Sound Source Localization Inside a Structure Under Semi-Supervised Conditions

Shunsuke Kita, *Graduate Student Member, IEEE*, and Yoshinobu Kajikawa, *Senior Member, IEEE*

***Abstract*—We propose a method for applying a sound source localization (SSL) model trained on simulated data in a real-world environment, with a domain transfer (DT) model for the SSL inside a structure. The DT model transfers real data into pseudo-simulation data. The SSL model trained on the simulation data is then adapted to the real data using the DT model. Our method consists of an SSL model and a DT model. The SSL model predicts the position of a sound source inside the structure, whereas the DT model transforms the data. Because our simulation is not perfect, real data are extrapolated for use with the SSL model. However, the data transformed by the DT model are interpolated within the feature space. The outcome is that the performance of the SSL model in the real world is improved. In our study, the frequency spectra of accelerometers observed on the outer surface of the structure are the model input. The goal is to predict the position of the sound source. The SSL model is built using deep and convolutional neural networks, and the DT model is built using either an autoencoder, a deep convolutional autoencoder, or pix2pix. The two-dimensional distributions of the t-distributed Stochastic Neighbor Embedding indicate that using pix2pix as the DT model shows the best performance. Furthermore, our method's performance for SSL is improved by 57% for the classification problem and by 27% for the regression problem when compared to the case where no transformation is applied.**

***Index Terms*—Sound source localization, domain transfer, acoustic-structure coupling, t-distributed stochastic neighbor embedding.**

I. Introduction

Sound source localization (SSL) is an important for reducing the noise of machines and electrical appliances. Currently, several SSL methods that use the correlation of time-frequency signals observed by multiple microphones have been proposed. Those methods are based on the time difference of arrival (TDOA) of acoustic signals [1], [2]. Many studies have reported improvements in TDOA problems such as in noise, reverberation, and the simultaneous emission of sound sources. In recent years, several methods have been proposed that incorporate deep learning and overcome different scenarios that are challenging for conventional methods [3], [4]. However, the applications of these methods are limited to circumstances where acoustic signals can be directly observed. These methods are not applicable for estimating, from outside, the position of a sound source inside a structure because the acoustic signals are observed as indirect sounds.

The SSL inside a structure is an important problem because it leads to essential solutions for product noise reduction. For example, if noise is generated owing to damage to a component inside the structure, disassembling the structure or placing measurement equipment inside the structure is not an option because it would change the structure's response system. In other words, the resonant frequency changes because the disassembly of the structure or placement of the measurement device causes a change in the volume of the acoustic space. Therefore, the SSL has to be conducted outside the structure under normal operating conditions. Specifically, it can detect the position of noise owing to component defects, deterioration, and interference that occur in mass-produced home appliances, mechanical products, and prototypes. Other applications can be applied to SSL in situations that cannot be observed directly. For example, SSL can be applied to estimate the position of noise generated in gas or water pipes. This research deals with the problem of estimating the location or position of sources that cannot be directly observed.

Methods based on deep neural networks (DNN) and computer-aided engineering (CAE) have been proposed for estimating the sound source inside a structure [5]. Our method successfully estimated the position of the sound source inside the structure from the signals observed by accelerometers installed on the outer surface of the structure, in both the simulation and real domains. However, our method still faces the challenge of applying a DNN trained in the simulation domain to the real domain. The main problem is that there are differences between the simulation data and actual experimental data. These differences occur because the simulation data poorly simulate the actual experimental conditions of a structure's geometry, material parameters, and nonlinearity. For both the indirect and direct sound, it is still difficult to apply the trained model built on simulation data to real data because the simulation is not perfect [6].

To solve the SSL problem, our study focuses on a method to apply models built with simulated data to real data. Adapting a trained model to another task or data is called "transfer learning (TL)," which has been studied in the fields of visual categorization and natural language processing by a large number of

Manuscript received 23 June 2022; revised 4 December 2022 and 6 February 2023; accepted 20 March 2023. Date of publication 5 April 2023; date of current version 10 April 2023. This work was supported by the JSPS KAKENHI under Grant 22K03991. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jon Barker. *(Corresponding author: Shunsuke Kita.)*

Shunsuke Kita is with the Division of Electronics and Mechanical Systems, Osaka Research Institute of Industrial Science and Technology, Osaka 594115, Japan (e-mail: kitas@orist.jp).

Yoshinobu Kajikawa is with the Faculty of Engineering Science, Kansai University, Osaka 5648680, Japan (e-mail: kaji@kansai-u.ac.jp).

Digital Object Identifier 10.1109/TASLP.2023.3263776

researchers [7], [8], [9], [10], [11]. With TL, the goal is to reduce the performance degradation caused by different distributions (called domain shifts) of the data used to train the model (source domain) and test data (target domain). Although there are a few studies on SSL in the TL, there are no methods for estimating the position of the sound source inside the structure. Previous studies on SSL have focused on classification problems and assumed weakly supervised labels and TL in visual categorization [12], [13], [14], [15]. We could not use these methods because our study included a regression problem for predicting the coordinates of a sound source. In addition, weakly supervised learning situations with missing or noisy labels were not targeted.

We focus on "transductive transfer learning" or "domain adaptation (DA)" because the task is the same and only the domain differs [16]. Within that learning condition, we use a "feature-representation-transfer approach" because the equivalency of conditional distributions in the source and target domains is not guaranteed [17]. The SSL in this study can use real data; that is, it is a semi-supervised condition. In the case of labeled and unlabeled data available in the target domain, there are discrepancy, adversarial, and reconstructive approaches for solving the DA with deep learning [18]. These methods mainly use invariant representations of the source and target data or assign pseudo labels to the target data.

We propose a DA method for SSL inside a structure under semi-supervised conditions. The datasets are labeled according to the area or position of the sound sources and not based on the data. Therefore, the model built in the simulation domain is used directly because the condition distributions differed between the simulation and real domains. The method consists of a domain transfer (DT) model and an SSL model. The DT model transforms real data into pseudo-simulation data and the SSL model predicts the position of the sound source from the transferred data.

The remainder of this paper is organized as follows. In Section II, the method used in our previous study on SSL inside a structure is summarized. In Section III, the proposed method for applying the models built in the simulation to real environments is described. Section IV presents the datasets of the simulation and actual experimental data. In Sections V and VI, the SSL and DT models are described. The results are described and discussed in Section VII, and Section VIII presents our conclusions.

II. Formularisation and Framework for SSL Inside a Structure

The finite discretization equation for the forced vibration of the acoustic–structural coupled problem is as follows:

$$\begin{bmatrix} \mathbf{M}_{\mathrm{S}} & 0 \\ \overline{\rho_0}\mathbf{R}^{\mathrm{T}} & \mathbf{M}_{\mathrm{F}} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{u}} \\ \ddot{\mathbf{p}} \end{bmatrix} + \begin{bmatrix} \mathbf{C}_{\mathrm{S}} & 0 \\ 0 & \mathbf{C}_{\mathrm{F}} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{p}} \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{\mathrm{S}} & -\mathbf{R} \\ 0 & \mathbf{K}_{\mathrm{F}} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{\mathrm{S}} \\ \mathbf{F}_{\mathrm{F}} \end{bmatrix}, \tag{1}$$

where $\mathbf{u}$, $\mathbf{p}$, $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{K}$ represent the displacement, sound pressure, mass matrix, damping matrix, and stiffness matrix, respectively. Suffixes S and F denote the structural and acoustic terms, respectively. Here, $\overline{\rho_0}$ denotes the mass density constant of the acoustic fluid, and matrix $\mathbf{R}$ denotes the coupled term. In this study, the external forces on the structure are not assumed ($\mathbf{F}_{\mathrm{S}} = 0$). The SSL inside the structure determines the input term for the inverse problem. There are three physical limitations to identifying the source inside the structure. First, the input-term estimation problem at resonance is ill-posed because the uniqueness of the solution is not guaranteed. Second, because the sound radiating from the structure is an indirect sound, the phase information of the internal sound source is lost. In the resonance state that causes noise, the resonance characteristics derived from the acoustic space are mixed with those derived from the structure. Third, because noise characteristics are determined by the resonance characteristics of the structure and its acoustic space, the structure cannot be disassembled. Therefore, in the framework of SSL inside the structure, the location or position of the sound source is stochastically estimated from observation data outside the structure, using machine learning techniques and simulations. We propose a new method using machine learning, which is required for SSL inside structures.

