, future research should systematically incorporate advanced techniques such as Bayesian deep learning across all frameworks. This integration would allow models to deliver not only a ToF prediction but also its full probability distribution, providing richer information for structural risk assessment.
5. Exploration of inverse problem solving for detailed damage diagnosis. This thesis has primarily focused on the forward problem of ToF extraction. A promising future direction is to leverage these high-fidelity physical models to solve inverse problems. For instance, by analyzing how damage alters the PhysRayGNN wavefield graph, it may be possible not only to locate the damage but also to determine its size, shape, and severity. This marks a significant step from simple "damage localization" to comprehensive "damage diagnosis."

Appendix *A.1* Physics-Informed GNN Training Strategy with Ray Tracing

Algorithm 1 Physics-Informed GNN Training Strategy with Ray Tracing

Input: Data folders (containing $\mathcal{D}_{\text{train}}^{E_p}$, $\mathcal{D}_{\text{test}}^{E_p}$), Excitation points E_{all}, Hyperparameters (N_{damage}, N_{main1}, N_{main2}, λ_{wave}, λ_{wavelet})
Output: Trained models $M_{\text{main}}^{E_p}$, Results $\mathcal{R}_{\text{all}}$

Variable Definitions:
E_{all}: Set of all excitation points.
$\mathcal{D}_{\text{train}}^{E_p}$, $\mathcal{D}_{\text{test}}^{E_p}$: Training and testing datasets for E_p.
$\mathcal{R}_{\text{all}}$: Collection of all evaluation results.
S: A sample from the dataset.
$\mathbf{d}_{\text{pred}}$, $\mathbf{d}_{\text{true}}$: Predicted and true damage positions.
$\mathbf{Sig}_S$: Signal data for sample S.
$\mathcal{L}$: Total loss (data and physics-based).
$\mathcal{R}_{\text{all}} \leftarrow \emptyset$
for each $E_p \in E_{\text{all}}$ **do**
  Load $\mathcal{D}_{\text{train}}^{E_p}$, $\mathcal{D}_{\text{test}}^{E_p}$ (normalized)
  if data invalid **then continue**
  end if
  Train Damage Position Initializer M_{init}:
  Initialize M_{init} and position scaler
  for epoch $= 1$ to N_{damage} **do**
    for each $S \in \mathcal{D}_{\text{train}}^{E_p}$ **do**
      $\mathbf{d}_{\text{pred}} \leftarrow M_{\text{init}}(\mathbf{X}_S^{\text{damage}}, \mathbf{P}_S^{\text{damage}})$
      Minimize $\text{MSE}(\mathbf{d}_{\text{pred}}, \text{Scaled}(\mathbf{d}_{\text{true}}))$
    end for
  end for
  Stage 1: Train GNN with True Positions & Ray Tracing:
  $M_{\text{main}} \leftarrow \text{InitModel}(E_p)$ with M_{init}
  for epoch $= 1$ to N_{main1} **do**
    for each $S \in \mathcal{D}_{\text{train}}^{E_p}$ **do**
      $\text{preds}, \mathbf{sig}^{\text{raw}}, \mathbf{sig}^{\text{wvlt}} \leftarrow M_{\text{main}}(\mathbf{Sig}_S, \mathbf{d}_{\text{true}})$ ▷ Build graph using ray tracing with true damage position
      $\mathcal{L} \leftarrow \mathcal{L}_{\text{data}} + \lambda_{\text{wave}}\mathcal{L}_{\text{wave}} + \lambda_{\text{wavelet}}\mathcal{L}_{\text{wavelet}}$
      Update M_{main} with $\mathcal{L}$
    end for
  end for
  Stage 2: Fine-tune GNN with Initializer Positions & Ray Tracing:
  for epoch $= 1$ to N_{main2} **do**
    for each $S \in \mathcal{D}_{\text{train}}^{E_p}$ **do**
      $\text{preds}, \mathbf{sig}^{\text{raw}}, \mathbf{sig}^{\text{wvlt}} \leftarrow M_{\text{main}}(\mathbf{Sig}_S)$ ▷ Build graph using ray tracing with M_{init} predicted position
      Compute and minimize $\mathcal{L}$ (same as Stage 1)
    end for
  end for
  Test Evaluation:
  for each $S \in \mathcal{D}_{\text{test}}^{E_p}$ **do**
    $\text{preds}_{\text{test}}, \mathbf{d}_{\text{pred}}, \text{attn} \leftarrow M_{\text{main}}(\mathbf{Sig}_S)$ ▷ Graph built with ray tracing using $\mathbf{d}_{\text{pred}}$
    Analyze: Graph visualization, attention vs propagation time, path statistics
    Record: MSE/MAE for ToF labels
  end for
  Save M_{main}, aggregate R_{E_p} to $\mathcal{R}_{\text{all}}$
end for
Save combined results and metrics
return $\mathcal{R}_{\text{all}}$

Appendix *A.2* Multi-Reflection Ray Tracing with Time Window Constraint

Algorithm 2 Multi-Reflection Ray Tracing with Time Window Constraint

Input: Source point $\mathbf{S}$, target point $\mathbf{T}$, plate boundaries $\mathcal{B}$, wave speed c, max reflections $r_{\max}$, time window T_w

Output: Valid paths $\mathcal{P} = \{P_1, ..., P_n\}$ with attributes

Initialize path set $\mathcal{P} \leftarrow \emptyset$
Compute direct path properties:
$d_{\text{direct}} \leftarrow \|\mathbf{S} - \mathbf{T}\|_2$
$t_{\text{direct}} \leftarrow d_{\text{direct}}/c$
if $t_{\text{direct}} \leq T_w$ **then**
    $\mathcal{P} \leftarrow \mathcal{P} \cup \{(d_{\text{direct}}, t_{\text{direct}}, 0, \text{"direct"}, [\mathbf{S}, \mathbf{T}])\}$
end if
if $r_{\max} \geq 1$ **then**
    for each boundary $b \in \{\text{top}, \text{bottom}, \text{left}, \text{right}\}$ **do**
        Compute reflection point $\mathbf{R}_b$ using image method:
        $\mathbf{I}_b \leftarrow \text{ImageProjection}(\mathbf{S}, b, \mathcal{B})$
        $\mathbf{R}_b \leftarrow \text{ReflectionPoint}(\mathbf{S}, \mathbf{I}_b, b, \mathcal{B})$
        Calculate reflected path length:
        $d_b \leftarrow \|\mathbf{I}_b - \mathbf{T}\|_2$
        $t_b \leftarrow d_b/c$
        if $t_b \leq T_w$ **then**
            $\mathcal{P} \leftarrow \mathcal{P} \cup \{(d_b, t_b, 1, \text{"single reflection"}, [\mathbf{S}, \mathbf{R}_b, \mathbf{T}]\}$
        end if
    end for
end if
if $r_{\max} \geq 2$ **then**
    for each boundary pair $(b_1, b_2) \in \mathcal{B}^2$ where $b_1 \neq b_2$ **do**
        Compute primary reflection $\mathbf{R}_1$ for b_1
        Compute secondary image $\mathbf{I}_{12} \leftarrow \text{ImageProjection}(\mathbf{I}_1, b_2, \mathcal{B})$
        $\mathbf{R}_2 \leftarrow \text{ReflectionPoint}(\mathbf{R}_1, \mathbf{I}_{12}, b_2, \mathcal{B})$
        Calculate double reflection path length:
        $d_{12} \leftarrow \|\mathbf{I}_{12} - \mathbf{T}\|_2$
        $t_{12} \leftarrow d_{12}/c$
        if $t_{12} \leq T_w$ **then**
            $\mathcal{P} \leftarrow \mathcal{P} \cup \{(d_{12}, t_{12}, 2, \text{"double reflection"}, [\mathbf{S}, \mathbf{R}_1, \mathbf{R}_2, \mathbf{T}]\}$
        end if
    end for
end if
return $\mathcal{P}$
procedure ImageProjection($\mathbf{P}, b, \mathcal{B}$) boundary b top
    $\mathbf{I} \leftarrow (P_x, 2\mathcal{B}_y^{\max} - P_y)$ bottom
    $\mathbf{I} \leftarrow (P_x, -P_y)$ left
    $\mathbf{I} \leftarrow (-P_x, P_y)$ right
    $\mathbf{I} \leftarrow (2\mathcal{B}_x^{\max} - P_x, P_y)$
    return I
end procedure
procedure ReflectionPoint($\mathbf{S}, \mathbf{I}, b, \mathcal{B}$)
    Calculate parametric reflection coefficient t:
    $t \leftarrow \frac{\mathcal{B}_b - S_b}{I_b - S_b}$ // $\mathcal{B}_b$ is boundary coordinate
    $\mathbf{R} \leftarrow \mathbf{S} + t(\mathbf{I} - \mathbf{S})$
    return Clipped $\mathbf{R}$ within $\mathcal{B}$
end procedure

