

Near-surface velocity estimation using shear-waves and deep-learning with a U-net trained on synthetic data

Taneesh Gupta ^a, Paul Zwartjes ^{b,*}, Udbhav Bamba ^a, Koustav Ghosal ^a, Deepak K. Gupta ^a

^a Transmute AI Research, the Netherlands

^b Aramco Research Center, Delft, the Netherlands



ARTICLE INFO

Keywords:

Near-surface
Dispersion curve
Rayleigh wave
velocity
U-Net

ABSTRACT

Estimation of good velocity models under complex near-surface conditions remains a topic of ongoing research. We propose to predict near-surface velocity profiles from surface-waves transformed to phase velocity-frequency panels in a data-driven manner using deep neural networks. This is a different approach from many recent works that attempt to estimate velocity from directly reflected body waves or guided waves. A secondary objective is to analyze the influence on the prediction accuracy of various commonly employed deep learning practices, such as transfer learning and data augmentations. Through numerical experiments on synthetic data as well as a real geophysical example, we demonstrate that transfer learning as well as data augmentations are helpful when using deep learning for velocity estimation. A third and final objective is to study lack of generalization of deep learning models for out-of-distribution (OOD) data in the context of our problem, and present a novel approach to tackle it. We propose a domain adaptation network for training deep learning models that uses a priori knowledge on the range of velocity values in order to constrain mapping of the output. The final comparison on field data, which was not part of the training data, show the deep neural network predictions compare favorably with a conventional velocity model estimation obtained with a dispersion curve inversion workflow.

1. Introduction

Since the advent of deep learning, almost all science engineering disciplines have benefited from the technique - either in terms of improved performance on their respective performance metrics or through reduction in the need of computational resources. Some typical examples include accelerated computational-fluid dynamics (Kochkov et al., 2021), efficient design optimization (Sosnovik and Oseledets, 2019), highly accurate prediction of protein structure (Tunyasuvanakool et al., 2021), among others. The field of geophysics has been no exception and a wide variety of problems have been solved with deep learning. For an overview of the recent developments, see the review paper by Yu and Ma (2021). There are roughly four categories of deep learning applications in seismic processing, although the boundaries between these categories are opaque. The first, and most mature, category of deep learning applications involves various types of machine vision based seismic interpretation, such as fault detection (Yang et al., 2020), seismic facies classification (Alsinan et al., 2021) and first break picking (Yuan et al., 2022). The second category of applications contains signal processing tasks such as seismic denoising (Bhowmick et al.,

2018), deblending (Baardman and Hegge, 2020), groundroll attenuation (Pham and Li, 2022) and seismic preconditioning (Ravasi, 2021). The third category deals with parametric inversion for rock properties (Das and Mukerji, 2020) or seismic impedance (Das et al., 2019). The fourth and final category of deep learning applications contains the so-called physics-informed neural networks (PINN). These PINNs employ standard feedforward neural networks with the parameters of a partial differential equation explicitly encoded into the network. An example of this is using deep neural networks for wavefield modelling (Huang and Alkhalifah, 2021). In the next section, we provide a brief description of the conventional estimation of near-surface velocities from surface waves. Following this, we present our approach based on deep learning, including brief descriptions of transfer learning, artificial augmentations and our domain adaptation strategy. We then evaluate the method on a synthetic dataset as well as real land seismic line. Lastly, we present the conclusions of the paper, followed by limitations and ways to circumvent them.

* Corresponding author.

E-mail address: paul.zwartjes@aramcooverseas.com (P. Zwartjes).

1.1. Deep learning in seismic velocity estimation

We focus in this work on the problem of near-surface velocity estimation from seismic shot gathers, which is traditionally solved as a parametric inversion problem. Geophysical inverse problems are very challenging because they are often ill-posed and computationally expensive. A potential advantage of deep learning for inverse problems is that via ‘suitably’ chosen training data, one can infuse prior knowledge into neural network training, as well as constrain the network predictions. On the computational side, it has now been abundantly demonstrated that the application of trained neural network is computationally efficient. The vast majority of computational effort is shifted to the training phase and can be performed prior to production processing. In recent years, several papers were published dealing with V_p velocity estimation from reflection seismic data using deep learning. In geophysical data processing, there is a wide spectrum of velocity definitions, both for V_p and V_s . On one end of that spectrum, we have the rock physics definition of velocity as used in quantitative inversion for medium parameters. Here, Zheng et al. (2019) have used supervised learning to train a CNN to perform pre-stack elastic inversion, estimating 1D V_p , V_s and density profiles from 2D shot gathers using a 1D earth model as labels. Biswas et al. (2019) apply a similar strategy to estimate 1D elastic profiles but use angle gathers instead of CMP gathers and propose explicit forward modelling from the estimated 1D profiles to ensure a physics driven neural network solution. On the other end of the velocity spectrum are the various ‘processing velocities’. Fabien-Ouellet and Sarkar (2020) used a recurrent neural network for estimating 1D RMS V_p velocity profiles on a gather-by-gather basis, basically replacing the conventional semblance-based workflow. Kazei et al. (2021) have a similar approach, but use a CNN. A more challenging task is to estimate 2D velocity model from shot gathers. Øye and Dahl (2019) use random 2.5D synthetic P-wave velocity models to generate acoustically modelled data which is then used to train a network to directly estimate such a V_p velocity model from the shot-gathers and conclude that the approach could be useful to generate smooth models. Duque et al. (2019) have a similar approach but use a generative adversarial network. Yang and Ma (2019) go one step further and aim to estimate 2D (acoustic) V_p velocity profiles from a collection of shots instead of a single shot. In a different domain, Araya-Polo et al. (2018) showed that it is possible to estimate the gridded V_p velocity models from directly the semblance plots. The results on synthetic data in all these papers are very encouraging, provided the test data is similar to the training data. Herein also lies the challenge, as there a very few papers that have successfully solved the generalization problem and actual application to field data. Deep neural networks have also been used to predict surface wave velocities from dispersion curves. In the field of seismology, Luo et al. (2022) and Wang et al. (2022) used a CNN to predict velocity profiles from actual dispersion curves in a global seismology problem, replacing the actual inversion from dispersion curve to velocity profile. For seismic shot records this leaves the challenge of actually obtaining these dispersion curves from noise spectra. Other authors have proposed method for estimating dispersion curves from phase velocity vs. frequency panels use segmentation neural networks, such as Dai et al. (2021) or directly from shot gathers (Chamorro et al., 2022).

In this paper, we tried to directly estimate the near surface V_p and V_s velocity from phase velocity vs. frequency amplitude spectra using a neural network. The goal of this study is primarily to analyze the influence of different deep neural network training operations (such as transfer learning and different augmentations) on prediction of velocity models. We first analyze the influence of pre-training of deep learning architectures on the overall performance for near-surface velocity estimation. To this end, we compare how a network trained from scratch on the geophysical data fairs against network which is first trained in a conventional manner on the popular ImageNet dataset (Deng et al., 2009) and then further optimized for our geophysical dataset. An important limitation of deep learning problems is their poor

generalization on out-of-distribution datasets. We study this aspect for our velocity estimation problem, and analyze on in-distribution and out-of-distribution samples.