As shown in Fig. 1, a framework for SSL inside the structure is implemented in the following three steps [5].

a) *Data generation by simulation:* A coupled acoustic-structure analysis is used to generate datasets that consist of data observed outside a structure and the position of a sound source. For example, the finite element method (FEM) is used to generate analytical data, such as acceleration signals on the exterior surface of the structure and acoustic signals around the structure corresponding to the acoustic excitation of the sound source position.

b) *Training of SSL model.:* The analytical data obtained from the coupled acoustic structure analysis are defined as input data for the DNN, and the positions of the sound sources paired with the input data are defined as the labels for the DNN. In other words, a combination (X, T) of the data observed outside the structure (input data X) and the location or position of the sound source (label data T) are treated as a dataset. The matrix $\mathbf{X}$ of D-dimensional input vectors $\mathbf{x}_i$ and the matrix $\mathbf{T}$ of K-dimensional label vectors $\mathbf{t}_i$ are given by

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_N)^{\mathrm{T}}, \tag{2}$$

$$\mathbf{T} = (\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3, \ldots, \mathbf{t}_N)^{\mathrm{T}}, \tag{3}$$

where the subscript N indicates the number of samples. In this step, the input-output relationships are learned by the DNN.

c) *Prediction of the sound source positions:* The trained DNN constructed with the simulation data is used in the real world for SSL.

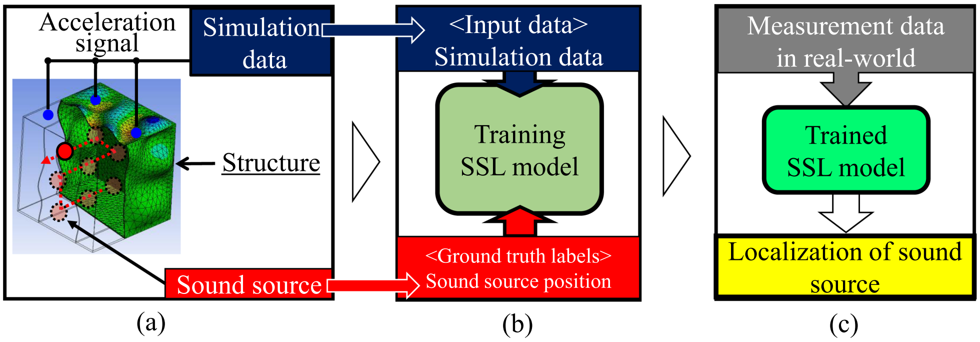

Acceleration signal
Simulation data
Structure
Sound source
<Input data> Simulation data
Training SSL model
<Ground truth labels> Sound source position
Measurement data in real-world
Trained SSL model
Localization of sound source
(a)
(b)
(c)

Fig. 1. Framework for SSL inside the structure. (a) Data generation by simulation, (b) training of SSL model, and (c) prediction of the sound source position.

Because this method is based on sampling data from a simulation, it applies to objects of various sizes, and the resolution of the SSL can be set as needed. Narrowing the sampling interval of the simulation improves the resolution near the decision boundary for the classification problem and reduces the variance of the RMSE for the regression problem owing to the increase in the number of data points. For example, if a regression problem requires an SSL performance of 20 mm or less, it can be handled by setting the sampling interval for the simulation to 20 mm or less, thereby allowing flexibility in the setup. In the product development design step, the clearance of the components placed inside the product is designed by considering the intersection of each element. The noise source parts can be identified by setting the sampling interval below the clearance in the simulation.

Furthermore, because this method uses multiple sensors, the transfer functions between the sensors as input data for the model enable SSL without depending on the characteristics of the sound source. Specifically, $Y_1 = G_1 S_0$ holds if the characteristic of the sound source is S_0, the signal observed by sensor “1” is Y_1, and the transfer function due to the path from the sound source to the observation point is G_1. Similarly, $Y_2 = G_2 S_0$ and $Y_3 = G_3 S_0$ for the other sensors. If Y_1 is the reference sensor, the ratios of the three observed signals are $Y_2/Y_1 = G_2/G_1$, $Y_3/Y_1 = G_3/G_1$, and $Y_2/Y_3 = G_2/G_3$, and these expressions are independent of S_0. By defining these three ratios as F_1, F_2, and F_3 and by using them as training data, SSL can be applied independently of the sound source characteristics. This methodology may be affected by noise from accelerometer observations and is not applied in this study. We plan to treat the noise as an optimization problem for the location and number of sensors. This is because the amplitude of the FSA measured at each observation point is different; therefore, the effect of noise is likely to be different for each point. In view of this, the present study focuses on the domain transfer problem without applying the division method because we consider that the domain transfer can be properly evaluated without canceling the characteristics of the sound sources.

III. Proposed Method:under Semi-Supervised Conditions for SSL Inside Structure

In general, the generalization performance is significantly lower when training and test data are sampled from populations with different distributions [7], [8], [19]. This is important when using machine-learning models in real-world scenarios. For the same reason, when the model trained in the simulation is applied to a real environment, the generalization performance of the trained model is low because the simulation does not perfectly reproduce the real environment. Therefore, the difference between the simulation and real data significantly decreases the SSL model’s performance that is trained during the simulation.

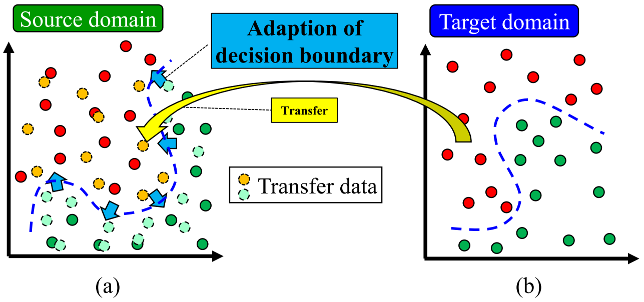


Fig. 2. Transformation and adaptation with the domain transfer model. (a) The simulation (source) domain, and (b) the real (target) domain.

In this study, the DT model is applied to reduce distributional discrepancies under semi-supervised conditions. DT models are incorporated into the framework of the SSL inside the structure. The DT model transfers real data into pseudo-simulation data so that the SSL model constructed in the simulation domain can handle real data. In machine learning, the simulation domain corresponds to the source and the real domain corresponds to the target [20]. Fig. 2 shows a schematic of data transfer and decision boundary adaptation in a situation where the simulation and real domain distributions and their respective decision boundaries are different. In general DA techniques, the direction of the transformation is from the source domain to the target domain. However, our method predicts the position of the sound source using the SSL model built into the simulation domain; therefore, the direction of the transformation is the opposite. In other words, the target data are transferred to the source data. This reverse transformation strategy is being studied further in a field called DA for semantic segmentation [21], which is a recent development.

The contribution of this study is to show a domain transformation method to adapt the SSL model built on simulation to the

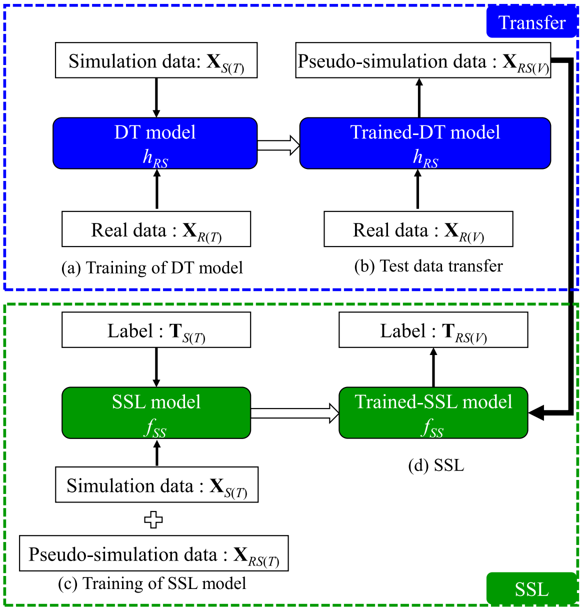

Transfer
Simulation data: $\mathbf{X}_{S(T)}$
Pseudo-simulation data : $\mathbf{X}_{RS(V)}$
DT model
h_{RS}
Trained-DT model
h_{RS}
Real data : $\mathbf{X}_{R(T)}$
Real data : $\mathbf{X}_{R(V)}$
(a) Training of DT model
(b) Test data transfer
Label : $\mathbf{T}_{S(T)}$
Label : $\mathbf{T}_{RS(V)}$
SSL model
f_{SS}
Trained-SSL model
f_{SS}
(d) SSL
Simulation data : $\mathbf{X}_{S(T)}$
+
Pseudo-simulation data : $\mathbf{X}_{RS(T)}$
(c) Training of SSL model
SSL

Fig. 3. Flowchart of the proposed method.

