Impact of data augmentation on supervised learning for a moving mid-frequency source

J. A. Castro-Correa, M. Badiey, T. B. Neilsen, et al.

Citation: The Journal of the Acoustical Society of America **150**, 3914 (2021); doi: 10.1121/10.0007284
View online: https://doi.org/10.1121/10.0007284
View Table of Contents: https://asa.scitation.org/toc/jas/150/5
Published by the Acoustical Society of America

JASA ARTICLE

Impact of data augmentation on supervised learning for a moving mid-frequency source

J. A. Castro-Correa,[1,a)] M. Badiey,[1,b)] T. B. Neilsen,[2,c)] D. P. Knobles,[3] and W. S. Hodgkiss[4]
[1]*Department of Electrical and Computer Engineering, University of Delaware, Newark, Delaware 19716, USA*
[2]*Department of Physics and Astronomy, Brigham Young University, Provo, Utah 84602, USA*
[3]*Knobles Scientific and Analysis, LLC, Austin, Texas 78731, USA*
[4]*Marine Physical Laboratory, Scripps Institution of Oceanography, University of California, San Diego, La Jolla, California 92093, USA*

ABSTRACT:
Two residual networks are implemented to perform regression for the source localization and environment classification using a moving mid-frequency source, recorded during the Seabed Characterization Experiment in 2017. The first model implements only the classification for inferring the seabed type, and the second model uses regression to estimate the source localization parameters. The training is performed using synthetic data generated by the ORCA normal mode model. The architectures are tested on both the measured field and simulated data with variations in the sound speed profile and seabed mismatch. Additionally, nine data augmentation techniques are implemented to study their effect on the network predictions. The metrics used to quantify the network performance are the root mean square error for regression and accuracy for seabed classification. The models report consistent results for the source localization estimation and accuracy above 65% in the worst-case scenario for the seabed classification. From the data augmentation study, the results show that the more complex transformations, such as time warping, time masking, frequency masking, and a combination of these techniques, yield significant improvement of the results using both the simulated and measured data. © *2021 Acoustical Society of America.*
https://doi.org/10.1121/10.0007284
(Received 30 June 2021; revised 26 October 2021; accepted 30 October 2021; published online 22 November 2021)
[Editor: Zoi-Heleni Michalopoulou] Pages: 3914–3928

I. INTRODUCTION

In ocean acoustics, a proper characterization of the environment is relevant to study the acoustic propagation under water. Because of the number of intrinsic parameters of the sediment and water column, the seabed characterization is primarily performed through geoacoustic inversions,[1] which use *prior* information about the environment along with generic algorithms. For instance, simulated annealing,[2,3] Gibbs sampling,[4] or maximum entropy[5] use prior knowledge about the waveguide to estimate the geophysical parameters based on the marginal distributions. Once the environment properties are defined, it is possible to estimate the location of the acoustic moving sources to perform tasks such as signal enhancement or underwater navigation. However, fluctuations in the environment make the localization and tracking of the moving sources difficult.[6] These source localization tasks are tackled using diverse approaches, such as match field processing (MFP),[7,8] multipath arrival estimations,[9,10] array waveguide invariant,[11,12] and Bayesian methods.[6] In addition, recent works present evidence of how deep learning models, such as convolutional neural networks (CNNs), can be used effectively to estimate the source localization parameters[13–15] while performing the seabed classification.[16] Here, these tasks are performed with a residual network (ResNet),[17] and the advantages of using data augmentation during the training on mid-frequency spectral levels are presented.

Recently, deep learning has become a popular approach to address numerous types of problems in different areas.[18–21] The deep learning models can learn patterns and features from the data with multiple levels of abstraction with the purpose of performing tasks such as feature extraction,[22] classification,[23] or regression.[18] These deep neural networks[24,25] perform better when larger datasets are used for the training.[26,27] However, vanishing or exploding gradients impede the convergence in the deep neural networks. This problem has been addressed by using normalization either at the beginning or intermediate layers,[24,28–30] allowing the networks to converge toward a global minimum using the gradient-based optimizations. Nevertheless, the aforementioned solution fails when very deep learning models are used. A large number of layers precludes updating the parameters in the first layers, yielding larger errors in the predictions.

ResNet algorithms are a type of neural network that includes extra components and solves the vanishing gradient problem.[17] ResNet architectures incorporate extra *residual* or *skip* connections, linking the convolutional layers.

[a)]Electronic mail: jcastro@udel.edu, ORCID: 0000-0002-2507-3535.
[b)]ORCID: 0000-0002-5869-336X.
[c)]ORCID: 0000-0002-9729-373X.

https://doi.org/10.1121/10.0007284

These residual connections do not add any learnable parameter or complexity to the model. The addition of such connections allows the gradients to flow through the network directly during the backward pass without going through the nonlinear activation functions. The nonlinear activation functions, by nature, are nonlinear and cause the gradients to explode or vanish.

Besides a powerful model to classify or predict, a good dataset is also needed to train a learning model. Because a neural network learns from the data and aims to generalize for any possible scenario, it is necessary to use a dataset with enough variability, which can best represent the environment. When the dataset is not large enough, the accuracy of the networks can be compromised. One of the most well-known approaches to address the lack of training samples and avoid overfitting is data augmentation, a regularization method that adds variability to the dataset. This concept refers to the application of random transformations to the dataset and is highly used in machine learning for image and speech recognition.[31–33] Not long ago, data augmentation techniques had been extended also to two-dimensional (2D) data spectrograms to study their incidence for the training of the learning models.[34–36]

It is because of such advances that deep learning algorithms have been implemented for both regression[37] and classification[38,39] in the underwater acoustics field. Bianco *et al.*[40] presented some insights about the different techniques and machine learning applications in ocean acoustics. Frederick *et al.*[41] implemented several learning algorithms for sediment classification in one-dimensional data, where the most complex models, such as CNN and ResNet, produced the best results. Later, Liu *et al.*[42] used a CNN for seabed classification. In their work, the CNN outperformed the conventional model-based methods for estimating the source depth and range of the emitting source. Recently, Neilsen *et al.*[43] implemented a CNN for both source localization and seabed classification using spectrograms from a moving mid-frequency source. In their approach, they used multitask learning for inferring the different targets and revealed some insights in the implications of testing such a model under variations in the sound speed profiles (SSPs) and seabed type.

This paper has two objectives: to show that ResNet is a viable method to estimate source localization parameters and classify seabed types and demonstrate that data augmentation can improve the ResNet performance by studying their effect of the predictions using mid-frequency spectrograms. Here, a proof of concept of deep learning algorithms for the source localization and environment classification applied to the mid-frequency towed tonal spectrograms is presented and also addressed in Ref. 43. The same experiments that were proposed in Neilsen *et al.*[43] are used in this work but they differ in the implementation of ResNet for source localization and seabed classification altogether as the use of data augmentation as a regularization technique to improve network performance. ResNets are applied to both the synthetic and at-sea data collected from the Seabed Characterization Experiment 2017 (SBCEX 2017). Two ResNet algorithms are implemented in this work. One algorithm is used to predict the source localization parameters, which are the source depth (z_s), closest point of approach or CPA range (r_{CPA}), and ship speed (v_{ship}); and a second algorithm is used to classify between the four canonical environments.[43] In addition, different data augmentation techniques are applied to the data during the training stage to improve the network performance on the learning of both the regression and classification parameters. The results show a favorable potential of the residual-based deep learning models to differentiate the seabed types and the source position based on the mid-frequency spectrograms and the benefits of using data augmentations during the training of the ResNet.

This manuscript is structured as follows. Section II presents the details of the experiment and measured data. Section III introduces the synthetic data used for the training and testing. The ResNet-18 architecture, as well as the data augmentation techniques used, are given in Sec. IV. Section V summarizes the results and discussion, which is followed by the conclusions in Sec. VI.

II. EXPERIMENTAL DATA FROM THE SBCEX 2017

The data used in this work were obtained from the SBCEX 2017 experiment which was performed in the New England Mud Patch region. The objective of the SBCEX 2017 was to infer the geoacoustic properties of a surface sediment layer composed of fine-grained mud in the 0.01–10 kHz frequency band.[44] Several of the analyses used short- and long-range propagation acoustic data as presented in Ref. 44. The data shown in Figs. 4(d)–4(f) were collected using the two vertical line arrays (VLAs) displayed in Fig. 1. The geographical positions of VLA1 and VLA2 were about 40° 28.207′ N 70° 35.8266′ W and 40° 26.5073′ N 70° 31.6299′ W, respectively, with an approximate distance of 6.7 km between them. During a portion of the SBCEX 2017, an ITC 2015 transducer was towed while emitting continuous sound waves (CW) every 0.5 s at frequencies of

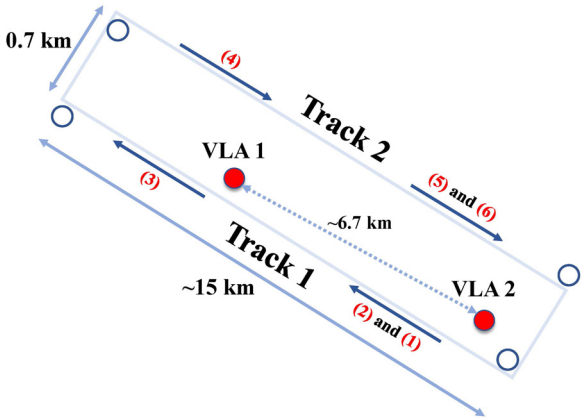

FIG. 1. (Color online) The tracks followed by the R/V Endeavor during the SBCEX 2017 on Julian day 83 from 00:56 to 07:32 UTC. The distance between VLA1 and VLA2 is about 6.7 km.

2, 2.5, 3, 3.5, and 4 kHz. This towed source experiment was made along a rectangular path with the two arrays positioned within a rectangle, whose perimeter was approximately 32 km, with a nominal source depth of 45 m and a ship speed of 3 kn. The tracks followed by the Research Vessel (R/V) Endeavor during the portion of the experiment that provides the data used in this work are depicted in Fig. 1.