To enrich our training set, there exist several popular perturbation methods that include geometric transformations, color space augmentations, kernel filters, mixing images, random erasing, feature space augmentation, adversarial training (Bai et al., 2021) and neural style transfer (Gatys et al., 2016). These methods have been shown to help boost network performance significantly on traditional computer vision problems. Among these, we investigate the influence of geometric transformations on our geophysical problem. Specific transformations include random cropping, translations, rotations, dilations and random masking. Since the absolute and relative location of energy in the phase velocity spectra is of critical importance, we also include two additional channels in the input data, namely the frequency and phase velocity coordinates.

A well-known problem of deep learning methods is that while their performance on in-distribution samples is good, their performance is bad on out-of-distribution ones. To circumvent this issue, we also present a domain adaption strategy for CNNs which uses a priori information on the bounds of seismic velocity to improve the generalization on out-of-distribution samples.

1.2. Near-surface velocity estimation with surface waves

Whereas the referenced works use vertically propagating P-waves and reflection seismic to estimate deep velocity profiles, we use the horizontally propagating surface waves and focus on the near-surface V_p and V_s velocities, down to roughly 80 m depth. Knowledge of the near-surface is important, both for geotechnical analysis (Socco and Strobbia, 2004) as well as for deep seismic exploration (Keho and Kelamis, 2012). In the former case, the objective could be to map the geometrical distribution of subsoil characteristics prior to construction work or to analyze soil stability. In the latter case, it is important that near-surface heterogeneity is accounted for, either via source and receiver statics or through more dynamic corrections using a detailed near-surface velocity model. Failure to account for near-surface velocity variations can result in structural deformation of deeper horizons and subsequently an over or underestimation of contained reservoir fluids. Surface waves are the dominant energy emitted by the seismic source. There are various types of surfaces waves, but near surface models are most often constructed using Rayleigh waves (83% of the time according to Socco et al. (2010)). Rayleigh waves propagate horizontally in a limited layer close to the surface, with the thickness of that layer roughly equal to one wavelength. The Rayleigh wave velocity, V_r , is slightly lower than the shear wave velocity V_s ($V_r \sim 0.9 V_s$). A complete description of the theory of surface wave propagation is beyond the scope of this paper, but a good introduction is given by Socco and Strobbia (2004). In that paper, the authors summarize that by writing the equation of motion for a laterally homogeneous medium, assuming a plane strain field, imposing boundary conditions of the waves in a half-space with a free surface and imposing the continuity of strain and stress at layer interfaces, a linear differential eigenvalue problem is obtained. In this formulation, the vector f , formed by two displacement eigenfunctions and two stress eigenfunctions, and the 4×4 matrix A , depending on the vertical distribution of the soil properties are related by the equation

$$df(z)/dz = A(z)f(z) \quad (1)$$

This expresses a linear differential eigenvalue problem that has a non-trivial solution only for special values of the wavenumber. The resulting equation is known as the Rayleigh secular equation, and can be written in an implicit form as

$$F_R[V_r(z), V_p(z), V_s(z), \rho(z), k_j, f] = 0 \quad (2)$$

where k is the wavenumber, f is the frequency, $V_r(z)$ the Rayleigh wave

phase velocity, $V_p(z)$ and $V_s(z)$ the compressional and shear wave velocities and $\rho(z)$ is the mass density. In general, it is impossible to solve Eq. (2) analytically and numerical methods are required. Given a set of model parameters ($V_p(z)$; $V_s(z)$; $\rho(z)$) and a specific frequency f , the roots of Eq. (2) are the phase velocities $V_r(z)$. In a vertically heterogeneous medium, this solution is a multi-valued function of frequency that represents the modal curves. In many cases, only the coordinates of the maximum of the fundamental Rayleigh mode are considered which gives us the kinematics information known as the ‘dispersion curve’. In classical dispersion curve inversion, the assumption is made that the fundamental mode is dominant and the energy information is ignored. However, that energy information depends strongly on the stratigraphy and therefore contains additional information (Socco and Strobbia, 2004). The Rayleigh wave velocity depends primarily on the shear wave velocity and layer thickness. The bulk density and P-wave velocity or Poisson’s ratio have less effect and they are often assumed to be known and held constant in the inversion (Xia et al., 1999). Because of these properties of the Rayleigh secular function in Eq. (2), the surface waves are most naturally analyzed via spectral analysis, an approach which dates back to the seventies (Nolet and Panza, 1976) and has culminated in the so-called \multi-channel analysis of surface waves (MASW) method (Park et al., 2007). The analysis can be performed either in the f-k domain, the f-p domain obtained from a slant-stack or the V_{ph} -frequency domain, where the phase velocity $V_{ph} = 2\pi f/k$. The spectral analysis approach obviously requires proper sampling along the spatial axis for unaliased recording of the surface waves, but is otherwise relatively easy in terms of preprocessing. The preprocessing is mainly some muting in the x-t and f-k domains to suppress noise and low S/N ratio data.

As mentioned above, the conventional approach is to pick the dispersion curve defined as the set of absolute maxima per frequency. This is the information used to derive the S-model through inversion (Socco et al., 2010). Manual picking is time consuming, highly subjective and represents a cumbersome task for modern large seismic surveys. Efforts have been made to automate dispersion curve picking, including machine learning techniques. Zheng and Miao (2014) presented an automatic method to pick dispersion curves in the frequency-phase velocity spectrum based on binarization and thinning (Zheng and Miao, 2014). Rovetta et al. (2020) discussed dispersion curve picking using a clustering algorithm. Despite the advancements in dispersion curve picking, the method still relies on the assumption that the required information to describe the subsurface is present in the Rayleigh wave fundamental mode, which, even if true, is complicated by the higher modes that can cross and overlap the fundamental mode.

Additionally, the fundamental mode is not always the mode with the highest amplitude which can confuse the amplitude-based pickers. In dispersion curve inversion, the model parameters are the shear velocity and layer thickness in case of a layered model inversion, or just the shear velocity in case of a gridded inversion. The low frequencies typically sense both the shallow as well as the deeper parts of the subsurface. The higher frequencies only penetrate the shallow part of the subsurface. The inversion problem is, therefore, mix-determined. Not all parts of our model are seen by all the data. Socco and Strobbia (2004) showed that there is no need to over-parameterize the model in terms of number of layers because a five-layer model could exhibit the same fundamental mode dispersion curve as a 35-layer model. This also explain why an inverted shear velocity model is typically much smoother than the actual velocity model. The maximum depth of penetration depends on wavelength and as a rule of thumb the depth is limited to one-half the maximum wavelength (Shtivelman, 1999). Note that the large wavelengths occur at low frequencies where the uncertainty is large, thus reducing the reliability of deep layers. Despite its long history and success, we see that the spectral analysis method for near-surface characterization has some weak points. These are to do with the very nature of dispersion curve inversion, namely that the individual modes can be uniquely identified and picked. Furthermore, in single or fundamental

modal analysis it could happen that higher modes cross and overlap or are stronger than the fundamental mode. Ideally, we would not want to pick the dispersion curves at all but rather provide the whole f-k or f- V_{ph} gather to the inversion or prediction algorithm. This is of course the problem that full waveform inversion (FWI) attempts to solve. However, FWI has its own challenges and is computationally very expensive, especially elastic FWI when applied to noisy land seismic data. We, therefore, set out to assess whether a deep neural network could bridge the gap between dispersion curve inversion and FWI and could learn from ‘seeing’ in the entire phase velocity spectrum via supervised learning what a reasonable estimate could be of V_s and V_p models.