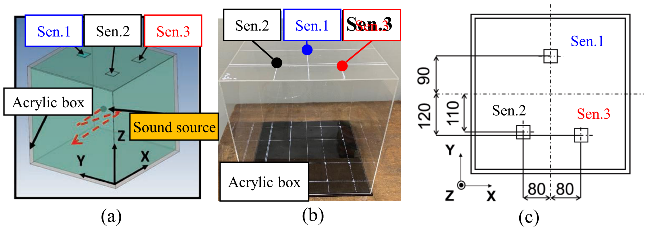


Fig. 4. Simulation and real domain setup. (a) Simulation. (b) Real. (c) Sensor placement positions.

real environment for the SSL inside structures, which has rarely been studied. Typical "source to target" domain transformation methods have been applied to photographic images and text data that are data-rich in both the source and target domains and have not been applied to SSLs inside structures. Our study requires the SSL in the target domain under the condition that a large amount of simulation data is available but real data is limited. Therefore, it is essentially impossible in this research to use "source to target." Because of this limitation, we use a "target to source" transformation direction. Furthermore, although our previous work [5] could not directly build SSL models with small amounts of real data, this inverse transformation contributes to the leveraging of SSL models trained on large amounts of simulated data.

Our method has the potential to use the numerous discriminative, regressive, and generative models that have been proposed. A flowchart of our method is shown in Fig. 3. The blue box represents the training and transfer phases of the DT model and the green box represent the training and prediction phases of the SSL model. The subscripts S and R denote the simulation and real domains, respectively, and $\mathbf{X}_S$ and $\mathbf{X}_R$ denote the input data. In addition, the paired sound source position labels are denoted by $\mathbf{T}_S$ and $\mathbf{T}_R$. The goal is to estimate $\mathbf{T}_R$ from $\mathbf{X}_R$ by using the SSL model built in the simulation domain. In most cases, $\mathbf{X}_S$ does not equal $\mathbf{X}_R$; and so the SSLs f_{SS} and f_{RR} for each domain are different; f_{SS} and f_{RR} are the models built in the simulation and real domains, respectively. Consequently, the SSL in the real domain using the SSL model trained in the simulation domain failed.

Therefore, the DT model h_{RS} is adapted to reduce the distributional discrepancies between the simulation and real domains. The model uses $N_{R(T)}$ pairs of simulation data $\mathbf{X}_{S(T)}$ and real data $\mathbf{X}_{R(T)}$ as the training data (Fig. 3(a)). The subscripts $S(T)$ and $R(T)$ represent the simulation and real training data, respectively. The DT model transforms the test data $\mathbf{X}_{R(V)}$ from the real data into pseudo-simulation data $\mathbf{X}_{RS(V)}$ (Fig. 3(b)). The subscripts $R(V)$ and $RS(V)$ denote the real and pseudo-simulated test data, respectively. The SSL model f_{SS} is built using the training dataset $(\mathbf{X}_{S(T)}, \mathbf{T}_{S(T)})$ with $N_{S(T)}$ datasets (Fig. 3(c)). By providing pseudo-simulation test data $\mathbf{X}_{RS(V)}$ as input data to the trained-SSL model, the sound source positions in the real data are predicted (Fig. 3(d)). The training and test data meet the following criteria.

$$N_{R(T)} + N_{R(V)} = N_R, \tag{4}$$

where N_R is the total number of real data and $N_{R(V)}$ is the real test data.

$$N_{S(T)} = N_S + N_{RS(T)}, N_{RS(T)} = N_{R(T)}, \tag{5}$$

where N_S is the total number of simulation data and $N_{RS(T)}$ are the pseudo-simulation training data. Pseudo-simulation data are not guaranteed to be transformed according to the decision boundaries of the SSL model. Therefore, the SSL model is trained on both the simulation and pseudo-simulation data to adapt to discriminative boundaries.

IV. Datasets of Simulation and Actual Experimental Data

A situation is assumed in which the acoustic excitation from "a single sound source" within the structure is measured using three accelerometers mounted on the outer surface of the structure. The frequency spectra of the accelerometer (FSA) are used as the observation data. The subject is an acrylic box as shown in Fig. 4. Fig. 4(a) shows the simulation and Fig. 4(b) shows a real domain. The datasets are FSAs observed by three accelerometers on the outer surface of the structure paired with the sound source position labels. These datasets are collected from both domains at the same sensor positions (Fig. 4(c)). The acoustic volume is 400 × 400 × 400 mm^3 and the thickness of the acrylic box is 3 mm.

The simulation conditions are listed in Table I. The simulation data are generated from a coupled acoustic structure analysis using FEM. The FEM solver is a full-harmonic analysis in ANSYS Mechanical [22]. (1) is solved using the FEM solver. The conditions for the position of the sound source are intervals of 50 mm for the simulation and 512 sound source points. The

TABLE I
CONDITIONS OF ANALYSIS

Acrylic young's modulus	27 MPa
Acrylic density	$1180\ \mathrm{kg/m^3}$
Acrylic damping ratio	0.8
Interval of sound source	50 mm
Number of sound source location	512
Observation	Sen.1 - Sen.3
Frequency range	0.01-1.5 kHz

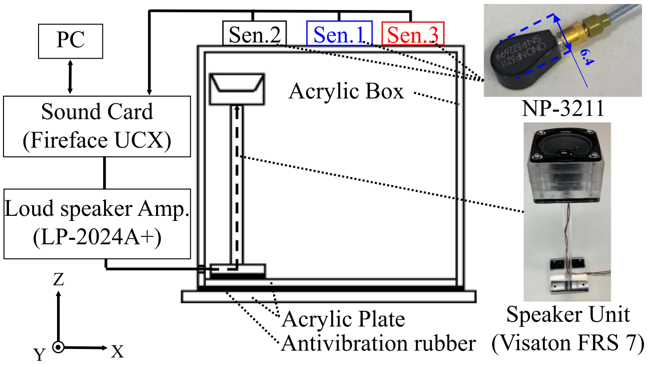


Fig. 5. Actual experimental setup.

TABLE II
CONDITIONS OF MEASUREMENT

Interval of sound source	100 mm
Number of sound source location	64
Observation	Sen.1 - Sen.3
Input signal	Swept sinusoidal
Frequency range	0.01-1.5 kHz
Sampling rate	4.8 kHz
Sound pressure	90 dB at 1 m
Sub band width	10 Hz

frequency range is 0.01–1.5 kHz, and the increment range of the data is 10 Hz.

The experimental conditions are shown in Fig. 5. In the actual experiment, one loudspeaker (Visation FRS 7) is placed inside the acrylic box as the sound source. The acoustic excitation of the structure is measured using three acceleration sensors (Ono Sokki Co. Ltd. NP-3211) installed on the outer surface of the structure. The sound waves of the sweep signal are generated by a loudspeaker via a sound card (Fireface UCX) and a loudspeaker amplifier (LP-2024-A +). The bottom of the structure and the loudspeaker are covered with anti-vibration sheets to reduce structure-borne sound. The experimental conditions are listed in Table II. The conditions for the position of the sound source are an interval of 100 mm between the actual experiment and 64 sound source points. The frequency range is 0.01–1.5 kHz, the same as the simulation domain. The time-series vibration data are measured at a sampling frequency of 4.8 kHz. The time-series acceleration data measured by the three sensors are transformed into FSA by applying a fast Fourier transform. The FSAs are sub banded into 150 bins by calculating the average of each band, which is used as the representative value. Therefore, the dimensions of the FSA per sensor in both the simulation and target domains are 150.