Appendix *A.3* Spectral Properties of the Parametric Activation Function

Theorem 1: Compact Support Characterization

The term $f(x) = \cos(cx) \cdot \mathrm{e}^{-ax} \cdot \max(0, 1 - \mathrm{e}^{-bx})$ satisfies the following properties:

1. **Localization Properties:** For any predefined error tolerance $\epsilon > 0$, there exists a finite interval $x \in (0, x_{\max})$ within which the function exhibits significant response.

$$|f(x)| \geq \epsilon \tag{A.3.1}$$

where the upper bound $x_{\max}$ is explicitly determined by:

$$x_{\max} = \min\left(\frac{1}{a}\ln\left(\frac{1}{\epsilon}\right), \frac{1}{b}\ln\left(\frac{1}{1-\epsilon}\right)\right) \tag{A.3.2}$$

2. **Lipschitz Continuity:** If $a \geq 0$ and $b \geq 0$, then $f(x)$ is Lipschitz continuous with Lipschitz constant:

$$L = |a| + |b| + |c| \tag{A.3.3}$$

Proof of Localization Properties:

1.Constraint from Exponential Decay Term:

The magnitude of $f(x)$ is bounded by:

$$|f(x)| \leq e^{-ax} \cdot \left(1 - e^{-bx}\right) \tag{A.3.4}$$

To ensure $|f(x)| \geq \epsilon$, it is sufficient to require both:

$$e^{-ax} \geq \epsilon \text{ and } 1 - e^{-bx} \geq \epsilon \tag{A.3.5}$$

2.Solving for x :

From $e^{-ax} \geq \epsilon$:

$$-ax \geq \ln \epsilon \Rightarrow x \leq \frac{1}{a} \ln \left(\frac{1}{\epsilon}\right) \tag{A.3.6}$$

From $1 - e^{-bx} \geq \epsilon$:

$$e^{-bx} \leq 1 - \epsilon \Rightarrow - bx \leq \ln (1 - \epsilon) \Rightarrow x \leq \frac{1}{b} \ln \left(\frac{1}{1-\epsilon}\right) \tag{A.3.7}$$

3.Intersection of Constraints:

The significant response interval is the intersection of the solutions from both inequalities:

$$x \in \left(0, \min\left(\frac{1}{a}\ln\left(\frac{1}{\epsilon}\right), \frac{1}{b}\ln\left(\frac{1}{1-\epsilon}\right)\right)\right) \quad (A.3.8)$$

Since $a > 0, b > 0$, and $\epsilon > 0$, the upper bound $x_{\max}$ is finite and positive. Thus, the localized sensitivity region exists.

Proof of Lipschitz continuity:

The proof consists of two parts

1. Case $1(x \leq 0)$

Here, $\max(0,1 - e^{-bx}) = 0$, so $f(x) = 0$. The derivative $f'(x) = 0$, trivially satisfying $|f'(x)| \leq L$.

2. Case $2(x > 0)$

The function simplifies to $f(x) = \cos(cx) \cdot e^{-ax} \cdot (1 - e^{-bx})$. Differentiating using the product rule:

$$f'(x) = -c\sin(cx)e^{-ax}(1 - e^{-bx}) - a\cos(cx)e^{-ax}(1 - e^{-bx}) + b\cos(cx)e^{-(a+b)x}$$

$$(A.3.9)$$

Bounding the absolute value of each term:

$$\begin{aligned}|f'(x)| &\le |c| \cdot e^{-ax}\left(1 - e^{-bx}\right) + |a| \cdot e^{-ax}\left(1 - e^{-bx}\right) + |b| \cdot e^{-(a+b)x} \\ &\le (|c| + |a|)e^{-ax} + |b|e^{-(a+b)x}\end{aligned} \tag{A.3.10}$$

For $a \ge 0, b \ge 0$, the maximum value of $|f'(x)|$ occurs at $x = 0^+$:

$$|f'(0^+)| = |a| + |b| + |c| \tag{A.3.11}$$

By the mean value theorem, $|f(x_1) - f(x_2)| \le L|x_1 - x_2|$, where $L = |a| + |b| + |c|$.

Remark 1. The localization property ensures that the activation function's response is confined to a tunable interval $x \in (0, x_{\max})$, controlled by parameters a and b. This allows neurons to precisely capture transient events in time-domain signals, enhancing temporal resolution and noise robustness. Meanwhile, Lipschitz continuity bounds the gradient magnitudes during backpropagation, preventing vanishing and exploding gradients and stabilizing training dynamics. By balancing localization and smoothness, the function adaptively optimizes interpretability and numerical robustness in deep networks.

Theorem 2: Spectral Impact of Parameters

The Fourier transform $\mathcal{F}\{f(x)\}(\omega)$ satisfies the following spectral properties:

1. **High-Frequency Decay Rate:**

For $\omega \gg a, b, c$, the magnitude spectrum decays asymptotically as:

$$|\mathcal{F}\{f(x)\}(\omega)| \approx \frac{2b}{\omega^2} \qquad (A.3.12)$$

By selecting bigger b, resulting in slower magnitude decay at high frequencies. This ensures better retention of high-frequency information compared to ReLU.

2. **Center Frequency Modulation:**

The dominant spectral energy is concentrated near $\omega = c$, modulated by the oscillation term $\cos(cx)$. Adjusting c shifts the spectral peak linearly.

3. **Bandwidth Control:**

The effective bandwidth $\Delta\omega$ of the spectrum is governed by the decay parameter a, satisfying:

$$\Delta\omega \propto a \qquad (A.3.13)$$

A larger a increases bandwidth but reduces temporal localization.

Proof of High-Frequency Decay Rate:

The Fourier transform of $f(x)$ is derived by splitting the integral into two terms:

$$\mathcal{F}\{f(x)\}(\omega) = \int_0^\infty \cos{(cx)} e^{-ax}\left(1 - e^{-bx}\right)e^{-j\omega x}dx = I_1 - I_2 \qquad (A.3.14)$$

Where

$$I_1 = \int_0^\infty \cos{(cx)} e^{-(a)x} e^{-j\omega x}dx,\ I_2 = \int_0^\infty \cos{(cx)} e^{-(a+b)x} e^{-j\omega x}dx \qquad (A.3.15)$$

Using the standard Fourier transform of a damped cosine function:

$$I_1 = \frac{a+j\omega}{(a+j\omega)^2+c^2},\ I_2 = \frac{a+b+j\omega}{(a+b+j\omega)^2+c^2} \qquad (A.3.16)$$

For $\omega \gg a, b, c$, expand both terms asymptotically:

$$I_1 \approx \frac{1}{j\omega} - \frac{2a}{\omega^2},\ I_2 \approx \frac{1}{j\omega} - \frac{2(a+b)}{\omega^2} \qquad (A.3.17)$$

Subtracting I_2 from I_1 :

$$\mathcal{F}\{f(x)\}(\omega) \approx \frac{2b}{\omega^2} \qquad (A.3.18)$$

Thus, the magnitude decays as $\mathcal{O}(1/\omega^2)$, with the coefficient $2b$ indicating that increasing b decelerates the decay.