2. Method

The goal of this research is to investigate the application of deep learning for the task of near-surface velocity estimation. The deep learning strategy is shown in the diagram in Fig. 1. For the conventional approach as well as the proposed deep learning method, we work with amplitude frequency spectrum. This is different from some of the recent works which use the shot gathers as direct input for the CNN models and attempt to estimate the velocity.

For the conventional method, as shown in Fig. 1, it is common to estimate dispersion curves and then use dispersion curve inversion to estimate the V_p and V_s curves, as described in the previous section. Our deep learning method alleviates the need for estimating the dispersion curves, through learning a proxy for it in a data-driven manner. The proposed pipeline is as follows. First the phase velocity V_s amplitude spectrum is passed through an augmentation module that applies several different transforms on the image to create new variants. Note that together with the input image, we also pass grid information containing the velocity and frequency coordinates of every point of the spectrum. This approach to preserving spatial information and its benefits has been previously studied by Liu et al. (2018) For the phase velocity spectra, every point on the image can be represented by a set of values (f; v). This grid information is fused in the form of additional channels and combined with the amplitude spectrum to form a 3-channel input for the transformation module. For an understanding of this, see the schematic diagram shown in Figure A-1 in the appendices of this paper. The advantage is that even after transformations applied, the spatial information related to each pixel of the image is still preserved. The transformed input is then fed into a CNN backbone from which the V_p and V_s curves are then regressed. Based on whether the domain-related information is to be input, the domain adaptation module is activated. The end-to-end deep learning pipeline is trained using several input-output combinations such that the regression loss on the velocity curves is optimized. With a diverse set of data samples, we hope that the network can be made sufficiently robust to generalize for unseen inputs.

2.1. Transfer learning

Deep learning methods excel at learning from large number of supervised examples, however, these methods do not typically generalize well for smaller datasets. For such cases, transfer learning helps through facilitating the transfer of knowledge from one task to another. This transfer of knowledge occurs in the form of initializing the weights of the network with values extracted from one previously learnt on a task that is usually more complex. A more complex dataset facilitates learning a more diverse set of discriminative features, thereby making the representation space richer. This generally helps in computer vision as kernels in the initial convolutional layers learn to generate high level features which are usually common across task (Yosinski et al., 2014) and reusing them helps to make optimization faster and better. For the geophysical problem studied in this paper, we use the pre-trained neural network weights and biases from the ImageNet task (Russakovsky et al., 2015) which used natural images divided into 1000 categories. Networks trained on ImageNet are generally rich in terms of features and

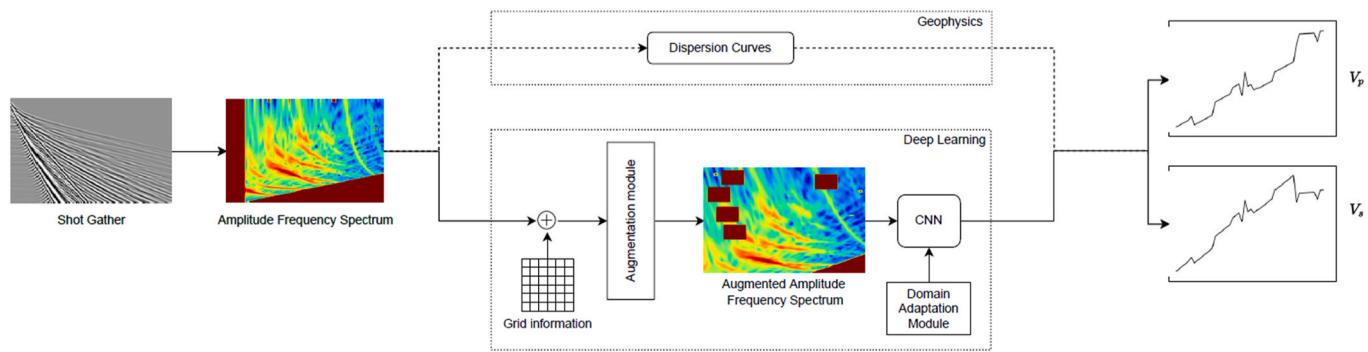


Fig. 1. Schematic representation for the estimation of V_p and V_s from shot gather based on the standard pipeline of involving estimation of dispersion curves followed by additional geophysical processing and inversion steps, and our deep learning-based data-driven pipeline. Note here that the domain adaptation module is optional and is only needed and we discuss both cases of with and without it in the paper.

have been deployed for various applications (Razavian et al., 2014; Liu and Deng, 2015). Although seismic data and natural images differ significantly in terms of the distributions in the space, it is of interest to explore whether their latent spaces have any overlap and whether transfer learning from ImageNet can help for our cause.

2.2. Artificial image augmentations

Deep neural networks rely heavily on big data to avoid the issue of overfitting. Unfortunately, for most applications, including geophysics, labelled data is relatively scarce, and data augmentation serves as a data-space solution to overcome this issue. Referred as image augmentation for our problem, it is a technique to create variation in existing images to artificially increase the size of dataset. We apply affine transformations to images such that the ground truth is not affected. Augmentation techniques that we explore in this paper are random cropping, random translations, random dilations, random rotations and random masking. These help the deep learning model by improving generalization and thus in turn reduce over-fitting. We briefly discuss these augmentations below.

2.2.1. Random cropping

This is an augmentation technique wherein random subsets are created from the original image. The input image is scaled to a larger size, followed by randomly cropping a part equal to the original size. An

example is shown in Fig. 2 (b). On conventional computer vision problems, this approach helps the model to generalize better because the object(s) of interest that are to be learnt by our model are not always wholly visible in the image or at the same scale in our training data. Random crops are a good choice for learning in such cases.

2.2.2. Random masking

It involves adding random patches of null or 0 values to the original image to mask information in those parts. An example is shown in Fig. 2 (c). This helps the network to not over-fit on certain regions and use whole feature space effectively. Specifically, for this example it acts as an additional denoising approach, as only the high amplitude values following dispersion curves contain predictive information about the subsurface model (red colors). The green/yellow/blue colors are lower amplitude values that are less relevant or irrelevant for the velocity prediction.

2.2.3. Random translations

This transformation involves simply moving the original image along f - or v -directions or both on the f - v grid space. From a general perspective, important features relevant to an image may not be localized in only a certain part of it, thus it's a good strategy to force CNNs to look around in all parts of the image. Inducing artificial augmentations to enrich the training dataset helps this cause. Mathematically, this operation can be represented as

