Convolutional neural network and level-set spectral element method for ultrasonic imaging of delamination cavities in an anisotropic composite structure

Boyoung Kim[a], Shashwat Maharjan[a], Fazle Mahdi Pranto[a], Bruno Guidio[a], Christoph Schaal[d,e], Chanseok Jeong[a,b,c,*]

[a]*School of Engineering and Technology, Central Michigan University, Mount Pleasant, MI 48859, USA*
[b]*Earth and Ecosystem Science Program, Central Michigan University, Mount Pleasant, MI 48859, USA*
[c]*Institute for Great Lakes Research, Central Michigan University, Mount Pleasant, MI 48859, USA*
[d]*Department of Mechanical Engineering, California State University, Northridge, CA 91330, USA*
[e]*Mechanical and Aerospace Engineering Department, University of California, Los Angeles, CA 90095, USA*

Abstract

We present a computational approach that incorporates a convolutional neural network (CNN) for detecting internal delamination in a layered 2D plane-strain anisotropic composite structure of transient elastodynamic fields. The two-dimensional spectral element method (SEM) is utilized to simulate the propagation of elastic waves in an orthotropic solid sandwiched by isotropic solids and their interaction with the internal delamination cavity. This work generates training data consisting of input-layer features (i.e., measured wave signals) and output-layer features (i.e., element types, such as void or regular, of all elements in a domain).

To accelerate training data generation, we utilize explicit time integration (e.g., the Runge-Kutta scheme) coupled with an SEM wave solver. Applying the level-set method additionally avoids having to perform an expensive re-meshing process for every possible configuration of the delamination cavities during the data-generation phase. The CNN is trained to classify each element as a non-void or void element from the measured wave signals. Clusters of identified void elements reconstruct targeted cavities. Once our neural network is trained using synthetic data, we analyze how effectively the CNN performs on synthetic measurement data. To this end, we use blind test data from a third-party simulator that explicitly models the traction-free boundary of cavities for anisotropic materials without the application of the level-set method.

Our numerical examples show that our approach can effectively detect the internal cavities in an anisotropic structure made of aluminum and carbon fiber-reinforced epoxy using the measured elastic waves without any prior information about the cavities' locations, shapes, and numbers. The presented method can be extended into a more realistic 3D setting and utilized for the nondestructive test of various anisotropic composite structures.

Keywords: Imaging of delamination cavities, Element-wise classification, Convolutional neural network, Machine learning, Level-set method, Spectral element method.

1. Introduction

The advancement of the manufacturing industry has brought composite materials to our daily lives. By formulating a composite, we adopt merits of each raw material while eliminating the limitations of each. One such composite material system is fiber metal laminates, where metal sheets are bonded to plies of fiber-reinforced epoxy to fabricate a new class of materials [1]. Fiber metal laminates are used in myriad of applications. Carbon fiber-reinforced metal laminates are increasingly used in the aviation industry for manufacturing aircraft because of their excellent mechanical properties, such as high strength-to-weight ratio, improved resistance to fatigue, higher damage tolerance, and enhanced mechanical and thermal properties over conventional materials [2, 3]. The weight proportion of composites is approximately 25% to 50%, respectively, in the modern aircraft, including the Airbus A380 and Boeing B787, of

*Corresponding author
Email address: jeong1c@cmich.edu (Chanseok Jeong)

which many crucial structural components are constructed of composites. Fiber metal composites are also used in various automotive components.

Because of the extensive usage of such composite materials, it is critical to ensure the integrity of composite structures during their entire life cycle. Delaminations, cracks, inclusions of foreign substance, and other defects could be introduced into composite structures during production or use of them. Among the defect types, delamination (i.e., the debonding of adjacent plies in laminated composites) is the most frequent structural anomaly for composite structures, particularly sandwiched ones. Because internal delamination significantly reduces the strength and fatigue resistance of composite components, it is of importance to accurately detect and characterize internal delamination in fiber metal laminate components in various industrial applications [4].

Various nondestructive testing (NDT) methods [5–7] are aimed to detect a variety of defects (e.g., cracks, delaminations, crushed and buckled cores, and core-skin disbonds). Ultrasound techniques (either active or passive) are widely adopted in industry due to their versatility. In particular, active guided ultrasound (e.g., [8, 9]) allows for long-range monitoring, as guided ultrasonic waves propagate along the structure (also referred to as waveguide) and tend not to lose the wave energy over the traveling distance. Although, in a structure made of composite materials, guided wave propagation is complex due to material anisotropy and the waves' dispersive and multi-modal nature, damage detection using guided waves is well-researched and has been applied to a variety of sandwich and other composite plate structures [10–20]. In addition to mere detection (e.g., examination of the existence of defects without discovering their locations or shapes) (level 1 [21]), localization (level 2) and shape-identification (level 3) are possible and implemented through various damage imaging techniques. However, most case studies in the past focused on very specific, often simplified, applications. To enable the accurate characterization of structural flaws, optimization-based methods have been developed, in particular, using the extended finite element method (XFEM) or the level-set method to avoid re-meshing while modeling defects of varying shapes during optimization processes [22–27]. Such optimization methods typically consume a large duration of computing time (e.g., days), limiting real-time detection of structural damages from measured dynamic responses.

As opposed to the optimization-based methods, studies have recently shown that machine learning (ML) drastically reduces the computing time for detecting of defects into seconds or less. For instance, Jiang et al. [28] used a combination of the level-set approach and the extreme learning machine (ELM) to find cavities in 2D solid structures. Once their neural network is informed of the number of cavities and their shape types (e.g., circular or elliptical), it effectively determines the coordinates of circular (or elliptical) cavities' centroids, radii, major/minor axes, and orientations of cavities. Humer et al. [29] used deep neural networks (DNNs) to predict wave damage interaction coefficients (WDICs) in order to identify damage in lightweight structures. Namely, they propose a hybrid neural network method that combines finite element (FE) simulations and DNNs to predict WDICs for different damage scenarios. Latête et al. [30] explored the use of CNNs to identify and locate flat bottom holes and side-drilled holes in an immersed test specimen using a single plane wave insonification. Mei et al. [31] developed a new approach to laser ultrasonic imaging of complex defects with different shapes, including delaminations and anomalies by using full-matrix capture and deep-learning extraction. The aforementioned papers demonstrate the superiority of ML approaches over optimization-based methods for defect detection in different types of materials.

Recently, Pranto et al. [32] presented an element-wise classification method to image cavities of random shapes and numbers in a homogeneous solid using elastic waves. They utilized a CNN to relate measured wave data at multiple sensors around a solid to a contour map of elements individually classified as either void or non-void elements. They demonstrated that ML model, trained by using synthetic data generated by finite element method (FEM) simulations, accurately predicts the sizes, thicknesses, and locations of penny-shaped defects in a solid. Inspired by the robust performance shown by Pranto et al. [32], this paper investigates the performance of a CNN, based on element-wise classification, for an anisotropic structure. This work incorporated the level-set method into the SEM, which can utilize an explicit time integration method to accelerate the data generation process. Namely, because the level-set method does not necessitate the task of generating explicit mesh around a cavity boundary, we save time for mesh generation. Furthermore, the SEM can further accelerate the calculation time than the regular FEM by the virtue of the diagonal mass matrix and the associated ease of matrix inversion during the time integration. Conversely, it can be very time-consuming if one uses a regular FEM with explicitly-generated mesh around a cavity boundary during the data generation process.

The remaining sections of the paper are structured as follows. In Section 2, we first present the governing equation to simulate the propagation of elastic waves in 2D plane-strain orthotropic materials, which are later considered to

be sandwiched by isotropic solids. Section 3 presents a level-set SEM forward wave solver for efficiently generating training data for the sandwiched composite structures with delamination cavities. In Section 4, to verify our level-set SEM wave solver, its computed displacement fields are compared with those obtained using ANSYS. Section 5 presents a CNN architecture that is trained using synthetic data sets for the classification of each element in the domain as a void or non-void element based on measured wave data. As a result, it provides an element-wise classification contour map that displays the probability of each element to be a void element. The positions, dimensions, and shapes of the cavities can be inferred from the contour map. In Section 6, the CNN's performance is assessed using test data sets produced by the SEM wave-solver and blind test data sets produced by ANSYS.

2. Problem Definition

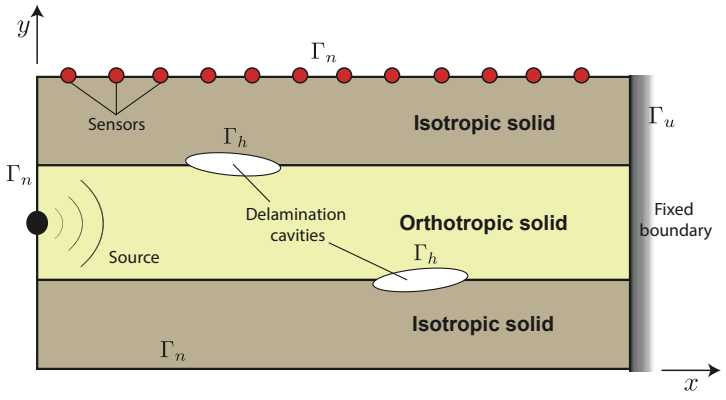


Figure 1: An illustration of an anisotropic layered medium (i.e., an orthotropic solid sandwiched by isotropic solid) with delamination cavities.

This paper considers a 2D layered domain (see Fig. 1), which is occupied by an orthotropic[1] material stacked between isotropic material with delamination cavities along the interfaces. Specifically, we study elastic waves in a 2D plane-strain anisotropic solid setting, where the governing equation reads:

$$\nabla \cdot \boldsymbol{\sigma} - \rho \ddot{\mathbf{u}} = \mathbf{0}, \quad \text{in } \Omega, \tag{1}$$

where $\mathbf{u} := \mathbf{u}(x, y, t) = [u_x, u_y]^T$ denotes the vector wave displacement field; $\boldsymbol{\sigma} := \boldsymbol{\sigma}(x, y, t)$ denotes the Cauchy stress tensor; $\nabla \cdot (\ )$ denotes the divergence operator; and $\rho := \rho(x, y)$ denotes the mass density. Initially, wave responses in time are zero (i.e., $\mathbf{u}$ and $\dot{\mathbf{u}}$ vanish at $t = 0$). The displacement field $\mathbf{u}$ is zero at Γ_u, and a cavity's surface Γ_h is free of traction while Γ_n is subjected to traction.

The orthotropic solid has three orthogonal planes of symmetry that are mutually perpendicular to each other. The nine elastic constants in orthotropic constitutive equations are comprised of three Young's moduli along each direction (E_1, E_2, E_3), the three Poisson's ratios $(\nu_{23}, \nu_{31}, \nu_{12})$, and the three shear moduli (G_{23}, G_{31}, G_{12}). The constitutive equation for a 2D plane-strain orthotropic material can be driven from that of a general 3D orthotropic solid as follows.

First, the three-dimensional strain-stress relationship for a general orthotropic material is the following:

$$\boldsymbol{\varepsilon} = \mathbf{S}\boldsymbol{\sigma} \rightarrow \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \varepsilon_{yz} \\ \varepsilon_{zx} \\ \varepsilon_{xy} \end{Bmatrix} = \begin{bmatrix} \frac{1}{E_1} & -\frac{\nu_{21}}{E_2} & -\frac{\nu_{31}}{E_3} & 0 & 0 & 0 \\ & \frac{1}{E_2} & -\frac{\nu_{32}}{E_3} & 0 & 0 & 0 \\ & & \frac{1}{E_3} & 0 & 0 & 0 \\ & & & \frac{1}{2G_{23}} & 0 & 0 \\ & \text{Sym.} & & & \frac{1}{2G_{13}} & 0 \\ & & & & 0 & \frac{1}{2G_{12}} \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \sigma_{yz} \\ \sigma_{zx} \\ \sigma_{xy} \end{Bmatrix}, \tag{2}$$

[1] We assume that the considered orthotropic solid is either a truly homogeneous material or a homogenized stack of layers.

where $\boldsymbol{\varepsilon}$ denotes the strain tensor; $\mathbf{S}$ is a compliance matrix; E_i denotes Young's modulus along axis i; ν_{ij} denotes the Poisson's ratio corresponds to a contraction in the j-th direction when an extension is applied in the i-th direction; and G_{ij} represents the shear modulus in the j-th direction on the plane of which normal vector is in the i-th direction. Assuming a plane strain condition where strains along the z-direction are zero, Eq. (2) can be reduced to

$$\boldsymbol{\varepsilon} = \mathbf{Q}\boldsymbol{\sigma} \rightarrow \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_{xy} \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{21} & Q_{22} & 0 \\ 0 & 0 & Q_{66} \end{bmatrix} \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{bmatrix} = \begin{Bmatrix} \frac{1-\nu_{13}\nu_{31}}{E_1} & -\frac{\nu_{12}(\nu_{21}-\nu_{23}\nu_{31})}{\nu_{21}E_1} & 0 \\ \text{Sym.} & \frac{1-\nu_{23}\nu_{32}}{E_2} & 0 \\ 0 & 0 & \frac{1}{2G_{12}} \end{Bmatrix} \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{bmatrix}, \tag{3}$$

where $\mathbf{Q}$ is a reduced compliance matrix, of which components can be calculated from $\mathbf{S}$ using the following equation for each component of the matrix [33]:

$$Q_{ij} = S_{ij} - \frac{S_{i3}S_{j3}}{S_{33}}, \qquad i, j = 1, 2, 6. \tag{4}$$

By calculating the inverse of the matrix $\mathbf{Q}$, we can obtain the plane-strain stiffness matrix, $\mathbf{C} = \mathbf{Q}^{-1}$.

The weak form of the governing equation can be derived by multiplying Eq. (1) by a test function $\mathbf{v} \in H_0^1(\Omega)$ and using the divergence theorem and integrating by parts:

$$\int_\Omega \rho\ddot{\mathbf{u}} \cdot \mathbf{v}^T \, \mathrm{d}\Omega + \int_\Omega \mathbf{v}^T\mathbf{D}^T\mathbf{C}\mathbf{D}\,\mathbf{u} \, \mathrm{d}\Omega = \int_{\Gamma_n} \mathbf{v}^T \cdot \mathbf{T} \, \mathrm{d}\Gamma, \tag{5}$$

where $\mathbf{D}$ is a differential operator matrix; and $\mathbf{T}$ is a traction on the loaded surface. Eq. (5) is turned into a time-dependent discrete form, of which matrices are built by using the SEM process with a convenient level-set modeling. The propagation of transient elastic waves for orthotropic composite structures can be calculated via the assembled system of the matrices and the force vector in the discrete form. The detail of such a numerical procedure is shown in the next Section 3.

3. Level-set SEM forward wave modeling

Our wave solver uses the level-set method [25, 26, 32, 34] to reduce computational time for generating training data by avoiding remeshing for different parameters of delamination cavities, which are updated in each iteration during data generation.