TABLE III
CONDITIONS OF THE SSL MODEL

Hidden layer	DNN : F400, F350, F300, F200, F100, F50 CNN : C256, C128, C64, C32, C16, F200, F150, F100, F50, F25, F20,
Activation	Hidden layer : ReLU Output : Linear (Reg.), Softmax (Class)
Loss function	Mean squared error (Reg.), Cross entropy error (Class)
Initialization	He normal
Batch size	50
Epochs	1000

The FSAs generated in both domains are defined as the data formats corresponding to the model as follows:

a) *Vector data:* The FSAs measured at the three positions are concatenated as a horizontal vector from Sens 1 to 3 in sequence when treated as vector data. Hence, the size of the observation data for one sound source point is 1×450.

b) *Image data:* The FSAs measured at three positions are transposed and concatenated horizontally. The size of the observation data for each sound source point is 150×3. Defining an array of data in this manner results in two-dimensional (2-D) data.

The input and label data for the DT model are FSAs. The input data for the SSL model are the FSA, and the label data changes depending on the problem [5]. In other words, in the case of the classification problem, the problem is to estimate which of the eight regions of the acoustic space where the sound source is located. In the case of a regression problem, the problem is to predict the X, Y, and Z coordinates.

V. EXPERIMENTAL SETUP USING SSL MODELS

SSL performance is tested by feeding pseudo-transformation data into the SSL model. The SSL model conditions are listed in Table III. A DNN is used when the input data are vectors, and a CNN is used when the input data are images. The optimization, preprocessing, and metrics are the same as those of the DT model. In the case of the classification problem, the total acoustic space is divided into eight acoustic sub volumes, and each acoustic space is labeled according to one of the K coding schemes. In the case of the regression problem, the X-, Y-, and Z-coordinates are directly defined as label data.

Accuracy (Acc.) shown in (6) is used to evaluate the accuracy of the classification problem, and the RMSE shown in (7) is used to solve the regression problem.

$$\text{Acc.} = \frac{\text{The number of correct answers}}{N_{RS(V)}}, \tag{6}$$

$$\text{RMSE} = \sqrt{\frac{(\mathbf{T}_{RS(V)} - f_{SS}(\mathbf{X}_{RS(V)}))^{\mathrm{T}}(\mathbf{T}_{RS(V)} - f_{SS}(\mathbf{X}_{RS(V)}))}{N_{RS(V)}}}, \tag{7}$$

where $N_{RS(V)}$ denotes the number of transformed test data points. The label data consider the X-, Y-, and Z-coordinates; hence, the RMSE is expressed by (7). The percentage of the actual experimental data used as the training data for the DT

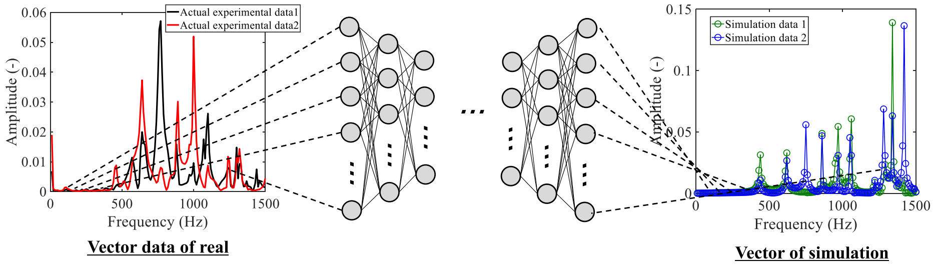


Fig. 6. DT model using the AE. The AE uses vector data as input and output data. Input data is the real data and output is the pseudo-simulation data.

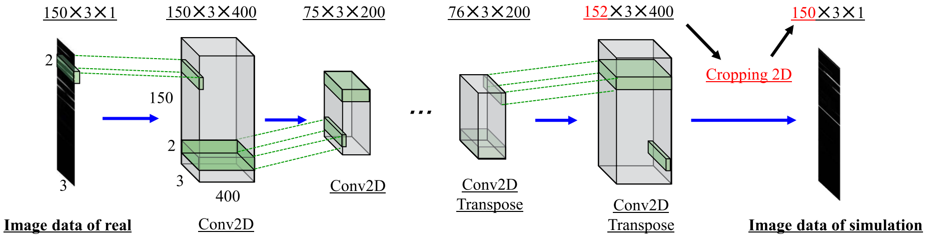


Fig. 7. DT model using DCAE. DCAE uses image data as input and output data. Input is the real data and the output is pseudo-simulation data.

model varies from 20 to 80% of the conditions, and the SSL performance is measured in each case. The SSL performance for real data is tested by predicting the SSL model on pseudo-simulated data, where the DT model transformed the real data into simulated data.

VI. Experimental Setup Using DT Models

The effectiveness of the proposed method is tested by evaluating the transformation performance of the DT model. In this study, the DT model is selected using two learning approaches: an encoder-decoder model and a generative model. The encoder-decoder model is built as an autoencoder (AE) [23] or a deep convolutional autoencoder (DCAE) [24]. The generative model is built using pix2pix [25] based on conditional generative adversarial nets (cGAN) [26]. cGAN is a model that allows generative adversarial nets (GAN) [27] to use conditional probabilities. The transfer performance of these models is evaluated using root mean square error (RMSE) and t-distributed Stochastic Neighbor Embedding (t-SNE) [28] distributions.

A. Encoder-Decoder Models

The input and output data for AE and DCAE are the vector and image data, respectively, and both DT models convert real data into pseudo-simulation data. Figs. 6 and 7 show an overview of the AE and DCAE. Both models are given FSAs of the real domain as the input and the simulation domain as the label. The difference between these models is whether a fully connected layer or a convolutional layer is used. The fully connected layer executed its task based on the extraction of features by linear summation over the input data and mapping by nonlinear activation functions. Therefore, it is not guaranteed to be equivariant or invariant [29], and it is not robust to either frequency peaks or notch deviations. By contrast, the convolutional layer can extract local features through filtering [30]. In addition, subsampling makes the convolutional layer robust against feature misalignments in an image.

TABLE IV
Conditions of AE and DCAE

Hidden layer	AE : F400, F350, F300, F250, F200, F100, F100, F200, F250, F300, F350, F400 DCAE : C400, C200, C100, CT100, CT200, CT400
Activation	Hidden layer : ReLU Output : AE (Linear), DCAE (Sigmoid)
Optimization	Adam : Learning rate = 0.001 (β_1 = 0.9, β_2 = 0.999)
Loss function	Mean squared error
Initialization	He normal
Batch size	5
Epochs	1000
Preprocessing	Min-max normalization
Metrics	Hold-out validation

The conditions for the AE and DCAE are listed in Table IV. “F” represents fully connected layers, “C” represents convolutional layers, and “CT” represents convolutional transpose layers. When both the input and label data are vector data, the AE is adopted as the DT model. When both the input and label data are image data, the DCAE is adopted. Batch normalization [31] is applied between the layers of each model. In the DCAE encoder, the 2-D convolution layers are set as (2, 3) kernel size, (1, 1) stride, and had the same padding. The hyperparameters of the convolutional layers transformed data of size (150, 3, 1) into (150, 3, 400) in the first layer. In the DCAE encoder, the 2-D convolution transpose layers are set to (2, 3) kernel size, (2, 1) stride, and the same padding. In the last decoder layer, cropping 2D is applied because the reconstruction size differs from the desired size owing to the effect of the input data

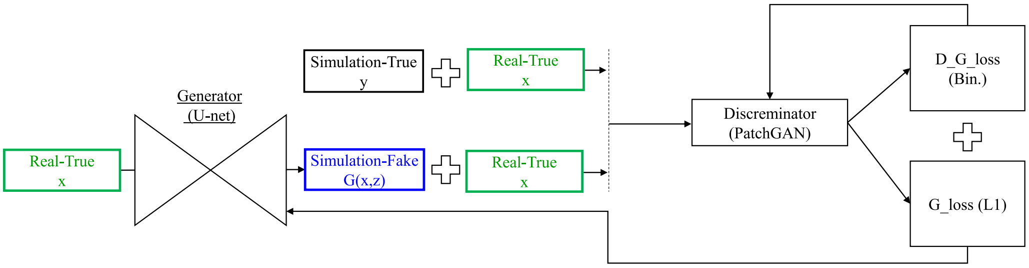


Fig. 8. DT model using pix2pix. Pix2pix consists of U-net and patchGAN frameworks.

size. In [32], [33], the frequency-response data are normalized from zero to $2^{16} - 1$ after a hyperbolic tangent transformation. In this study, because the measured data are between 0 and 1, the only preprocessing step performed is min-max normalization for each dataset. Masking data augmentation is applied to build each model [34], [35].