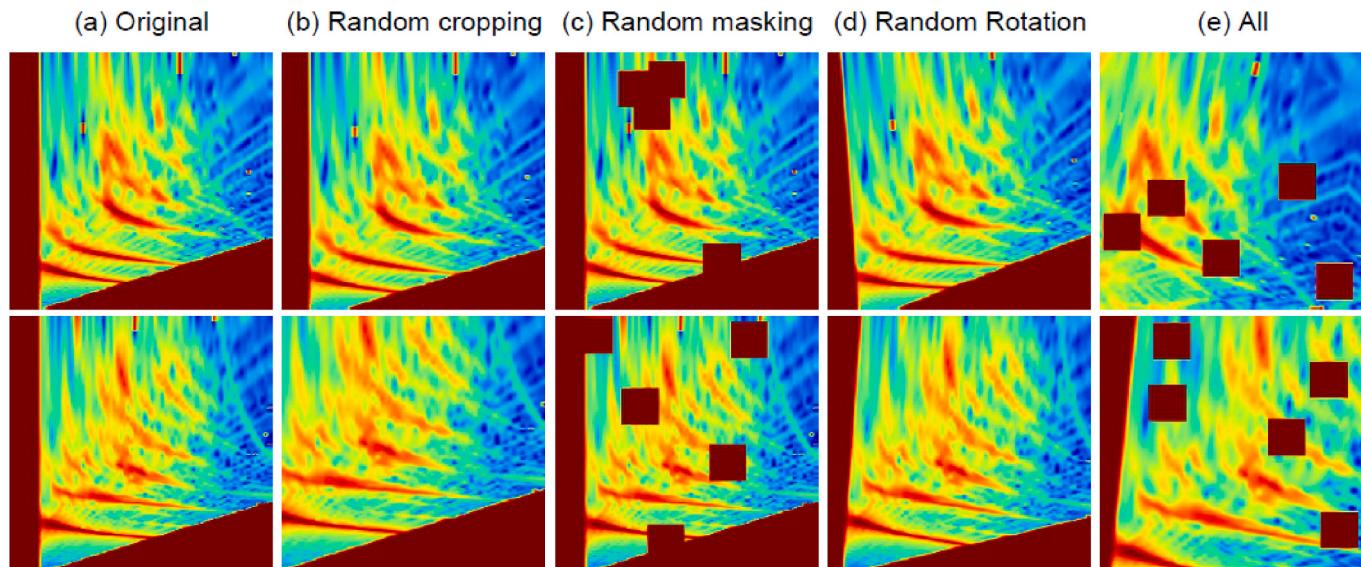


Fig. 2. Example original amplitude-frequency spectra and their augmented variations obtained with random cropping, random masking and random rotations.

$$\begin{bmatrix} f_{new} \\ v_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} f_{new} \\ v_{new} \\ 1 \end{bmatrix} \quad (3)$$

2.2.4. Random rotations

Input image is rotated in clockwise or anti-clockwise direction around its center by a random angle θ to produce the augmented image as shown in Fig. 2 (d). Important to note is that this operation does not preserve the dimensions of the original image. If your image is a square, rotating it at right angles will preserve the image size. If it is a rectangle, rotating it by 180° would preserve the size. To circumvent this issue, a crop of original size is taken from the rotated image and the empty region is filled with zeros. Mathematically the rotation operation can be defined as

$$\begin{bmatrix} f_{new} \\ v_{new} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} f_{old} \\ v_{old} \end{bmatrix} \quad (4)$$

2.2.5. Random dilations

This transformation involves scaling the original image inward or outward. For outward scaling, the final image size of the image is larger than the original image size. Following this, a sub-image of the size of the original image is cropped. Similarly, scaling inward reduces the size of the original. The empty pixels around the boundary in this case are filled with zero. Note that all the aforementioned augmentation methods are also applied to the coordinate channels, so the absolute and relative positions of the input data are preserved regardless of the rotation and dilation.

2.3. Domain adaptation

A very common problem with deep neural networks is their lack of generality on out-of-distribution datasets. In other words, a model trained on even very large training data can still fail easily, if the distribution of the validation set is different from the training set. As stated earlier, a common assumption for the conventional deep learning pipeline is that the underlying distribution of the training and test sets are similar. From the practical point of view, it implies that the training set should be sufficiently rich, and should comprise samples from all possible scenarios. However, for geophysical problems, it is not realistic since data samples coming in from a different survey can be significantly different from the data used to train the models. We propose here a domain adaptation strategy for training our deep learning networks, which allows our networks to fit relatively better on the out-of-distribution (OOD) samples at the cost of small drop in performance on in-distribution (ID) samples. Abbreviated as p-CNN, this approach takes a priori knowledge on the permissible range of V_p and V_s as additional input to learn the mapping function of a convolutional neural network (CNN). This input can vary significantly across data distributions, and providing it as an additional input induces invariance into the CNN with respect to the mapping range for V_p and V_s while preserving the overall spatial trend of the output. An understanding of this can be obtained from the schematic representation shown in Fig. 1. We later describe in the appendices of this paper how this information is embedded into the original network. Two modifications are proposed to the original network in this regard. First, we alleviate the one-to-one mapping between the input and output through adding random shifts in the output V_p and V_s curves. This forces the network to not learn very strictly the range of values for V_p and V_s for every input phase velocity spectrum. Rather the focus is only on learning the overall trend of the output curves. Second, the approximate ranges of V_p and V_s are explicitly provided to the network, and the network learns during the training process to fit the output curves within these ranges. With this modification, we hope that if the network has learned the trend of the output curves, it can map on out-of-distribution samples as well when the additional information on the range is provided to it.

Mathematically, the implementation of p-CNN can be expressed as follows. For the training dataset D stated earlier, p-CNN learns a mapping function $g(.)$ such that

$$g(x, y_{min}, y_{max}) = y, \quad (5)$$

where x and y denote the input and output of the network, respectively, and y_{min} and y_{max} denote lower and upper bounds on the output, respectively. As stated earlier, it is assumed that y_{min} and y_{max} serve as a priori information obtained other sources and are used to force the model to predict a feasible local solution within the bounds.

2.4. Synthetic training dataset

The synthetic dataset consists of a total of 7000 shot gathers, which provides us 7000 phase-velocity spectrum images. The data was generated with 2D elastic finite-difference modelling (Thorbecke and Draganov., 2011) from 1D models with a random number of layers, with random thickness. The layers were assigned random interval V_p values. The assigned interval V_p velocity values are defined as random perturbations on a V_p velocity trend with depth. Those velocity trends are based on the average V_p values we have observed on the uphole data acquired along the real seismic line. Per layer we randomly assign a V_p/V_s ratio in the range indicated in Fig. 3 and this is used to define an interval V_s . Of this dataset, we use 5600 shots for model training and the rest 1400 for validation. Since this validation set is created using random splitting of the 7000 samples, it corresponds as in-distribution (ID) dataset. The training and ID validation set contain V_p and V_s values in the ranges of 349–1950 m/s and 223–1327 m/s, respectively. In addition, we also use an out-of-distribution (OOD) validation set. The ID and OOD are defined based on the range of V_p/V_s values in the respective datasets. Fig. 3 shows the distributions of V_p , V_s and V_p/V_s samples for the ID and OOD samples, and the distinction between ID and OOD is clear from the V_p/V_s distributions.

2.5. Evaluation metrics

To evaluate the performance of the trained deep learning models, we use two evaluation metrics: mean squared error (MSE) and mean absolute error (MAE). Mathematically, MAE can be stated as

$$MAE_{V_p} = \frac{1}{n} \sum_{i=1}^n |V_p^{pred} - V_p^{gt}| \quad (6)$$

$$MAE_{V_s} = \frac{1}{n} \sum_{i=1}^n |V_s^{pred} - V_s^{gt}| \quad (7)$$

$$MAE_{tot_p} = MAE_{V_p} + MAE_{V_s} \quad (8)$$

here, MAE_{tot} denotes the total MAE score for p-wave and s-wave velocities combined. Similarly, for mean squared error, we have

$$MSE_{V_p} = \frac{1}{n} \sum_{i=1}^n (V_p^{pred} - V_p^{gt})^2 \quad (9)$$

$$MSE_{V_s} = \frac{1}{n} \sum_{i=1}^n |V_s^{pred} - V_s^{gt}| \quad (10)$$

$$MSE_{tot_p} = MSE_{V_p} + MSE_{V_s} \quad (11)$$

2.6. Implementation details

2.6.1. Model architecture

A first application of this idea used a encoder-type sequence of residual blocks, followed by a fully connected neural network as described in Zwartjes (2020). We achieved better results with a pre-trained