In the weak form of the governing equation per each element, the displacement field $\mathbf{u}$ in each finite element is estimated by using the finite element approximation and an enrichment function $V(x, y)$. Namely, $\mathbf{u}$ in an element is approximated as:

$$\mathbf{u}^h(x, y, t) = V(x, y) \sum_{i=1}^{N_n} \phi_i(x, y)\mathbf{u}_i(t), \tag{6}$$

with a nodal displacement $\mathbf{u}_i$, a local shape function ϕ_i at the i-th local node, N_n denoting the total nodes in an element, and:

$$V(x, y) = \begin{cases} 1, & \text{if } \ x, y \ \in \text{an element categorized as a non-void element,} \\ 0, & \text{if } \ x, y \ \in \text{an element categorized as a void element.} \end{cases} \tag{7}$$

In this study, the approximation in Eq. (6) is made with 9-node quadrilateral elements. An element is considered a void element if its centroid and 4 additional nodes (out of a total of 9 nodes) fall within the cavity boundary. In all other cases, elements are categorized as non-void elements. An illustrative example of the element classification is shown in Fig. 2. Thus, we can rewrite the weak form, Eq. (5), to the following time-dependent discrete form per element by utilizing Eq. (6):

$$\begin{bmatrix} \mathbf{K}^e_{xx} & \mathbf{K}^e_{xy} \\ \mathbf{K}^e_{yx} & \mathbf{K}^e_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{u}^e_x \\ \mathbf{u}^e_y \end{bmatrix} + \begin{bmatrix} \mathbf{M}^e_{xx} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}^e_{yy} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{u}}^e_x \\ \ddot{\mathbf{u}}^e_y \end{bmatrix} = \begin{bmatrix} \mathbf{F}^e_x \\ \mathbf{F}^e_y \end{bmatrix}, \tag{8}$$

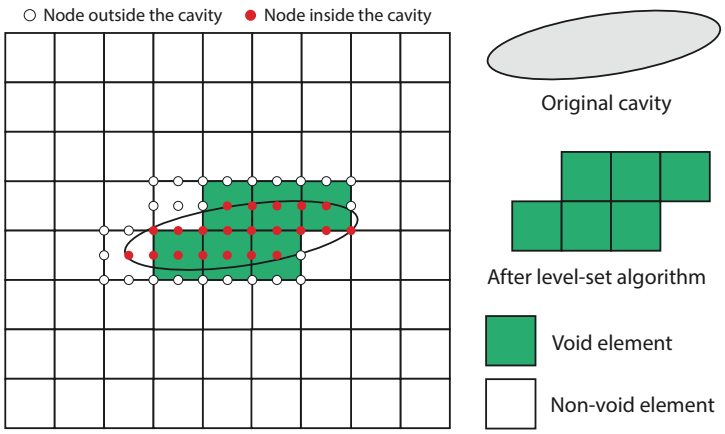


Figure 2: Level-set classification of void and non-void elements for a 9-node quadrilateral element. On the right, we can see a comparison between the original cavity and its level-set algorithm application.

where

$$\mathbf{K}^e_{xx} = \int_{\Omega^e} V(x,y)\left(C_{11}\frac{\partial \mathbf{\Phi}}{\partial x}\frac{\partial \mathbf{\Phi}^T}{\partial x} + C_{33}\frac{\partial \mathbf{\Phi}}{\partial y}\frac{\partial \mathbf{\Phi}^T}{\partial y}\right)\mathrm{d}\Omega, \tag{9a}$$

$$\mathbf{K}^e_{xy} = \int_{\Omega^e} V(x,y)\left(C_{12}\frac{\partial \mathbf{\Phi}}{\partial x}\frac{\partial \mathbf{\Phi}^T}{\partial y} + C_{33}\frac{\partial \mathbf{\Phi}}{\partial y}\frac{\partial \mathbf{\Phi}^T}{\partial x}\right)\mathrm{d}\Omega, \tag{9b}$$

$$\mathbf{K}^e_{yx} = \int_{\Omega^e} V(x,y)\left(C_{12}\frac{\partial \mathbf{\Phi}}{\partial x}\frac{\partial \mathbf{\Phi}^T}{\partial y} + C_{33}\frac{\partial \mathbf{\Phi}}{\partial y}\frac{\partial \mathbf{\Phi}^T}{\partial x}\right)\mathrm{d}\Omega, \tag{9c}$$

$$\mathbf{K}^e_{yy} = \int_{\Omega^e} V(x,y)\left(C_{22}\frac{\partial \mathbf{\Phi}}{\partial y}\frac{\partial \mathbf{\Phi}^T}{\partial y} + C_{33}\frac{\partial \mathbf{\Phi}}{\partial x}\frac{\partial \mathbf{\Phi}^T}{\partial x}\right)\mathrm{d}\Omega, \tag{9d}$$

$$\mathbf{M}^e_{xx} = \mathbf{M}^e_{yy} = \int_{\Omega^e} V(x,y)\rho\mathbf{\Phi}\mathbf{\Phi}^T\mathrm{d}\Omega, \tag{9e}$$

$$\mathbf{F}^e_x = \int_{\Gamma^e_n} \mathbf{\Phi}\mathbf{T}_x\mathrm{d}\Gamma, \tag{9f}$$

$$\mathbf{F}^e_y = \int_{\Gamma^e_n} \mathbf{\Phi}\mathbf{T}_y\mathrm{d}\Gamma. \tag{9g}$$

In Eqs. (9a) to (9d), C_{ij} is a component of the aforementioned plane-strain stiffness tensor computed per Eq. (3); $\mathbf{\Phi}$ is a vector of shape functions (please refer to Appendix A, which details the shape functions associated with our spectral element modeling); and $\mathbf{T}_i$ denotes the prescribed traction of the surface Γ_n.

It should be noted that the stiffness and mass matrices of a void element become zero because of the enrichment function $V(x,y)$ [34]. We derive the time-dependent discrete equation by globally assembling the element matrices and element force vector, resulting in a solution vector representing global displacements:

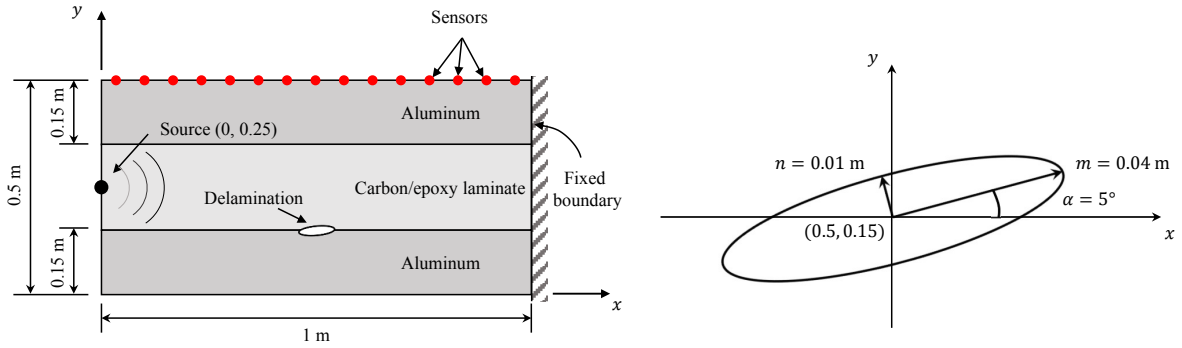
$$\mathbf{K}\mathbf{u} + \mathbf{M}\ddot{\mathbf{u}} = \mathbf{F}, \tag{10}$$

where $\mathbf{u} = [\mathbf{u}_x; \mathbf{u}_y]$ is the global solution vector. We resolve Eq. (10) by using the explicit time integration method, of which details are shown in Appendix B.

4. Verification of the forward wave solver

To verify our in-house level-set SEM wave solver, the numerical solution was compared with a reference solution obtained using ANSYS Mechanical, a finite element analysis software. A three-layered structure with the size of

1×0.5 m with one delamination cavity was considered as depicted in Fig. 3(a). The domain is subjected to fixed boundary conditions on the right surface. The delamination cavity is elliptical with major and minor axes of $m = 0.04$ m and $n = 0.01$ m, respectively. Its centroid is located at $(x = 0.05,\ y = 0.015)$ m, and its major axis is inclined at an angle of $\alpha = 5^\circ$ with respect to the horizontal axis, as shown in Fig. 3(b). In our level-set SEM forward wave solver, we discretize the domain with a structured mesh of 9-node quadrilateral elements of size 0.01 m. Fig. 4 shows close-up views of the delamination in the domain for both our SEM-based level-set forward solver and an FEM software ANSYS. To obtain the reference solution, we create an unstructured mesh of 8-node quadrilateral elements with an average element size of 0.005 m using the automatic mesh generation feature of ANSYS.



(a) The domain for verification.

(b) The dimension of the delamination cavity.

Figure 3: A stamp with a delamination cavity for the verification of our in-house level-set SEM wave solver.

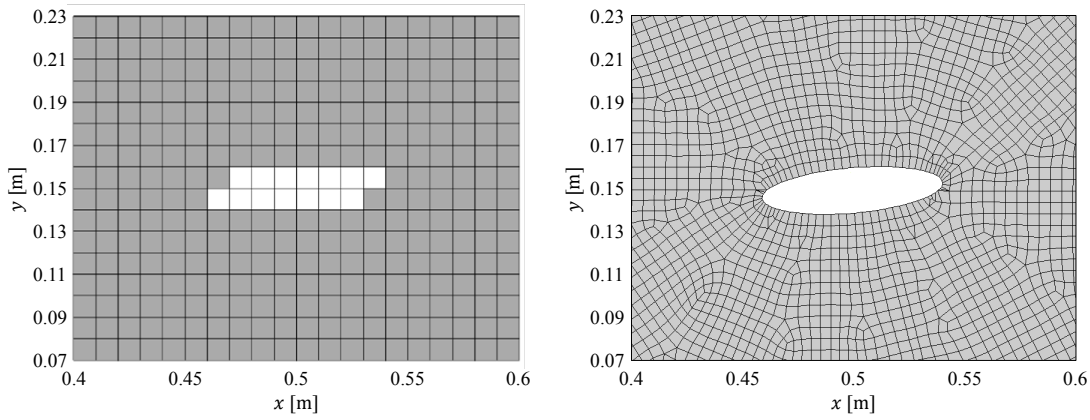


Figure 4: Close-up views of (a) the structured mesh for our level-set and SEM-based forward wave solver (white elements represent void elements) with an element size of 0.01 m and (b) the unstructured mesh for ANSYS for an element size of 0.005 m.

The top and bottom layers are assumed to be aluminum, while the middle layer is carbon/epoxy laminate, and their material properties are presented in Table 1. It indicates that aluminum is isotropic, and carbon/epoxy laminate is orthotropic.

A Ricker-pulse wave source with a peak amplitude of 5 kN/m and a central frequency of $f = 50$ kHz, as shown in Fig. 5, is applied at $x = 0$ m and $y = 0.25$ m as a point (i.e., z-directional line loading in a 2D) wave source based on Eq. (11) such that the positive value of the signal, $F(t)$, is directed in the normal direction toward the interior of the domain. The time step used in the simulation is $\Delta t = 0.5\ \mu$s, and the total analysis time is 350 μs. The displacement in the y direction is calculated for a total of 41 sensors located on the upper surface at intervals of 0.02 m between

$x = 0.1$ m and $x = 0.9$ m at $y = 0.5$ m.

$$F(t) = \begin{cases} \dfrac{5000\left[\left(0.25\left(2\pi f t - 3\sqrt{6}\right)^2\right) - 0.5\right] e^{-0.25\left(2\pi f t - 3\sqrt{6}\right)^2} - 13.e^{-13.5}}{0.5 + 13.e^{-13.5}}, & \text{if} \quad t \leq \frac{6\sqrt{6}}{2\pi f}, \\ 0, & \text{if} \quad t > \frac{6\sqrt{6}}{2\pi f}. \end{cases} \tag{11}$$

Table 1: Material properties of aluminum and carbon/epoxy laminate.

Elastic properties	Aluminum	Carbon/epoxy laminate [35]
ρ (kg/m^3)	2,700	1,550
E_{xx} (GPa)	69	150
E_{yy} (GPa)	69	10
E_{zz} (GPa)	69	10
ν_{xy}	0.33	0.33
ν_{yx}	0.33	0.022
ν_{yz}	0.33	0.44
ν_{zy}	0.33	0.44
ν_{xz}	0.33	0.33
ν_{zx}	0.33	0.022
G_{xy} (GPa)	25.9	9

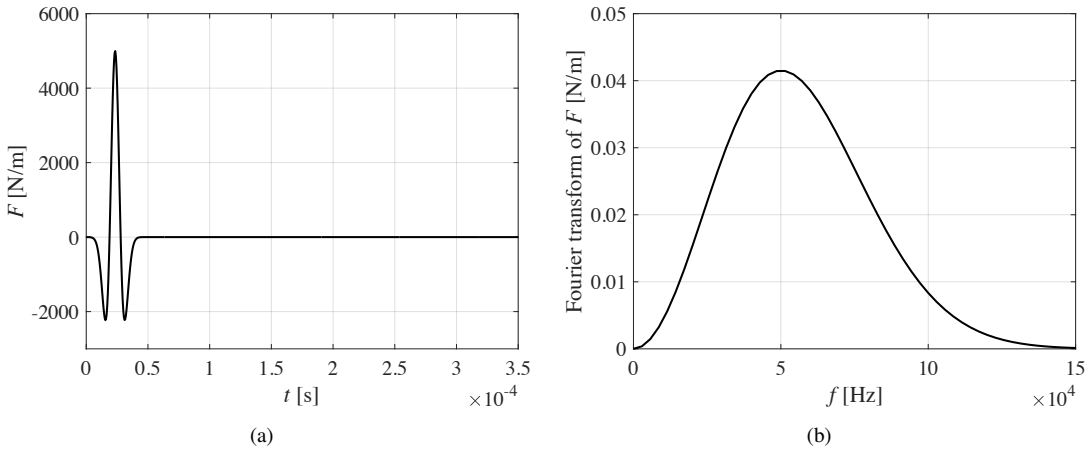
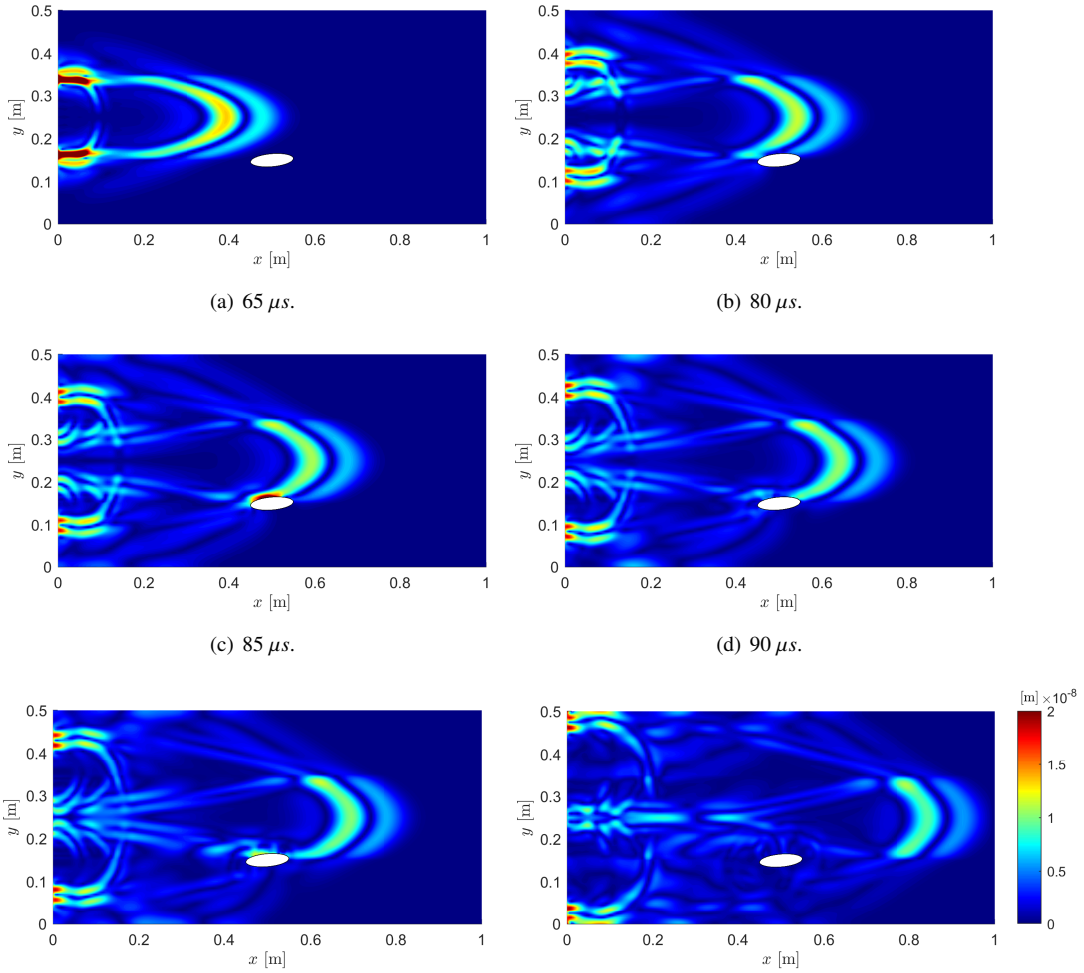


Figure 5: Time history and Fourier spectrum of the Ricker pulse.