Proof of Center Frequency Modulation:

The term $\cos(cx)$ introduces oscillations in the time domain, whose Fourier transform generates peaks at $\omega = \pm c$. The exponential decay e^{-ax} modulates these peaks into Lorentzian profiles. The spectral energy is dominated by the interaction between $\cos(cx)$ and e^{-ax}:

$$\mathcal{F}\{\cos(cx)e^{-ax}\}(\omega) = \frac{a+j\omega}{(a+j\omega)^2+c^2} \qquad (A.3.19)$$

The peak magnitude occurs near $\omega = c$, and increasing c shifts the peak linearly to higher frequencies.

Proof of Bandwidth Control:

The exponential term e^{-ax} corresponds to a Lorentzian spectrum in the frequency domain:

$$\mathcal{F}\{e^{-ax}\}(\omega) = \frac{1}{a+j\omega} \tag{A.3.20}$$

The peak magnitude occurs at $\omega = 0$:

$$|\mathcal{F}\{g(x)\}(0)| = \frac{1}{a} \tag{A.3.21}$$

Let $\omega = \Gamma/2$ be the frequency where the magnitude is half the peak value:

$$\frac{1}{\sqrt{a^2+(\Gamma/2)^2}} = \frac{1}{2} \cdot \frac{1}{a} \tag{A.3.22}$$

The FWHM of the Lorentzian profile is often approximated as:

$$\Gamma \approx 2a \tag{A.3.23}$$

Therefore, increasing a broadens the bandwidth but shortens the temporal localization window $x_{max} \propto 1/a$, in accordance with the time-frequency uncertainty principle.

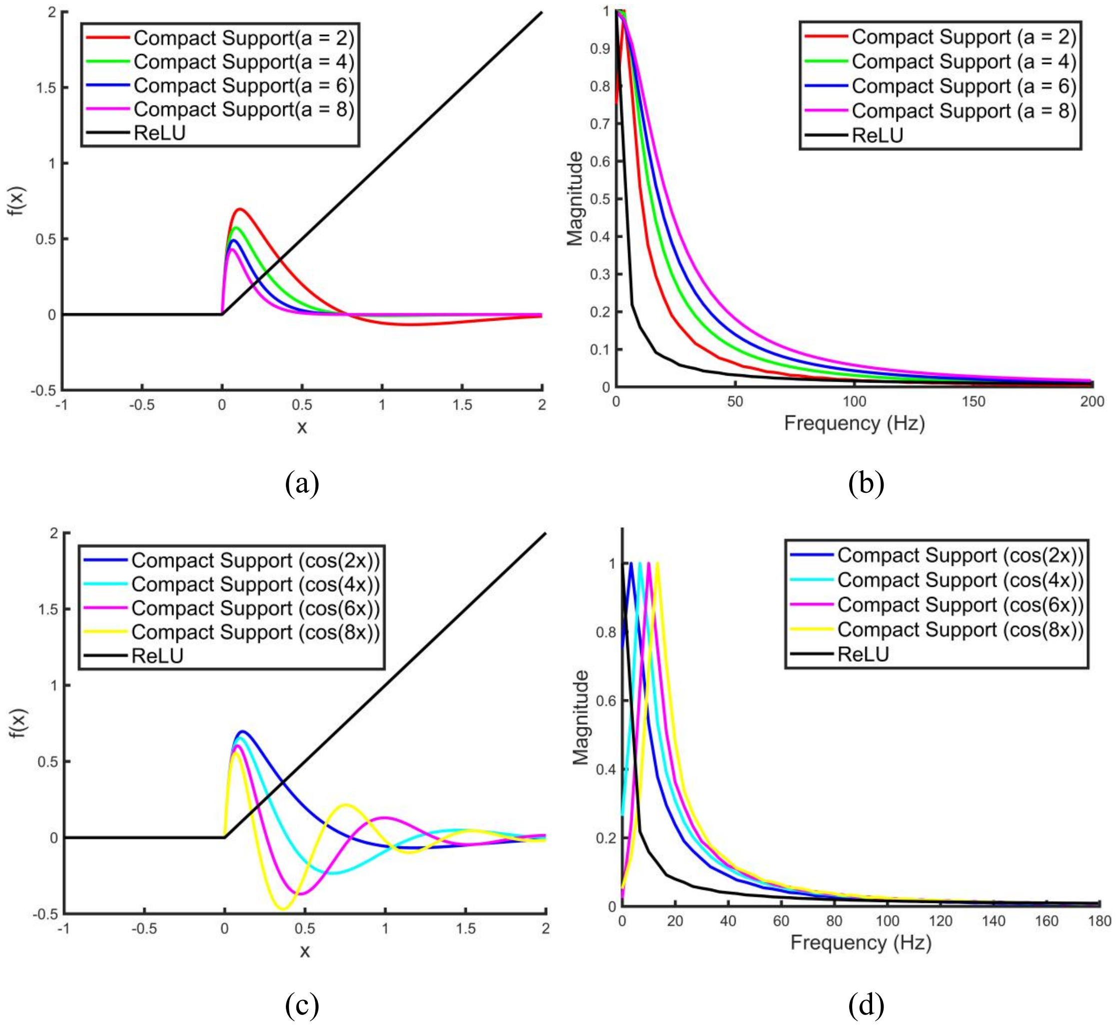


Figure *A.3.*1 Time and frequency domain analysis of parametric activation function (PAF): (a) Time domain representation for parameter a with the fixed parameter b and c; (b) Frequency domain representation corresponding to (a); (c) Time domain representation for parameter c with the fixed parameter a and b; (d) Frequency domain representation corresponding to (c).

Remark 2. Figure *A.3.*1 provides a visual analysis of the time-frequency characteristics of the activation function under parameter adjustments. In Figures *A.3.*1(a) and (b), increasing parameter a significantly shortens the decay duration of the time-domain signal, concentrating energy within a narrower temporal window. Correspondingly, the frequency-domain responsing demonstrates that a larger a broadens the spectral bandwidth and decelerates high-frequency decay.

In Figures *A.3.*1(c) and (d), the impact of parameter c is highlighted. The time-domain signal shows that increasing c introduces higher-frequency oscillations, resulting in a denser waveform. The corresponding frequency-domain response reveals that the spectral energy peak shifts linearly from low-frequency regions to higher frequencies, with stable peak magnitudes.

Experimental observations closely match theoretical derivations: the temporal decay rate, spectral bandwidth broadening, and oscillation frequency collectively demonstrate the flexibility of this activation function in joint time-frequency optimization. In contrast, traditional ReLU's fixed decay characteristics cannot achieve such dynamic control, as its frequency-domain energy remains dispersed and unable to effectively learn high-frequency information. This tunability provides critical advantages for complex tasks, by adaptively adjusting parameters a, b, and c, networks can capture transient features in the time domain while focusing on essential frequency components, thereby improving learning efficiency and robustness.