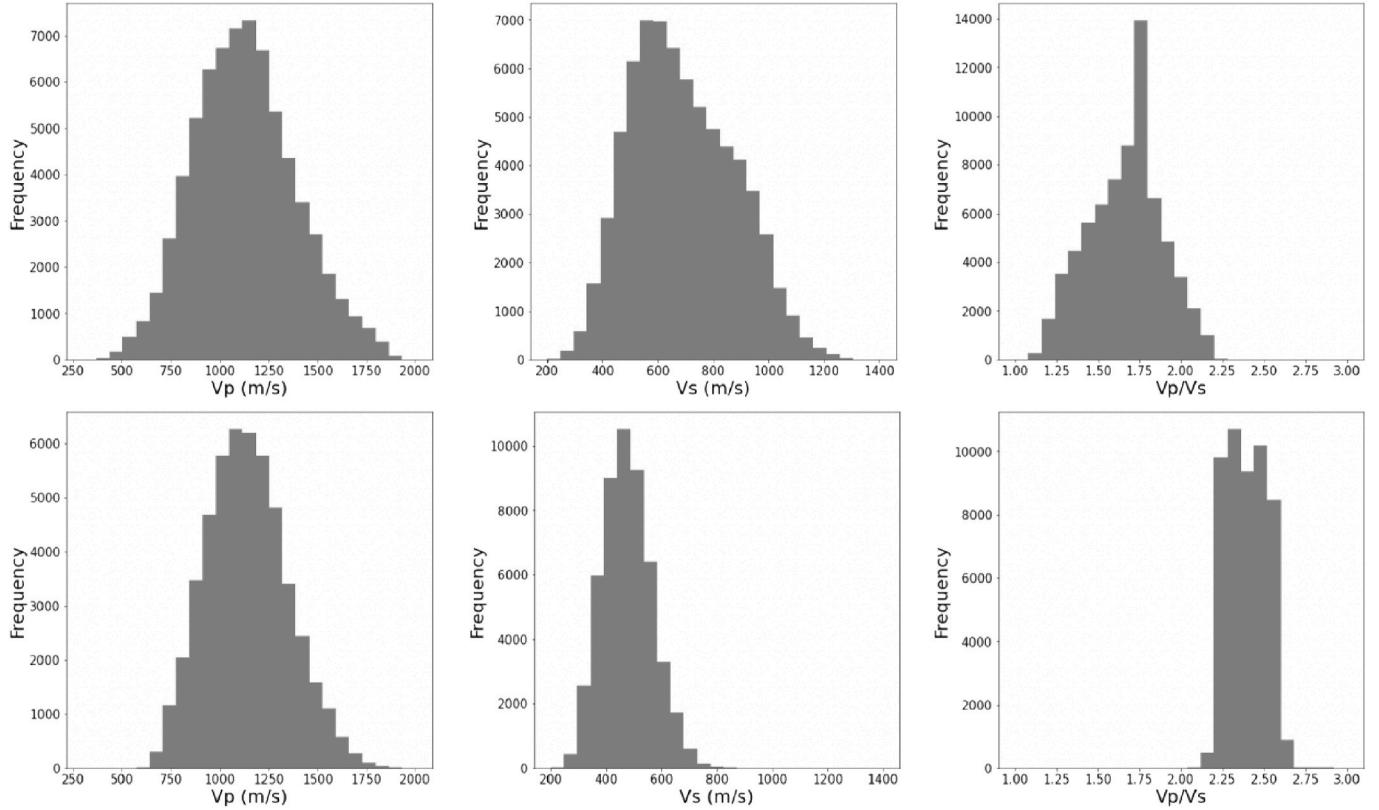


Fig. 3. Histograms showing distribution of V_p , V_s and V_p/V_s for the in-distribution (ID) (top) and out-of-distribution (OOD) (bottom) validation samples. Note that ID and OOD samples are chosen based on the overlap of V_p/V_s with that of the training samples, where the ID samples are chosen randomly from the train set and have a similar distribution.

ResNet-18 architecture. For all our experiments, we use residual networks (He et al., 2016). These are chosen as backbone for learning high dimensional features due to their capability of solving gradient vanishing problem with the use of skip connection. For an understanding on the underlying mechanism that makes residual networks successful, see the appendix of this paper. We have an 18-layer residual network as our backbone contributing to 1.8 billion FLOPs. To adapt to our task, we have replaced the fully-connected layer of our backbone with two linear layers for V_p and V_s , respectively, each of input and output dimensions of 512 and 50, respectively. All experiments in the paper use ResNet18 with customized fully-connected layers at the end for predicting V_p and V_s . More details can be obtained from the schematic diagram shown in Fig. A-1. For the domain adaptation module, the architecture differs slightly, and we describe it below.

2.6.2. Domain adaption architecture

For this task as well, the backbone architecture is ResNet18. For adding the prior knowledge on V_p and V_s bounds, we have two meta architectures. Each of these comprises one linear layer, a batch-normalization layer and a ReLU activation module. For V_p as well as V_s , the input comprises 2 neurons each, expressing the upper and lower bounds, and the corresponding meta-architectures encode them into 50 neurons. The output 50 neurons from both architectures are then concatenated to the output of the CNN backbone and fed into the two linear heads to compute V_p and V_s .

2.6.3. Training and inference

For training as well as inference, we use 3-channel images, where first channel includes amplitude-frequency spectrum for a shot gather and the other two channels contain x- and y-related grid information. To incorporate transfer learning, we use pretrained ResNet-18 with pre-training done on ImageNet dataset. Two regression heads are added for

predicting V_p and V_s and each of these contain 50 neurons. For optimization, we use Adam optimizer with hyperparameter setting as $\beta_1 = 0.9$, $\beta_2 = 0.99$, $\varepsilon = 10^{-8}$. The learning rate is initialized as 10^{-3} and updated based on the ‘ReduceLROnPlateau’ policy. All the models are trained for 500 epochs, and a new model is saved whenever there is an improvement observed on the ID validation set. During inference, no augmentations are used. For the domain adaptation module, training and inference are performed in different ways. During training, we take the minimum and maximum values of V_p and V_s and perturb them with additive noise of up to 100 m/s. Further, a constant shift of up to 500 m/s is added to V_p and 80% of the sampled value for V_s to induce the invariance in the network. During inference, the minimum and maximum values of V_p and V_s are directly used as bounds for input to the network.