In Fig. 6, we show the contour plot of amplitudes of displacement at different timesteps generated by our level-set SEM wave solver. Fig. 7 shows that the wave responses in the y-direction from our level-set SEM wave solver are in excellent agreement with those from ANSYS at the sensor locations on the top of the domain. This comparison verifies the capability of our level-set SEM solver for accurately simulating the propagation of elastic waves in a composite structure with a delamination cavity.

With an element length of 0.01 m in our level-set solver, the smallest distance between nodes is $r_{min} = 0.005$ m. We use the time step size Δt of 0.5 μs satisfying the Courant-Friedrichs-Lewy (CFL) condition, Eq. (B.10), for the maximum wave velocity of the considered materials presented in Appendix C. Since the CFL condition is satisfied, the explicit time integration in our SEM-based forward wave solver leads to accurate solutions without non-physical oscillations or numerical instabilities in the solution.

Note that, using an element size of 0.005 m, it takes 15 min for the ANSYS simulation to complete. For matching element sizes of 0.01 m, the ANSYS simulation completes in 5 min, while our level-set SEM wave solver finishes the computation in only 1 min, demonstrating superior computational efficiency of our wave solver.



(a) 65 μs. (b) 80 μs.

(c) 85 μs. (d) 90 μs.

(e) 95 μs. (f) 110 μs.

Figure 6: Contour plot displaying the elastic wave's displacement field's amplitudes in an anisotropic structure made of aluminum and carbon/epoxy laminate with a delamination cavity.

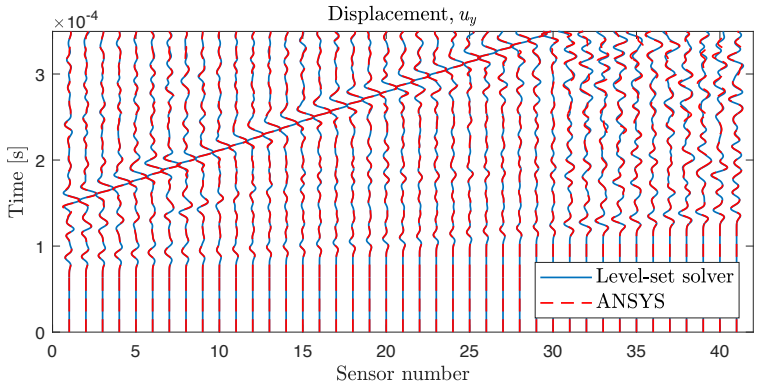


Figure 7: Comparison of displacement-time histories obtained using our level-set SEM wave solver and ANSYS.

5. CNN modeling

5.1. Data generation

To train our CNN, our data-generation simulation should (*i*) randomly populate void elements in a domain, (*ii*) run our SEM forward wave solver, and (*iii*) record the measured wave signals at sensors. To create void elements in a random but realistic manner in step (*i*), we capture (via the level-set method) void elements within a set of cavities that are generated with a random number, location, and configurations.

We use a three-layered domain of size, for instance, $16.44\lambda \times 8.22\lambda$ in meters in Examples 1 and 2, where λ denotes the wavelength of the shear wave of aluminum. Here, λ is 0.0608 m for the central frequency f of the Ricker of 50 kHz leading to a domain of 1 m × 0.5 m as in the verification example. The solid is made of the same carbon/Epoxy laminate sandwiched by aluminum (see Table 1). This plane-strain setting domain is discretized using an element size of 0.1644λ (0.01 m and 0.001 m for f of 50 kHz (λ = 0.0608 m) and 500 kHz (λ = 0.00608 m) in Examples 1 and 2, respectively). Thus, a structured background mesh is generated, discretizing the entire domain with 5,000 square elements. We use a Ricker-pulse wave source, $F(t)$ in Eq. (11), with its peak amplitude of 5 kN/m.

We populate the elliptical shaped cavities using our in-house randomizer to make an arbitrary number of cavities with different sizes and angles at various locations within the interior domain. Each elliptical cavity's actual boundary is defined by the following equation:

$$\frac{[(x - x_0)\cos(\alpha) + (y - y_0)\sin(\alpha)]^2}{m^2} + \frac{[(x - x_0)\sin(\alpha) - (y - y_0)\cos(\alpha)]^2}{n^2} = 1, \tag{12}$$

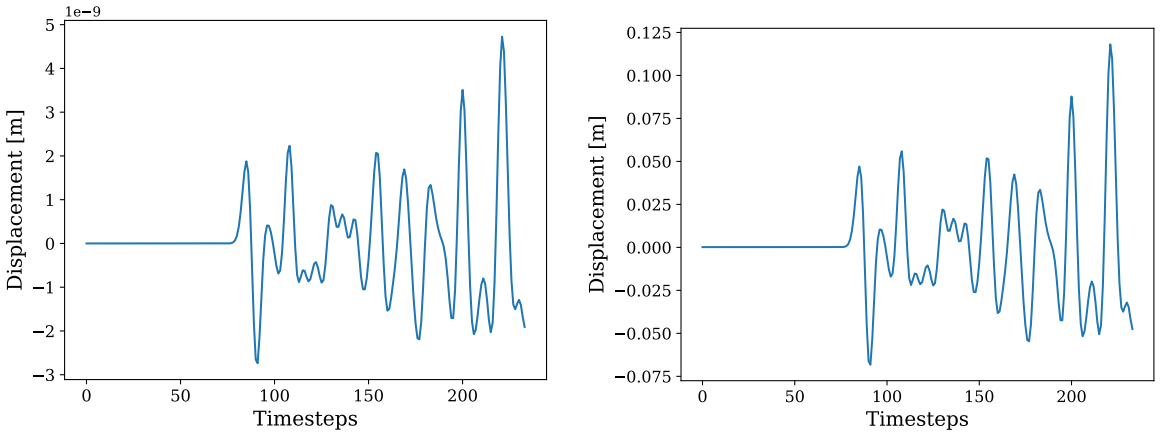
where (x_0, y_0) denotes the coordinates of the center; m and n are its major and minor axes, respectively; and α is the angle of the major axis with respect to the x-axis (see Fig. 3(b)). In each data set, our randomizer makes a random set of ellipses in a way such that the parameters are unrelated to each other. Despite the possibility of overlap, we avoid double-counting void elements. Because our CNN model is designed to find targeted void elements, overlapping of elliptical cavities does not affect how well our predictions work.

To generate the data set, we create elliptical cavities with a major axis between 0.49λ and 1.32λ (0.03 m and 0.08 m for f of 50 kHz), a minor axis between 0.16λ and 0.25λ meters (0.01 m and 0.015 m for f of 50 kHz), and an orientation between -5° and 5°. We randomly place our cavities along the two material interfaces to get training data that is free of bias.

Our data generation process randomizes the location and size of the target cavities in each data set, and we classify the elements based on our level-set approximation so that output feature data are labeled. We assign a value of one to void elements and zero to non-void elements and store the assigned values in the training data. At the same time, we run our in-house wave solver to record the displacement, u_y, of wave responses at the sensors. We generate a total of 6,000 data sets in three sets of 2,000, each accounting for 0, 1, and 2 cavities in total, respectively. The total observation time for each forward iteration is divided by 700 time steps, and each time step's size is $T/40$, where T is $\frac{1}{f}$. For f of 50 kHz, the time step size is 0.5 μs. We have 41 sensors located at the top surface of the domain. It takes 65 seconds for our level-set SEM wave solver to generate one data set on a single CPU and MATLAB. Thus, a total of 109 hours were spent to complete the data generation process on a single CPU.

5.2. Data Preparation

We preprocess the generated 6,000 data to render them suitable for training, validation, and testing. Each data comprises of vertical-displacement values of measured wave signals at 41 sensors spanning a duration of 35μs for f of 50 kHz. Consequently, the input data consists of 41 sensors multiplied by 701 timesteps. To train a neural network, data were computed on a system comprised of an 8-core CPU with 64 gigabytes of DDR4 RAM and a 12 gigabytes memory NVIDIA Titan V GPU. To reduce the computational cost of training our CNN, we extract every third-time step, resulting in only 234 timesteps at each sensor. This reduction in timesteps significantly mitigates the computational cost while having no impact on performance during training, validation, and testing. We allocated 80% (4800) of the 6,000 data for training, 10% (600) for validation, and the remaining 10% (600) for testing.



(a) Before normalization.

(b) After normalization.

Figure 8: Input displacement data for training CNN.

Figure 8(a) presents a sample of the input data at one sensor. It is evident from Figure 8(a) that the data encompasses magnitudes, spanning from -10^{-9} to 10^{-9}. To facilitate the efficient learning of feature-relationships between input- and output-layer data by our CNN, we normalize the data using Eq. (13), limiting the range of the data between -0.5 and 0.5. We use this normalized data as the input-layer data for our CNN.

$$A^n_{ijk} = \frac{A_{ijk} - A^{\text{mean}}_{\text{train}}}{A^{\text{max}}_{\text{train}} - A^{\text{min}}_{\text{train}}}, \tag{13}$$

where A^n_{ijk} is the normalized displacement data set; the equation involves subtracting the mean value $A^{\text{mean}}_{\text{train}}$ from the un-normalized input data matrix's value A_{ijk} and dividing the result by the range $(A^{\text{max}}_{\text{train}} - A^{\text{min}}_{\text{train}})$ of the training data set.

5.3. CNN Architecture

We utilize a CNN architecture to identify important feature-relationship between our input- and output-layer data. Our CNN architecture consists of a Convolution, Max Pooling, Batch Normalization, Flatten, and Dense layer. We feed our preprocessed displacement data consisting of 41 sensors each with 234-time steps as input which then passes the data along to the convolution layer. We use $n_{\underline{filters}}$ of filter size $n_{\underline{filtersize}}$ whose initial values are initialized using the "He uniform" initializer [36]. The convolutional filters work in a sliding-dot product-like operation to extract a convolution map from each displacement value recorded at different sensor positions. The convoluted output is passed through an activation function, in our case Leaky Rectified Linear Unit (LeakyReLU), to grasp the non-linear patterns between the provided input and target output data. Without non-linear activation functions, a neural network, regardless of its depth, would essentially operate as a linear model, limiting its ability to learn from the intricate patterns and relationships present in the data. The subsequent Max Pooling layer receives this convolution map and extracts maximum values from $n_{\underline{poolsize}}$ to make a further generalization based on prominent features. The feature map from the previous layer is batch normalized to make the training more stable. The batch-normalized feature map is then transformed into a one-dimensional input to feed the data into a densely connected hidden layer. We utilize 4500 neurons and utilize LeakyReLU as the activation function. The output from this penultimate layer, containing $n_{\underline{units}}$, is passed onto the output layer to predict 5000 probabilistic values using the sigmoid activation function. Each value represents the probability, between 0 and 1, of an element in a two-dimensional domain of being void. Our CNN architecture is presented in Figure 9. We employed the "binary cross-entropy" loss function to calculate the disparity between the predicted and target outputs which is shown in Eq. (14). The actual hyperparameter combinations are

shown in

$$\mathcal{L} = -\frac{1}{M}\sum_{i=1}^{M}\left(\frac{1}{N}\sum_{i=1}^{N} Y_i \log(P(Y_i)) + (1 - Y_i)\log(1 - P(Y_i))\right), \tag{14}$$

where Y_i represents the target label, where "1" indicates voids and "0" indicates non-void elements, for the i-th element; N represents the total number of elements; M represents the total number of training data; and $P(Y_i)$ is the predicted probability of the i-th element being void, ranging from 0 to 1.

For optimization, we used the "Nadam" optimizer to back-propagate using TensorFlow provided automated differentiation to adjust the learnable parameters based on the calculated loss value. To assess our training and validation performance, we employed "Precision" as our evaluation metric. Additionally, we present the test set's performance across all four evaluation metrics: "Accuracy", "Precision", "Recall", and "F1-score".

5.4. Hyperparameter Search and Optimization

To automate the laborious and time-consuming process of hyperparameter search, we conducted an extensive hyperparameter search using custom algorithms, where the values of $n_{\underline{filters}}$, $n_{\underline{filtersize}}$, $n_{\underline{poolsize}}$, and $n_{\underline{units}}$ are optimized. We tested various combinations of layers and learnable parameters and then fine-tuned the best-performing combination manually to obtain optimal results on the validation data. To prevent overfitting, we use TensorFlow's callback mechanisms to halt training when there is no observed improvement in model performance based on validation loss (binary cross-entropy). Our presented CNN model is well-optimized and produces outstanding results on both the blind test data generated using ANSYS and the test data as shown in the upcoming Section 6.

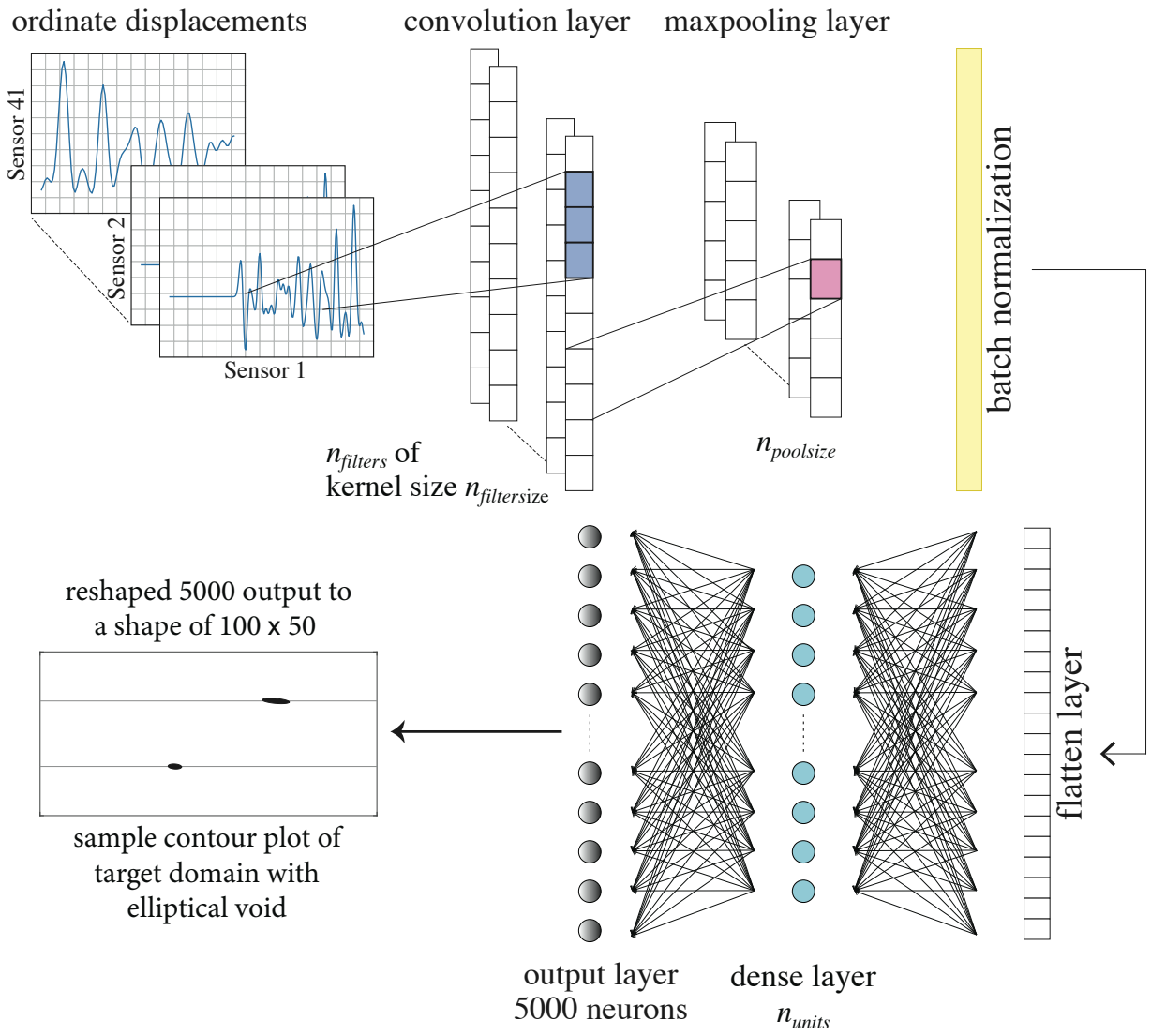


Figure 9: Our CNN architecture.