B. *Generative Models*

An overview of pix2pix is presented in Fig. 8. Pix2pix is included in cGAN, which is the conditional model of GAN. Its structure comprises a generator composed of U-NET [36] and a discriminator composed of patchGAN. Both the generator and discriminator are convolution-BatchNorm-ReLu methods. The goal of this model, similar to the encoder-decoder model, is to generate pseudo-simulation data that are similar to the simulated data. In pix2pix, two images are paired and trained using adversarial training. Through adversarial learning, a generator can generate fake images that appear as true images. *I.e.*, the generator is responsible for transforming the real data into more realistic simulated fake data. In contrast, the discriminator receives concatenated simulation-true and real-true data or concatenated simulation fake and real data generated by the generator.

The pix2pix generator generates an image $G(x, z)$ from image x and noise z such that the discriminator cannot distinguish between true and fake images. The pix2pix's discriminator takes a pair of images x and conditional y (or $G(x, z)$) as input and determines whether the image is fake or authentic. The loss of binary classification is given by

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z)))], \quad (8)$$

where "G" represents the generator and "D" represents the discriminator. "G" attempts to maximize this loss while "D" minimizes it in the adversarial training manner. The L1 distance s adopted for the generator loss.

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[||y - G(x, z)||_1] \ . \quad (9)$$

Consequently, the final loss function is.

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G) \ . \quad (10)$$

Note that the loss function of the discriminator uses the binary classification loss. In other words, the loss function is adapted to determine only whether the image generated by the generator is fake or authentic, and not a human-designed loss function such as AE or DCAE. Generally, it is difficult to design a loss function that best represents a dataset. For example, MSE cannot be used to evaluate the resonant frequency deviations, as is clear from its definition.

TABLE V
CONDITIONS OF PIX2PIX

Discriminator	C256, C128, C64, C32, C16, C1
Generator	En.:C64, C128, C256, C512, C512, C512, C512, C512 De.:CT512, CT512 ,CT512, CT512, CT256, CT128, CT64
Activation(Discriminator)	Hidden layer：LeakyReLU Output：sigmoid
Activation(Generator)	Hidden layer：LeakyReLU (En.), ReLU (De.) Output：tanh
Optimization	Adam : Learning rate = 0.0002 (β_1 = 0.5, β_2 = 0.999) λ = 100
Loss function	binary crossentropy+L1
Initialization	Random Normal
Batch size	Instance normalization:1
Epochs	1000
Preprocessing	[-1 1] normalization
Metrics	Hold-out validation

The conditions for pix2pix are listed in Table V. The pix2pix's discriminator takes the form of a patchGAN. The receptive field of the input data for one pixel of the output is a 6×3 patch. The pix2pix generator is in the form of a U-NET, and its structure is the same as [25]. As with AE and DCAE, min-max normalization is applied for preprocessing, followed by pix2pix specific [-1 1] normalization. This process is based on the tanh activation function of the generator. In the discriminator and generator, the 2-D convolution layers are set to (2, 3) kernel size, (1, 1) stride, and the same padding. Furthermore, the noise z consist of dropouts. The AE, DCAE, and pix2pix transfer performances are not only evaluated by the RMSE but also visualized by t-SNE.

Note that the paired data is determined based on the X-, Y-, and Z-coordinates. As shown in Tables I and II, the number of sound source positions is 512 in the simulation and 64 in the real environment. The pair data are determined by selecting X-, Y-, and Z-coordinates that are the same as or close to each other.

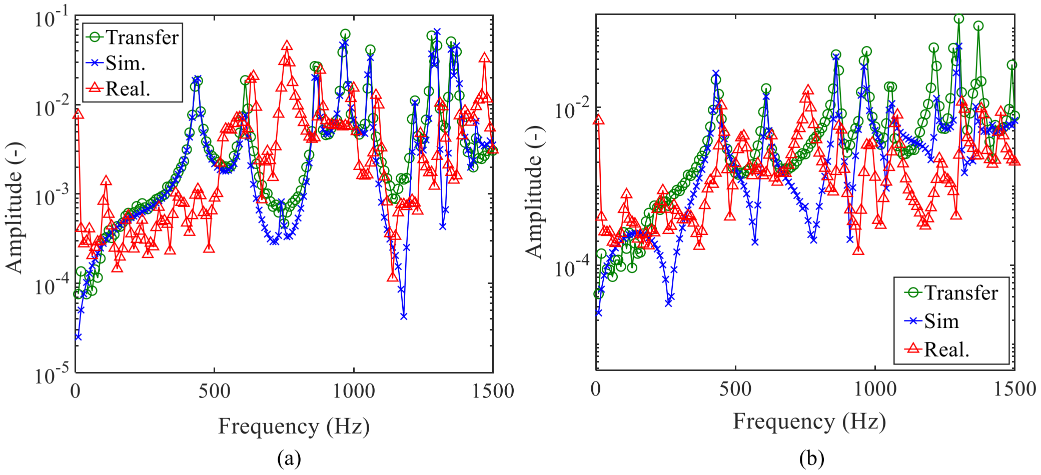


Fig. 9. FSA transformation of training and test data. The semi-supervised data is 70% of the total actual experimental data. (a) Training data. (b) Test data.

VII. Results and Discussion

First, the transformation performance of the DT model is described. AE, DCAE, and pix2pix are selected as DT models, and their transformative performance is visualized in two dimensions with t-SNE and quantified using the RMSE. Then, we describe the SSL performance. Here, we show the relationship between the amount of semi-supervised data and the SSL performance for each of the classification and regression problems. Finally, a comparison of the SSL performance of the proposed method to that of the conventional method and the non-adaptative case (from our previous work) is made.

A. Data Transformation Performance With DT Model

Fig. 9 shows an example of data transformation by AE. The red, blue, and green solid lines represent real, simulated, and transformed data, respectively. Fig. 9(a) and (b) show the training and testing data, respectively. This figure shows that the transformed data are shifted in the resonance frequency and transformed closer to the simulation data. To evaluate the transformation performance of each model quantitatively, the RMSEs values are shown in Fig. 10. Each solid line in the figure represents the RMSE for AE, DCAE, and pix2pix for each of the training and test datasets. This result indicates the conclusion that pix2pix's transformation performance is worse than DCAE's for the training data. Furthermore, the RMSE of the test data is the largest for pix2pix, and the RMSE did not seem to decrease as the number of semi-supervised datasets increased. However, the visualization of t-SNE leads to different conclusions regarding the transformation performance. Fig. 11(a) shows the visualization of t-SNE for the transformation performance of the AE. The red, blue, and green plots represent real, simulated, and transformed data, respectively. The numbers in the plots represent the classes in the classification problem. The number of classes in the classification problem is eight, implying that numbers $0-7$ are given. The subscripts “T” and “V” represent training and test data, respectively. Clearly, visualization by t-SNE shows that domain matching by AE is not possible. This can be understood from the fact that AE cannot learn the local features of the data, and the transformed data have negative values. Fig. 11(b) shows the t-SNE visualization of the data transformed using DCAE. This distribution shows that the transformation by DCAE enables domain matching in most of the data (most of the plots overlap in the well-matched, and thus, a zoomed-in view is shown in the figure). Furthermore, the transformation by pix2pix appears to match better (Fig. 11(c)). Although evaluation using the RMSE is useful because of its quantitative aspect, the RMSE is not necessarily correct for transformational performance. As the RMSE cannot evaluate the amount of frequency misalignment, it is inappropriate as an indicator for evaluating the characteristics of the FSA. Similarly, the MSE set as the loss function for DCAE is also considered inappropriate. The measurement of the similarity between the two resonance frequencies is summarized in [37]. A method that uses these metrics as loss functions should also be considered.