3. Results

3.1. Baseline model

Results for 30 V_s velocity profiles are shown in Fig. 4. The plots show the interval V_s velocity as a function of depth, from the surface, on the left-hand side of each figure, to bottom on the right-hand side. The results demonstrate that the trends of each profile are captured quite well, but the network can only predict a smooth version of the true model and completely misses the high-frequency variation in interval velocity vs. depth. Another goal of this study is to demonstrate that deep learning, as well as additional operations such as augmentation and transfer learning help to estimate near-surface velocities in a data-driven manner. In this regard, it should be sufficient to show that the error in predictions is lower than that of a random baseline. Thus, we use the mean of the sample values at very point as the baseline and refer to it as random baseline. We discuss below results for the in-distribution (ID) as well as

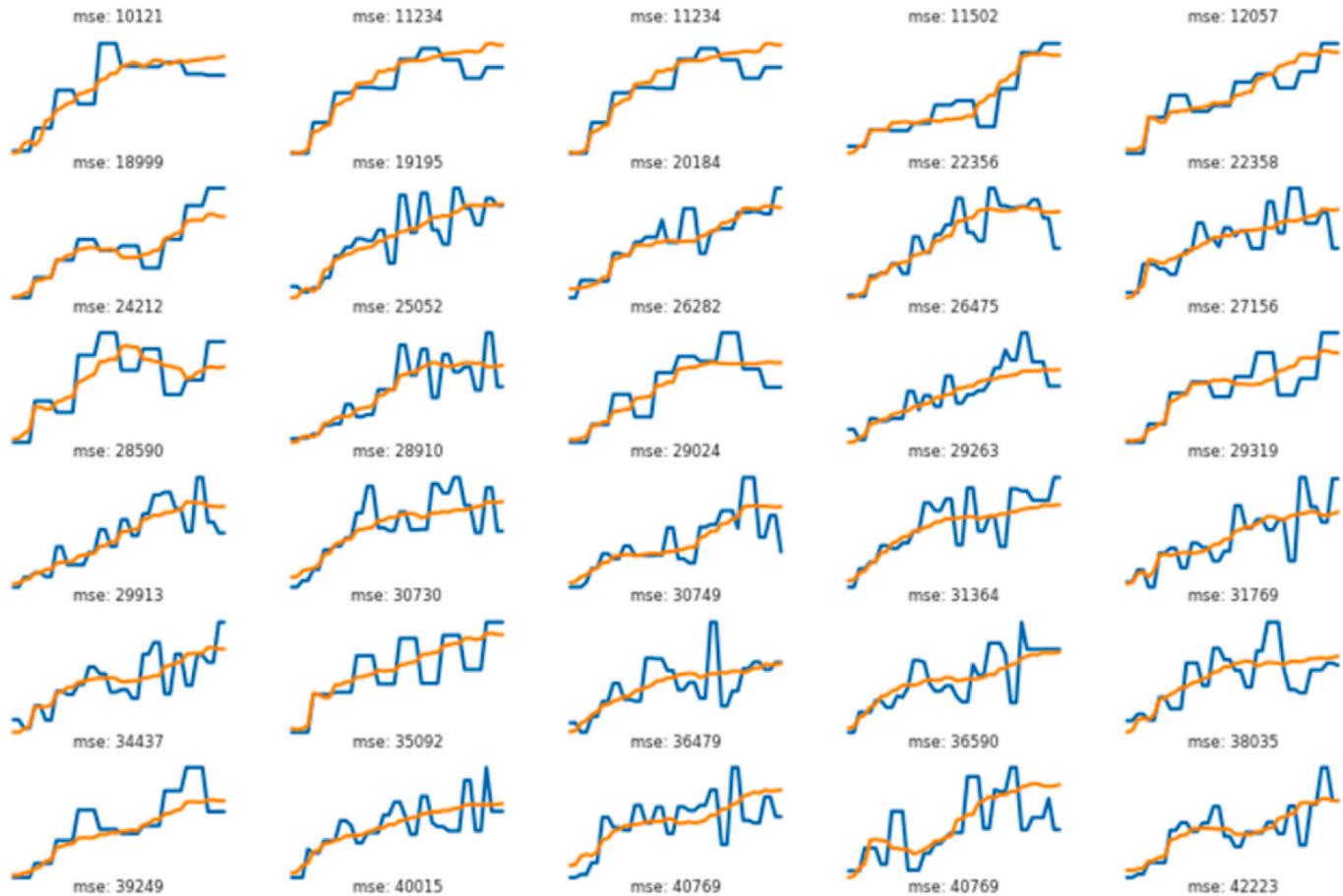


Fig. 4. True (blue) vs. predicted (orange) V_s velocity profiles for 30 synthetic examples. The plots show the interval V_s velocity as a function of depth, from the surface to bottom (left to right) of each profile. The trend of each velocity profile is captured quite well, but the network can only predict a smooth version of the true model and completely misses the high-frequency variation in interval velocity vs. depth.

out-of-distribution (OOD) samples for the simulated dataset. Related results are reported in Table 1.

Compared to the random baseline, we show that all variants of the presented deep learning model show significant improvements in MAE as well as MSE for the predicted velocities. This confirms that it is indeed possible to learn the mapping from the amplitude-frequency spectrum to V_p and V_s , and the fact that these reductions in error are generally more than 70% for MAE and as much as 80% for MSE confirm that the models are able to learn this mapping very well. This is further evident from the examples presented in Fig. 8. We see here while the random baseline is merely a mean of the training data samples at every point of the sequence, the results of deep learning tend to model the inherent trend

of the data. In the later results, we analyze how different variants of our deep learning solution affect the overall prediction.

3.2. Transfer learning

The cases of transfer learning are reported as ‘pretrained’ in Table 1. As can be seen, when models are trained first on ImageNet dataset already, the results on our dataset are slightly more improved and this is more prominent for the OOD samples. As has been motivated earlier, ImageNet is a very diverse dataset facilitating to learn a rich set of features. This added knowledge helps to improve the convergence of the model on our dataset, thus making it easier to generalize better on OOD

Table 1

Error measures expressed in terms of mean absolute error (MAE) and mean-squared error (MSE) for the prediction of V_p and V_s from amplitude spectra using several variants of deep learning models for in-distribution (ID) and out-of-distribution (OOD) datasets. All CNN models reported here use ResNet-18 architecture. Suffix ‘-aug’ states that the augmentation module is active and each input image undergoes the employed transforms. For pretrained cases, models trained on ImageNet dataset have been used. Further the prefix ‘p’ implies that the domain adaptation module has been used.

Approach	in-distribution data				out-of-distribution data			
	V_p (mae)	V_p (mse)	V_s (mae)	V_s (mse)	V_p (mae)	V_p (mse)	V_s (mae)	V_s (mse)
Random baseline	164	40,920	113	20,288	170	43,921	209	57,485
CNN	56	5501	30	1607	100	21,871	75	9956
CNN-aug	51	4630	27	1351	92	20,041	70	8925
CNN (pretrained)	55	5260	29	1549	94	20,359	73	9245
CNN-aug (pretrained)	50	4557	26	1324	92	19,999	64	7225
p -CNN	62	6363	33	1858	75	8772	52	4914
p -CNN-aug	61	6179	33	1815	69	7619	43	3408
p -CNN (pretrained)	61	6230	32	1821	72	8312	43	3148
p -CNN-aug (pretrained)	57	5535	30	1608	66	7147	42	2994

samples.

3.3. Augmentations

The effect of artificial augmentations is clearly evident from the reduction in errors in Table 1. Compared to the non-augmented versions, we see that adding the augmentations to the training process improves the performance of the model for all the scenarios. We see relative reductions of more than 5% for almost all cases. It clearly shows that adding the augmentations during the training makes the model more robust, thereby improving its generalization on the test set. Examples reflecting this improvement are shown in Fig. 8.

3.4. Domain adaptation

We discuss here the results related to using domain information in the model. As shown in Table 1, adding the domain information reduces the performance on the ID samples by a small margin, however, this in turn leads to significantly improved performance on the OOD samples where relative error reductions are around 35% for each of the cases. As

shown in Fig. 8, adding the domain information helps to bring the predicting velocities closer to the ground-truth while also accurately predicting the shapes correctly.