Appendix *A.4* Physics-Constrained Adaptive MSE-Quantile Loss (PCAMQ-Loss) Algorithm

Algorithm 1 Physics-Constrained Adaptive MSE-Quantile Loss (PCAMQ-Loss)

Input: Ground truth y_{true}, prediction y_{pred}, calibrated wave speed $c_{\text{calibrated}}$, initial weight $w_{\text{init}} = 0.5$, balance factor $\beta = 0.1$

Output: Total loss L_{total}

Initialize dynamic weight $w \leftarrow w_{\text{init}}$
Record weight history $\mathcal{H} \leftarrow \emptyset$
procedure QuantileLoss($y_{\text{true}}, y_{\text{pred}}, q = 0.45$)
    Convert tensors: $y_{\text{pred}} \leftarrow$ `tf.convert_to_tensor`(y_{pred}, `dtype=tf.float64`)
    $y_{\text{true}} \leftarrow$ `tf.cast`($y_{\text{true}}, y_{\text{pred}}$`.dtype`)
    Compute residual: $e \leftarrow y_{\text{pred}} - y_{\text{true}}$
    $L_q \leftarrow \frac{1}{N}\sum_{i=1}^{N} \max(q|e_i|, (1-q)|e_i|)$
    return L_q
end procedure
procedure PhysicsGuidedMSE($y_{\text{true}}, y_{\text{pred}}, c_{\text{calibrated}}$)
    $d_{\text{pred}} \leftarrow y_{\text{pred}} \cdot c_{\text{calibrated}}$
    $d_{\text{true}} \leftarrow y_{\text{true}} \cdot c_{\text{calibrated}}$
    $L_{\text{physics}} \leftarrow \frac{1}{N}\sum(d_{\text{pred}} - d_{\text{true}})^2$
    return L_{physics}
end procedure
procedure DynamicWeight($L_{\text{physics}}, L_q, \beta$)
    $\Delta \leftarrow L_{\text{physics}} - L_q$
    $w_{\text{adj}} \leftarrow 1 + \beta \cdot \tanh(\Delta)$
    if $L_{\text{physics}} > L_q$ **then**
        $w \leftarrow w \cdot (0.995 \cdot w_{\text{adj}})$
    else
        $w \leftarrow w \cdot (1.002 / w_{\text{adj}})$
    end if
    $w \leftarrow \text{clip}(w, [0.01, 0.8])$
    return w
end procedure
procedure TotalLoss($y_{\text{true}}, y_{\text{pred}}, c_{\text{calibrated}}$)
    $L_q \leftarrow \text{QuantileLoss}(y_{\text{true}}, y_{\text{pred}})$
    $L_{\text{physics}} \leftarrow \text{PhysicsGuidedMSE}(y_{\text{true}}, y_{\text{pred}}, c_{\text{calibrated}})$
    $w \leftarrow \text{DynamicWeight}(L_{\text{physics}}, L_q, \beta = 0.1)$
    $L_{\text{total}} \leftarrow w \cdot L_{\text{physics}} + L_q$
    return L_{total}
end procedure

Appendix *A.5* Loss Landscape Visualization and Hessian Curvature Analysis Algorithm

Algorithm 2 Loss Landscape Visualization and Hessian Curvature Analysis

Input: Trained model, training data, calibration parameters:
- Save interval $k = 5$
- PCA components $n = 2$
- Grid resolution $r = 500$
- RBF parameters $\epsilon = 0.3$

Output: 3D loss landscape plots, curvature statistics

procedure DataCollection
  Initialize `LossLandscapeCallback` with $k = 5$
  During training:
  for each epoch $t \in [1, T]$ **do**
    if $t \mod k = 0$ **then**
      Flatten and concatenate all layer weights θ_t
      Record θ_t and current loss $\mathcal{L}_t$
    end if
  end for
  Output: Weight matrix $\Theta \in R^{m \times p}$, loss vector $\mathbf{L} \in R^m$
end procedure
procedure DimensionalityReduction(Θ)
  Apply incremental PCA with chunk size 5000:
  for each chunk $\Theta_{\text{chunk}} \subset \Theta$ **do**
    PCA.partial_fit(Θ_{chunk})
  end for
  Project weights: $\Theta_{\text{PCA}} \leftarrow$ PCA.transform(Θ)
  Output: $\Theta_{\text{PCA}} \in R^{m \times 2}$
end procedure
procedure GridInterpolation($\Theta_{\text{PCA}}, \mathbf{L}$)
  Generate parameter grid:
  $x_{\text{grid}} \leftarrow \text{linspace}(\min(\Theta_{\text{PCA}}[:,0]), \max(\Theta_{\text{PCA}}[:,0]), r)$
  $y_{\text{grid}} \leftarrow \text{linspace}(\min(\Theta_{\text{PCA}}[:,1]), \max(\Theta_{\text{PCA}}[:,1]), r)$
  $X_{\text{grid}}, Y_{\text{grid}} \leftarrow \text{meshgrid}(x_{\text{grid}}, y_{\text{grid}})$
  Interpolate loss values:
  $\mathcal{L}_{\text{interp}} \leftarrow \text{RBF}(\Theta_{\text{PCA}}, \mathbf{L}, \text{function='inverse'})(X_{\text{grid}}, Y_{\text{grid}})$ LinAlgError
  $\mathcal{L}_{\text{interp}} \leftarrow \text{griddata}(\Theta_{\text{PCA}}, \mathbf{L}, (X_{\text{grid}}, Y_{\text{grid}}), \text{method='linear'})$
  Handle NaN: $\mathcal{L}_{\text{interp}}[\text{isnan}(\mathcal{L}_{\text{interp}})] \leftarrow \text{mean}(\mathbf{L})$
  Output: $\mathcal{L}_{\text{interp}}$
end procedure
procedure Visualization($X_{\text{grid}}, Y_{\text{grid}}, \mathcal{L}_{\text{interp}}$)
  Create 3D surface plot:
  `ax.plot_surface`($\text{X}_{\text{grid}}, Y_{\text{grid}}, \text{L}_{\text{interp}}$, cmap='plasma')
  Create filled contour plot:
  `ax.contourf`($\text{X}_{\text{grid}}, Y_{\text{grid}}, \text{L}_{\text{interp}}$, levels=20)
  Create line contour plot:
  `ax.contour`($\text{X}_{\text{grid}}, Y_{\text{grid}}, \text{L}_{\text{interp}}$, levels=20)
end procedure
procedure CurvatureAnalysis($\Theta_{\text{PCA}}, \mathbf{L}$)
  Find optimal point: $\theta^* \leftarrow \Theta_{\text{PCA}}[\arg\min(\mathbf{L})]$
  Generate perturbations: $\theta_{\text{perturbed}} \leftarrow \theta^* + \mathcal{N}(0, 0.1^2)$
  for each $\theta \in \{\theta^*\} \cup \theta_{\text{perturbed}}$ **do**
    Compute Hessian matrix H:

$$H_{ij} = \frac{\mathcal{L}(\theta + \epsilon e_i + \epsilon e_j) - 2\mathcal{L}(\theta) + \mathcal{L}(\theta - \epsilon e_j)}{\epsilon^2} \quad (\epsilon = 10^{-6})$$

    Calculate eigenvalues $\lambda_1, \lambda_2 \leftarrow \text{eig}(H)$
  end for
  Compute statistics:
  Mean Hessian $\mu_H \leftarrow \frac{1}{N}\sum_{i=1}^{N} H^{(i)}$
  Std. dev. $\sigma_H \leftarrow \sqrt{\frac{1}{N}\sum_{i=1}^{N}(H^{(i)} - \mu_H)^2}$
end procedure

Reference

[1] B. Wang, S. Zhong, T.-L. Lee, K. S. Fancey, and J. J. A. i. m. e. Mi, "Non-destructive testing and evaluation of composite materials/structures: A state-of-the-art review," vol. 12, no. 4, p. 1687814020913761, 2020.

[2] J. Jodhani, A. Handa, A. Gautam, and R. J. M. T. P. Rana, "Ultrasonic non-destructive evaluation of composites: A review," vol. 78, pp. 627-632, 2023.

[3] A. P. Sarvazyan, M. W. Urban, J. F. J. U. i. m. Greenleaf, and biology, "Acoustic waves in medical imaging and diagnostics," vol. 39, no. 7, pp. 1133-1146, 2013.

[4] A. P. Sarvazyan, O. V. Rudenko, S. D. Swanson, J. B. Fowlkes, S. Y. J. U. i. m. Emelianov, and biology, "Shear wave elasticity imaging: a new ultrasonic technology of medical diagnostics," vol. 24, no. 9, pp. 1419-1435, 1998.

[5] R. W. King, M. Owens, and T. T. Wu, *Lateral electromagnetic waves: Theory and applications to communications, geophysical exploration, and remote sensing*. Springer Science & Business Media, 2012.

[6] A. Ishimaru, *Electromagnetic wave propagation, radiation, and scattering: from fundamentals to applications*. John Wiley & Sons, 2017.