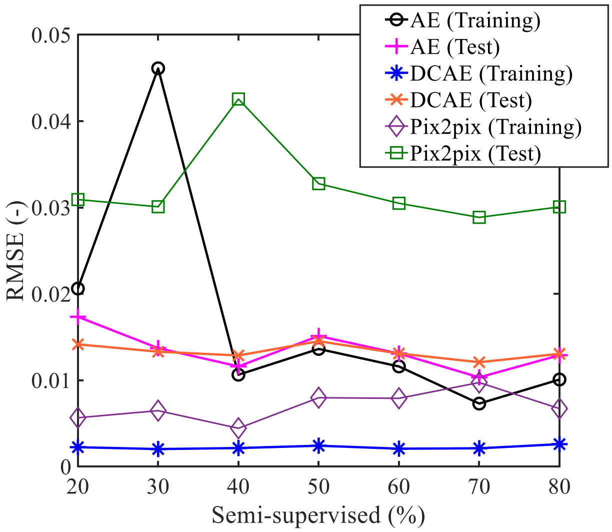


Fig. 10. RMSE of semi-supervised conditions.

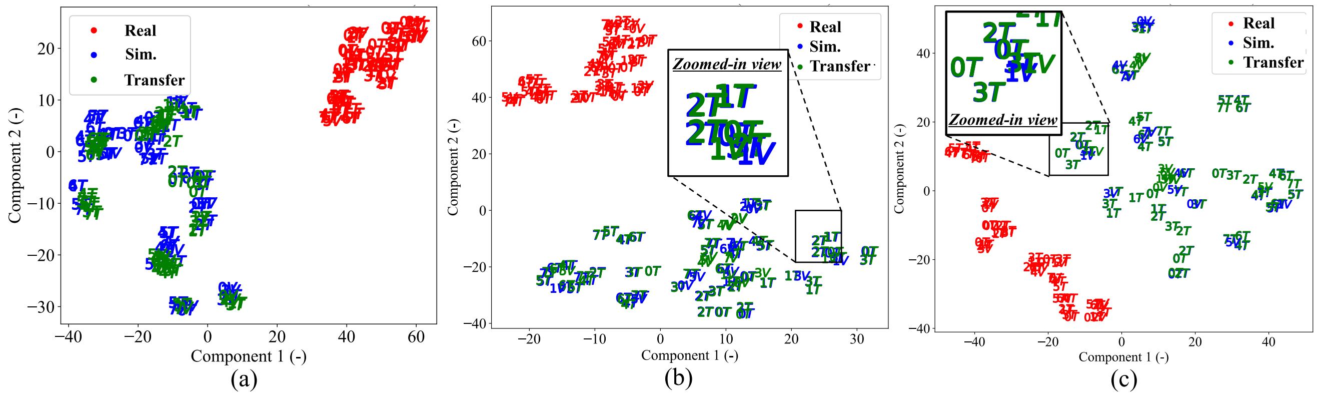


Fig. 11. Visualization of t-SNE. The subscripts “T” and “V” are the training data and test data, respectively. Visualization of transformed data with (a) AE, (b) DCAE, and (c) pix2pix. The numbers in the plots represent the classes in the classification problem, with eight classes of values from 0 to 7.

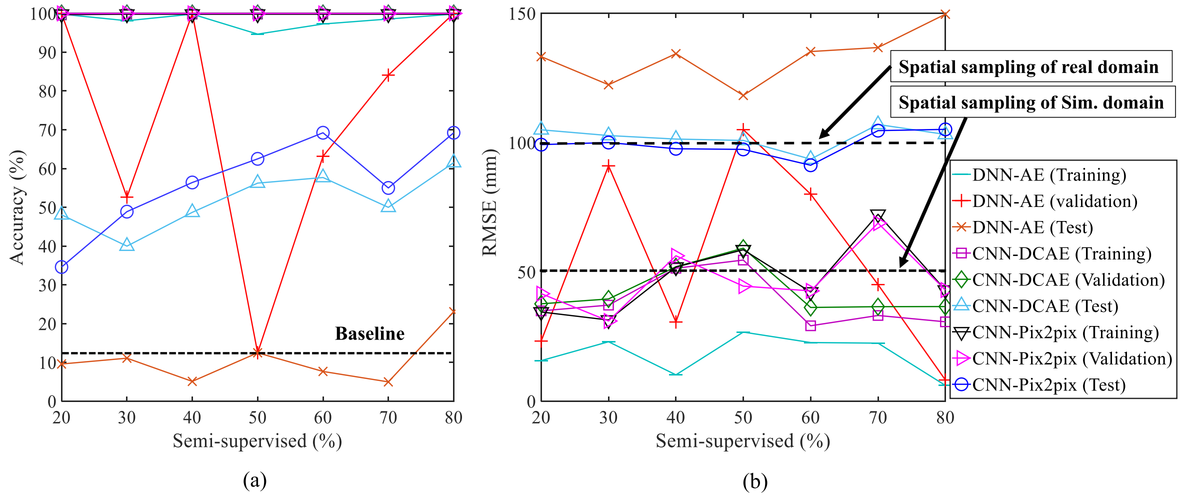


Fig. 12. Results of classification and regression problems. (a) Classification. (b) Regression. Legend follows SSL model - DT model.

TABLE VI
PROPOSED METHOD VS. CONVENTIONAL METHOD (CROSS-CORRELATION) AND NON-ADAPTATION

Prob. (Criteria)	Model or method	Training performance	Test performance
Class. (Acc.)	CC	-	-
	Non-adaptation	99.75%	15.38%
	DNN-AE	99.82%	23.07%
	CNN-DCAE	99.82%	61.54%
	CNN-Pix2pix	99.82%	69.23%
Reg.(RMSE)	CC	-	235.07
	Non-adaptation	128.18	141.94
	DNN-AE	6.12	149.67
	CNN-DCAE	30.72	103.17
	CNN-Pix2pix	42.98	105.13

The learning curves for the stability of each model are shown in the Appendix.

B. SSL Performance With SSL Model

The performance of the SSL model in terms of classification and regression is shown in Fig. 12. The training data given to the model are all simulation data, and the real data are semi-supervised; the validation data are semi-supervised real data, and the test data are the real data for testing. Although the validation data should be unknown, in this case, we use real training data to focus on learning domain-specific data in more detail. We use validation data to check for model overfitting on the simulation data because the simulation data are always larger than the real data. For the classification problem (Fig. 12(a)), all models are nearly 100% accurate for the training data but differed in accuracy for the validation and test data. The accuracies of CNN-DCAE and CNN-pix2pix for the validation data are 100% consistent. However, the accuracy of DNN-AE concerning the validation data is unstable. The SSL performance for the test data is similar for the CNN-pix2pix and CNN-DCAE. In this dataset, CNN-pix2pix is the highest and DNN-AE is the poorest and below the baseline. The baseline is the probability that a subvolume is selected for a random selection of eight classes. In this case, it is 20%. In the CNN-pix2pix and CNN-DCAE cases, the accuracy improves as the amount of semi-supervised data increases and exceeds the baseline in all conditions. In particular, the performance of pix2pix shows an accuracy of approximately 70% for a semi-supervised data rate of 80%. The accuracy without the DT model is 12% [5]; therefore, the improvement in the accuracy of the DT model is approximately 58%.

In the regression problem (Fig. 12(b)), the RMSE of DNN-AE for the training data is less than that of the spatial sampling of the sound sources in the simulation domain with high performance, but poor for the validation and test data. This result is similar to that obtained for the classification problem. The RMSEs of CNN-DCAE and CNN-pix2pix for the training and validation data are approximately equal to the spatial sampling of the sound source in the simulation domain. The RMSEs for the test data are almost identical for the CNN-DCAE and CNN-pix2pix. In this dataset, CNN-pix2pix has the lowest value. In addition, these values are close to the spatial sampling of the real domain. The RMSE without the DT model is 142 mm [5], whereas the performance improves to 100 mm when pix2pix is applied. These results indicate that the regression model is still underperforming and could require tuning the structure and hyperparameters of the CNN. However, tuning the model is inefficient, and it is considered more important for the DT model to directly learn the labels of the SSL model.