3.5. 1D vs 2D earth models

As was mentioned in Section 1.2, the method of V_p and V_s velocity estimation from phase velocity vs. frequency panels is based on the assumption of 1D models, i.e. laterally homogenous. The synthetic training data was created through elastic finite-difference modelling using 1D, laterally homogeneous earth models as input. As a test for robustness, we have modelled seismic shot gathers along a simple, 2D laterally inhomogeneous medium consisting of 4 layers. Three layers are laterally homogeneous, while the 2nd layer from the top has a velocity that increases along the line from 500 to 800 m/s. The top row in Fig. 5 shows the true velocity profile and the predicted velocity profiles obtained by applying the trained neural network to each phase velocity vs. frequency spectrum of derived from the synthetic shot gathers. The bottom row shows a 1D vertical profile in the center of the models, with at each depth a box-plot of the velocity range. The depth sampling is 2m.

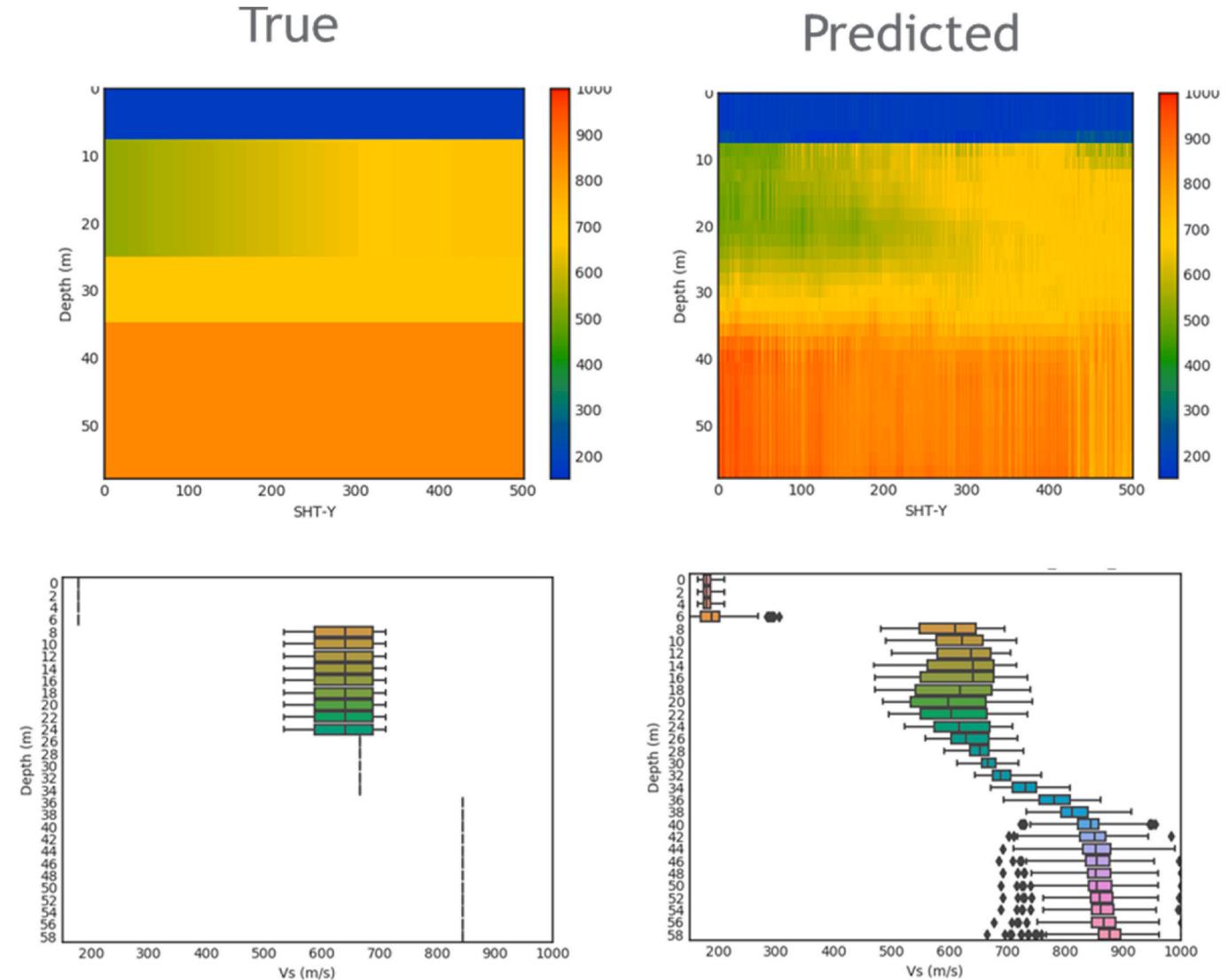


Fig. 5. The true (left) and predicted (right) velocity profile for the case of seismic data generated from laterally inhomogeneous earth models. The 2nd layer from the top is laterally inhomogeneous whereas the other layers are all of constant value. The neural network was trained on data generated from 1D earth models, but was applied to 400 phase velocity vs. frequency spectra derived from the modelled seismic data. The bottom row shows a 1D vertical profile in the center of the models, with at each depth a box-plot of the velocity range. The depth sampling is 2m.

The wavelet used in modelling was the same for both training and testing data. Although the prediction shows more variation than the true model, the trend and range of predicted values lie close to the true values.

3.6. Real land seismic line

The real data used in this study corresponds to an 80 km land seismic line, consisting of 6104 shots spaced 25 m apart. The receiver spacing per shot was 15 m. Along the line, uphole velocity measurements were recorded at five locations. These were converted to V_p interval velocity to compare with the neural network predictions. The pre-processing applied to the seismic line was a simple $t^{0.25}$ gain and offset were limited to 2000 m so that we have 133 channels per shot. The phase velocity vs. frequency panels were constructed using a 2D Fourier transform and stretching the k-axis using $v = \omega/k$. The maximum frequency used in the panels is 25 Hz, the maximum phase velocity is 1850 m/s. Subsequently the phase velocity V_s , frequency panels with a running average of 31 shots (775 m). Fig. 6 shows a seismic shot and the raw and processed phase velocity vs. frequency panels. Fig. 7 shows histograms of the interval V_p , V_s and V_p/V_s of the training data and the predicted data. The range of velocities used to create the training data was determined by velocities suggested by the uphole measurements. Fig. 9 shows the predicted V_p , V_s and V_p/V_s along the seismic line. The predictions were made using velocity constraints during prediction which were obtained from the input data. The minimum expected V_s per gather was defined as the minimum observed phase velocity in the input data, whereas the maximum V_s velocity was simply set to a value 1000 m/s higher. The V_p constraints were defined as 1.8 times the V_s constraints, a number obtained for other near surface work in the area. Fig. 10 shows a comparison of the predicted V_p from Fig. 9 and the V_p predicted by a ‘conventional’ dispersion curve inversion workflow in a study by Rovetta et al. (2020). Note the good agreement between the two methods, in terms of lateral and vertical velocity variation of overall range of velocity values. The bottom image in Fig. 10 shows the comparison of the neural network V_p predictions using a network trained without velocity constraints. These predictions lack the lateral velocity variations and show a mismatch in range of predicted velocities. Finally, Fig. 11 shows the phase velocity V_s , frequency spectra at the locations’ uphole measurements and, in the middle row, the predicted V_p and V_s velocities using the constrained network and the bottom row the predictions using the unconstrained network. The uphole interval velocity derived from the pairs of uphole depth and times is also shown. Although there is not a perfect match, we see overall a better agreement between the predictions from the constrained network and the uphole velocities.