[7] M. Fazli and S. A. L. J. A. r. i. e. s. Maftunzada, "THE ROLE OF ELECTROMAGNETIC WAVES IN MODERN COMMUNICATIONS," vol. 5, no. 7-8, pp. 80-89, 2024.

[8] C. Pendão and I. J. S. Silva, "Optical fiber sensors and sensing networks: Overview of the main principles and applications," vol. 22, no. 19, p. 7554, 2022.

[9] M. Ramakrishnan, G. Rajan, Y. Semenova, and G. J. S. Farrell, "Overview of fiber optic sensor technologies for strain/temperature sensing applications in composite materials," vol. 16, no. 1, p. 99, 2016.

[10] L. Thévenaz, M. Niklès, A. Fellay, M. Facchini, and P. A. Robert, "Truly distributed strain and temperature sensing using embedded optical fibers," in *Smart Structures and Materials 1998: Sensory Phenomena and Measurement Instrumentation for Smart Structures and Materials*, 1998, vol. 3330, pp. 301-314: SPIE.

[11] T. Chong, X. Tang, C. Leng, M. Yogeswaran, O. Ng, and Y. J. P. C. S. Chong, "Sensor technologies and simultaneous localization and mapping (SLAM)," vol. 76, pp. 174-179, 2015.

[12] A. Diosi and L. Kleeman, "Advanced sonar and laser range finder fusion for simultaneous localization and mapping," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, 2004, vol. 2, pp. 1854-1859: IEEE.

[13] A. Diosi, G. Taylor, and L. Kleeman, "Interactive SLAM using laser and advanced sonar," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 1103-1108: IEEE.

[14] S. Zheng, J. Wang, C. Rizos, W. Ding, and A. J. R. S. El-Mowafy, "Simultaneous localization and mapping (slam) for autonomous driving: concept and analysis," vol. 15, no. 4, p. 1156,

2023.

[15] H. Wang, M. Cao, H. Jiang, and L. Xie, "Feasible computationally efficient path planning for UAV collision avoidance," in *2018 IEEE 14th International Conference on Control and Automation (ICCA)*, 2018, pp. 576-581: IEEE.

[16] G. Naveenkumar, M. Suriyaprakash, and T. Prem Anand, "Autonomous Drone Using Time-of-Flight Sensor for Collision Avoidance," in *Intelligent Communication Technologies and Virtual Mobile Networks*: Springer, 2023, pp. 57-73.

[17] L. Bascetta, G. Magnani, P. Rocco, R. Migliorini, and M. Pelagatti, "Anti-collision systems for robotic applications based on laser time-of-flight sensors," in *2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2010, pp. 278-284: IEEE.

[18] G. A. Koulieris, K. Akşit, M. Stengel, R. K. Mantiuk, K. Mania, and C. Richardt, "Near-eye display and tracking technologies for virtual and augmented reality," in *Computer Graphics Forum*, 2019, vol. 38, no. 2, pp. 493-519: Wiley Online Library.

[19] V. Blahnik and O. J. A. O. T. Schindelbeck, "Smartphone imaging technology and its applications," vol. 10, no. 3, pp. 145-232, 2021.

[20] W. Kutschera, A. T. Jull, M. Paul, and A. J. R. o. M. P. Wallner, "Atom counting with accelerator mass spectrometry," vol. 95, no. 3, p. 035006, 2023.

[21] M. J. J. o. P. G. N. Williams and P. Physics, "Particle identification using time of flight," vol. 39, no. 12, p. 123001, 2012.

[22] J. Liu, Q. Sun, Z. Fan, and Y. Jia, "TOF lidar development in autonomous vehicle," in *2018 IEEE 3rd Optoelectronics Global Conference (OGC)*, 2018, pp. 185-190: IEEE.

[23] T. Yang, Y. Li, Y. Ruichek, and Z. J. I. T. o. I. T. S. Yan, "Performance modeling a near-infrared tof lidar under fog: A data-driven approach," vol. 23, no. 8, pp. 11227-11236, 2021.

[24] A. Mehranian *et al.*, "Deep learning–based time-of-flight (ToF) image enhancement of non-ToF PET scans," vol. 49, no. 11, pp. 3740-3749, 2022.

[25] J. Kriegsmann, M. Kriegsmann, and R. J. I. j. o. o. Casadonte, "MALDI TOF imaging mass spectrometry in clinical pathology: a valuable tool for cancer diagnostics," vol. 46, no. 3, pp. 893-906, 2015.

[26] C. Bamji *et al.*, "A review of indirect time-of-flight technologies," vol. 69, no. 6, pp. 2779-2793, 2022.

[27] M. Georgiev, R. Bregović, and A. J. I. T. o. I. P. Gotchev, "Time-of-flight range measurement in low-sensing environment: Noise analysis and complex-domain non-local denoising," vol. 27, no. 6, pp. 2911-2926, 2018.

[28] A. Burr, "The multipath problem: An overview," in *IEE Colloquium on Multipath Countermeasures*, 1996, pp. 1/1-1/7: IET.

[29] W. M. Gifford, D. Dardari, and M. Z. J. I. T. o. S. P. Win, "The impact of multipath information on time-of-arrival estimation," vol. 70, pp. 31-46, 2020.

[30] M. Chamseddine, J. Rambach, D. Stricker, and O. Wasenmuller, "Ghost target detection in 3d radar data using point cloud based deep neural network," in *2020 25th International*

Conference on Pattern Recognition (ICPR), 2021, pp. 10398-10403: IEEE.

[31] J. Soubielle, I. Fijalkow, P. Duvaut, and A. J. I. T. o. S. P. Bibaut, "GPS positioning in a multipath environment," vol. 50, no. 1, pp. 141-150, 2002.

[32] T. Kos, I. Markezic, and J. Pokrajcic, "Effects of multipath reception on GPS positioning performance," in *Proceedings ELMAR-2010*, 2010, pp. 399-402: IEEE.

[33] S. M. J. I. T. o. A. Garber and E. Systems, "High resolution sonar signals in a multipath environment," no. 6, pp. 431-440, 2008.

[34] D. T. Bradley, "Architectural Acoustics: Measurement and Modeling of Scattering Effects."

[35] T.-T. Nguyen, C.-H. Cheng, D.-G. Liu, and M.-H. J. E. Le, "A fast cross-correlation combined with interpolation algorithms for the LiDAR working in the high background noise," vol. 11, no. 7, p. 985, 2022.

[36] T. Chenevert *et al.*, "Ultrasonic computed tomography of the breast. Improvement of image quality by use of cross-correlation time-of-flight and phase-insensitive attenuation measurements," vol. 152, no. 1, pp. 155-159, 1984.

[37] M. Barroso-Romero, D. Gagar, S. Pant, and M. Martinez, "Wave mode identification of acoustic emission signals using phase analysis," in *Acoustics*, 2019, vol. 1, no. 2, pp. 450-472: MDPI.

[38] M. Mitra, S. J. S. M. Gopalakrishnan, and Structures, "Guided wave based structural health monitoring: A review," vol. 25, no. 5, p. 053001, 2016.

[39] A. Raghavan, "Guided-wave structural health monitoring," 2007.

[40] F. Yan, R. L. Royer Jr, J. L. J. J. o. i. m. S. Rose, and Structures, "Ultrasonic guided wave imaging techniques in structural health monitoring," vol. 21, no. 3, pp. 377-384, 2010.

[41] M. Abbas and M. J. S. Shafiee, "Structural health monitoring (SHM) and determination of surface defects in large metallic structures using ultrasonic guided waves," vol. 18, no. 11, p. 3958, 2018.

[42] V. Memmolo, E. Monaco, N. Boffa, L. Maio, and F. J. C. S. Ricci, "Guided wave propagation and scattering for structural health monitoring of stiffened composites," vol. 184, pp. 568-580, 2018.

[43] J. L. J. K. e. m. Rose, "Ultrasonic guided waves in structural health monitoring," vol. 270, pp. 14-21, 2004.

[44] P. Cawley, "Practical guided wave inspection and applications to structural health monitoring," 2007.