Six spectrograms were generated from the data collected during a section of the experiment on Julian Day 83 from 00:56 to 07:32 UTC when the R/V Endeavor passed close to VLA1 and VLA2, following tracks 1 and 2, respectively, as shown in Fig. 1. The length of each of the tracks was 15 km, whereas the extracted portions of the tracks used to generate the spectrograms were approximately 6.83 km long. Four of the samples were recorded at VLA1 when the ship was moving *away* (#3) and *toward* (#2) the VLA in track 1 and at the time the vessel was moving *away* (#5) and *toward* (#4) the VLA but following track 2. The last two samples were measured at VLA2 when the ship moved *away* (#1) in track 1 and traveled *toward* (#6) the VLA following track 2 (each numbered event is depicted in Fig. 1). These six samples were extracted based on the CPA range between the R/V Endeavor and each VLA as sketched in Figs. 1 and 2. That is, the first time step of each sample corresponds to the CPA (r_{CPA}). The CPA range of each data sample was estimated using the Doppler shift as described in Sec. II B of Ref. 43, the ship speed was retrieved from the Global Positioning System (GPS) data, and the source depth was measured using a pressure sensor during the experiment. The spectrograms were generated using a window of 1 s and 50% overlap and smoothed using a spline fit interpolation because of the high variation in the tonal levels.[43]

Each resulting data sample is arranged in $1 \times 80 \times 8850$ sized spectrograms, whose dimensions correspond to the number of input channels, frequencies per hydrophone (5 frequencies $\times$ 16 receivers), and time steps. The extracted spectrograms cover 4425 s each (approximately 74 min) as the time steps are sampled at 0.5 s intervals, i.e., 8850×0.5 s $= 4425$ s. These six data samples are used to test the ResNet algorithms in this work.

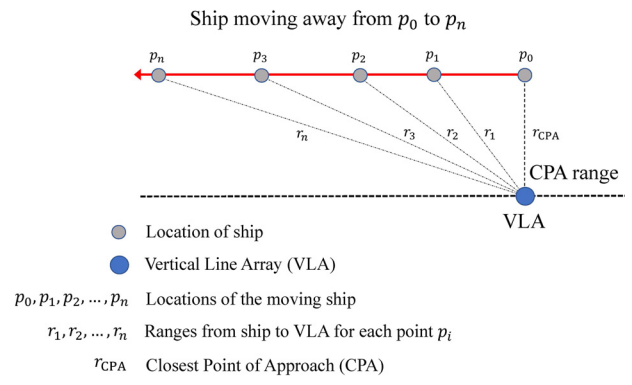

FIG. 2. (Color online) A schematic showing an acoustic source moving from right to left (p_0 to p_n) relative to a vertical line array (VLA) depicting the CPA range. The points $p_1, p_2, \ldots, p_n$ show the different positions relative to a VLA when the ship is moving along the red track.

III. SYNTHETIC DATA USING ORCA

The synthetic data are employed to train and test the ResNet algorithms implemented here. These synthetic data are generated using the range-independent ORCA normal mode model[45] and experimental geometry in the SBCEX 2017.

The data are generated considering a scenario in which 16 receivers placed in a vertical array recorded the sound emitted by a towed source broadcasting CW tones at 2, 2.5, 3, 3.5, and 4 kHz every 0.5 s. The simulation variables involved are presented in Table I. The receiver depths, frequency, and water depth are fixed during the simulation, whereas the other parameters varied inside the limits presented in Table I. To ensure that the parameter space is sampled evenly, the data generation is performed for n_r random and n_s specified values (equally spaced between the minimum and maximum values). For the source depth z_s, $n_r = 10$ and $n_s = 10$; in the case of the ship speed v_{ship}, $n_r = 5$ and $n_s = 5$, whereas for the CPA range r_{CPA}, $n_r = 5$ and $n_s = 6$, as shown in Table I.

For each of the combinations of the source parameters z_s, v_{ship} and r_{CPA}, the channel response is calculated for three out of the ten different SSPs collected during the SBCEX 2017 and four canonical environments. The SSP profiles used to generate the training data and the synthetic test cases A–D related to the SSPs mismatch are shown in Fig. 3. The seabed types used are (#1) the *deep mud* reported by Knobles *et al.*;[46] (#2) the *mud over sand* seabed, inferred during the SBCEX 2017 (Ref. 47); (#3) the *sandy silt* environment from the New England Bight and reported by Potty *et al.*;[48] and (#4) the *sand* sediment, obtained by Zhou *et al.* in 2009.[49] The four seabeds are sorted from highest to lowest bottom loss to create labels for the machine learning algorithm. Additional details and the geoacoustic models are presented in Fig. 5 in Ref. 43. In total, the combination of source parameters, seabed types, and SSPs yields 6600 data samples, where 3000 are generated using n_r random parameters and 3600 are generated with n_s specified values according to Table I. The spectrograms have a dimension of $1 \times 80 \times 8850$, corresponding to the number of input channels, frequencies $\times$ hydrophones, and time steps.

TABLE I. The parameters used in ORCA for the generation of the training data. For z_s, v_{ship}, and r_{CPA}, ($\mathbf{n}_r$) and (n_s) show the specified and random parameter values, respectively, used in the simulation.

Parameter	Value
Receivers depth	15–71.25 m (at 3.75 m spacing)
Frequency	2, 2.5, 3, 3.5, and 4 kHz
Water depth	75 m
Source depth (z_s)	5–65 m · (**10**) (*10*)
Ship speed (v_{ship})	1–5 kn · (**5**) (*5*)
CPA range (r_{CPA})	100–1100 m · (**5**) (*6*)
Seabed types	(1) deep mud, (2) mud over sand, (3) sandy silt, and (4) sand

https://doi.org/10.1121/10.0007284

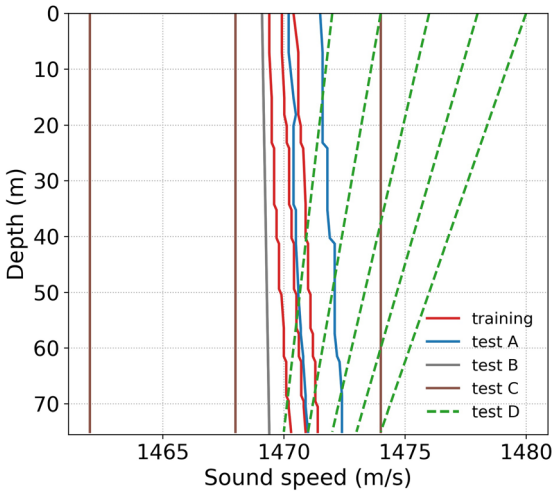

FIG. 3. (Color online) The SSPs used for the synthetic training and testing datasets. Adapted from Ref. 43.

Some of the extracted samples for both the synthetic and measured data are presented in Fig. 4. The top row in Fig. 4 corresponds to the data generated using the ORCA model for one of the source depths z_s (in meters), CPA ranges r_{CPA} (in meters), and ship speeds v_{ship} (in knots) listed in Table I and one of the four aforementioned seabed types. The bottom row in Fig. 4 shows three of the six measured samples from the experiment for the case of *moving away* from VLA1 (track 1), *moving toward* VLA2 (track 2), and *moving toward* VLA1 (track 2). The vertical axes on the spectrograms correspond to the frequency channels at 2, 2.5, 3, 3.5, and 4 kHz for each of the 16 hydrophones (5×16), whereas the horizontal axis shows the signal arrival time in seconds. The spectrograms used to train the machine learning models are given in dB ref 1 μPa and are not scaled or normalized.

IV. METHODOLOGY

A. ResNets

The ResNets are inspired by the structure of the visual systems. These deep learning architectures incorporate extra components called *residual connections*, which prevent the vanishing gradient issue exhibited in the deep neural networks.[17] Additionally, clear empirical evidence shows that training with residual connections accelerates the training time of the networks significantly in comparison with the current architectures.[17,50] As in other deep learning approaches,[51] the ResNet algorithms detect and learn patterns from the data by using different components such as kernels, nonlinearity/activation functions, batch normalization, and pooling layers. Formally, to initiate the training, any supervised learning model requires data that are generated according to some probability distribution P over a domain set $\mathcal{X}$, and that can be labeled according to a label set $\mathcal{Y}$. Thus, a dataset is defined as $Z^n = \{(x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$, and n is the number of samples in the dataset used to update the set of weights or parameters W during the learning process. In this manner, each data sample $x_i \in \mathbb{R}^{C_s \times H_s \times W_s}$, where C_s denotes the input channels, H_s is the height, and W_s is the width of the spectrogram. In the case of regression, each prediction is given by $y_i \in \mathbb{R}^{1\times d}$, where d denotes the number of targets for the regression task. Similarly, for the classification tasks, each element in the label set is defined as $y_i \in \Phi = \{\phi_1, \ldots, \phi_j\}_{j=1}^m$, where ϕ_j corresponds to a specific class. For both the prediction and classification tasks, any learner h (such as CNN or ResNet) aims to map the training set successfully with the label set, i.e., $h : \mathcal{X} \to \mathcal{Y}$, where h belongs to a hypothesis set $\mathcal{H}$.

The ResNet algorithms are multistage architectures that do not learn from the data by the conventional underlying

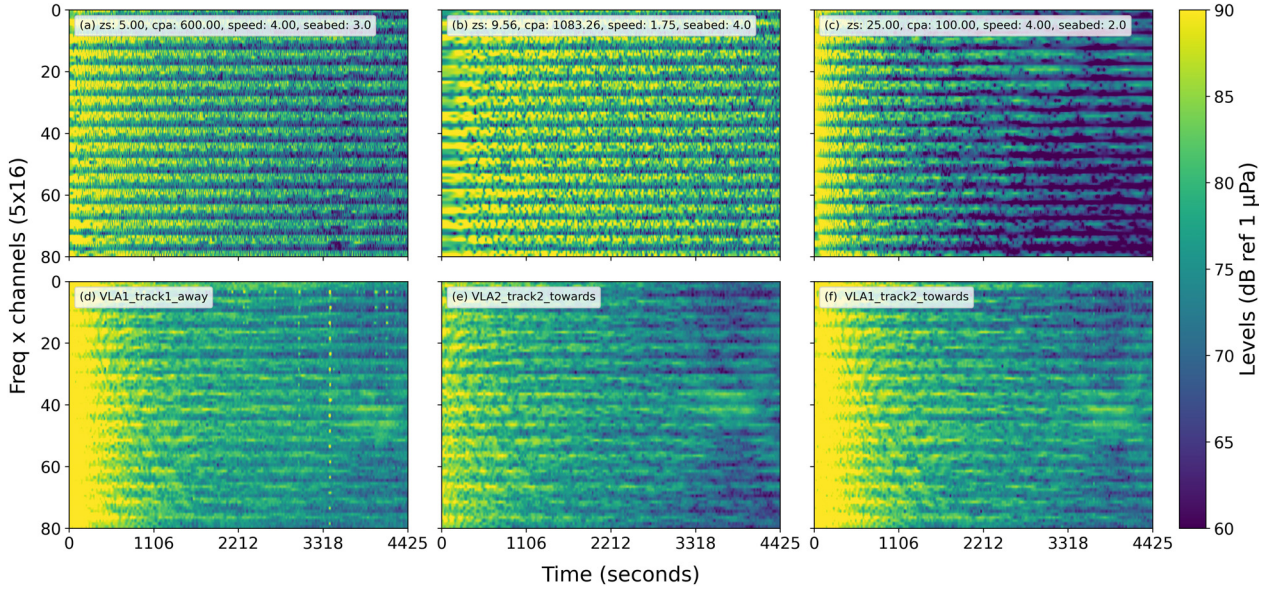

FIG. 4. (Color online) Samples of the synthetic data and measured spectrograms. The top panels [(a)–(c)] present the simulated data generated using ORCA for different z_s, v_{ship}, r_{CPA}, and seabed type. The bottom panels show the measured data samples from the SBCEX 2017 when the R/V Endeavor was (d) moving *away* from VLA1 in track 1, (e) moving *toward* VLA2 in track 2, and (f) moving *toward* VLA1 in track 2.