5.5. Evaluation metrics

In this section, we show how the different evaluation metrics change during training and how our trained CNN model performs on test and blind-test data sets, the latter generated using ANSYS. We can evaluate our CNN's performance on training, validation, and two sets of test data sets using the following evaluation metrics:

$$\text{Accuracy} = \frac{tp + tn}{tp + fn + tn + fp} \times 100\ [\%], \tag{15a}$$

$$\text{Precision} = \frac{tp}{tp + fp} \times 100\ [\%], \tag{15b}$$

$$\text{Recall} = \frac{tp}{tp + fn} \times 100\ [\%], \tag{15c}$$

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \times 100\ [\%], \tag{15d}$$

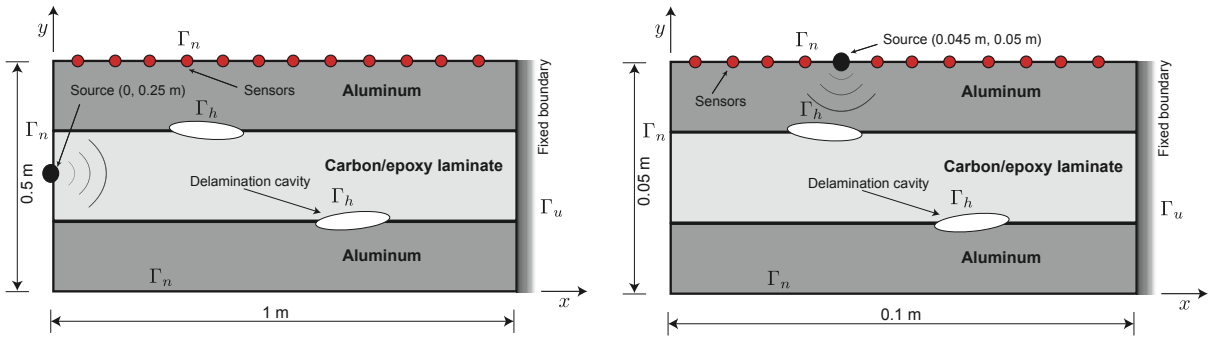
where tp, tn, fp, and fn represent, respectively, the counts of true-positive, true-negative, false-positive, and false-negative classifications of all the elements in a particular data set. Specifically, we count tp when our CNN correctly classifies a target void element as a void; tn when our CNN correctly classifies a target non-void element as a non-void element; fp when our CNN incorrectly classifies a target non-void element as a void element; and fn when our CNN incorrectly classifies a target void element as a non-void element.

We note that Accuracy was not an ideal choice as our domain primarily comprised non-void elements, which are significantly easier to predict and can falsely represent our CNN's performance. In contrast, Precision assesses how effectively our CNN identifies void elements (tp) compared to falsely classifying a non-void element as a void element. On the other hand, Recall measures our CNN's performance in identifying void elements (tp) compared to incorrectly identifying a void element as a non-void element (fn). Moreover, the inclusion of the F1-score complements this assessment, serving as a harmonic mean between Precision and Recall. Opting for a single metric wouldn't provide a holistic representation of our model's predictive capabilities. Hence, we present all four metrics for a comprehensive evaluation.

6. Numerical Results

6.1. Case A: Examining the scalability of the presented method with respect to the scale of the domain size, wave frequency, and source location.

We consider two examples shown in Fig. 10. The first one (shown in Fig. 10(a)) considers the wave source at the left boundary and the central frequency f of the Ricker pulse 50 kHz. The second one (shown in Fig. 10(b)) utilizes the source on the top surface with f of 500 kHz. To accommodate two different frequencies, ten folds apart each other, the size of the domain in the first example is set to be 1 m by 0.5 m while that of the second example is 0.1 m by 0.05 m. The orders of magnitudes of horizontal-direction dimensions of cavities in the blind tests are up to 0.04 m and 0.004 m in Examples 1 and 2, respectively. These examples demonstrate that our presented examples are scalable to various sizes of domains, sizes of delamination cavities, the frequency regime of waves, and the location of a wave source.



(a) Large domain (Example 1).

(b) Small domain (Example 2).

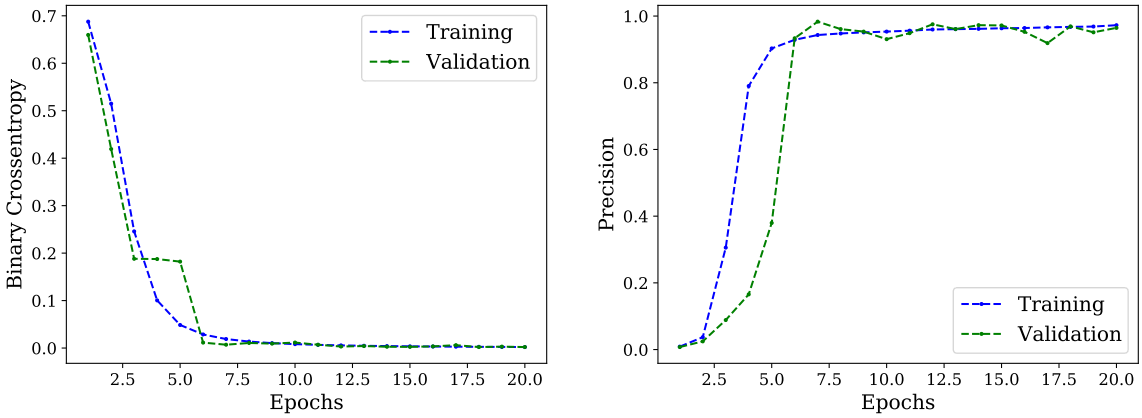
Figure 10: A schematic diagram of the data generation domain for Examples 1 and 2.

6.1.1. Performance during training for Examples 1 and 2

As this research problem involves a binary classification task, we show the convergence of binary cross-entropy loss function in Eq. (14) and Precision in Eq. (15b) over epochs in Fig. 11. Aside from the convergence of the loss function, we employed "Precision" as our exemplary independent evaluation metric[2], showing the progress of the training process in Fig. 11(b). In Example 1, Fig. 11 shows that the loss function converges to zero and the Precision metric reaches 100% within 20 epochs. If the curves were indicative of overfitting, the loss and Precision curves for

[2]Additionally, we present the performance of our CNN model across all four evaluation metrics: "Accuracy", "Precision", "Recall", and "F1-score" when we subsequently examine test data and perform a blind test.

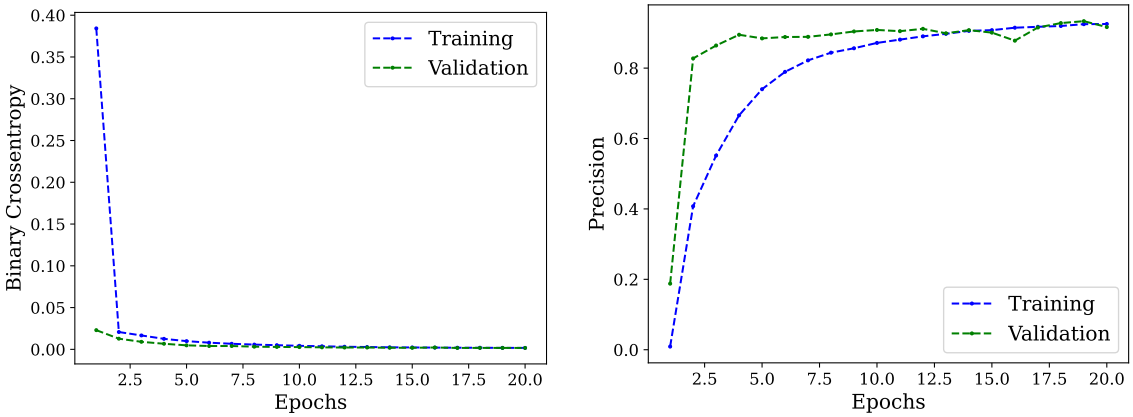
training and validation data sets would not overlap, suggesting a lack of generalization during the learning phase. In Fig. 12, we observe the similar behavior of loss function and Precision curves over the epochs in Example 2 where we use a smaller domain, a source at the top of a higher frequency, and smaller cavity sizes than Example 1.



(a) Binary cross-entropy loss function.

(b) Precision.

Figure 11: Convergence curve of the loss function and Precision over 20 epochs for Example 1.



(a) Binary cross-entropy loss function.

(b) Precision.

Figure 12: Convergence curve of the loss function and Precision over 20 epochs for Example 2.

6.1.2. Performance on test data set for Examples 1 and 2

Tables 2 and 3 present the overview of our trained CNN's performance on the 600 test data sets and show the performance of our trained CNN both in larger and smaller domains (Examples 1 and 2). To provide a detailed analysis, Tables 2 and 3 individually identify the best and worst predictions for the three types of test data sets (i.e., 1 data set with zero target cavities, data sets with one target cavity, data sets with two target cavities).

- **Data set with zero target cavities**: Notably, we achieve perfect Accuracy for the domain that contains no void elements for both Examples 1 and 2. However, we are unable to compute the remaining three metrics due to the lack of tn, fp, and fn terms, which renders the expressions in Eqs. (15b), (15c), and (15d) indeterminate.

- **Data sets with one target cavity**: For the domain with one delamination cavity in Example 1, our CNN achieves an impressive Accuracy of 99.99%, a Precision of 100.00%, a Recall of 96.15%, and an F1-score of 98.04% for

the best prediction. Even the worst prediction for Example 1 resulted in an Accuracy of 99.95%, a Precision of 93.75%, a Recall of 94.94%, and an F1-score of 94.34%. Our CNN also shows an outstanding performance with an Accuracy of 99.95%, a Precision of 96.82%, a Recall of 91.03%, and an F1-score of 93.84% as shown in the best prediction result of Example 2. The worst prediction result of Example 2 also shows high values of the metrics: an Accuracy of 99.90%, a Precision of 84.51%, a Recall of 93.75%, and an F1-score of 88.89%.

- **Data sets with two target cavities**: Moreover, for the domain with two delamination cavities, our CNN achieves an Accuracy of 99.98%, a Precision of 97.83%, a Recall of 100.00%, and an F1-score of 98.90% for the best prediction in Example 1. On the other hand, the worst prediction for Example 1 results in an Accuracy of 99.84%, a Precision of 93.07%, a Recall of 91.26%, and an F1-score of 92.16%. Example 2 also shows a great performance of our CNN as evidenced in the best prediction in Example 2 with an Accuracy of 99.88%, a Precision of 94.18%, a Recall of 89.91%, and an F1-score of 92% and the worst prediction in Example 2 with an Accuracy of 99.79%, a Precision of 82.57%, a Recall of 88.24%, and an F1-score of 85.31%.

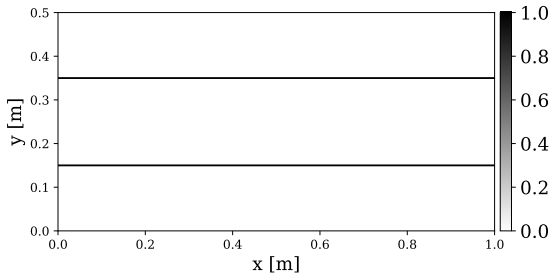
Table 2: The test data set's (Example 1) evaluation metrics for CNN under Case A.

Figures	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Fig. 13 - Prediction (zero cavities)	100			
Fig. 14 - Best prediction (one cavity)	99.99	100.00	96.15	98.04
Fig. 15 -Worst prediction (one cavity)	99.95	93.75	94.94	94.34
Fig. 16 - Best prediction (two cavities)	99.98	97.83	100.00	98.90
Fig. 17 - Worst prediction (two cavities)	99.84	93.07	91.26	92.16

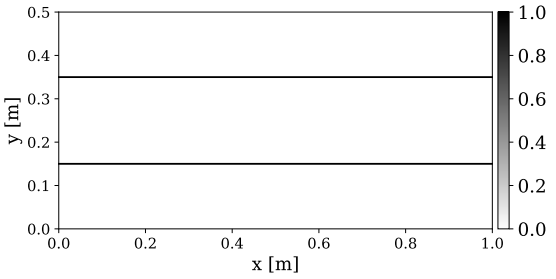
Table 3: The test data set's (Example 2) evaluation metrics for CNN under Case A.

Figures	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Fig. 13 - Prediction (zero cavities)	100			
Fig. 14 - Best prediction (one cavity)	99.95	96.82	91.03	93.84
Fig. 15 -Worst prediction (one cavity)	99.90	84.51	93.75	88.89
Fig. 16 - Best prediction (two cavities)	99.88	94.18	89.91	92.00
Fig. 17 - Worst prediction (two cavities)	99.79	82.57	88.24	85.31

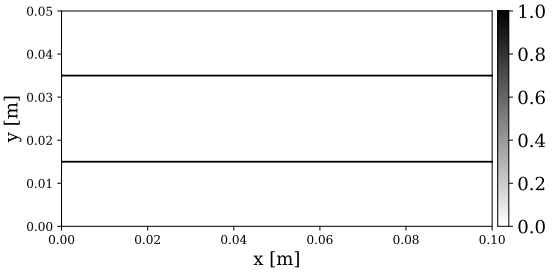
To provide a more intuitive understanding of the predictions, we present the visual representation of predicted cavities in each type of test data set consisting of 0, 1, and 2 targeted cavities in Figures 13 to 17, respectively.



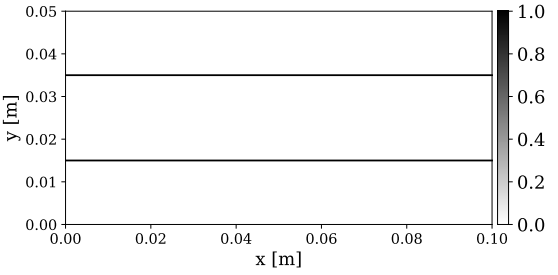
(a) Target domain (Example 1).



(b) Predicted domain (Example 1).

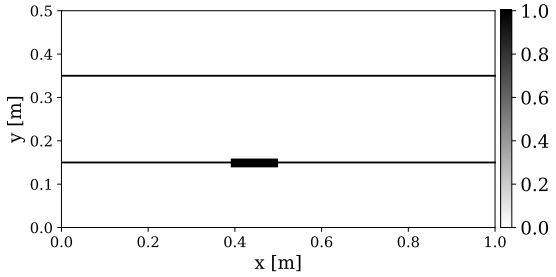


(c) Target domain (Example 2).

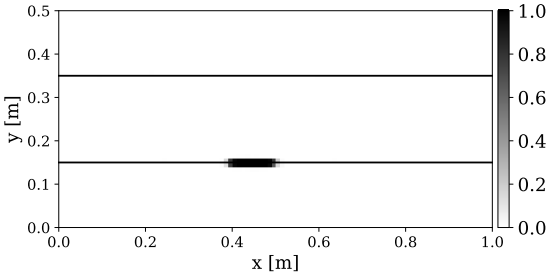


(d) Predicted domain (Example 2).