Table VI shows the proposed method, conventional method, and non-adaptation SSL performance. The CC in the table represents the conventional method (cross-correlation method), and nonadaptation is the result shown in our previous paper. Our proposed method shows results for the case with 80% ratio of semi-supervised data. The adaptation models are effective in both tasks: CNN-Pix2pix for the classification problem and CNN-DCAE for the regression problem. In our previous work, an SSL experiment using small amounts of data without data augmentation in the real-world domain failed to learn the SSL model, resulting in poor performance. However, the use of the DT model improves the SSL performance without applying data augmentation to real-environment data.

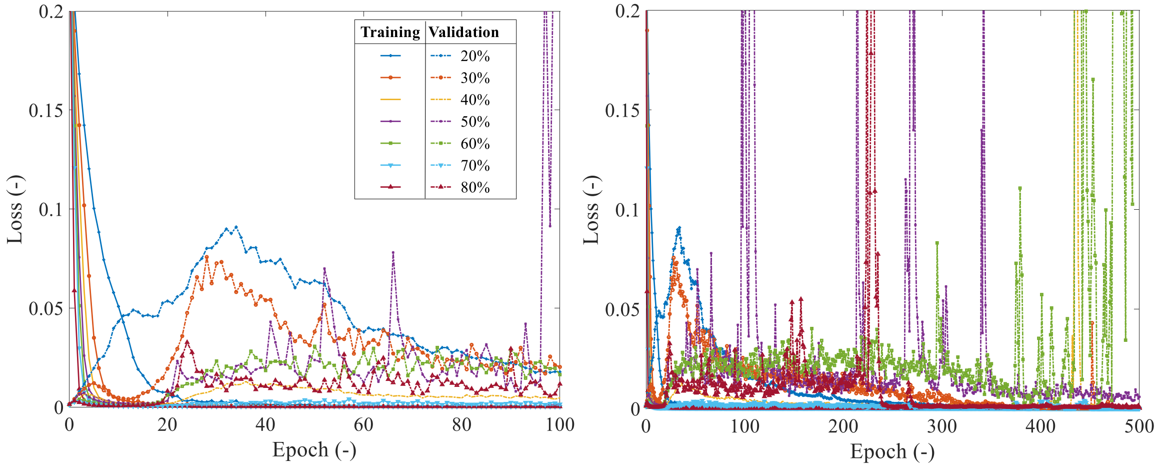
VIII. Conclusion

The proposed method of transforming real data into pseudo-simulation data using the DT model improves the performance of the SSL inside the structure. The 2-D distribution of t-SNE indicates that DCAE and pix2pix exhibit better transfer performance than AE for the FSA of the exterior of the structure. Both models incorporate a convolution-based layer, which is superior to a fully connected layer because it enables the learning of local features. This can be understood from the stability of the learning curves for both models. The SSL classification problem is less accurate when using the DT model with AE and more accurate when using DCAE or pix2pix than the baseline. In the test on the dataset used in this study, the overall accuracy is higher when pix2pix is used than when the DCAE is used. This seems to be because pix2pix utilizes binary cross-entropy as the loss function, whereas DCAE sets it as MSE. MSE cannot evaluate frequency response deviations, whereas binary cross-entropy is a simple metric that discriminates between true and real values. In particular, the performance of pix2pix shows an accuracy of approximately 70% for a semi-supervised data rate of 80%. The accuracy without the DT model is 12%; therefore, the improvement in the accuracy with the DT model is approximately 58%. Similarly, with the regression problem, the RMSE is lower when DCAE or pix2pix is used than when the AE is used. The RMSE without the DT model is 142 mm, whereas the performance improves to 100 mm when the DT model is applied. However, the RMSE of the SSL model using both transformations for the training data is approximately equal to the sound source spatial sampling of the simulation domain and requires further improvement. This indicates that the CNN structure and hyperparameters must be tuned.

Going forward, we aim to build a model that combines SSL and DT models. The proposed method separates the SSL model from the DT model, and the DT model cannot directly learn the discriminative bounds to solve the SSL problem. In other words, the DT model does not directly learn the domain transformations that are important for the SSL. Therefore, we aim to construct an SSL method that utilizes the discriminator of the GAN. Furthermore, we plan to develop a method for SSL under unsupervised conditions.

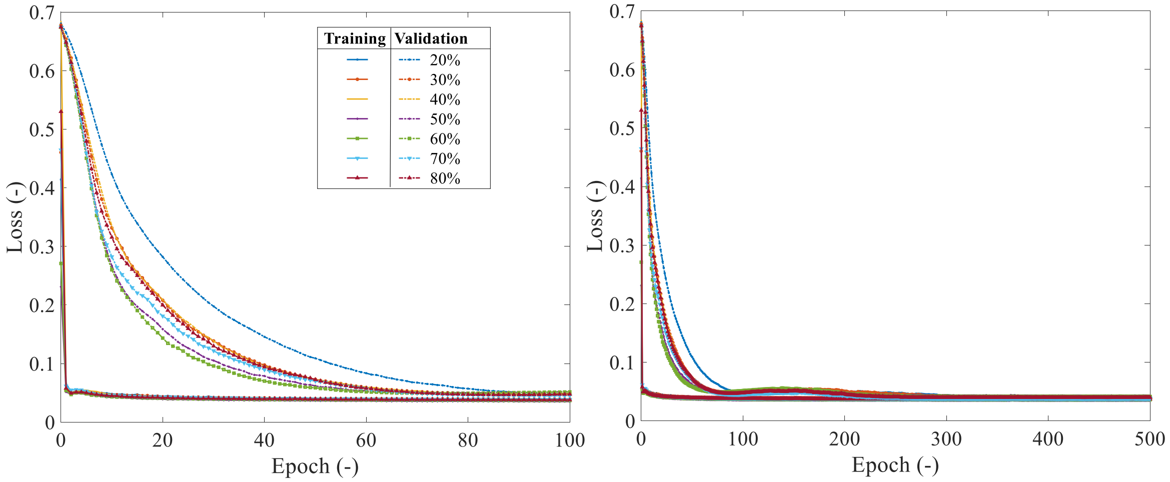
Appendix
Stability of Learning Curve for Each Model

Fig. 13 shows the learning curve of AE for each semi-supervised system. Fig. 13(a) and (b) show the learning curves up to 100 and 500 epochs, respectively. This figure shows that the loss to the training data tends to converge faster as the number of semi-supervisors increases. However, the loss of test data oscillates significantly. After 500 epochs, there is no convergence, and the amplitude of oscillations is larger. This indicates that the transformation by AE fails to learn the features. In contrast, the learning curve of DCAE is stable without loss of oscillation, unlike AE (Fig. 14). Similar to the AE, the number of epochs leading to convergence tends to decrease as the number of training data increases. These results demonstrate that DCAE using convolutional layers is effective for domain transformations. Fig. 15 shows the respective loss progress in adversarial training of the generator and discriminator. The solid blue line is the loss to the generator, and “G” represents the generator. The solid red line represents the loss associated with the discriminator, and its value is expressed as an average of the true and fake data. “D” represents the discriminator. During training for up to 500 epochs, both losses oscillate and converge. After 500 epochs, the losses of both models are close to each other, indicating that these models are in equilibrium (Fig. 15(a)). The loss of the test data is lower in the discriminator than in the generator, and the convergence is approximately 600 epochs. No mode collapse is observed in the output image of the generator.