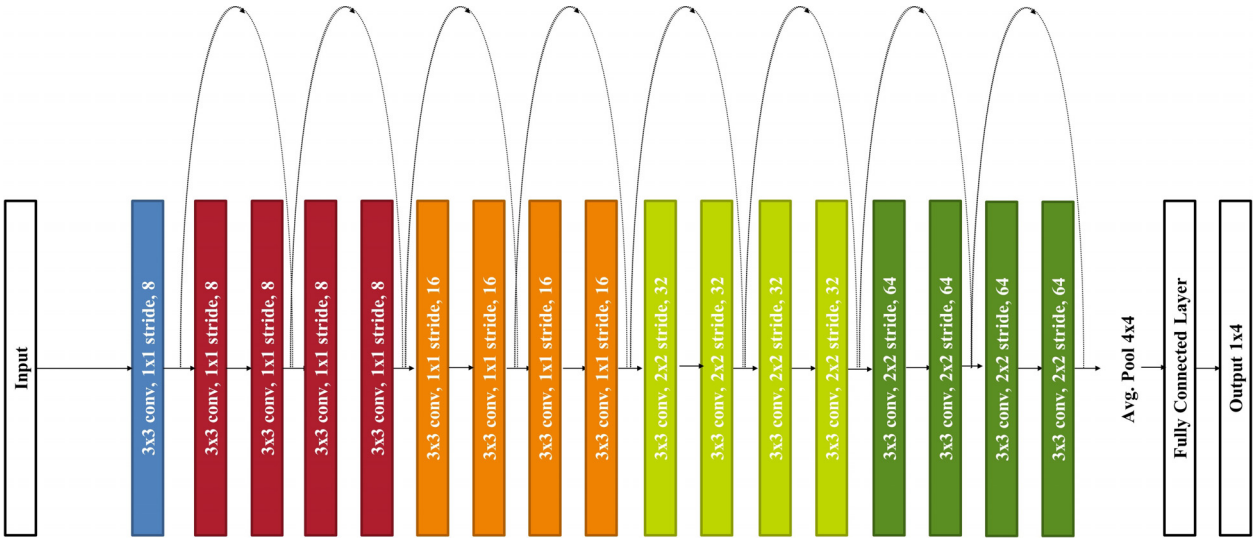

FIG. 5. (Color online) The ResNet-18 architecture used to classify the seabeds and predict the source location by means of the towed tonal spectrograms is shown. The colors represent the distinct *basic blocks* in the architecture. Each curved line represents a *residual connection*. Two different models are implemented with minor changes in the output layer depending on whether classification or regression is used.

mapping found in the plain networks (such as CNN) but from a residual mapping produced by the extra residual connections. In the plain networks, the conventional underlying mapping is denoted as $H^{\ell}(x_i)$, which is the consequence of applying an input x_i to the layer ℓ. The addition of the residual connections in the ResNet algorithms changes the way that the network learns. In this case, instead of fitting the underlying mapping $H^{\ell}(x_i)$, the residual mapping $F^{\ell}(x_i)$ is fitted. Equation (1) shows the relationship between the residual and underlying mappings,

$$F^{\ell}(x_i) := H^{\ell}(x_i) - x_i, \tag{1}$$

where x_i is the original input, and $H^{\ell}(x_i)$ denotes the underlying mapping at the layer ℓ. In Eq. (1), the original mapping is recast into $F^{\ell}(x_i) + x_i$. Theoretically, training in the ResNet is faster due to the fact that if $H^{\ell}(x_i)$ is an optimal mapping, then the residual mapping $F^{\ell}(x_i)$ is pushed to zero and makes the learning of the residuals easier. If $H^{\ell}(x_i)$ is far from x_i, then the residual mapping is similar to those found in the conventional plain networks.[17]

In this work, 18-layer ResNet architectures (ResNet-18), as the one shown in Fig. 5, are implemented. The ResNet-18 depicted in Fig. 5 consists, at first, of a convolutional layer with eight kernels (convolutional filters) of size 3×3 and a stride (horizontal and vertical displacement of the kernel) of 1×1. Subsequently, the architecture incorporates 8 *basic blocks* that yield 16 convolutional layers (each basic block adds a residual connection at the top; see Fig. 5). For these layers, the number of kernels is duplicated every two basic blocks as presented in Fig. 5. The first 2 blocks in red have 8 kernels, the next blocks in orange have 16 kernels, the blocks in the light green color have 32 kernels, and the last two blocks in green have 64 kernels. The kernel size in all of the blocks is 3×3, whereas the stride is 1×1 for the first four blocks and 2×2 for the last blocks. After the basic blocks, an average pooling of size 4×4 is applied to the feature map to reduce the dimensionality of the model. Finally, a fully connected (FC) layer is used to generate the desired outputs for the regression or classification. The Softmax function (normalized exponential function) is applied at the output layer but only for the classification task. Notice that each of the convolutional layers of this model is followed by a rectified linear activation function (ReLU) and a batch normalization layer. Table II summarizes the components of the 18-layer ResNets implemented here. The layer conv1 refers to the blue layer in Fig. 5. Likewise, the layers conv2_x, conv3_x, conv4_x, and conv5_x are represented by the colors red, orange, light green, and dark green in Fig. 5.

As mentioned earlier, the ResNet-18 is used to solve the problem presented in Ref. 43, which combines the source localization and seabed classification tasks. The input of the

TABLE II. The ResNet-18 architecture for the regression and classification. The description column includes the [kernel size, number of kernels], $\times 2$ if the entire block is completed twice, and the stride.

Layer name	Description
Conv1	$[3 \times 3,\ 8]$, stride 1
Conv2_x	$\begin{bmatrix} 3 \times 3,\ 8 \\ 3 \times 3,\ 8 \end{bmatrix} \times 2$, stride 1
Conv3_x	$\begin{bmatrix} 3 \times 3,\ 16 \\ 3 \times 3,\ 16 \end{bmatrix} \times 2$, stride 2
Conv4_x	$\begin{bmatrix} 3 \times 3,\ 32 \\ 3 \times 3,\ 32 \end{bmatrix} \times 2$, stride 2
Conv5_x	$\begin{bmatrix} 3 \times 3,\ 64 \\ 3 \times 3,\ 64 \end{bmatrix} \times 2$, stride 2
Pool1	Average pooling, 4×4
Flatten layer	4353 features
Output layer	$\begin{cases} 3-\text{d FC layer (regression)} \\ 4-\text{d FC layer} + \text{Softmax (classication)} \end{cases}$

https://doi.org/10.1121/10.0007284

ResNet-18 architecture presented in Fig. 5 consists of spectrograms of size $1 \times 80 \times 1107$ (after downsampling and smoothing the raw spectrograms as explained in Sec. V A). The output of the network corresponds to the seabed types (for only the classification) and source localization parameters (for only the regression). The target variables are the source depth z_s, the CPA range r_{CPA}, the ship speed v_{ship}, and the seabed type.

Here, two ResNet-18 architectures are used. The first network is employed for inferring only the seabed type via classification, whereas the second ResNet-18 performs regression over the three source localization parameters (z_s, r_{CPA}, and v_{ship}). As described in Sec. III, the seabed types are ordered depending on their bottom loss, which states a relationship in the sound speed ratio r_c, as explained in Ref. 43.

B. Data augmentation

One of the main problems in ocean acoustics is the lack of adequate labeled training data that help networks to learn the important features in a manner that allows better generalization in realistic scenarios. To solve this problem, data augmentation is used to add variability to the data by applying small random transformations to individual data samples at each epoch. Data augmentation is a regularization and preprocessing technique used to improve the performance and generalization capabilities of the machine learning models.[52] One of the classical approaches to implement data augmentation into a dataset is to apply a small amount of random noise to each sample;[53] other authors use different transformations such as rotation, scaling, cropping, and translation.[54] In this study, the effect on the training process of nine different data augmentation techniques is investigated using the ResNet-18 architectures. The data augmentation techniques addressed in this work are the dropout of some of the points, the addition of uniform noise, time stretching, addition or subtraction of dB levels to the spectrogram, flipping, random time warping, time and frequency masking, and a combination of time warping with time and frequency masking. Each one of the transformations results in visible changes of the spectrograms, as presented in Fig. 6. The nine data augmentation techniques are explained briefly below.

(1) **Dropout**: Dropout is commonly used to prevent overfitting during the training by having the network learn how to ignore anomalies in the data samples.[55] Here, a batch is chosen from the dataset with a probability of $p \in [0, 1]$. Then, a fixed percentage of the points $q \in [0, 100]\%$ of each sample is randomly dropped out to zero, as shown in Fig. 6(b). This augmentation can be related to the missing points in the spectrogram or background noise presented during the at-sea experiments.