[45] M. C. J. E. o. s. m. Bhardwaj, "Non-destructive evaluation," vol. 2, pp. 690-714, 2002.

[46] J. A. Epaarachchi and G. C. Kahandawa, *Structural health monitoring technologies and next-generation smart composite structures*. CRC Press, 2016.

[47] B. Zima, M. J. A. o. c. Rucka, and m. engineering, "Guided waves for monitoring of plate structures with linear cracks of variable length," vol. 16, no. 3, pp. 387-396, 2016.

[48] H. Liu *et al.*, "Ultrasonic guided wave damage localization method for composite fan blades based on damage-scattered wave difference," vol. 33, no. 10, p. 105011, 2024.

[49] V. Giurgiutiu, *Structural health monitoring with piezoelectric wafer active sensors*. Academic press, 2014.

[50] H. Mei, M. F. Haider, R. Joseph, A. Migot, and V. J. S. Giurgiutiu, "Recent advances in piezoelectric wafer active sensors for structural health monitoring applications," vol. 19, no. 2, p. 383, 2019.

[51] R. Guan, Y. Lu, W. Duan, X. J. S. c. Wang, and h. monitoring, "Guided waves for damage identification in pipeline structures: A review," vol. 24, no. 11, p. e2007, 2017.

[52] V. Memmolo, L. Maio, and F. J. S. Ricci, "Assessment of damage in composite pressure vessels using guided waves," vol. 22, no. 14, p. 5182, 2022.

[53] R. Soleimanpour, M. H. Yassin, A. Mehdizadeh, and D. J. A. A. Chronopoulos, "Locating cracks in isotropic plates using nonlinear guided waves," vol. 211, p. 109522, 2023.

[54] L. Capineri and A. J. S. Bulletti, "Ultrasonic guided-waves sensors and integrated structural health monitoring systems for impact detection and localization: A review," vol. 21, no. 9, p. 2929, 2021.

[55] L. Yu, V. J. S. S. Giurgiutiu, and Systems, "Advanced signal processing for enhanced damage detection with piezoelectric wafer active sensors," vol. 1, no. 2, pp. 185-215, 2005.

[56] S. Shan, J. Qiu, C. Zhang, H. Ji, and L. J. S. H. M. Cheng, "Multi-damage localization on large complex structures through an extended delay-and-sum based method," vol. 15, no. 1, pp. 50-64, 2016.

[57] G. Lu *et al.*, "A multi-delay-and-sum imaging algorithm for damage detection using piezoceramic transducers," vol. 28, no. 9, pp. 1150-1159, 2017.

[58] C. B. Nzouatchoua *et al.*, "Damage localization on composite structures based on the delay-and-sum algorithm using simulation and experimental methods," vol. 23, no. 9, p. 4368, 2023.

[59] Q. Wang, M. Hong, Z. J. S. M. Su, and Structures, "A sparse sensor network topologized for cylindrical wave-based identification of damage in pipeline structures," vol. 25, no. 7, p. 075015, 2016.

[60] A. Demma, "The interaction of guided waves with discontinuities in structures," University of London London, UK, 2003.

[61] P. Rajagopal, "Towards higher resolution guided wave inspection: scattering studies," Imperial College London (University of London), 2007.

[62] S. Banerjee, F. Ricci, E. Monaco, A. J. J. o. s. Mal, and vibration, "A wave propagation and vibration-based approach for damage identification in structural components," vol. 322, no. 1-2, pp. 167-183, 2009.

[63] O. Mesnil and M. J. U. Ruzzene, "Sparse wavefield reconstruction and source detection using compressed sensing," vol. 67, pp. 94-104, 2016.

[64] Z. Zhang, L. Zeng, and N. J. S. H. M. Zhang, "Damage imaging using multipath-scattered Lamb waves under a sparse reconstruction framework," vol. 23, no. 4, pp. 2013-2025, 2024.

[65] T.-H. J. S. Yu, "Plate waves scattering analysis and active damage detection," vol. 21, no. 16, p. 5458, 2021.

[66] C. Zhou, "A diagnostic imaging technique for identification of structural damage using hybrid features of Lamb wave signals," 2011.

[67] M. Bilodeau, N. Quaegebeur, P. J. N. Masson, and E. International, "Design of a guided wave absorber for Structural Health Monitoring system development," vol. 88, pp. 33-41, 2017.

[68] D. Chen, W. Li, Z. Dong, R. Fu, Q. J. S. M. Yu, and Structures, "Multi-bolt looseness monitoring using guided waves: a cross-correlation approach of the wavelet energy envelope," vol. 33, no. 12, p. 125019, 2024.

[69] M. R. Taha, A. Noureldin, J. L. Lucero, and T. J. J. S. h. m. Baca, "Wavelet transform for structural health monitoring: a compendium of uses and features," vol. 5, no. 3, pp. 267-295, 2006.

[70] Y. Kankanamge, Y. Hu, X. J. E. E. Shao, and E. Vibration, "Application of wavelet transform in structural health monitoring," vol. 19, no. 2, pp. 515-532, 2020.

[71] M. Azimi, A. D. Eslamlou, and G. J. S. Pekcan, "Data-driven structural health monitoring and damage detection through deep learning: State-of-the-art review," vol. 20, no. 10, p. 2778, 2020.

[72] A. Malekloo, E. Ozer, M. AlHamaydeh, and M. J. S. H. M. Girolami, "Machine learning and structural health monitoring overview with emerging technology and high-dimensional data source highlights," vol. 21, no. 4, pp. 1906-1955, 2022.

[73] A. Sattarifar and T. J. E. A. o. A. I. Nestorović, "Damage localization and characterization using one-dimensional convolutional neural network and a sparse network of transducers," vol. 115, p. 105273, 2022.

[74] M. Schaller, D. Schlör, and A. J. a. p. a. Hotho, "ModeConv: a novel convolution for distinguishing anomalous and normal structural behavior," 2024.

[75] X. Li *et al.*, "Interpretable deep learning: Interpretation, interpretability, trustworthiness, and beyond," vol. 64, no. 12, pp. 3197-3234, 2022.

[76] Y. Xu, S. Kohtz, J. Boakye, P. Gardoni, P. J. R. E. Wang, and S. Safety, "Physics-informed machine learning for reliability and systems safety applications: State of the art and challenges," vol. 230, p. 108900, 2023.

[77] C. Meng, S. Griesemer, D. Cao, S. Seo, Y. J. M. L. f. C. S. Liu, and Engineering, "When physics meets machine learning: A survey of physics-informed machine learning," vol. 1, no. 1, p. 20, 2025.

[78] S.-P. Zhu *et al.*, "Physics-informed machine learning and its structural integrity applications: state of the art," vol. 381, no. 2260, p. 20220406, 2023.

[79] Z. Hao *et al.*, "Physics-informed machine learning: A survey on problems, methods and applications," 2022.

[80] S. J. Raymond and D. B. J. a. p. a. Camarillo, "Applying physics-based loss functions to neural networks for improved generalizability in mechanics problems," 2021.

[81] J. Willard, X. Jia, S. Xu, M. Steinbach, and V. J. a. p. a. Kumar, "Integrating physics-based modeling with machine learning: A survey," vol. 1, no. 1, pp. 1-34, 2020.

[82] X. Xu and C. J. C. S. Liu, "Physics-guided deep learning for damage detection in CFRP composite structures," vol. 331, p. 117889, 2024.

[83] S. A. Zargar and F.-G. J. S. H. M. Yuan, "Physics-informed deep learning for scattered full wavefield reconstruction from a sparse set of sensor data for impact diagnosis in structural health monitoring," vol. 23, no. 5, pp. 2963-2979, 2024.

[84] K. O'shea and R. J. a. p. a. Nash, "An introduction to convolutional neural networks," 2015.

[85] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. J. I. t. o. n. n. Monfardini, "The graph neural network model," vol. 20, no. 1, pp. 61-80, 2008.

[86] Z. Niu, G. Zhong, and H. J. N. Yu, "A review on the attention mechanism of deep learning," vol. 452, pp. 48-62, 2021.