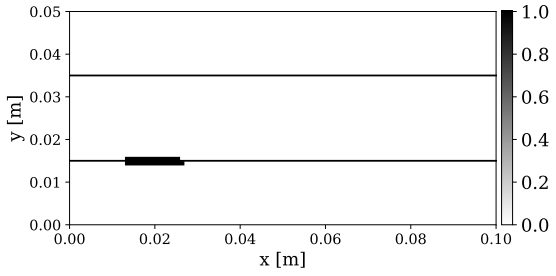
Figure 13: Prediction for zero target cavities in test data set under Case A.



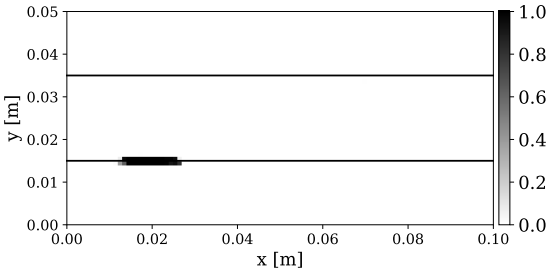
(a) Target domain (Example 1).



(b) Predicted domain (Example 1).

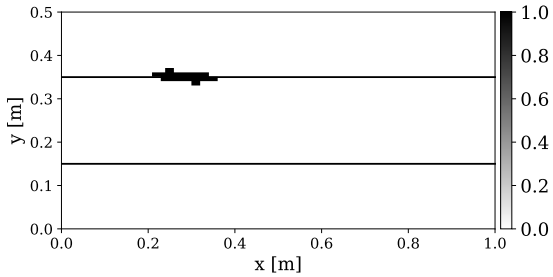


(c) Target domain (Example 2).

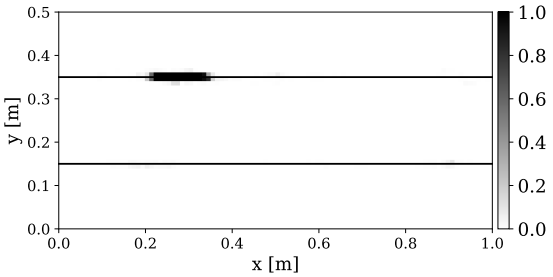


(d) Predicted domain (Example 2).

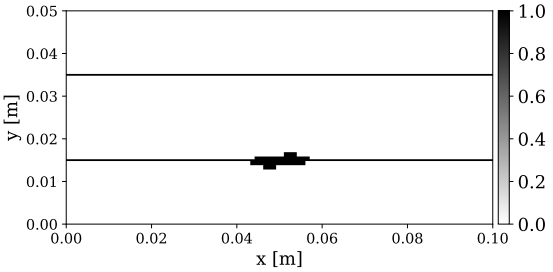
Figure 14: Best prediction for one target cavity in test data set under Case A.



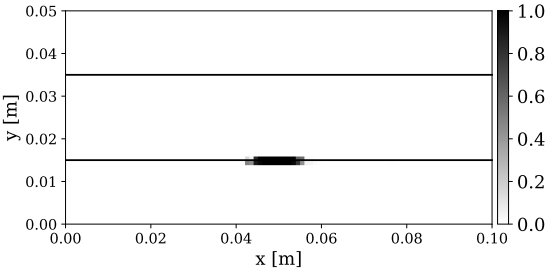
(a) Target domain (Example 1).



(b) Predicted domain (Example 1).

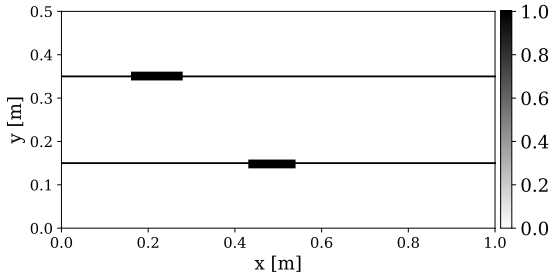


(c) Target domain (Example 2).

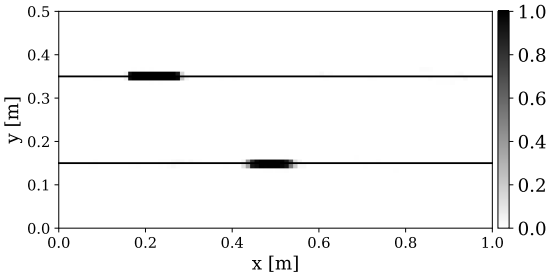


(d) Predicted domain (Example 2).

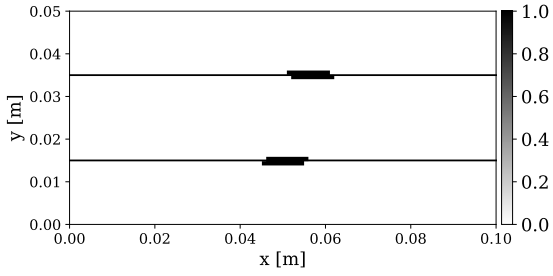
Figure 15: Worst prediction for one target cavity in test data set under Case A.



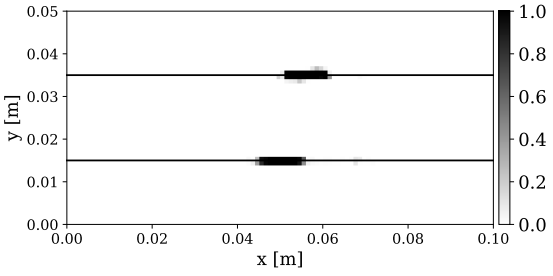
(a) Target domain (Example 1).



(b) Predicted domain (Example 1).

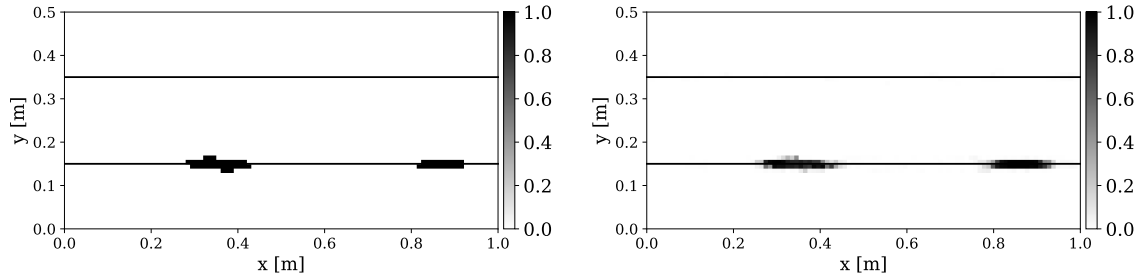


(c) Target domain (Example 2).



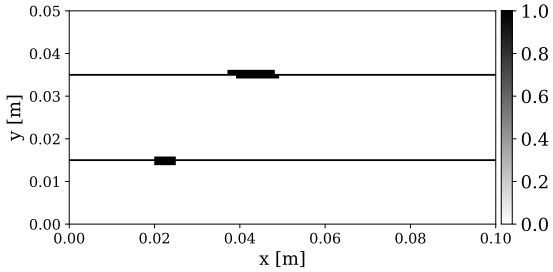
(d) Predicted domain (Example 2).

Figure 16: Best prediction for two target cavities in test data set under Case A.

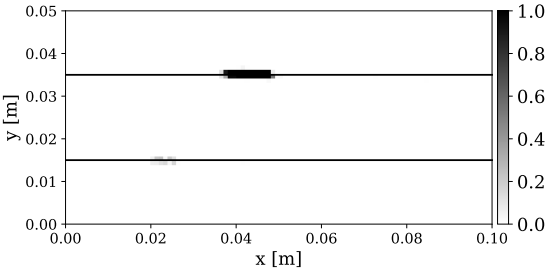


(a) Target domain (Example 1).

(b) Predicted domain (Example 1).



(c) Target domain (Example 2).



(d) Predicted domain (Example 2).

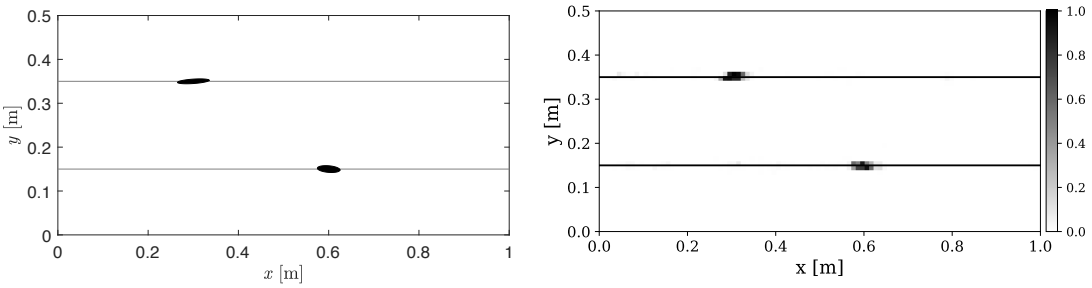
Figure 17: Worst prediction for two target cavities in test data set under Case A.

6.1.3. Performance on blind test data sets for Examples 1 and 2

The previous Section 6.1.2 employed test data sets that were generated by a level-set SEM wave solver to investigate the performance of our CNN. In this section, we present the performance of our CNN on blind test data that was generated using ANSYS Mechanical. By running the CNN using the measurement data from the third-party simulator that models the boundary of cavities explicitly (i.e., more accurately than our level-set wave solver), we avoid an inverse crime. Namely, we virtually mimic a situation where our numerical simulator uses experimental data, which are true to real physics. Our blind tests also demonstrate how our CNN solver predicts delamination cavities when it is fed by the measurement data from ANSYS in consideration of (*i*) elliptical cavities of even smaller major axes than those in the training data and (*ii*) cavities with non-elliptical shapes. Namely, we test the performance of our CNN inversion solver for cavities of non-elliptical shapes that it has never been exposed to during the training phase. To obtain the blind test data set, we created an unstructured mesh of 8-node quadrilateral elements, each having an average element size of 0.0025 m for Example 1 and 0.00025 m for Example 2, via ANSYS's automatic mesh generation feature. Such ANSYS modeling is considered to be, again, more veracious to the real physics than that of the level-set solver, particularly in modeling of a cavity with a zero-traction boundary.

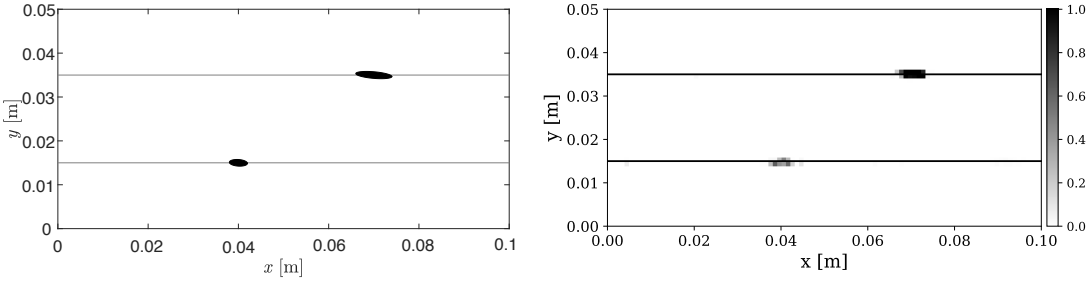
We utilize our CNN to identify delamination cavities in samples of blind test data sets from ANSYS. Namely, the measured wave signals from ANSYS modeling for the samples are fed into our trained CNN, which, in turn, yields the contour maps of the predicted void elements. They are, eventually, compared to the geometries of targeted cavities in ANSYS visually and numerically. Our CNN exhibited great performance on the samples (Figures 18 - 22). The Samples A1 to A3 correspond to the instances when the targeted delamination cavities, modeled by ANSYS, are elliptical, but with even smaller major axes (e.g., 0.01 to 0.02 m in Example 1 of a larger domain) than those (e.g., 0.03 to 0.08 m in Example 1) in the training data. In Samples A4 and A5, the targeted cavities are of non-elliptical shapes (e.g., Gaussian distribution or saw-tooth shape). Figs. 18 to 22 illustrate the target and prediction domains for all Samples A1 to A5. For brevity, the targeted and predicted cavities of Samples A1, A3, and A5 are presented only for Example 1 while those of Samples A2 and A4 are depicted for Example 2.

The numerical results of the evaluation metrics for all five blind test samples for both Examples 1 and 2 are presented in Tables 4 and 5. To this end, we map the geometries of targeted delamination cavities from ANSYS into a structured grid and identify equivalent targeted void elements in each sample. In the next Section 6.2, we compare the blind test results for various central frequencies of the source while its location is fixed.



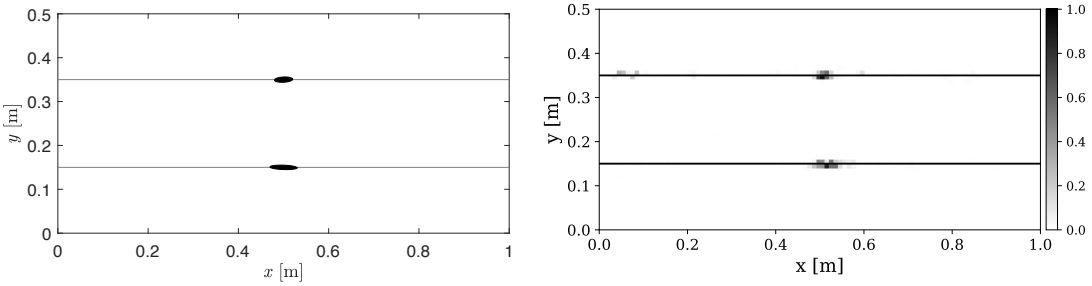
(a) Target domain. (b) Predicted domain.

Figure 18: Blind test (Sample A1) for Example 1 under Case A.



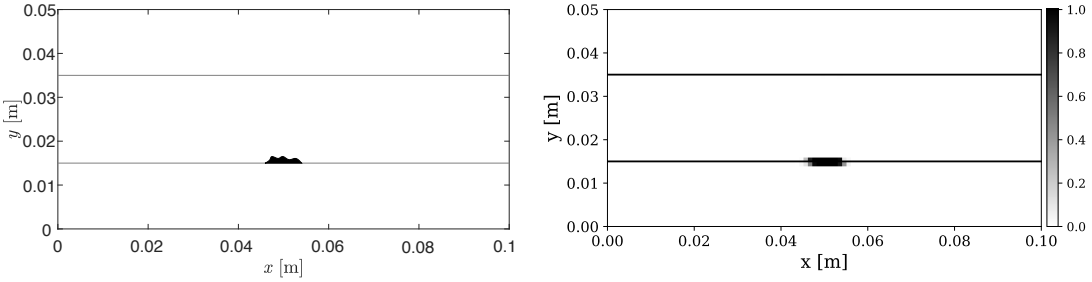
(a) Target domain. (b) Predicted domain.

Figure 19: Blind test (Sample A2) for Example 2 under Case A.



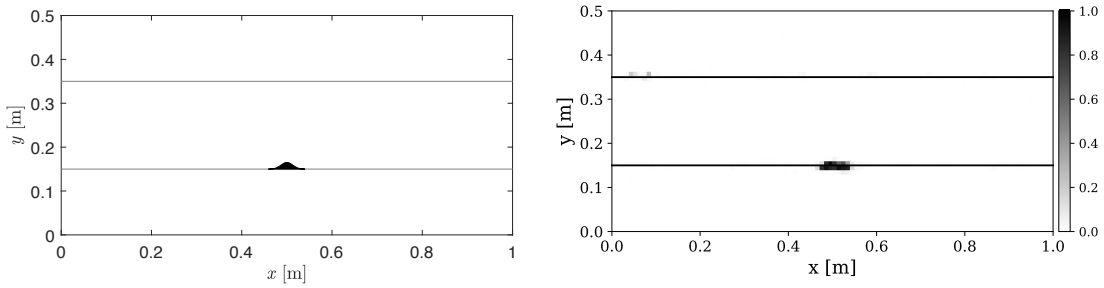
(a) Target domain. (b) Predicted domain.

Figure 20: Blind test (Sample A3) for Example 1 under Case A.



(a) Target domain. (b) Predicted domain.

Figure 21: Blind test (Sample A4) for Example 2 under Case A.



(a) Target domain.