(2) **Uniform noise**: The incorporation of this type of augmentation emulates potential noise-related artifacts on *in situ* measurements. In this case, a batch from the data is selected with a probability p. Then, random noise drawn from a uniform distribution U is applied to each spectrogram in the batch at every epoch. A random matrix $H \sim U(\mu, \sigma)$ with mean $\mu = 0$ and standard deviation $\sigma = 1$ is used here. The random noise is calculated by multiplying a scalar

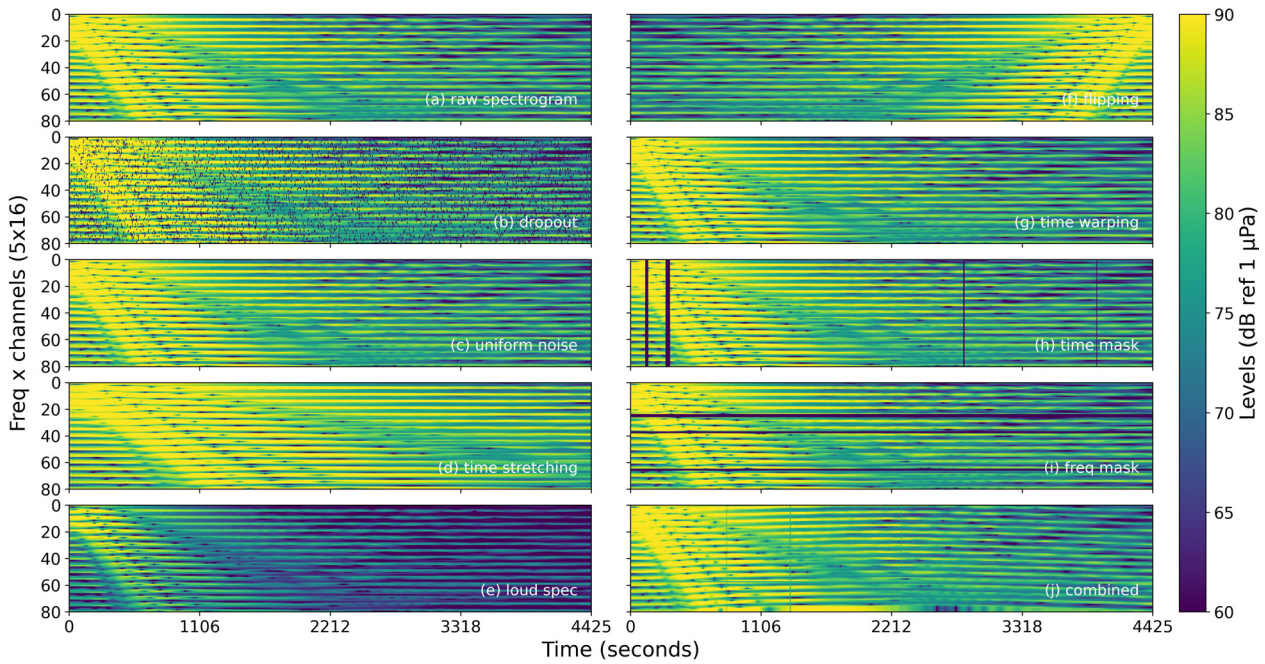

FIG. 6. (Color online) Examples of the data augmentation applied to the synthetic data sample with $z_s = 5$ m, $r_{\text{CPA}} = 100$ m, and speed $v_{\text{ship}} = 1$ kn using seabed #1. The (a) raw sample from the training data with no data augmentation included; (b) dropout with $q = 0.05$; (c) uniform noise, $H \sim U(0, 1)$ with $c = 60$ dB; (d) time stretching with $d = 2$; (e) loudness augmentation by subtracting $c = 10$ dB to the spectrogram; (f) vertical flipping; (g) time warping with $\tau = 50$; (h) time masking with $w_t = 20$ and $n_t = 5$; (i) frequency masking with $w_f = 5$ and $n_f = 5$; and (j) a combination of (g), (i), and (h) with $\tau = 25, w_f = 5, w_t = 20, n_f = 5$, and $n_t = 5$.

factor c (in dB) with the matrix H. Thus, the noise level for each spectrogram is $N_{\text{noise}} = c \times H$ dB. Figure 6(c) presents a spectrogram with $c = 60$ dB.

(3) **Time stretching**: Time stretching is the process of changing the speed or duration of an audio signal without affecting its pitch. Such alterations in the data can be explained by changes in the characteristics of the waveguide propagation. As was done with the previous transformations, a batch is selected with a probability p. Then, the samples are stretched by a constant factor d and cropped to fit in the network. For convenience, d is set to $d \geq 1$ to ensure that the modified spectrogram has the original size without using any padding. Figure 6(d) displays the stretched spectrogram with a factor of $d = 2$.

(4) **Loudness**: To study the changes in the predictions for the CPA range (r_{CPA}), which causes the levels of the signal to change, loudness was also used as a data transformation. In acoustics, loudness is related to the sound pressure levels. The changes produced by this augmentation are explained by the variations in the acoustic channel, distances, or failures in the source or receiver. To implement this transformation, a batch is picked up randomly with a probability $p \in [0, 1]$. Then, a constant level $\pm c$ in dB is added or subtracted (with the same probability) to each of the spectrograms in the batch. In Fig. 6(e), a constant level of $c = 10$ dB has been subtracted from the raw sample.

(5) **Flipping**: Similar to conventional image classification,[54] flipping is used as one of the transformations in this work. Even though this transformation cannot be either physically explained or directly produced by the environment, it is employed to test the invariant filters used in the ResNet. For this augmentation, a batch is chosen with a probability p. Then, the spectrograms are flipped along the horizontal or vertical axis with the same probability, i.e., $p_h = p_v = 0.5$, where p_h and p_v are the probabilities of horizontally and vertically flipping the spectrogram, respectively. A modified spectrogram vertically flipped is shown in Fig. 6(f).

(6)–(9) **Time warping, time masking, frequency masking, and combined transformation**: These techniques are adapted from Park *et al.*,[34] where spectrograms are modified to perform speech recognition tasks. Each of these augmentations has a physical interpretation in the ocean acoustics context. The time warping transformation helps the network to be robust against deformations in the time direction produced by a degradation in the acoustic channel. The time masking and frequency masking augmentations prepare the network for a partial loss of small segments of speech and a partial loss of frequency information in the data collected from the at-sea experiments. In addition, the combined case, which incorporates all three of the previous augmentations (time warping, frequency masking, and time masking), is used due to the outstanding results shown in Ref. 34.

In the case of time warping, only a portion of the spectrograms is stretched in time while the frequency is kept constant.[34] For each sample, a time warp parameter τ is set. Then, a segment with a maximum width τ is randomly selected from the spectrogram to be stretched. An example of this transformation is shown in Fig. 6(g), where the stretching was performed at approximately $t = 1106$ s of the recorded signal. Time masks are small blocks with a maximum width w_t, where the values inside the mask are dropped out. The parameter w_t is randomly selected within the interval $[0, \zeta]$, where ζ is a positive integer value. In total, there are at most n_t masks that are applied across the x axis to each spectrogram, as shown in Fig. 6(h). Similarly, up to n_f frequency masks with a height of h_f are applied across the y axis, as depicted in Fig. 6(i). The last transformation presented in Fig. 6(j) is a combination of the time warping along with the time and frequency masking. Each of the spectrograms in the training set is modified by applying time warping, followed by the addition of masking blocks of consecutive frequency channels and masking blocks across the time direction. These nine data augmentation techniques are implemented to help the network to be robust against deformations across the time and frequency axes.

As previously mentioned, data augmentation randomly modifies the spectrograms at each epoch. These random transformations induce changes in the patterns and features in the data learned by the network, constraining the way the network picks up meaningful information from the training set. In other words, each of these augmentations indirectly adds a different regularization constraint during the training that helps the network avoid overfitting by learning more than only the fixed patterns.

V. RESULTS AND DISCUSSION

Using the ResNet-18 architecture presented in Fig. 5 and Table II, two different output layers are considered to perform the source localization and environment classification. The first network, referred to as "*4-class*," produces classification outputs corresponding to the four canonical environments mentioned in Sec. III and Ref. 43. The network called "*3-reg*" uses regression to find the three source labels z_s, r_{CPA}, and v_{ship}. In this section, the performance of both of the modes is studied for different test scenarios, which employ the synthetic and measured data. Additionally, data augmentation is also applied to both of the architectures. This section also presents the results after applying the augmentations discussed in Sec. IV B.

To effectively report the results, appropriate metrics were chosen to quantify the performance of the networks adequately. For the case of the model implementing regression, the root mean square error (RMSE), shown in Eq. (2), was used as a metric to evaluate the predictions,

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}, \tag{2}$$

whereas for the classification, the accuracy or percentage of correct predictions presented in Eq. (3) was reported as

https://doi.org/10.1121/10.0007284

$$\text{accuracy} = \frac{\sum_{i=1}^{n} \mathbb{1}(\hat{y}_i = y_i)}{n} \times 100\%, \tag{3}$$

where $\mathbb{1}(\cdot)$ is the indicator function, n is the total number of samples in the dataset, y_i is the ith label or ground truth, and $\hat{y}_i$ is the ith prediction such that $\hat{y}_i = h(y_i)$, where h is the model used (ResNet-18).

A. Preprocessing

Before the training, the entire dataset was passed through a preprocessing stage in which the original simulated data were downsampled by a factor of eight across the horizontal axis to reduce the dimensionality without losing generalization capabilities in the network. The new size of each spectrogram after this preprocessing was $1 \times 80 \times 1107$. Afterward, the spectrograms were smoothed horizontally with a median filter of size 1×8. The resulting spectrograms (in dB ref 1 μPa) after downsampling and smoothing were used as input data for the training. Additionally, the range for each of the labels corresponding to the source parameters shown in Table I, $z_s \in [5, 65]$ m, $r_{\text{CPA}} \in [0, 1200]$ m, $v_{\text{ship}} \in [0, 5]$ kn, was normalized on a scale from 0 to 100.

B. Training and validation

The ResNet-18 architectures were trained using the 6600 preprocessed synthetic samples. The simulated dataset was split into two subsets in which 95% of the samples were used for the training and the remaining 5% were used for the validation. To train the network, multiple parameters were selected. First, the gradient-based Adam optimizer[56] was used with a maximum of 200 epochs. In addition, early stopping with patience $n_p = 25$ was used to prevent overfitting in the model. With early stopping, the validation error was measured at each epoch to finish the training when the error no longer decreased during $n_p = 25$ consecutive epochs. The initial learning rate for both of the networks was 0.01 with the implementation of the cosine annealing scheduler,[57] which changes the learning rate every epoch. The batch size used for the training was a relevant factor to consider because of the risk of overfitting. A small batch size tends to overfit the model, whereas a large batch size yields higher prediction errors.[58,59] In this study, after testing different values, the batch size for both of the networks was set to 32. For updating the weights and parameters in the network, a cost or loss function $L(\mathbf{y}, \hat{\mathbf{y}})$ was chosen. In the case of the classification, the cross-entropy loss in Eq. (4) was used as a loss function for the optimization

$$L_{\text{cross-entropy}}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{n} y_i \log \hat{y}_i, \tag{4}$$

where the sum of the entropy is calculated for the n samples in the simulated dataset. In the case of the regression, the mean squared error [Eq. (5)] was selected as the cost function

$$L_{\text{MSE}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n} \sum_{i=1}^{n} \|y_i - \hat{y}_i\|_2^2, \tag{5}$$

where $\|\cdot\|_2$ represents the ℓ_2-norm and n is the number of samples. In Eqs. (4) and (5), y_i is the true label, and $\hat{y}_i$ is the prediction for the ith spectrogram in the simulated dataset.