[87] Z.-Q. J. Xu, Y. Zhang, T. J. C. o. A. M. Luo, and Computation, "Overview frequency principle/spectral bias in deep learning," pp. 1-38, 2024.

[88] A. Jacot, F. Gabriel, and C. J. A. i. n. i. p. s. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks," vol. 31, 2018.

[89] Y. Zhang, T. Luo, Z. Ma, and Z.-Q. J. J. C. P. L. Xu, "A linear frequency principle model to understand the absence of overfitting in neural networks," vol. 38, no. 3, p. 038701, 2021.

[90] M. Zhu and H. Zhou, "Time of Flight Measurement Method Combining Threshold Method and Cross-Correlation Method," in *Journal of Physics: Conference Series*, 2023, vol. 2656, no. 1, p. 012014: IOP Publishing.

[91] W. Pan, N. Ma, and Y. J. R. S. Wang, "An Envelope Travel-Time Objective Function for Reducing Source–Velocity Trade-Offs in Wave-Equation Tomography," vol. 14, no. 20, p. 5223, 2022.

[92] D. Dai and Q. J. S. P. He, "Structure damage localization with ultrasonic guided waves based on a time–frequency method," vol. 96, pp. 21-28, 2014.

[93] M. Salmanpour, Z. Sharif Khodaei, M. J. S. C. Aliabadi, and H. Monitoring, "Transducer placement optimisation scheme for a delay and sum damage detection algorithm," vol. 24, no. 4, p. e1898, 2017.

[94] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. J. N. R. P. Yang, "Physics-informed machine learning," vol. 3, no. 6, pp. 422-440, 2021.

[95] H. V. Tetali, J. B. Harley, and B. D. J. I. T. o. S. P. Haeffele, "Wave physics-informed matrix factorizations," vol. 72, pp. 535-548, 2024.

[96] P. J. Baddoo, B. Herrmann, B. J. McKeon, J. Nathan Kutz, and S. L. J. P. o. t. R. S. A. Brunton, "Physics-informed dynamic mode decomposition," vol. 479, no. 2271, p. 20220576, 2023.

[97] Y. Wu, B. Sicard, and S. A. J. E. S. w. A. Gadsden, "Physics-informed machine learning: A comprehensive review on applications in anomaly detection and condition monitoring," vol. 255, p. 124678, 2024.

[98] K. Shukla, P. C. Di Leoni, J. Blackshire, D. Sparkman, and G. E. J. J. o. N. E. Karniadakis, "Physics-informed neural network for ultrasound nondestructive quantification of surface breaking cracks," vol. 39, no. 3, p. 61, 2020.

[99] C. R. Farrar and K. Worden, *Structural health monitoring: a machine learning perspective*. John Wiley & Sons, 2012.

[100] Y. Bao and H. J. S. h. m. Li, "Machine learning paradigm for structural health monitoring," vol. 20, no. 4, pp. 1353-1372, 2021.

[101] S. Zhang, C. M. Li, W. J. M. S. Ye, and S. Processing, "Damage localization in plate-like structures using time-varying feature and one-dimensional convolutional neural network," vol. 147, p. 107107, 2021.

[102] S. Zhang, C. M. Li, J. Yang, W. J. M. S. Ye, and S. Processing, "Effective combination of modeling and experimental data with deep metric learning for guided wave-based damage localization in plates," vol. 172, p. 108979, 2022.

[103] X. Wang *et al.*, "Ultrasonic guided wave imaging with deep learning: Applications in corrosion mapping," vol. 169, p. 108761, 2022.

[104] M. Rautela and S. Gopalakrishnan, "Deep learning frameworks for wave propagation-based damage detection in 1d-waveguides," in *Proceedings of 11th International Symposium on NDT in Aerospace*, 2019, vol. 2, pp. 1-11.

[105] C. Sbarufatti, G. Manson, K. J. J. o. S. Worden, and Vibration, "A numerically-enhanced machine learning approach to damage diagnosis using a Lamb wave sensing network," vol. 333, no. 19, pp. 4499-4525, 2014.

[106] M. Lin, S. Guo, S. He, W. Li, and D. J. C. S. Yang, "Structure health monitoring of a composite wing based on flight load and strain data using deep learning method," vol. 286, p. 115305, 2022.

[107] M. Heesch, M. Dziendzikowski, K. Mendrok, and Z. J. S. Dworakowski, "Diagnostic-quality guided wave signals synthesized using generative adversarial neural networks," vol. 22, no. 10, p. 3848, 2022.

[108] C. Liu, Y. Chen, and X. J. E. S. w. A. Xu, "Structural digital Twin for damage detection of CFRP composites using meta transfer Learning-based approach," vol. 261, p. 125458, 2025.

[109] X. Liu, Z. Jiang, and Z. Yan, "Improvement of accuracy in damage localization using frequency slice wavelet transform," *Shock and Vibration,* vol. 19, no. 4, pp. 585-596, 2012.

[110] S.-i. Amari, "Backpropagation and stochastic gradient descent method," *Neurocomputing,* vol. 5, no. 4-5, pp. 185-196, 1993.

[111] Z. Zhang, "Improved adam optimizer for deep neural networks," in *2018 IEEE/ACM 26th international symposium on quality of service (IWQoS)*, 2018, pp. 1-2: Ieee.

[112] Q. Ma, E. Chen, Z. Lin, J. Yan, Z. Yu, and W. W. Ng, "Convolutional multitimescale echo state network," *IEEE Transactions on Cybernetics,* vol. 51, no. 3, pp. 1613-1625, 2019.

[113] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807-814.

[114] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, 2015, pp.

448-456: pmlr.

[115] H. Gholamalinezhad and H. Khosravi, "Pooling methods in deep neural networks, a review," *arXiv preprint arXiv:2009.07485,* 2020.

[116] R. Kruse, S. Mostaghim, C. Borgelt, C. Braune, and M. Steinbrecher, "Multi-layer perceptrons," in *Computational intelligence: a methodological introduction*: Springer, 2022, pp. 53-124.

[117] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to linear regression analysis*. John Wiley & Sons, 2021.

[118] G. C. J. W. I. R. C. S. McDonald, "Ridge regression," vol. 1, no. 1, pp. 93-100, 2009.

[119] J. Ranstam and J. A. J. J. o. B. S. Cook, "LASSO regression," vol. 105, no. 10, pp. 1348-1348, 2018.

[120] Y.-Y. Song and L. J. S. a. o. p. Ying, "Decision tree methods: applications for classification and prediction," vol. 27, no. 2, p. 130, 2015.

[121] A. Maćkiewicz, W. J. C. Ratajczak, and Geosciences, "Principal components analysis (PCA)," vol. 19, no. 3, pp. 303-342, 1993.

[122] M. Awad, R. Khanna, M. Awad, R. J. E. l. m. T. Khanna, concepts,, a. f. engineers, and s. designers, "Support vector regression," pp. 67-80, 2015.

[123] D. Lee and H. S. J. A. i. n. i. p. s. Seung, "Algorithms for non-negative matrix factorization," vol. 13, 2000.

[124] A. J. P. D. N. P. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," vol. 404, p. 132306, 2020.

[125] L. R. Medsker, L. J. D. Jain, and applications, "Recurrent neural networks," vol. 5, no. 64-67, p. 2, 2001.

[126] J. Chung, C. Gulcehre, K. Cho, and Y. J. a. p. a. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014.

[127] Y. LeCun, Y. Bengio, and G. J. n. Hinton, "Deep learning," vol. 521, no. 7553, pp. 436-444, 2015.

[128] A. J. A. i. N. I. P. S. Vaswani, "Attention is all you need," 2017.

[129] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700-4708.

[130] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

[131] A. Krizhevsky, I. Sutskever, and G. E. J. A. i. n. i. p. s. Hinton, "Imagenet classification with deep convolutional neural networks," vol. 25, 2012.