(b) Predicted domain.

Figure 22: Blind test (Sample A5) for Example 1 under Case A.

Table 4: The evaluation metrics for CNN (Example 1) for the ANSYS generated blind test data set under Case A.

Samples	Targeted delamination cavities	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
A1 - Fig. 18	Two cavities in different interfaces	99.88	91.67	68.75	78.57
A2	Two cavities in different interfaces	99.88	88.46	71.88	79.31
A3 - Fig. 20	Two cavities in different interfaces	99.87	87.50	63.64	73.68
A4	One cavity (saw-tooth shape)	99.83	80.00	55.56	65.57
A5 - Fig. 22	One cavity (Gaussian distribution shape)	99.83	81.48	52.38	63.77

Table 5: The evaluation metrics for CNN (Example 2) for the ANSYS generated blind test data set under Case A.

Samples	Targeted delamination cavities	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
A1	Two cavities in different interfaces	99.84	91.67	61.11	73.33
A2 - Fig. 19	Two cavities in different interfaces	99.87	88.46	69.70	77.97
A3	Two cavities in different interfaces	99.85	84.38	61.36	71.05
A4 - Fig. 21	One cavity (saw-tooth shape)	99.81	86.00	51.81	64.66
A5	One cavity (Gaussian distribution shape)	99.81	87.04	48.45	62.25

6.2. Case B: Examining the performance of the method with respect to the frequency of the load for Example 1

In this section, we examine how well our CNN performs on blind test data in relation to the load frequency in Example 1. We explore three central frequencies (30 kHz, 40 kHz, and 50 kHz) of the Ricker-pulse wave source. We train three distinct CNN architectures with varied hyperparameter configurations, tailoring each to the training data for its respective frequency. A dedicated hyperparameter search is conducted for each wave source frequency problem to find optimal settings. Following a similar approach as outlined in Section 6.1.3, we generate blind test data sets for five samples using ANSYS. Our CNN demonstrates great performance across various central frequencies of the load for all samples.

The numerical results for the evaluation metrics of all five blind test Samples B1 to B5 are outlined in Tables 6 and 7 for central frequencies of 40 kHz and 30 kHz, respectively. Earlier results for the central frequency of 50 kHz are available in Table 4 in Section 6.1.3. Notably, we consistently observe positive outcomes across all three load frequencies, underscoring the broad applicability of our proposed CNN-based approach, irrespective of the source wave frequency. We notice that training with a 50 Hz frequency leads to slightly better performance, as evidenced by improvements in the F1-score. Figure 23 illustrates the target and prediction domains for only Sample B1 for brevity.

We notice the slightly less accurate performance of our CNN when (*i*) targeted cavities are not elliptical shaped than otherwise; (*ii*) targeted cavities are located near the top or bottom surface or the fixed boundary; and (*iii*) targeted cavities are vertically oriented. This could be improved by introducing a variety of cavities of different sizes and

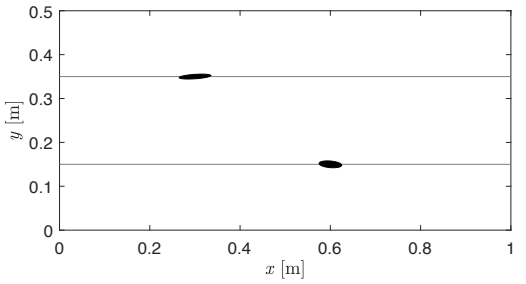
shapes in our training data sets. In the next Section 6.3, we examine that the presented method can be improved if the training data are further diversified in terms of the target cavity profiles.

Table 6: The evaluation metrics for CNN (Example 1, central frequency of 40 kHz) for the ANSYS generated blind test data set under Case B.

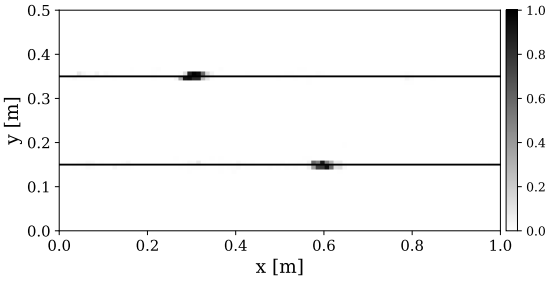
Samples	Targeted delamination cavities	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
B1 - Fig. 23(c)	Two cavities in different interfaces	99.79	100	41.51	58.67
B2	Two cavities in different interfaces	99.80	100	48.00	64.86
B3	Two cavities in different interfaces	99.79	100	44.21	61.31
B4	One cavity (saw-tooth shape)	99.80	100	37.50	54.55
B5	One cavity (Gaussian distribution shape)	99.79	100	32.26	48.78

Table 7: The evaluation metrics for CNN (Example 1, central frequency of 30 kHz) for the ANSYS generated blind test data set under Case B.

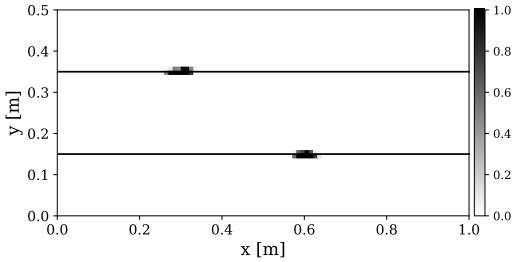
Samples	Targeted delamination cavities	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
B1 - Fig. 23(d)	Two cavities in different interfaces	99.81	77.27	41.46	53.97
B2	Two cavities in different interfaces	99.81	86.11	49.21	62.63
B3	Two cavities in different interfaces	99.81	88.10	46.84	61.16
B4	One cavity (saw-tooth shape)	99.82	100	40.00	57.14
B5	One cavity (Gaussian distribution shape)	99.81	100	34.48	51.28



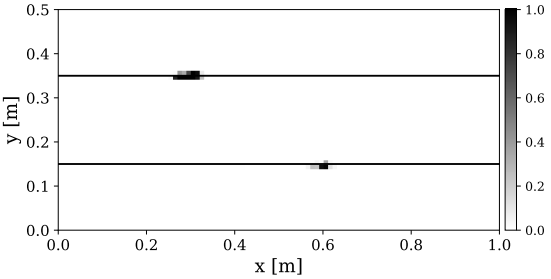
(a) Target domain (Sample B1).



(b) Predicted domain (frequency = 50 kHz).



(c) Predicted domain (frequency = 40 kHz).



(d) Predicted domain (frequency = 30 kHz).

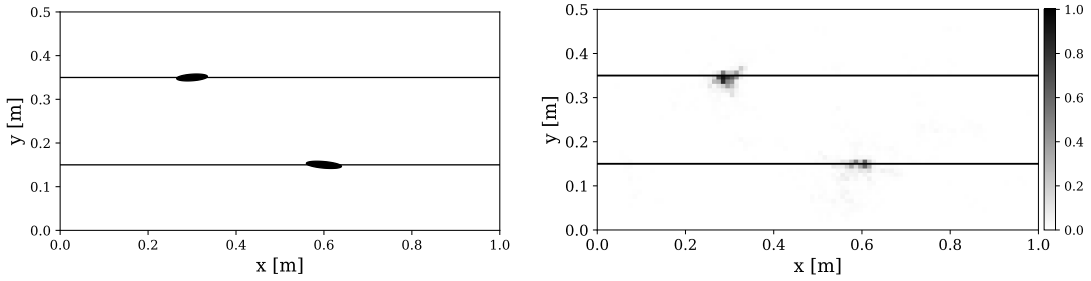
Figure 23: Blind test (Sample B1) for various central frequencies of the load under Case B.

6.3. Case C: Examining the effect of diversifying training data in terms of target cavity profiles for Examples 1 and 2

We train the CNNs using the data sets with more diversified target cavity profiles for Examples 1 and 2 than Case A. As distinction from Case A, the angle ranges from -88° to 88° in Case C, being more inclusive than the previous

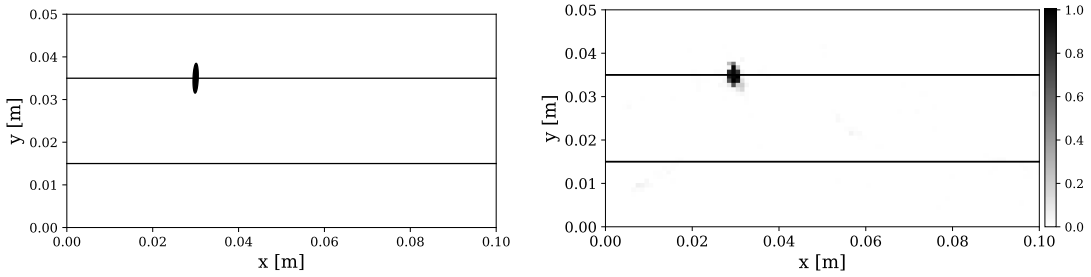
range of -5° to 5° in Case A. The cavities are located throughout the entire domain in Case C (Case A populates them only along the interfaces), with their centers at $1.32\lambda \le x \le 16.45\lambda$ and $1.32\lambda \le y \le 8.22\lambda$ (0.08 m $\le x \le$ 1 m and 0.08 m $\le y \le$ 0.42 m for Example 1; 0.008 m $\le x \le$ 0.1 m and 0.008 m $\le y \le$ 0.042 m for Example 2). The target profiles of training data include not only horizontally oriented cavities but also vertically oriented cavities, near-surface cracks, or those near the fixed boundary. We generate the blind test data sets, produced by ANSYS, simulating domains with two horizontally oriented cavities at different interfaces (Sample C1), a vertically oriented cavity at the interface (Sample C2), a cavity near the surface (Sample C3), and a cavity near the fixed boundary (Sample C4). Except for the aforementioned newly diversified profiles of target cavities in training data sets, all other conditions (e.g., boundary condition, source profile, sensor locations) remain the same as Case A.

To assess the effect of diversifying the cavity profiles of training data sets, we show the prediction results of blind tests. Figs. 24 and 26 depict the results of Samples C1 and C3 only for Example 1 while Figs. 25 and 27 present those of Samples C2 and C4 only for Example 2. The models have improved in detecting the locations of cavities because our CNN model is trained using data sets that consider both horizontally and vertically oriented cavities near the interface, surface, or fixed boundary. In Tables 8 and 9, the numerical results of the evaluation metrics for four blind test samples are presented for both Examples 1 and 2.



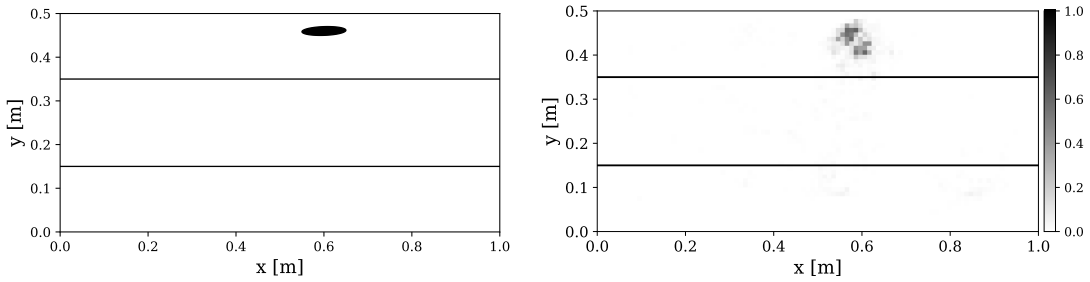
(a) Target domain. (b) Predicted domain.

Figure 24: Blind test (Sample C1) for Example 1 under Case C: Two horizontally oriented delamination cavities.



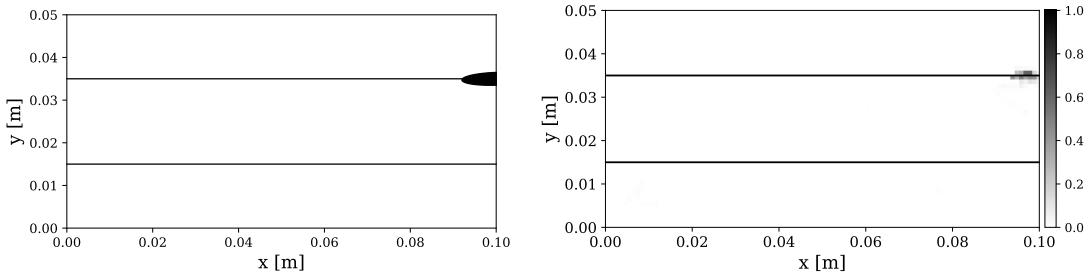
(a) Target domain. (b) Predicted domain.

Figure 25: Blind test (Sample C2) for Example 2 under Case C: One vertically oriented cavity.



(a) Target domain. (b) Predicted domain.

Figure 26: Blind test (Sample C3) for Example 1 under Case C: One cavity near the surface.



(a) Target domain. (b) Predicted domain.

Figure 27: Blind test (Sample C4) for Example 2 under Case C: One delamination cavity near the fixed boundary.

Table 8: The evaluation metrics for CNN for the ANSYS generated blind test data set for Example 1 under Case C.

Samples	Targeted cavities	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
C1 - Fig. 24	Two delamination cavities in different interfaces	99.72	45.00	75.00	56.25
C2	One vertically oriented cavity in the interface	99.81	57.14	69.57	62.75
C3 - Fig. 26	One cavity near the surface	99.72	49.18	54.55	51.72
C4	One delamination cavity near the fixed boundary	99.78	60.00	64.29	62.07

Table 9: The evaluation metrics for CNN for the ANSYS generated blind test data set for Example 2 under Case C.

Samples	Targeted cavities	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
C1	Two delamination cavities in different interfaces	99.58	15.00	42.86	22.22
C2 - Fig. 25	One vertically oriented cavity in the interface	99.71	32.14	47.37	38.30
C3	One cavity near the surface	99.74	42.62	60.47	50.00
C4 - Fig. 27	One delamination cavity near the fixed boundary	99.73	33.33	60.00	42.86

6.4. Case D: Examining the performance of the method for a domain (Example 3) of a different dimensional ratio from Examples 1 and 2

In this section, we train the new CNN model with a different dimensional ratio from Examples 1 and 2. We reduce the thickness of aluminum from 0.015 m to 0.005 m while the thickness of the Carbon/epoxy laminate remains as 0.02 m, as illustrated in Fig. 28. Similar to Example 2 in Case C, we create elliptical cavities with a major axis between

0.003 m and 0.008 m, a minor axis between 0.001 m and 0.0015 m, and an orientation between -88° and 88°. The center of cavities is located at $0.004 \text{ m} \le x \le 0.1$ m and $0.004 \text{ m} \le y \le 0.026$ m. We use the central frequency of the Ricker wave source as 500 kHz and apply it in the y-direction at (0.045 m, 0.03 m). To train the model, we collect the displacement data in the y-direction at a total of 41 sensors located on the upper surface at intervals of 0.002 m between $x = 0.01$ m and $x = 0.09$ m at $y = 0.03$ m. The blind test data sets for four samples are generated using ANSYS, as in Section 6.3, considering cavities near the interface, surface, or fixed boundary.

Figures 29 to 32 demonstrate the ability of our CNN model to detect cavities in the domain for Example 3. Moreover, we find that transitioning to a different dimensional ratio does not significantly impact the CNN's performance—it consistently identifies the presence of single or double cavities and accurately locates their positions within the domain. Additionally, the model continues to successfully reconstruct the cavity's shapes by predicting the associated elements. The numerical results for the prediction of blind test data for this dimensional ratio are summarized in Table 10.