To test that the models were learning properly and verify their robustness, ten different instances of each network were trained without applying any transformation (i.e., without data augmentation). Here, the term *instance* refers to a *copy* of the model with different random initial weights. For each of the nine different data augmentation techniques discussed in Sec. IV B, only five instances were trained to study the models *4-class* and *3-reg* due to the number of transformations. In total, 100 trainings were done for the data augmentation analysis based on the five instances, tasks, and augmentations.

During each epoch, the training and validation errors were measured to optimize the model properly. PyTorch was used as the framework for implementing the models and transformations. The computational time to train each model was about 4 h using a GPU (graphic processing unit) Nvidia Quadro RTX 5000 with 16 GB of VRAM (virtual random access memory). The total number of learnable parameters for the *4-class* networks was 188 667, whereas the *3-reg* model for the source localization had 193 020 parameters.

C. Testing

Ten instances of the ResNet-18 models used for the seabed classification (*4-class*) and source localization (*3-reg*) are tested in the three following scenarios: (1) *SSP mismatch*, (2) *sediment mismatch*, and (3) *measured field data* from the SBCEX 2017. Sections III C 1–III C 3 present cumulative results of the RMSE and accuracy for the ten instances in each case.

1. Sound speed profile mismatch

To evaluate the effect of the changes in the SSPs, the synthetic test cases A–D use different SSPs than those used in the training, as shown in Fig. 3. The seabeds used to generate the test cases correspond to the same four canonical environments from the training. Test A comprises 2000 samples and presents SSPs similar to the training with slight changes in the magnitude of the sound speed across the water column. Test B has 2000 samples and uses an isovelocity SSP, where the water depth varies about ± 0.5 m. Test C has 3000 samples and uses three isovelocity SSPs (lower and higher than those used in training). Test D counts on 5000 samples and is simulated with the downward refracting SSPs. Test D is the most distinct environment out of the previous four test cases and represents a challenge for the network. The SSPs used to simulate tests A–D and the training data are displayed in Fig. 3.

The effects of the SSP mismatch are studied using test cases A–D and the ten trained instances of ResNet-18 without applying any data augmentation to the dataset. The

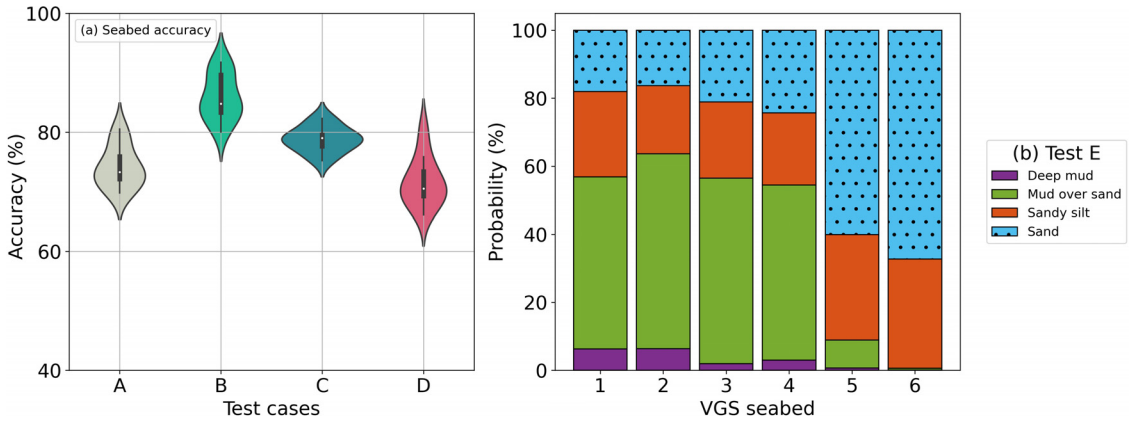

FIG. 7. (Color online) The accuracy in the synthetic test cases using the *4-class* ResNet-18 for (a) tests A, B, C, and D, and (b) the accuracy for test E, which comprises the viscous grain shearing (VGS) parameterization, are shown.

seabed classification accuracy for all of the ten cases in the *4-class* ResNet-18 study are presented in Fig. 7. The statistical distribution and accuracy over the ten instances are displayed as violin plots. One advantage of the violin plot is that it provides a graphical representation of the distribution of the obtained results. Most predictions are shown to be concentrated and not sparse (see the bar inside the violin plots, which represents the interquartile range). The best performance is obtained in test B, representing the case of a small variation in the water depth (± 0.5 m). The average accuracy for test B is about 90%. In tests A, C, and D, the mean accuracy is above 65% for all of the cases.

On the other hand, Fig. 8 shows the predictions for the ten instances of the *3-reg* architectures. The RMSE error is reported for the parameters of the source depth z_s in Fig. 8(a), CPA range r_{CPA} in Fig. 8(b), and ship speed v_{ship} in Fig. 8(c) for the different tests cases. The RMSE for cases A–D are primarily less than 15 m in the case of z_s. As expected, the error increases as the SSP mismatch increases. In the case of r_{CPA} [Fig. 8(b)], the errors for tests A–D are consistently about 100 m off, and this fact suggests that the SSP mismatch does not represent a big issue for the generalization in the proposed ResNet. For the v_{ship} predictions [Fig. 8(c)], the outcome of tests A–C are consistently less 0.4 kn. Although the predictions for v_{ship} in test D (i.e., the downward refracting SSPs) are nicely grouped, the error is higher than that for the other three cases, indicating that the SSPs not included in the training data have a major effect on the ship speed predictions.

For most of the cases, the violin plots shown in Figs. 7 and 8 present a low variance in the RMSE for the ten instances, indicating that the networks tend to produce similar predictions for all of the ten instances. Out of all of the parameters in both the classification and regression models, the source depth presents a bigger challenge and is the most sensitive parameter of the four. The predictions for both v_{ship} and r_{CPA} are consistent across the ten instances with a classification accuracy for all four of the cases above 65%.

2. Seabed mismatch

To study the impact of the seabed mismatch on the ResNet-18 predictions, test E incorporates a different type of sediment. For this case, a viscous grain shearing (VGS) parameterization of the seabed is employed instead of the four canonical environments used for the training and test cases A–D. Test E has 4536 samples and uses six distinct different parameterizations of the VGS model. The six VGS

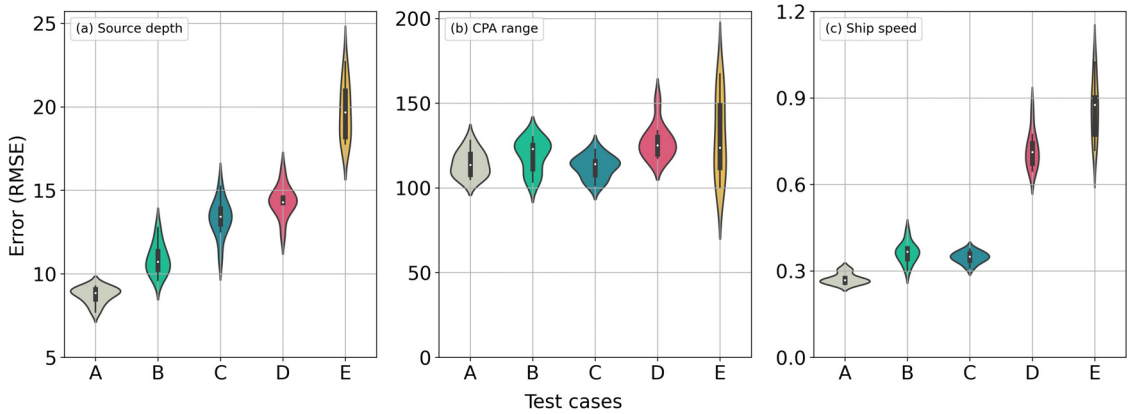

FIG. 8. (Color online) The root mean square error (RMSE) for the synthetic test cases A–E using the *3-reg* ResNet-18 for the (a) source depth, (b) CPA range, and (c) ship speed.

https://doi.org/10.1121/10.0007284

environments are mainly based from the inversions performed during several experiments and correspond to mud and sandy environments. VGS #1, reported by Knobles *et al*., corresponds to lossy *mud over sand*. VGS #2 or *deep mud over sand* is based on the inversions performed in the Gulf of Mexico.[46] VGS #3 refers as to medium loss *mud over sand* similar to that in VGS #4, which corresponds to low loss *mud over sand*. VGS #5 is *sandy silt* obtained via the geoacoustic inversions by Potty *et al*.,[48] whereas VGS #6 is *coarse sand*, gathered from the New Jersey sand ridge experiment.[49]

The performance of the networks in the presence of seabed mismatch can be studied using VGS parameterization because the geophysical properties are quite different from those used in the training. The results of applying the *4-class* network to test E are presented in Fig.7(b). Although there is not a one-to-one correlation between the VGS seabeds and the four seabed labels used in the training (#1, #2, #3, and #4), the classification tries to link the data samples in test case E to those four labels. Therefore, for test E, a stacked bar plot is used to show the performance of the network. The bars display the probability (in percentage) of selecting one of the four trained labels for each of the VGS environments. As mentioned earlier, the VGS seabeds #1–#4 correspond to muddy environments and can be associated with the (1) *deep mud* and (2) *mud over sand* sediments used to train the networks. Similarly, the VGS seabeds #5 and #6 correspond to sandy environments akin to the (3) *sandy silt* and (4) *sand* sediments from the training. The classification predictions in Fig. 7(b) agree with the expected behavior, in which the VGS seabed types are labeled according to their similarity with the properties of the four canonical environments.