[132] A. C. Belkina, C. O. Ciccolella, R. Anno, R. Halpert, J. Spidlen, and J. E. Snyder-Cappione, "Automated optimized parameters for T-distributed stochastic neighbor embedding improve visualization and analysis of large datasets," *Nature communications,* vol. 10, no. 1, p. 5415,

2019.

[133] Y. Chen *et al.*, "Deep autoencoder for interpretable tissue-adaptive deconvolution and cell-type-specific gene analysis," *Nature Communications,* vol. 13, no. 1, p. 6735, 2022.

[134] Y. Ma, Z.-Q. J. Xu, and J. Zhang, "Frequency principle in deep learning beyond gradient-descent-based training," *arXiv preprint arXiv:2101.00747,* 2021.

[135] R. Zheng, G. Wang, and K. Ho, "Accurate semidefinite relaxation method for elliptic localization with unknown transmitter position," *IEEE Transactions on Wireless Communications,* vol. 20, no. 4, pp. 2746-2760, 2020.

[136] A. Nokhbatolfoghahai, H. M. Navazi, R. M. J. J. o. I. M. S. Groves, and Structures, "Use of delay and sum for sparse reconstruction improvement for structural health monitoring," vol. 30, no. 18-19, pp. 2919-2931, 2019.

[137] L. Qiu, S. Yuan, X. Zhang, Y. J. S. M. Wang, and Structures, "A time reversal focusing based impact imaging method and its evaluation on complexcomposite structures," vol. 20, no. 10, p. 105014, 2011.

[138] F. Ciampa and M. J. S. H. M. Meo, "Impact detection in anisotropic materials using a time reversal approach," vol. 11, no. 1, pp. 43-49, 2012.

[139] P. Kudela, M. Radzieński, W. J. M. S. Ostachowicz, and S. Processing, "Identification of cracks in thin-walled structures by means of wavenumber filtering," vol. 50, pp. 456-466, 2015.

[140] Y. Ren, L. Qiu, S. Yuan, Z. J. M. S. Su, and S. Processing, "A diagnostic imaging approach for online characterization of multi-impact in aircraft composite structures based on a scanning spatial-wavenumber filter of guided wave," vol. 90, pp. 44-63, 2017.

[141] S. H. M. Rizvi and M. J. E. R. E. Abbas, "From data to insight, enhancing structural health monitoring using physics-informed machine learning and advanced data collection methods," vol. 5, no. 3, p. 032003, 2023.

[142] B. Kunzer, M. Berges, and A. J. a. p. a. Dubrawski, "The digital twin landscape at the crossroads of predictive maintenance, machine learning and physics based modeling," 2022.

[143] H. Ali and R. Marchant, "Integrating Physics-Informed Machine Learning for Enhanced Reliability and Failure Prediction in Engineering Systems," 2025.

[144] L. Pinello, L. Lomazzi, G. Jacopo, M. Giglio, and F. J. T. e.-j. o. n. t. Cadini, "Physics-Informed Machine Learning for Structural Damage Diagnosis in Aluminium Plates," vol. 29, no. 7, pp. 1-10, 2024.

[145] V. Rathod and P. Ramuhalli, "Physics-informed neural networks for identification of material properties using standing waves," in *Nondestructive Characterization and Monitoring of Advanced Materials, Aerospace, Civil Infrastructure, and Transportation XVI*, 2022, vol. 12047, pp. 179-188: SPIE.

[146] K. Hara, D. Saito, and H. Shouno, "Analysis of function of rectified linear unit used in deep learning," in *2015 international joint conference on neural networks (IJCNN)*, 2015, pp. 1-8: IEEE.

[147] J. L. Ba, J. R. Kiros, and G. E. J. a. p. a. Hinton, "Layer normalization," 2016.

[148] M.-C. Popescu, V. E. Balas, L. Perescu-Popescu, N. J. W. T. o. C. Mastorakis, and Systems, "Multilayer perceptron and neural networks," vol. 8, no. 7, pp. 579-588, 2009.

[149] A. Vaswani *et al.*, "Attention is all you need," vol. 30, 2017.

[150] L. Wen, X. Li, L. J. N. C. Gao, and Applications, "A transfer convolutional neural network for fault diagnosis based on ResNet-50," vol. 32, no. 10, pp. 6111-6124, 2020.

[151] K. Simonyan and A. J. a. p. a. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.

[152] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, 2019, pp. 6105-6114: PMLR.

[153] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132-7141.

[154] Y. Song, S. Shan, Y. Zhang, and L. J. S. H. M. Cheng, "Physics-guided neural network for structural health monitoring with lamb waves through boundary reflection elimination," p. 14759217241305050, 2025.

[155] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. J. s. Bengio, "Graph attention networks," vol. 1050, no. 20, pp. 10-48550, 2017.

[156] W. Hamilton, Z. Ying, and J. J. A. i. n. i. p. s. Leskovec, "Inductive representation learning on large graphs," vol. 30, 2017.

[157] K. Xu, W. Hu, J. Leskovec, and S. J. a. p. a. Jegelka, "How powerful are graph neural networks?," 2018.

[158] A. Rai, Z. Ahmad, M. J. Hasan, and J.-M. J. S. Kim, "A novel pipeline leak detection technique based on acoustic emission features and two-sample kolmogorov–smirnov test," vol. 21, no. 24, p. 8247, 2021.

[159] T.-K. Nguyen, Z. Ahmad, D.-T. Nguyen, J.-M. J. M. S. Kim, and S. Processing, "A remaining useful lifetime prediction model for concrete structures using Mann-Whitney U test state indicator and deep learning," vol. 222, p. 111795, 2025.

[160] D. L. Whaley III, "The interquartile range: Theory and estimation," East Tennessee State University, 2005.

[161] R. Zheng, G. Wang, and K. J. I. T. o. W. C. Ho, "Accurate semidefinite relaxation method for elliptic localization with unknown transmitter position," vol. 20, no. 4, pp. 2746-2760, 2020.

[162] J. S. Hall and J. E. J. S. H. M. Michaels, "Multipath ultrasonic guided wave imaging in complex structures," vol. 14, no. 4, pp. 345-358, 2015.

[163] Z. Zhang, H. Pan, X. Wang, and Z. J. S. Lin, "Deep learning empowered structural health monitoring and damage diagnostics for structures with weldment via decoding ultrasonic guided wave," vol. 22, no. 14, p. 5390, 2022.

[164] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050-1059: PMLR.

[165] D. Zhang, D. J. F. o. i. d. m. A. Zhang, Features, Classification, and Retrieval, "Wavelet transform," pp. 35-44, 2019.

[166] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3-19.

[167] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 315-323: JMLR Workshop and Conference Proceedings.

[168] F. J. P. r. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," vol. 65, no. 6, p. 386, 1958.

[169] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, 2014, pp. 818-833: Springer.

[170] Y. Dai, F. Gieseke, S. Oehmcke, Y. Wu, and K. Barnard, "Attentional feature fusion," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 3560-3569.

[171] X. Zhang, "Gaussian Distribution," ed, 2010.

[172] S. Hochreiter and J. J. N. c. Schmidhuber, "Long short-term memory," vol. 9, no. 8, pp. 1735-1780, 1997.

[173] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (GRU) neural networks," in *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, 2017, pp. 1597-1600: IEEE.

[174] W. H. L. Pinaya, S. Vieira, R. Garcia-Dias, and A. Mechelli, "Autoencoders," in *Machine learning*: Elsevier, 2020, pp. 193-208.

[175] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818-2826.

[176] L. Van Der Maaten, E. O. Postma, and H. J. J. J. o. m. l. r. Van Den Herik, "Dimensionality reduction: A comparative review," vol. 10, no. 66-71, p. 13, 2009.

[177] R. Pascanu, Y. N. Dauphin, S. Ganguli, and Y. J. a. p. a. Bengio, "On the saddle point problem for non-convex optimization," 2014.

[178] Y.-C. J. B. Chen and Epidemiology, "A tutorial on kernel density estimation and recent advances," vol. 1, no. 1, pp. 161-187, 2017.

[179] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618-626.