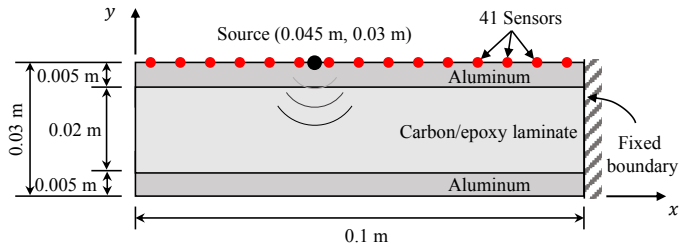
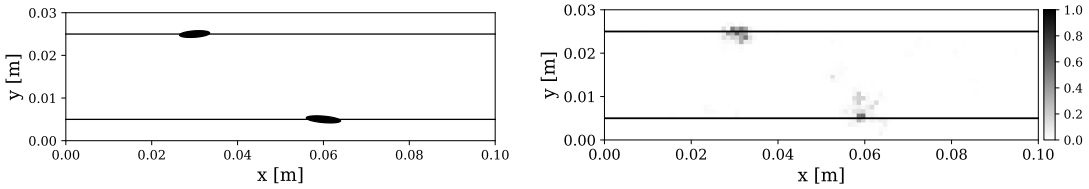
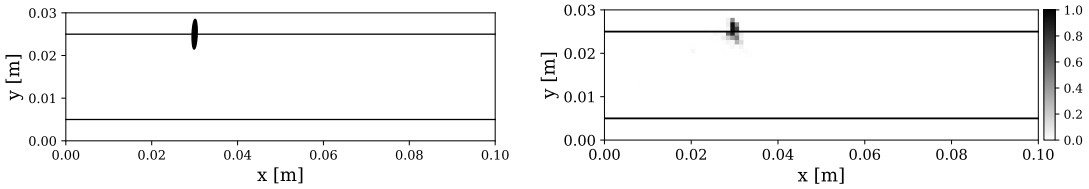


Figure 28: A schematic diagram of the data generation domain for Example 3.



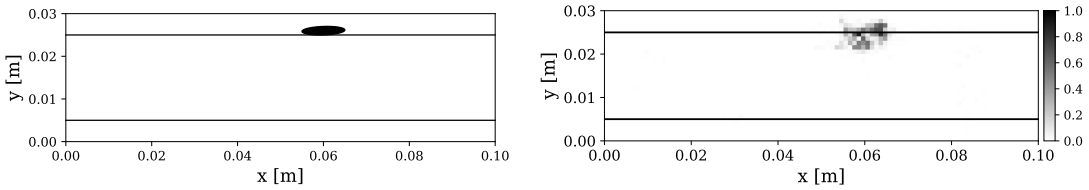
(a) Target domain. (b) Predicted domain.

Figure 29: Blind test (Sample D1) for Example 3 under Case D: Two horizontally oriented cavities.



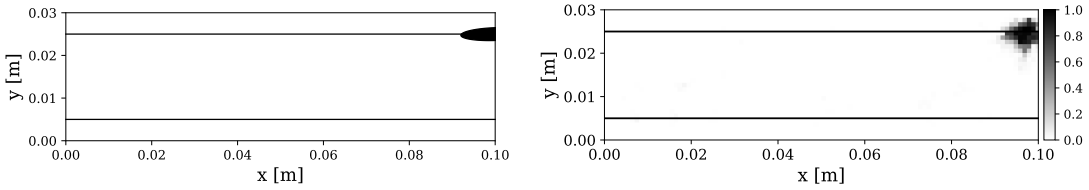
(a) Target domain. (b) Predicted domain.

Figure 30: Blind test (Sample D2) for Example 3 under Case D: One vertically oriented cavity.



(a) Target domain. (b) Predicted domain.

Figure 31: Blind test (Sample D3) for Example 3 under Case D: One cavity near the surface.



(a) Target domain. (b) Predicted domain.

Figure 32: Blind test (Sample D4) for Example 3 under Case D: One cavity near the fixed boundary.

Table 10: The evaluation metrics for CNN (Example 3) for the ANSYS generated blind test data set under Case D.

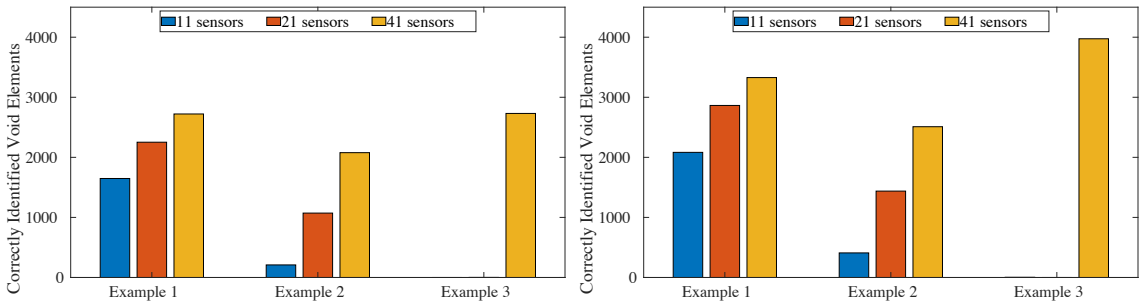
Samples	Targeted cavities	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
D1 - Fig. 29	Two delamination cavities in different interfaces	99.47	25.00	83.33	38.46
D2 - Fig. 30	One vertically oriented cavity in the interface	99.60	32.14	64.22	42.86
D3 - Fig. 31	One cavity near the surface	99.42	50.82	43.66	46.97
D4 - Fig. 32	One delamination cavity near the fixed boundary	99.48	55.56	48.08	51.55

6.5. Case E: Examining the performance of the method with respect to the number of sensors.

In this section, we discuss the outcomes of our parametric study, focusing on the correlation between the number of sensors and the predictive performance of the neural network. To evaluate the model's predictive accuracy, we analyze true positives (correct identification of void elements) and false positives (incorrect identification of non-void elements as void elements) in the test data set of Examples 1 and 2 under Case C and Example 3 under Case D in Figures 33 and 34, respectively.

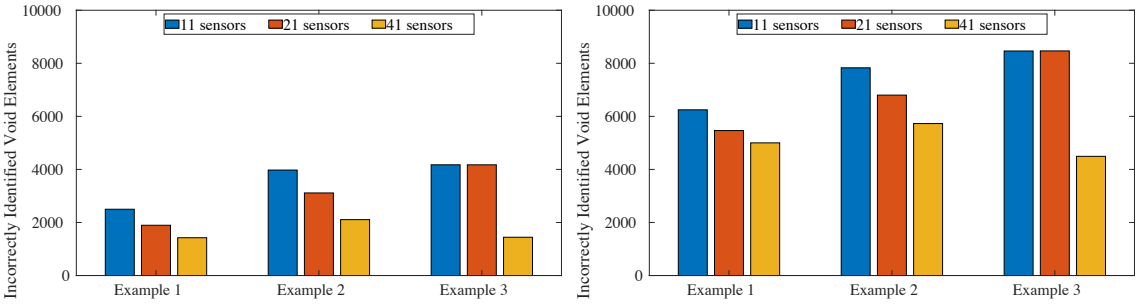
Figures 33(a) and 33(b) illustrate the correct identification of void elements (true positives) in domains containing 1 and 2 cavities, respectively. Notably, an increase in the number of sensors, exemplified by the yellow bars (41 sensors), consistently leads to improved void element identification compared to trained models with fewer sensors (11 sensors in blue and 21 sensors in orange). The use of fewer sensors can significantly hinder the neural network's capability to predict void elements.

Further analysis of Figures 34(a) and 34(b), which depict false positives (instances where non-void elements are incorrectly identified as void elements), shows that employing more sensors is associated with a decrease in misidentifying void elements. This observation suggests that a greater number of sensors facilitates the neural network in learning intricate feature relationships between input- and output-layer data, contributing to enhanced void element identification.



(a) Correctly identified void elements (one target cavity). (b) Correctly identified void elements (two target cavities).

Figure 33: Parametric study investigating the impact of sensor configurations on the predictive performance of our CNN in correctly identifying void elements (i.e., the number of true positives) when there are (a) one or (b) two target cavities.



(a) Incorrectly identified void elements (one target cavity). (b) Incorrectly identified void elements (two target cavities).

Figure 34: Parametric study investigating the impact of sensor configurations on the predictive performance of our CNN in incorrectly identifying void elements (i.e., the number of false positives) when there are (a) one or (b) two target cavities.

7. Conclusion

We discuss a novel method to detect delamination cavities in a 2D plane-strain anisotropic structure subjected to elastic waves. This paper presents the following methodological contributions.

- In order to accelerate the data generation process, we utilize the SEM method endowed with the explicit time integration and the level-set method. Namely, we rapidly solve a series of forward problems with varying sizes, shapes, and locations of delamination cavities located across an orthotropic material (e.g., carbon fiber-reinforced epoxy) and an isotropic layer (e.g., aluminum) without remeshing for varying parameters of the geometries of targeted cavities. In that, we systematically capture (via the level-set method) void elements within a set of cavities that are populated with a random number, location, and configurations. The binary information (i.e., void or non-void element) of all the elements and the measured wave signals at sensors are utilized to train our CNN.

- Once the CNN learns from these training data sets, our CNN categorizes each element in the domain as either a void or non-void one. Clusters of identified void elements reconstruct targeted cavities although they are arbitrary in their numbers and shapes. The advantage of such element-wise classification is that a neural network does not need to be informed of the prior information about the target cavities (e.g., their number and their shapes). Therefore, our CNN can identify targeted cavities, which are arbitrary in terms of shapes, numbers, locations, and size using only measurement data.

This work also utilizes blind test data produced by ANSYS Mechanical, which simulates the boundaries of the delamination cavities explicitly by using an unstructured mesh of fine resolution. Thus, we do not commit an inverse

crime in this work by mimicking a situation where our numerical simulator uses experimental data. The blind test also examines the performance of our CNN inversion solver for cavities of non-elliptical shapes (e.g., Gaussian distribution or saw-tooth shape) that it has never been exposed to during the training phase.

The following summarizes the findings from our numerical results.

- For domains with different scales in domain size, cavity sizes, and wave frequencies, our training consistently demonstrates the convergence of the loss function to a minimum, ensuring great performance for accurate cavity detection, also validated through visual inspection and numerical results. In blind tests, our CNN solver effectively detects delamination cavities, considering (*i*) elliptical cavities of even smaller major axes than those in the training data and (*ii*) cavities with non-elliptical shapes.

- From a numerical simulation, examining the performance of our CNNs with respect to only the central frequency of a wave source, we found that the accuracy of the CNN is not significantly affected by changing the central frequency (e.g., 30, 40, and 50 kHz).

- The performance of our CNN is improved once we consider target cavities of a broader range of shapes (e.g., a larger range of inclined angles) and locations (e.g., not just along the interfaces but also around the top and bottom surfaces and the fixed boundary) in our training data sets.

- Our CNN model successfully detects cavities near the interface, surface, and fixed boundary in a sandwiched composite structure of various dimensional ratios.

- The evaluation of our CNN's correct and incorrect identification of void elements indicates that an increase in the number of sensors consistently improves the accuracy of cavity identification.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Award CMMI-2053694. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors are also grateful for the support by the Faculty Research and Creative Endeavors (FRCE) Research Grant-48058 at Central Michigan University. This paper is contribution # TBD of the Central Michigan University Institute for Great Lakes Research. The authors greatly appreciate anonymous reviewers, who have helped improve the manuscript.

Data Availability

Some or all data, models, or code generated or used during the study are available from the corresponding author by request.

Appendix A. On the Spectral Element Modeling

To improve computational efficiency in terms of computing time and memory usage, the global mass matrix, **M**, can be diagonalized due to its efficient matrix-vector multiplication and matrix inversion compared to the non-diagonal mass matrix. While the lumped mass model is a straightforward way to diagonalize the mass matrix, we utilize the SEM [37], which utilizes high-order polynomials as basis functions with the Gauss-Lobatto-Legendre (GLL) numerical integration within the framework of a typical finite element modeling, to avoid the approximation error of the mass lumping but still diagonalize **M**. Thus, the SEM leads to a higher accuracy of the solution than the mass lumping.

We use three integration points for each direction in the quadratic 9-node element to perform the GLL numerical integration in this study. The basis function ψ for each direction in the local coordinate (ξ and η in a 2D) can be expressed using the second-order Legendre polynomial. For instance, when ξ is considered,

$$\psi_p(\xi) = \frac{1}{6L_2(\xi_p)} \frac{(1-\xi^2)L_2'(\xi)}{\xi - \xi_p}, \qquad p = 1, 2, 3, \quad -1 \le \xi \le 1, \tag{A.1}$$

where ξ_p is the p-th local node's coordinate (i.e., $\xi_1 = -1$, $\xi_2 = 0$, and $\xi_3 = 1$) and $L_2(\xi)$ denotes the second-order Legendre polynomial defined as:

$$L_2(\xi) = \frac{1}{2}\left(3\xi^2 - 1\right). \tag{A.2}$$

The shape function ϕ for the quadratic 9-node element in the local coordinate (ξ, η), $\xi, \eta \in [-1, 1]$, is built as follows:

$$\phi_i(\xi, \eta) = \psi_p(\xi)\psi_q(\eta), \qquad p, q = 1, 2, 3, \text{ and } i = 1, \ldots, 9. \tag{A.3}$$

The combination of p and q values for each i in Eq. (A.3) are determined per Fig. A.35.

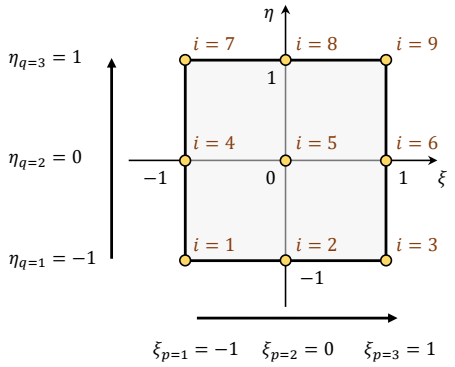


Figure A.35: The quadratic 9-node element in the local coordinate.

Table A.11 presents the location of nodes, integration points, and the corresponding weight factors (w_p and w_q) for the GLL quadrature used in this paper.

Table A.11: **Location of integration points and weight factors for the GLL quadrature.**

p, q	Location of integration points ξ_p and η_q	Weight factors w_p and w_q
1	-1	1/3
2	0	4/3
3	1	1/3

The element-level calculation of $\mathbf{M}^e$ in the local coordinate system is as follows:

$$M_{ij}^e = \int_{\Omega^e} \rho\phi_i(x,y)\phi_j(x,y)\mathrm{d}\Omega = \int_{-1}^{1}\int_{-1}^{1} \rho\phi_i(\xi,\eta)\phi_j(\xi,\eta)\,|\mathbf{J}|\,\mathrm{d}\xi\mathrm{d}\eta, \qquad i,j = 1,\ldots,9, \tag{A.4}$$

where $|\mathbf{J}|$ is the determinant of the Jacobian matrix. By using three integration points in each direction with the GLL quadrature method, Eq. (A.4) becomes

$$M_{ij}^e = \sum_{p=1}^{3}\sum_{q=1}^{3} w_p w_q \rho\phi_i(\xi_p,\eta_q)\phi_j(\xi_p,\eta_q)\,|\mathbf{J}|\,, \tag{A.5}$$

and $\mathbf{M}^e$ is characterized as:

$$M_{ij}^e = \begin{cases} M_{ij}^e = 0, & i \neq j \\ M_{ij}^e \neq 0, & i = j. \end{cases} \tag{A.6}$$

The global mass matrix, $\mathbf{M}$ in Eq. (B.1), can be obtained from the assembly of $\mathbf{M}^e$, and then $\mathbf{M}$ is diagonalized.