For the *3-reg* network, the seabed is not labeled, and regression is performed solely for the source parameters. The resulting predictions in Fig. 8 show the predictions of the *3-reg* model for the three source localization parameters z_s, r_{CPA}, and v_{ship}. The source location estimations obtained in test E are comparable with the median value of the predictions from test D, except in the case of z_s. However, the distributions are wider, representing the increased uncertainty resulting from the seabed mismatch.

As reported in the SSP mismatch, estimations of the source depth yield the highest relative RMSE compared to the rest of the source localization labels. The seabed classification predictions are affected by changes in the water column and sediment. The error presented in the ResNet-18 predictions increases when the VGS parameterization is used, as is shown in Figs. 7 and 8. These results reveal the limitations of the machine learning techniques for extrapolation. The generalization and predictions can be improved by training the network using a larger dataset containing multiple and variate representative seabeds.

3. VLA measured data

The ability of ResNet-18 to make accurate predictions on the at-sea observations is also evaluated using the six measured samples described in Sec. III. Ten instances of the *4-class* and *3-reg* models without any data augmentation are tested using the data collected from the mid-frequency towed source in the SBCEX 2017.

The seabed presented in this New England Mud Patch resembles the characteristics of classes #1 and #2 used for the training. Figure 9 presents the results for the *4-class* network, where most of the predictions are within the range (1–3). All of the distributions corresponding to the six samples are very similar, which indicates that the network classifies the environments consistently because the data correspond to points in the same geographic area. However, the wide distributions indicate a large variation across the results from the ten instances of the *4-class* ResNet-18 model.

Likewise, the results of the ten instances of the *3-reg* network used to infer the z_s, r_{CPA}, and v_{ship} are presented in Fig. 10. The dashed lines in Fig. 10 show the expected values for the source parameters. Some ResNet-18 predictions reach the expected values for each label; however, the remaining predictions underestimate the ground truth values. The estimations of the source depths present the widest distributions, reiterating the high sensitivity of z_s for the generalization. The spread and variation of the predictions are likely because the four canonical seabeds do not exactly capture the seabed properties presented in the SBCEX 2017.

As discussed in Sec. V A, the synthetic data used here were preprocessed but not normalized. The differences and offsets in the predictions could be explained by the spline fit (i.e., envelope in the signal) applied in the measured data during the extraction process[43] and the information that the skip connections in ResNet provide. Compared with the study presented in Ref. 43, the predictions for the *3-reg* model are closer to the expected values for the CPA range r_{CPA}, farther from the true values for ship speed v_{ship}, and equivalent in the case of the source depth z_s. In the case of the *4-class* network, ResNet-18 performs similarly to the CNN, and both architectures yield the expected seabed types.

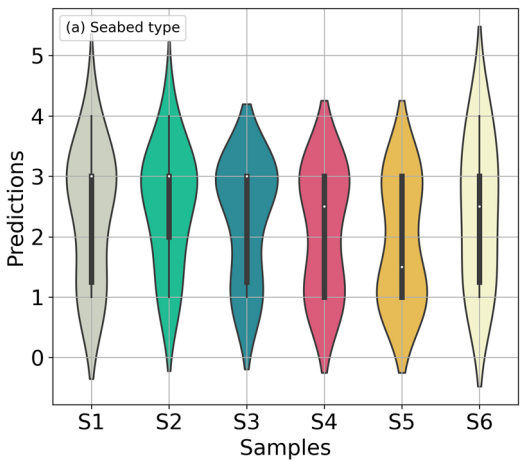

FIG. 9. (Color online) The predictions for the six measured samples using the *4-class* ResNet-18 to infer the seabed type.

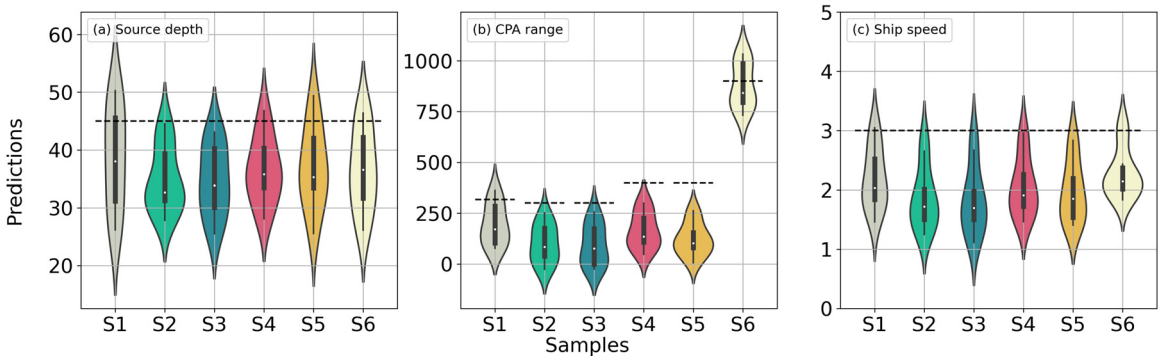

FIG. 10. (Color online) The predictions for the six measured samples using the *3-reg* ResNet-18 to predict the (a) source depth z_s, (b) CPA range r_{CPA}, and (c) ship speed v_{ship}. The dashed line in each subplot indicates the expected values for the variable.

D. Data augmentation

One of the objectives of this study is to assess the effects of data augmentation in the predictions while it is applied during the training of the ResNet. The impact of the data augmentation in the ResNet-18 performance was addressed by the training models with and without the transformations discussed in Sec. IV B. To evaluate the data augmentation, the models for the seabed classification (i.e., *4-class*) and regression (i.e., *3-reg*) were studied. For each task, five instances were trained using each of the nine augmentations during the training. These trained networks were then tested using cases A–E and the at-sea data. In total, 100 trainings were performed to study the impact of the data augmentation (*i.e., 2 tasks* × *5 instances* × *10 augmentations*).

The data augmentation was handled by selecting random batches from the training data with a probability $p = 0.5$ to apply one of the nine transformations described in Sec. IV B. The dropout was applied to 5% of the values in each spectrogram from the batch ($q = 0.05$). The uniform noise incorporated a level of 60 dB ($c = 60$), multiplied by a random matrix coming from a uniform distribution $N_{\text{noise}} = 60\ \text{dB} \times H$ such that $H \sim U(0, 1)$. A factor $d = 2$ was used for time stretching, where the spectrograms in the batch were stretched in time by $d = 2$ and then cropped to half of the original size. For the loudness, a ±10 dB level was randomly applied on each of the selected spectrograms, i.e., $c = \pm 10$ dB. In the case of flipping, each spectrogram in the batch was flipped randomly with the probabilities $p_v = p_h = 0.5$. Time warping was applied to a random window $\tau = 50$ to all of the spectrograms in the batch. Similarly, the random time and frequency masks were used for all of the samples in the batch, where $n_t = n_f = 5$, $w_t = 20$, and $w_f = 5$. Finally, the combined transformation using warping and the time and frequency masking was applied to the randomly selected spectrograms with values of $\tau = 20$, $n_t = n_f = 5$, $w_t = 20$, and $w_f = 5$ to all of the training data. These nine transformations were only applied during the training, whereas test cases A–E and the at-sea data remained the same (i.e., without any modification).

The results of the networks trained with and without data augmentation applied to synthetic tests A–E are shown in Fig. 11. The RMSE is calculated for each test set and the five instances of the *3-reg* model for the source depth, CPA range, and ship speed [see Figs. 11(a)–11(c)]. The seabed accuracy from the *4-class* architecture is shown in Fig. 11(d). The confidence intervals shown at the top of each bar in Fig. 11 represent the standard deviation (σ) of the errors and accuracy. In this paper, the standard deviation is expressed as $\sigma = \sqrt{\sum_{i=1}^{k} (y_i - \hat{y}_i)^2 / k}$, where k is the total number of instances, y_i is the ground truth value, and $\hat{y}_i$ is the estimation made by the ith instance.

Some of the implemented augmentations change the structure of the data, as shown in Fig. 6. For instance, the dropout, addition of loudness, uniform noise, and time and frequency masks change the levels in the data samples. Time stretching, flipping, warping, and the combined augmentation produce modifications in the shapes of the striation patterns presented in the spectrograms.

The bar plots displayed in Fig. 11 allow for an easy comparison of the overall effect of the nine data augmentations implemented here. The addition or subtraction of dB levels (loudness) produces the highest errors in the regression tasks for the CPA range and ship speed. This result is explained because these spectrogram levels are highly correlated with the proximity between the emitter and the receiver. Other transformations, such as dropout, uniform noise, and time warping, do not yield significant improvements in the predictions compared to the case without data augmentation. In contrast, stretching and flipping do improve the performance most of the time for both scenarios. Time stretching works because it enhances the big striation patterns in the sample, and flipping helps the training by adding some randomness to the spectrograms without changing their level or shape. Time and frequency masking also provide improvements in the generalization of the model. The addition of small zeroed-out blocks into the time and frequency axes allows the network not to depend every time on the same patterns to "*make a decision*" and predict a value. Finally, the combined transformation implements that time warping and the frequency and time masks are superior to the other augmentations in almost all of the scenarios shown in Fig. 11. The power of such a