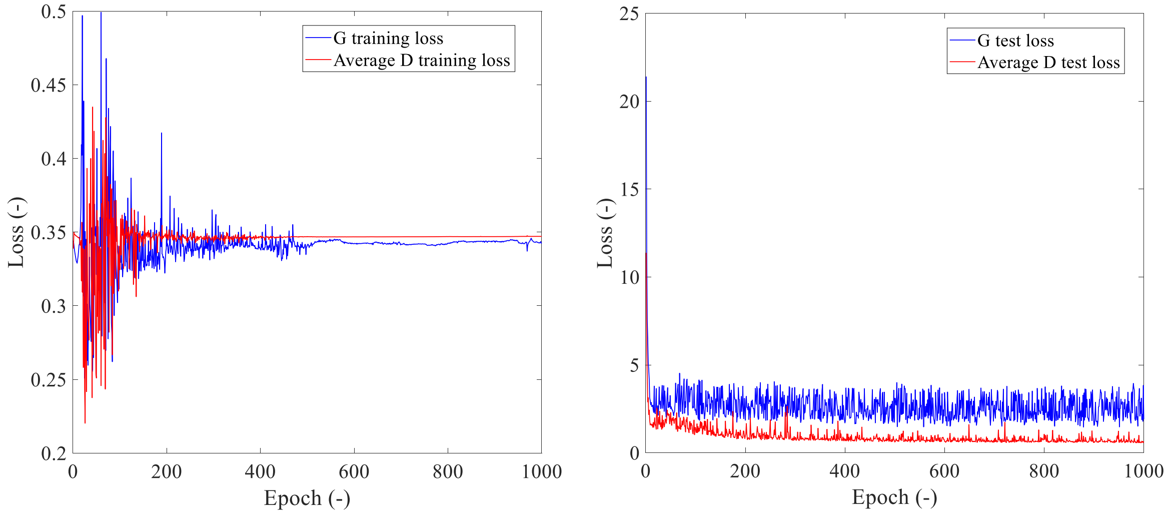
(a) (b)

Fig. 13. Learning curves of AE. (a) 0-100 epochs, (b) 0-500 epochs.



(a) (b)

Fig. 14. Learning curves of DCAE. (a) 0-100 epochs, (b) 0-500 epochs.



(a) (b)

Fig. 15. Learning curves of pix2pix. (a) Training. (b) Test.

References

[1] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 24, no. 4, pp. 320–327, Aug. 1976.

[2] G. C. Carter, "Coherence and time delay estimation," *Proc. IEEE*, vol. 75, no. 2, pp. 236–255, Feb. 1987.

[3] P.-A. Grumiaux, S. Kitić, L. Girin, and A. Guérin, "A survey of sound source localization with deep learning methods," *J. Acoust. Soc. Amer.*, vol. 152, no. 1, pp. 107–151, 2022.

[4] Z.-M. Liu, C. Zhang, and P. S. Yu, "Direction-of-arrival estimation based on deep neural networks with robustness to array imperfections," *IEEE Trans. Antennas Propag.*, vol. 66, no. 12, pp. 7315–7327, Dec. 2018.

[5] S. Kita and Y. Kajikawa, "Fundamental study on sound source localization inside a structure using a deep neural network and computer-aided engineering," *J. Sound Vib.*, vol. 513, 2021, Art. no. 116400.

[6] N. Poschadel, R. Hupke, S. Preihs, and J. Peissig, "Direction of arrival estimation of noisy speech using convolutional recurrent neural networks with higher-order ambisonics signals," in *Proc. IEEE 29th Eur. Signal Process. Conf.*, 2021, pp. 211–215.

[7] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *J. Stat. Plan. Inference*, vol. 90, no. 2, pp. 227–244, 2000.

[8] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera, "A unifying view on dataset shift in classification," *Pattern Recognit.*, vol. 45, no. 1, pp. 521–530, 2012.

[9] L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 5, pp. 1019–1034, May 2015.

[10] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer learning using computational intelligence: A survey," *Knowl.-Based Syst.*, vol. 80, pp. 14–23, 2015.

[11] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *Proc. 27th Int. Conf. Artif. Neural Netw.*, 2018, pp. 270–279.

[12] R. Takeda, Y. Kudo, K. Takashima, Y. Kitamura, and K. Komatani, "Unsupervised adaptation of neural networks for discriminative sound source localization with eliminative constraint," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 3514–3518.

[13] R. Takeda and K. Komatani, "Unsupervised adaptation of deep neural networks for sound source localization using entropy minimization," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2017, pp. 2217–2221.

[14] W. He, P. Motlicek, and J.-M. Odobez, "Neural network adaptation and data augmentation for multi-speaker direction-of-arrival estimation," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 29, pp. 1303–1317, 2021.

[15] G. Le Moing et al., "Data-efficient framework for real-world multiple sound source 2D localization," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 3425–3429.

[16] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[17] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J. Big Data*, vol. 3, no. 1, pp. 1–40, 2016.

[18] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[19] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning.* Cambridge, MA, USA: MIT Press, 2008.

[20] S. Kita and Y. Kajikawa, "Study on sound source localization inside a structure using a domain transfer model for real-world adaption of a trained model," in *Proc. Int. Congr. Noise Control Eng.*, 2022, pp. 1239–1248.

[21] J. Yang, W. An, S. Wang, X. Zhu, C. Yan, and J. Huang, "Label-driven reconstruction for domain adaptation in semantic segmentation," in *Proc. IEEE Eur. Conf. Comput. Vis.*, 2020, pp. 480–498.

[22] Ansys, "Mechanical APDL theory reference, release 18.2, 256–258," 2017, Accessed: Aug. 11, 2021. [Online]. Available: https://www.mm.bme.hu/gyebro/files/ans_help_v182/ans_thry/thy_acou2.html

[23] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[24] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Proc. 21st Int. Conf. Artif. Neural Netw.*, 2011, pp. 52–59.

[25] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1125–1134.

[26] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*. [Online]. Available: https://arxiv.org/pdf/1411.1784.pdf

[27] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[28] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE.," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

[29] C. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer, 2006.

[30] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3320–3328.

[31] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[32] T. Dare, "Experimental force reconstruction using a neural network and simulated training data," in *Proc. Int. Congr. Noise Control Eng.*, 2020, pp. 5682–5688.

[33] T. Dare, "Experimental force reconstruction on plates of arbitrary shape using neural networks," in *Proc. Int. Congr. Noise Control Eng.*, 2021, pp. 3407–3416.

[34] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 13001–13008.

[35] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," 2017, *arXiv:1708.04552*. [Online]. Available: https://arxiv.org/pdf/1708.04552.pdf

[36] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, 2015, pp. 234–241.

[37] D. Lee, T.-S. Ahn, and H.-S. Kim, "A metric on the similarity between two frequency response functions," *J. Sound Vib.*, vol. 436, pp. 32–45, 2018.

Shunsuke Kita (Graduate Student Member, IEEE) received the M.E. degree from Kansai University, Osaka, Japan, in 2011. He is currently with the Osaka Research Institute of Industrial Science and Technology (ORIST), Izumi, Japan, the research division of electronic and mechanical systems. His research interests include machine learning and acoustic-structural coupled analytical methods.

Yoshinobu Kajikawa (Senior Member, IEEE) received the B.Eng. and M.Eng. degrees in electrical engineering from Kansai University, Osaka, Japan, in 1991 and 1993, respectively, and the D.Eng. degree in communication engineering from Osaka University, Osaka, in 1997. In 1993, he joined Fujitsu Ltd., Kawasaki, Japan, and engaged in research on active noise control. In 1994, he joined Kansai University, where he is currently a Professor. He has authored or co-authored more than 200 articles in journals and conference proceedings and has more than ten patents. His research interests include signal processing in audio and acoustic systems. He is a Fellow of IEICE and Member of APSIPA, the Acoustical Society of America, and ASJ. He is also the President-Elect of IEICE Engineering and Science Society, a Members-at-Large of APSIPA, and Member of the Applied Signal Processing Systems TC of IEEE SPS. He is an Associate Editor for *IET Signal Processing* and *Applied Sciences*. He was the Editor-in-Chief of *IEICE Transactions on Fundamentals of Electronics, Communications, and Computer Sciences*. He was the recipient of the 2012 Sato Prize Paper Award from the Acoustical Society of Japan, Best Paper Award in APCCAS 2014, 2017 Sadaoki Furui Prize Paper Award from the Asia Pacific Signal and Information Processing Association, and 2019 Best Paper Award from IEICE.