Appendix B. Explicit Time Integration Method

We introduce an explicit time integration method, leveraged by the diagonal mass matrix naturally arisen from the SEM, to solve nodal displacements and velocities. The system of the global matrices and the force vector in Eq. (10) can be transformed into a system of first-order differential equations, by assuming $\mathbf{y} = \dot{\mathbf{u}}$ and multiplying by the inverse of the mass matrix into Eq. (10), as:

$$\begin{bmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{y}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{u} \\ \mathbf{y} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1}\mathbf{F} \end{bmatrix}, \tag{B.1}$$

where $\mathbf{y} = [\dot{\mathbf{u}}_x; \dot{\mathbf{u}}_y]$. By substituting $\dot{\mathbf{s}} = [\dot{\mathbf{u}}; \dot{\mathbf{y}}]$, $\mathbf{s} = [\mathbf{u}; \mathbf{y}]$ into Eq. (B.1), the second-order differential equation system in Eq. (10) can be transformed into a system of first-order differential equations in every discrete time step n:

$$\dot{\mathbf{s}}_n = \mathbf{J}\mathbf{s}_n + \mathbf{F}_n^{\mathrm{eff}}, \tag{B.2}$$

where $\mathbf{J} = \left[\mathbf{0}, -\mathbf{I}; -\mathbf{M}^{-1}\mathbf{K}, \mathbf{0}\right]$, and $\mathbf{F}^{\mathrm{eff}} = \left[\mathbf{0}; \mathbf{M}^{-1}\mathbf{F}\right]$.

We adopt the explicit time integration scheme, the fourth-order Runge-Kutta method, to solve for the global solution vector of the first-order system in Eq. (B.2), for each discrete time step. The displacements and velocities for the next step ($\mathbf{s}_n$) can be calculated from a previous time step ($\mathbf{s}_{n-1}$) using the Runge-Kutta method with a time step Δt as seen below:

$$\mathbf{s}_n = \mathbf{s}_{n-1} + \frac{\Delta t}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4), \tag{B.3}$$

where the vector $\mathbf{k}_i$ are called i-th stage derivative of the 4th-order Runge-Kutta integration scheme as:

$$\mathbf{k}_1 = \frac{\partial \mathbf{s}}{\partial t}(t_{n-1}, \mathbf{s}_{n-1}) = \mathbf{J}\mathbf{s}_{n-1} + \mathbf{F}_{n-1}^{\mathrm{eff}}, \tag{B.4}$$

$$\mathbf{k}_2 = \frac{\partial \mathbf{s}}{\partial t}\left(t_{n-1} + \frac{\Delta t}{2}, \mathbf{s}_{n-1} + \frac{\Delta t}{2}\mathbf{k}_1\right) = \mathbf{J}\left(\mathbf{s}_{n-1} + \frac{\Delta t}{2}\mathbf{k}_1\right) + \mathbf{F}_{n-0.5}^{\mathrm{eff}}, \tag{B.5}$$

$$\mathbf{k}_3 = \frac{\partial \mathbf{s}}{\partial t}\left(t_{n-1} + \frac{\Delta t}{2}, \mathbf{s}_{n-1} + \frac{\Delta t}{2}\mathbf{k}_2\right) = \mathbf{J}\left(\mathbf{s}_{n-1} + \frac{\Delta t}{2}\mathbf{k}_2\right) + \mathbf{F}_{n-0.5}^{\mathrm{eff}}, \tag{B.6}$$

$$\mathbf{k}_4 = \frac{\partial \mathbf{s}}{\partial t}(t_n, \mathbf{s}_{n-1} + \Delta t\mathbf{k}_3) = \mathbf{J}(\mathbf{s}_{n-1} + \Delta t\mathbf{k}_3) + \mathbf{F}_n^{\mathrm{eff}}, \tag{B.7}$$

and

$$\mathbf{F}_{n-0.5}^{\mathrm{eff}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1}\mathbf{F}\left(t_{n-1} + \frac{\Delta t}{2}\right) \end{bmatrix}. \tag{B.8}$$

A time step size is selected to satisfy the following CFL condition for convergence, useful for explicit time integration schemes:

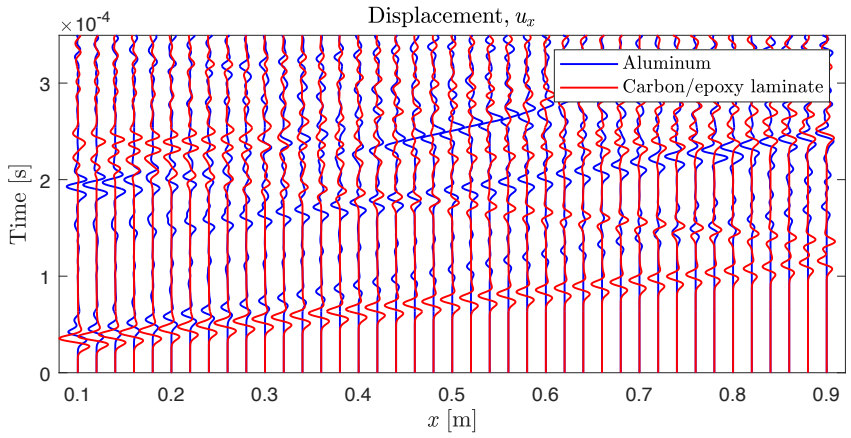
$$C = \frac{v_{max}\,(\Delta t)}{r_{min}} \leq C_{max}, \tag{B.9}$$

where C is the Courant number, v_{max} is the largest wave speed of the material(s) used in the model, Δt is the time step size, r_{min} is the smallest distance between nodes in the finite element model, and C_{max} is the maximum allowable Courant number. Choosing $C_{max} = 1.0$, the time steps used in this study meet the following condition:

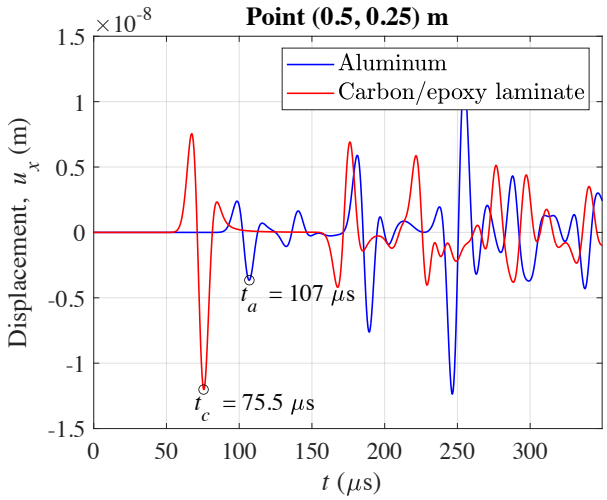
$$\Delta t \leq \frac{r_{min}}{v_{max}}. \tag{B.10}$$

Appendix C. Evaluation of the maximum wave velocity of the considered materials

We could analytically compute the maximum wave velocity of the considered materials and ensure that the CFL condition (defined in Eq. (B.9)) is satisfied. Namely, the maximum wave velocity of aluminum is calculated $v_a = \sqrt{\frac{E(1-\nu)}{\rho(1+\nu)(1-2\nu)}} = 6153$ m/s while that of carbon/epoxy laminate is $v_{xx} = \sqrt{\frac{E_{xx}}{\rho}} = 9837$ m/s [38]. To confirm our analytical evaluation, we simulate wave propagation in a homogeneous domain with aluminum or carbon/epoxy laminate. The size of the homogeneous domain is 1×0.5 m, identical to that of Fig. 3(a), with no cavities. The boundary and loading conditions are the same as the simulation for verification of our level-set SEM solver. Fig. C.36(a) presents a comparison of displacement-time histories in the x-direction at the middle of the domain ($y = 0.25$ m) for both the aluminum domain and carbon/epoxy laminate. The comparison is conducted at intervals of 0.02 m between $x = 0.1$ m and $x = 0.9$ m at $y = 0.25$ m. Fig. C.36(b) presents displacement-time histories for both domains at point $(x = 0.5, y = 0.25)$ m. In the case of the aluminum domain, the peak of the Ricker-pulse wave source initiates at $(0, 0.25)$ m after $t_i = 24$ μs, and reaches point $(0.5, 0.25)$ m at the arrival time of about $t_a = 107$ μs, taking a total of about 83 μs. On the other hand, for the carbon/epoxy laminate, the arrival time t_c is 75.5 μs, taking a total of 51.5 μs. The maximum wave velocity of each material can be numerically estimated by using the aforementioned travel time for each. The numerically estimated velocities of the materials are very close to their analytical counterparts, mentioned above.



(a) Along the x-axis at y = 0.25 m.



(b) At the sampling location (0.5 m, 0.25 m).

Figure C.36: Comparison of displacement-time histories of the aluminum domain and carbon/epoxy laminate.

References

[1] A. Asundi, A. Y. Choi, Fiber metal laminates: an advanced material for future aircraft, Journal of Materials processing technology 63 (1997) 384–394.

[2] P. Zuluaga-Ramírez, M. Frövel, T. Belenguer, F. Salazar, Non contact inspection of the fatigue damage state of carbon fiber reinforced polymer by optical surface roughness measurements, NDT & E International 70 (2015) 22–28.

[3] N. Chawla, K. Chawla, Metal-matrix composites in ground transportation, JoM 58 (2006) 67–70.

[4] M. Martarelli, P. Chiariotti, M. Pezzola, P. Castellini, Delamination detection in composites by laser ultrasonics, in: AIP Conference Proceedings, volume 1600, American Institute of Physics, 2014, pp. 405–412.

[5] N. Takeda, S. Minakuchi, Y. Okabe, Smart composite sandwich structures for future aerospace application-damage detection and suppression: A review, Journal of Solid Mechanics and Materials Engineering 1 (2007) 3–17.

[6] A. Katunin, K. Dragan, M. Dziendzikowski, Damage identification in aircraft composite structures: A case study using various non-destructive testing techniques, Composite structures 127 (2015) 1–9.

[7] R. Gupta, D. Mitchell, J. Blanche, S. Harper, W. Tang, K. Pancholi, L. Baines, D. G. Bucknall, D. Flynn, A review of sensing technologies for non-destructive evaluation of structural composite materials, Journal of Composites Science 5 (2021) 319.

[8] M. Mitra, S. Gopalakrishnan, Guided wave based structural health monitoring: A review, Smart Materials and Structures 25 (2016) 053001.

[9] N. Nazeer, M. Ratassepp, Z. Fan, Damage detection in bent plates using shear horizontal guided waves, Ultrasonics 75 (2017) 155–163.

[10] Z. Su, L. Ye, Y. Lu, Guided Lamb waves for identification of damage in composite structures: A review, Journal of sound and vibration 295 (2006) 753–780.

[11] S. K. Chakrapani, D. Barnard, V. Dayal, Finite element simulation of core inspection in helicopter rotor blades using guided waves, Ultrasonics 62 (2015) 126–135.

[12] C. Schaal, A. Mal, Core-skin disbond detection in a composite sandwich panel using guided ultrasonic waves, Journal of Nondestructive Evaluation, Diagnostics and Prognostics of Engineering Systems 1 (2017) 011006. doi:10.1115/1.4037544.

[13] S. Sikdar, S. Banerjee, Identification of disbond and high density core region in a honeycomb composite sandwich structure using ultrasonic guided waves, Composite Structures 152 (2016) 568–578.

[14] K. Balasubramaniam, S. Sikdar, R. Soman, P. Malinowski, Multi step structural health monitoring approaches in debonding assessment in a sandwich honeycomb composite structure using ultrasonic guided waves, Measurement 194 (2022) 111057.

[15] S. M. H. Hosseini, C. Willberg, A. Kharaghani, U. Gabbert, Characterization of the guided wave propagation in simplified foam, honeycomb and hollow sphere structures, Composites Part B: Engineering 56 (2014) 553–566.

[16] R. K. Munian, D. R. Mahapatra, S. Gopalakrishnan, Lamb wave interaction with composite delamination, Composite Structures 206 (2018) 484–498.

[17] S. Finnveden, Spectral finite element analysis of the vibration of straight fluid-filled pipes with flanges, Journal of Sound and Vibration 199 (1997) 125–154. doi:10.1006/jsvi.1996.0602.

[18] L. Gavrić, Computation of propagative waves in free rail using a finite element technique, Journal of Sound and Vibration 185 (1995) 531–543. doi:10.1006/jsvi.1995.0398.

[19] B. R. Mace, D. Duhamel, M. J. Brennan, L. Hinke, Finite element prediction of wave motion in structural waveguides, The Journal of the Acoustical Society of America 117 (2005) 2835–2843. doi:10.1121/1.1887126.

[20] C. Schaal, S. Zhang, H. Samajder, A. Mal, An analytical study of the scattering of ultrasonic guided waves at a delamination-like discontinuity in a plate, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science (2017) 1–14. doi:10.1177/0954406217700176.

[21] D. Balageas, C.-P. Fritzen, A. Güemes, Structural health monitoring, ISTE Ltd, London, UK, 2006.

[22] D. Rabinovich, D. Givoli, S. Vigdergauz, XFEM-based crack detection scheme using a genetic algorithm, International Journal for Numerical Methods in Engineering 71 (2007) 1051–1080.

[23] D. Rabinovich, D. Givoli, S. Vigdergauz, Crack identification by 'arrival time' using XFEM and a genetic algorithm, International Journal for Numerical Methods in Engineering 77 (2009) 337–359.

[24] H. Waisman, E. Chatzi, A. W. Smyth, Detection and quantification of flaws in structures by the extended finite element method and genetic algorithms, International Journal for Numerical Methods in Engineering 82 (2010) 303–328.

[25] E. N. Chatzi, B. Hiriyur, H. Waisman, A. W. Smyth, Experimental application and enhancement of the XFEM–GA algorithm for the detection of flaws in structures, Computers & Structures 89 (2011) 556–570.

[26] J. Jung, C. Jeong, E. Taciroglu, Identification of a scatterer embedded in elastic heterogeneous media using dynamic XFEM, Computer Methods in Applied Mechanics and Engineering 259 (2013) 50–63.

[27] L. Zhang, G. Yang, D. Hu, X. Han, An approach based on level set method for void identification of continuum structure with time-domain dynamic response, Applied Mathematical Modelling 75 (2019) 446–480.

[28] S. Jiang, L. Zhao, C. Du, Combining dynamic XFEM with machine learning for detection of multiple flaws, International Journal for Numerical Methods in Engineering 122 (2021) 6253–6282.

[29] C. Humer, S. Höll, C. Kralovec, M. Schagerl, Damage identification using wave damage interaction coefficients predicted by deep neural networks, Ultrasonics 124 (2022) 106743. URL: https://www.sciencedirect.com/science/article/pii/S0041624X22000555. doi:https://doi.org/10.1016/j.ultras.2022.106743.

[30] T. Latête, B. Gauthier, P. Belanger, Towards using convolutional neural network to locate, identify and size defects in phased array ultrasonic testing, Ultrasonics 115 (2021) 106436. URL: https://www.sciencedirect.com/science/article/pii/S0041624X21000731. doi:https://doi.org/10.1016/j.ultras.2021.106436.

[31] Y. Mei, J. Chen, Y. Zeng, L. Wu, Z. Fan, Laser ultrasonic imaging of complex defects with full-matrix capture and deep-learning extraction, Ultrasonics 129 (2023) 106915. URL: https://www.sciencedirect.com/science/article/pii/S0041624X22002219. doi:https://doi.org/10.1016/j.ultras.2022.106915.

[32] F. M. Pranto, S. Maharjan, C. Jeong, Level-set and learn: Convolutional neural network for classification of elements to identify an arbitrary number of voids in a 2D solid using elastic waves, Journal of Engineering Mechanics 149 (2023) 04023035.
[33] P. Vanucci, Anisotropic elasticity. lecture notes in applied and computational mechanics, 2018.
[34] H. Sun, H. Waisman, R. Betti, A multiscale flaw detection algorithm based on XFEM, International Journal for Numerical Methods in Engineering 100 (2014) 477–503.
[35] G. Sun, Z. Zhou, Non-contact detection of delamination in layered anisotropic composite materials with ultrasonic waves generated and detected by lasers, Optik 127 (2016) 6424–6433.
[36] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1026–1034.
[37] D. Komatitsch, J.-P. Vilotte, The spectral element method: an efficient tool to simulate the seismic response of 2D and 3D geological structures, Bulletin of the seismological society of America 88 (1998) 368–392.
[38] G. Baum, C. Habeger, E. Fleischman, Measurement of the orthotropic elastic constants of paper, in: The role of fundamental research in paper-making, Trans. of the VIIth Fund. Res. Symp. Cambridge, 1981. doi:10.15376/frc.1981.1.453.