https://doi.org/10.1121/10.0007284

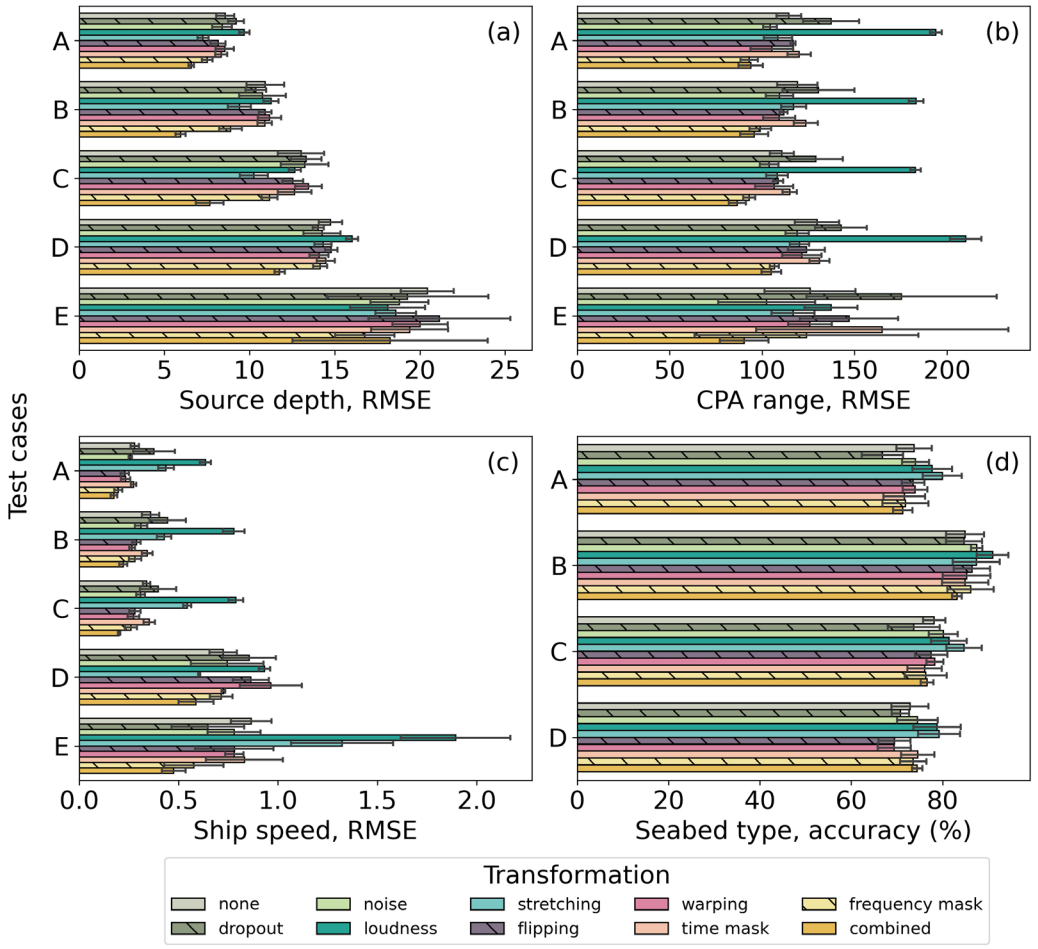

FIG. 11. (Color online) The performance of ResNet-18 algorithms with and without data augmentation. The bars show a comparison of the error and accuracy between the trained models without data augmentation (*none* in the figure) and the nine different transformation used. The results are presented for the (a) source depth z_s, (b) CPA range r_{CPA}, (c) ship speed v_{ship}, and (d) seabed type. The results in (a), (b), and (c) are from ResNet trained for only regression outputs, whereas (d) contains the results for ResNet trained for only classification.

transformation dwells in its ability to modify the patterns and shapes presented in the spectrogram without altering the pressure levels.

Similarly, the networks trained with and without transformations were applied to the six real data samples from the SBCEX 2017. The results for the regression and classification are presented in Fig. 12 as bar plots, representing the mean RMSE (relative to the expected values) of the predictions of the five instances for the *4-class* and *3-reg* architectures. In this case, the effects of the data augmentation are similar to the results obtained with the simulated test cases reported in Fig. 11. In most cases, the combined transformations of time warping and the time and frequency masking outperform the results obtained with the other data augmentation techniques. The combined data augmentation helps the network to be robust against deformations in the time direction, partial loss of frequency information, and partial loss of small segments of speech in the signal.[34]

When using the at-sea spectrograms, the application of the data augmentation produces similar results to those seen in test cases A–E. For the source depth estimation, time warping and loudness yield lower errors than those cases when no transformation is applied. Regarding the CPA range predictions, only the combined augmentation significantly improves the network performance most of the time. For ship speed, uniform noise, frequency masking, and the combined transformation present the lowest errors, whereas the loudness presents the highest RMSE. As for the seabed type in Fig. 12(d), the mean predicted seabed class over the five instances is reported and most agree with the expected value of seabed class #2 (*mud over sand*). The majority of the transformations yield labeled seabed types around 1.5–2.5, which is close to the expected value estimated from the SBCEX 2017. Only a few instances of the frequency and time masking transformations produce values below or above such an interval. In general, the combined augmentation yields lower errors when the synthetic data are used, whereas noise presents improvements when the measured data are employed. Here, data augmentation acts as a regularizer and reduces the overfitting when training the ResNet-18 models. The predictions presented in Figs. 11 and 12 show the effectiveness of incorporating data augmentation during the training step of ResNet-18.

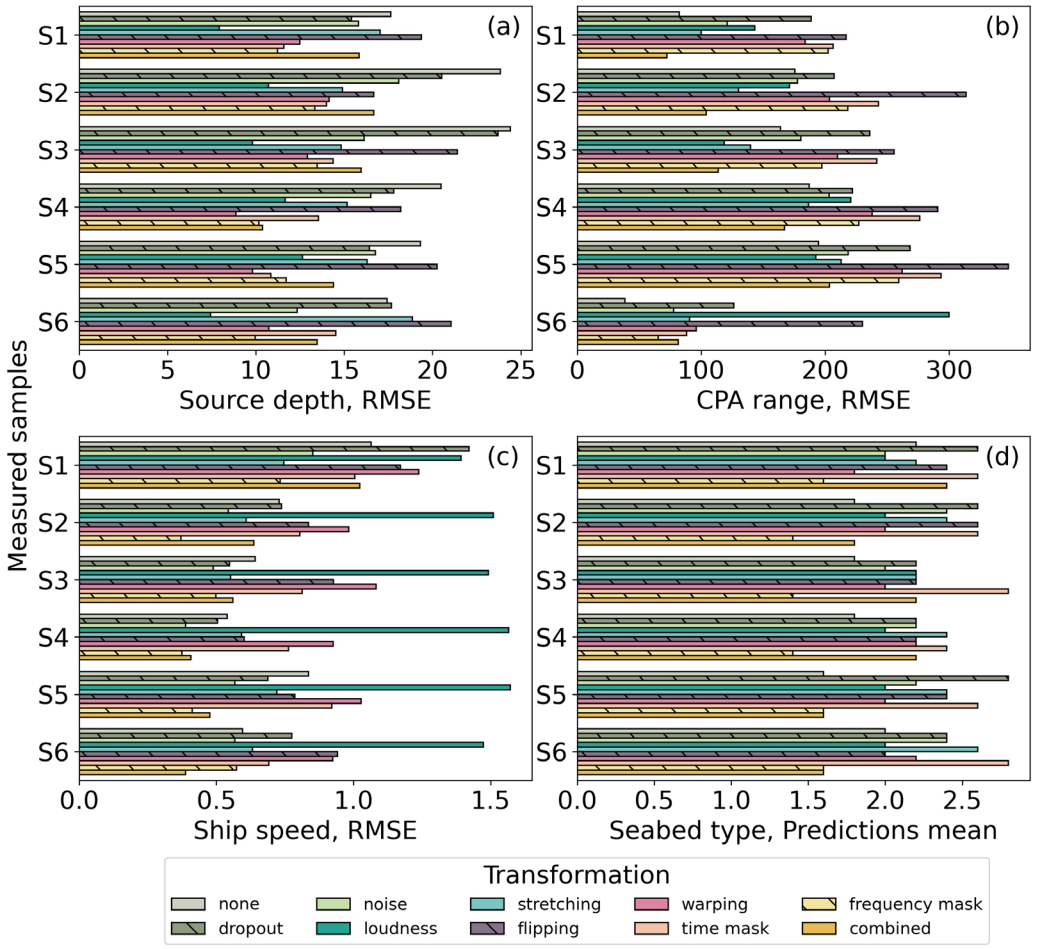

FIG. 12. (Color online) The performance of ResNet-18 algorithms with data augmentation for six measured samples for the SBCEX 2017. The RMSE are presented for the (a) source depth z_s, (b) CPA range r_{CPA}, and (c) ship speed v_{ship}. (d) The mean of the predictions for the seabed type of five instances similar to Fig. 11 are presented.

VI. CONCLUSIONS

This paper presents the application of ResNets (ResNet-18) for both source localization and seabed classification using towed tonal source spectrograms. Two architectures were implemented, one to perform the seabed classification and another to predict the source parameters (source depth, CPA range, and ship speed). Both of the architectures were tested using the synthetic and measured data collected from the SBCEX 2017. After applying the networks to the synthetic test cases and at-sea measured data, both of the architectures yielded results comparable to previous work.[43] These ResNets proved to be a viable alternative for the source localization and seabed classification as were the CNNs in Ref. 43. The implemented ResNet-18 reported comparable results using a much lesser number of hyperparameters (193 020 versus 32 462 036) while keeping a similar training time (about 3–4 h).

ResNet-18 was applied to the testing data with the SSP and seabed mismatch. Tests A–D incorporated variations in the SSP, whereas test E changed the sediment structure. Substantial changes in the environment yielded higher generalization errors as presented in test cases D and E. The networks related to the seabed classification reached an accuracy of over 65% using four seabed types. An extension of this work beyond four seabed types is needed to better represent the variety of sediments in the seafloor.

Two types of models were applied to the SBCEX 2017 experimental data. In the case of seabed classification, the *4-class* architecture consistently predicts a class within the range of 1 and 3, in agreement with the seabed type obtained by the geoacoustic inversions during the experiment.[44] For regression, the *3-reg* model (which performs regression for all four labels) consistently predicts the source depth, CPA range, and ship speed slightly lower than the expected values. However, the source depth is the most difficult label to predict because the predictions of this parameter present the largest percentage of relative error. Finally, all of the ResNet-18 models in this paper show improvement in the CPA range and source depth predictions but degradation in the ship speed estimations compared with previous studies.[43] These implemented ResNets estimate the CPA ranges much closer to the expected values, underestimate the ship speed, and produce comparable results for the source depth and seabed type for the measured data samples.

The results obtained by applying ResNet-18 are shown to be consistent with the findings obtained in previous work (Fig. 10 in Ref. 43).

Data augmentation is a key tool for improving the machine learning performance by preventing overfitting in the models. In this work, multiple data transformations have been applied to different testing sets to study the influence of such changes on the accuracy of the models. Out of the nine augmentations implemented during the training, only the loudness transformation impacted the results negatively. The dropout, uniform noise, and time warping augmentations did not contribute to significant improvements for the network predictions. In contrast, time stretching, flipping, time masking, frequency masking, and the combined transformation positively impacted the ability of the trained networks to generalize to synthetic datasets with environmental mismatch and the measured data samples. The use of those five transformations reduced the errors for the source depth, CPA range, and ship speed predictions by about 20% and increased the model accuracy for the seabed classification by approximately 1%–4% compared to the case in which the data augmentation was not implemented. The improvement in the predictions showed that data augmentation helps deep networks to focus only on the most relevant features in the data during the training. The results demonstrated that these data transformations are a reliable regularization technique to improve the ResNet performance for seabed classification and source localization using mid-frequency spectrograms, outperforming predictions obtained without data augmentation.

ACKNOWLEDGMENTS

This work was funded by the Office of Naval Research Contract No. N00014-19-C-2001.

[1]M. Siderius, P. L. Nielsen, J. Sellschopp, M. Snellen, and D. Simons, "Experimental study of geo-acoustic inversion uncertainty due to ocean sound-speed fluctuations," J. Acoust. Soc. Am. **110**(2), 769–781 (2001).
[2]S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," Science **220**(4598), 671–680 (1983).
[3]S. Kirkpatrick, "Optimization by simulated annealing: Quantitative studies," J. Stat. Phys. **34**(5-6), 975–986 (1984).
[4]Z.-H. Michalopoulou, "Gibbs sampling optimization in underwater sound problems," in *Conference Proceedings of MTS/IEEE Oceans 2001. An Ocean Odyssey* (IEEE, Honolulu, HI, 2001), Vol. 2, pp. 782–785.
[5]G. L. Bilbro and D. E. Van den Bout, "Maximum entropy and learning theory," Neural Comput. **4**(6), 839–853 (1992).
[6]S. E. Dosso and M. J. Wilmut, "Bayesian focalization: Quantifying source localization with environmental uncertainty," J. Acoust. Soc. Am. **121**(5), 2567–2574 (2007).
[7]A. B. Baggeroer, W. A. Kuperman, and H. Schmidt, "Matched field processing: Source localization in correlated noise as an optimum parameter estimation problem," J. Acoust. Soc. Am. **83**(2), 571–587 (1988).
[8]M. B. Porter and A. Tolstoy, "The matched field processing benchmark problems," J. Comput. Acoust. **02**(03), 161–185 (1994).
[9]E. K. Westwood and D. P. Knobles, "Source track localization via multipath correlation matching," J. Acoust. Soc. Am. **102**(5), 2645–2654 (1997).
[10]M. S. Ballard, "Estimation of source range using horizontal multipath in continental shelf environments," J. Acoust. Soc. Am. **134**(4), EL340–EL344 (2013).
[11]C. Cho, H. C. Song, and W. S. Hodgkiss, "Robust source-range estimation using the array/waveguide invariant and a vertical array," J. Acoust. Soc. Am. **139**(1), 63–69 (2016).
[12]C. Cho and H.-C. Song, "Impact of array tilting on source-range estimation based on the array/waveguide invariant," J. Acoust. Soc. Am. **140**(4), 3172 (2016).
[13]A. Kujawski, G. Herold, and E. Sarradj, "A deep learning method for grid-free localization and quantification of sound sources," J. Acoust. Soc. Am. **146**(3), EL225–EL231 (2019).
[14]Y. Liu, H. Niu, and Z. Li, "A multi-task learning convolutional neural network for source localization in deep ocean," J. Acoust. Soc. Am. **148**(2), 873–883 (2020).
[15]Y. Liu, H. Niu, Z. Li, and M. Wang, "Deep-learning source localization using autocorrelation functions from a single hydrophone in deep ocean," JASA Express Lett. **1**(3), 036002 (2021).
[16]D. F. Van Komen, T. B. Neilsen, D. B. Mortenson, M. C. Acree, D. P. Knobles, M. Badiey, and W. S. Hodgkiss, "Seabed type and source parameters predictions using ship spectrograms in convolutional neural networks," J. Acoust. Soc. Am. **149**(2), 1198–1210 (2021).
[17]K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 770–778.
[18]Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature **521**(7553), 436–444 (2015).
[19]L. Deng and D. Yu, "Deep learning: Methods and applications," Found. Trends Signal Process. **7**(3–4), 197–387 (2014).
[20]I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning* (MIT Press, Cambridge, 2016).
[21]Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, Paris, France (IEEE, 2010), pp. 253–256.
[22]H. Liang, X. Sun, Y. Sun, and Y. Gao, "Text feature extraction based on deep learning: A review," EURASIP J. Wireless Commun. Network. **2017**(1), 1–12 (2017).
[23]Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," IEEE J. Sel. Topics Appl. Earth Obs. Remote Sens. **7**(6), 2094–2107 (2014).
[24]K. He, X. Zhang, S. Ren, and 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.
[25]S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv:1502.03167 (2015).
[26]K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv:1409.1556 (2014).
[27]C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 1–9.
[28]A. M. Saxe, J. L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," arXiv:1312.6120 (2013).
[29]X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (2010), pp. 249–256.
[30]P. Y. Simard, Y. A. LeCun, J. S. Denker, and B. Victorri, "Transformation invariance in pattern recognition–tangent distance and tangent propagation," in *Neural Networks: Tricks of the Trade*, edited by G. Montavon, G. B. Orr, and K.-R. Müller (Springer, Berlin, Heidelberg, 1998), pp. 239–274.
[31]L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," arXiv:1712.04621 (2017).
[32]J. Wei and K. Zou, "EDA: Easy data augmentation techniques for boosting performance on text classification tasks," arXiv:1901.11196 (2019).
[33]Z. Hussain, F. Gimenez, D. Yi, and D. Rubin, "Differential data augmentation techniques for medical imaging classification tasks," in *AMIA Annual Symposium Proceedings* (American Medical Informatics Association, Washington, DC, 2017), Vol. 2017, p. 979.

[34]D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," arXiv:1904.08779 (2019).
[35]A. Sakai, Y. Minoda, and K. Morikawa, "Data augmentation methods for machine-learning-based classification of bio-signals," in *2017 10th Biomedical Engineering International Conference (BMEiCON)* (IEEE, Hokkaido, Japan, 2017), pp. 1–4.
[36]J. Schlüter and T. Grill, "Exploring data augmentation for improved singing voice detection with neural networks," in *ISMIR* (2015), pp. 121–126.
[37]H. Niu, Z. Gong, E. Ozanich, P. Gerstoft, H. Wang, and Z. Li, "Deep-learning source localization using multi-frequency magnitude-only data," J. Acoust. Soc. Am. **146**(1), 211–222 (2019).
[38]D. F. Van Komen, T. B. Neilsen, K. Howarth, D. P. Knobles, and P. H. Dahl, "Seabed and range estimation of impulsive time series using a convolutional neural network," J. Acoust. Soc. Am. **147**(5), EL403–EL408 (2020).
[39]C. D. Escobar-Amado, T. B. Neilsen, J. A. Castro-Correa, D. F. Van Komen, M. Badiey, D. P. Knobles, and W. S. Hodgkiss, "Seabed classification from merchant ship-radiated noise using a physics-based ensemble of deep learning algorithms," J. Acoust. Soc. Am. **150**(2), 1434–1447 (2021).
[40]M. J. Bianco, P. Gerstoft, J. Traer, E. Ozanich, M. A. Roch, S. Gannot, and C.-A. Deledalle, "Machine learning in acoustics: Theory and applications," J. Acoust. Soc. Am. **146**(5), 3590–3628 (2019).
[41]C. Frederick, S. Villar, and Z.-H. Michalopoulou, "Seabed classification using physics-based modeling and machine learning," J. Acoust. Soc. Am. **148**(2), 859–872 (2020).
[42]W. Liu, Y. Yang, M. Xu, L. Lü, Z. Liu, and Y. Shi, "Source localization in the deep ocean using a convolutional neural network," J. Acoust. Soc. Am. **147**(4), EL314–EL319 (2020).
[43]T. B. Neilsen, C. Escobar-Amado, M. C. Acree, W. S. Hodgkiss, D. F. Van Komen, D. P. Knobles, M. Badiey, and J. Castro-Correa, "Learning location and seabed type from a moving mid-frequency source," J. Acoust. Soc. Am. **149**(1), 692–705 (2021).
[44]P. S. Wilson, D. P. Knobles, and T. B. Neilsen, "Guest editorial an overview of the seabed characterization experiment," IEEE J. Ocean. Eng. **45**(1), 1–13 (2020).
[45]E. K. Westwood, C. T. Tindle, and N. R. Chapman, "A normal mode model for acousto-elastic ocean environments," J. Acoust. Soc. Am. **100**(6), 3631–3645 (1996).
[46]D. P. Knobles, R. A. Koch, L. A. Thompson, K. C. Focke, and P. E. Eisman, "Broadband sound propagation in shallow water and geoacoustic inversion," J. Acoust. Soc. Am. **113**(1), 205–222 (2003).
[47]D. P. Knobles, P. S. Wilson, J. A. Goff, L. Wan, M. J. Buckingham, J. D. Chaytor, and M. Badiey, "Maximum entropy derived statistics of sound-speed structure in a fine-grained sediment inferred from sparse broadband acoustic measurements on the New England Continental Shelf," IEEE J. Ocean. Eng. **45**, 161–173 (2020).
[48]G. R. Potty, J. H. Miller, and J. F. Lynch, "Inversion for sediment geoacoustic properties at the New England Bight," J. Acoust. Soc. Am. **114**(4), 1874–1887 (2003).
[49]J.-X. Zhou, X.-Z. Zhang, and D. P. Knobles, "Low-frequency geoacoustic model for the effective properties of sandy seabottoms," J. Acoust. Soc. Am. **125**(5), 2847–2866 (2009).
[50]C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Thirty-First AAAI Conference on Artificial Intelligence* (2017).
[51]Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE **86**(11), 2278–2324 (1998).
[52]K. Imai and D. A. Van Dyk, "A Bayesian analysis of the multinomial probit model using marginal data augmentation," J. Econometrics **124**(2), 311–334 (2005).
[53]L. Tóth, G. Kovács, and D. Van Compernolle, "A perceptually inspired data augmentation method for noise robust CNN acoustic models," in *International Conference on Speech and Computer* (Springer, Leipzig, Germany, 2018), pp. 697–706.
[54]C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," J. Big Data **6**(1), 1–48 (2019).
[55]N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," J. Mach. Learn. Res. **15**(56), 1929–1958 (2014).
[56]Z. Zhang, "Improved Adam optimizer for deep neural networks," in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)* (IEEE, Banff, AB, Canada, 2018), pp. 1–2.
[57]A. Gotmare, N. S. Keskar, C. Xiong, and R. Socher, "A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation," arXiv:1810.13243 (2018).
[58]N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," arXiv:1609.04836 (2016).
[59]I. Kandel and M. Castelli, "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset," ICT Express **6**(4), 312–315 (2020).