4. Discussion

In this article we have explored the feasibility of predicting a shallow 1D velocity profile from a phase velocity vs frequency spectrum of surface waves. The results show that this is feasible but that a smooth version of the interval velocity will be predicted. This is similar to the results of conventional dispersion curve inversion techniques, since Socco and Strobbia (2004) showed that an inverted shear velocity model is typically much smoother than the actual velocity model. Instead of

predicting interval velocity profile, we should aim to predict the average velocity profile since this is smooth by definition. Although the training data was created from 1D earth models, we obtained encouraging results in the application to synthetic data created from an earth model with a velocity layer that gradually changes in the horizontal direction.

We have taken care in designing the training data to stay close to the velocity profiles expected on the real seismic line. It is well known the neural networks typically generalize poorly and do well only when the new data is similar to the training data. In that sense, the neural network approach lacks the flexibility of the model-based inversion approach. Rather than attempt a brute force modelling to cater for all possible 1D earth models, we would advocate a physics-based approach where an additional loop is included to verify whether the predicted models are physically realistic. As an intermediate solution we have explored the use of data augmentation techniques to increase model robustness. These augmentation techniques are common place in the field on computer vision, and result on improve prediction metrics. Further improvements may be obtained by geophysically inspired augmentations, such as variations in noise, frequency content, seismic wavelet, etc. Our first architecture was a encoder-style convolutional network, but better results were obtained with a standard ResNet-18 network. Additional improvement may be obtained by exploring for a more optimal neural network architecture.

5. Conclusion

In this paper, we have presented a deep learning methodology to estimate near-surface velocities from shot gathers in a data-driven manner. We have computed them from phase velocity-frequency panels. We analyzed the influence of some of the popular operations (such as transfer learning, data augmentations) commonly employed in deep learning for the problem of velocity estimation. Through numerical experiments on a simulated synthetic dataset, we have demonstrated that transfer learning as well as data augmentations are helpful when using deep learning for velocity estimation. We have further addressed the critical issue of lack of generation of deep learning models on out-of-distribution samples in the context of the posed problem. Through numerical experiments we demonstrated that using a priori information on the velocity bounds can help to improve the convergence of the learning process on OOD samples. We applied the proposed method to phase velocity spectra from a real land seismic line and produced a near-surface velocity model that is similar to a velocity model obtained via conventional dispersion curve inversion. Our results are encouraging and demonstrate that such an approach could be of value for other geophysical problems as well, where application of empirical models on new OOD datasets has always been a challenge. Overall, based on the results and the associated discussions presented in the paper, we conclude that deep learning could be a potential tool to create proxy models for near-surface velocity estimation, and more research in this direction should be of interest.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

APPENDIX. NETWORK ARCHITECTURE

One of the biggest challenges in training deep convolutional neural nets is the issue of vanishing gradients. This issue along with accuracy saturation problem associated with training deep neural networks are circumvented with the use of residual networks. Residual networks use skip connections or shortcuts which act like highway for gradients and features between layers (Fig. A-1). These shortcuts generally skip over two or three convolutional layer which contain batch normalization and nonlinearities (ReLU) in between them. An understanding of this can be obtained from the equation stated below. Residual block is defined as $f_i(\cdot)$ consisting of double- or triple-convolution layer with a batch-normalization layer and a ReLU

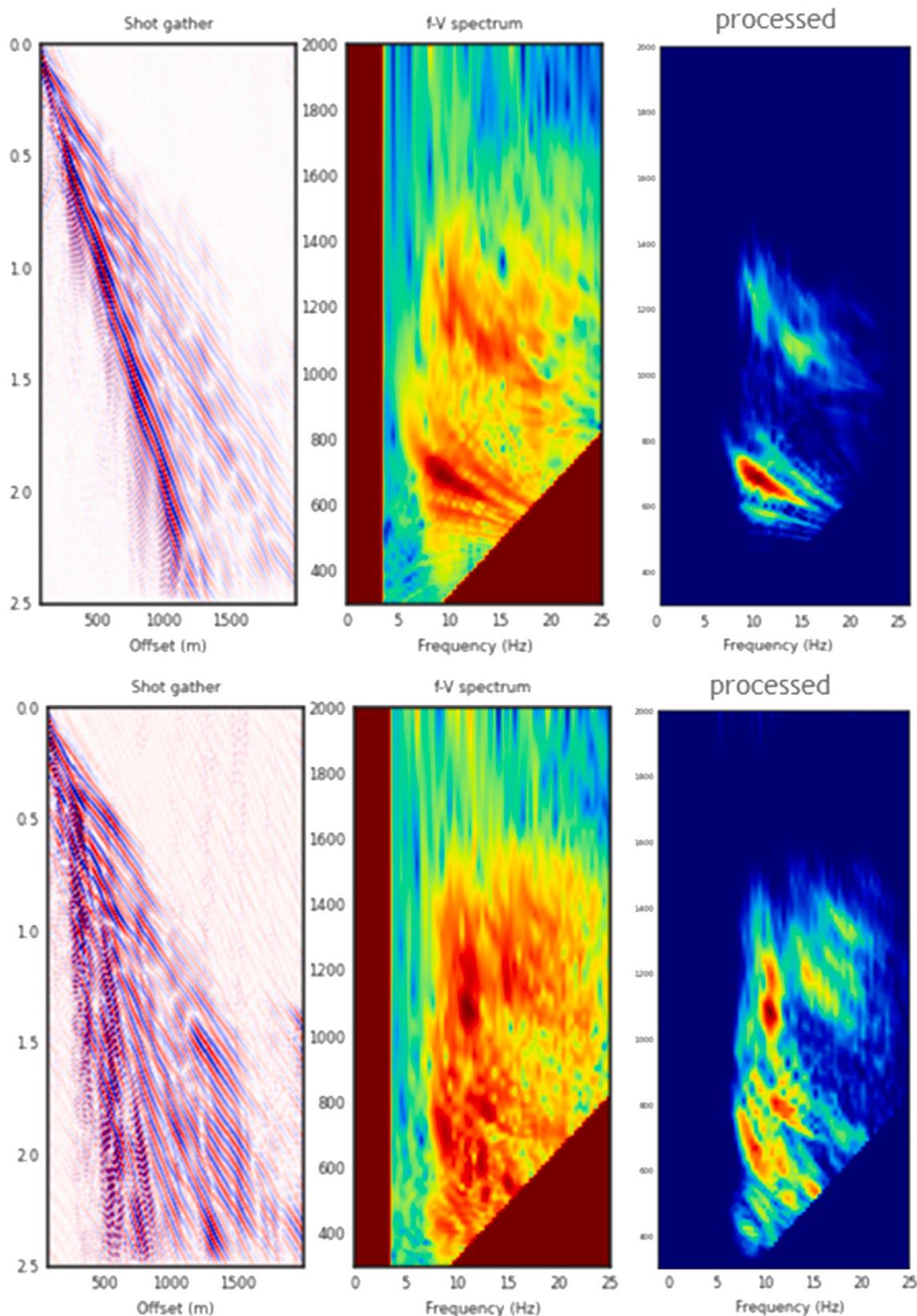


Fig. 6. Shows for two different locations the seismic shot gather (left), the raw phase velocity vs. frequency spectrum (middle) and the processed spectrum (right). The processing applied was a running average over 15 adjacent raw panels, followed by standardization with the mean and standard deviation from the synthetic data. A mute was applied to exclude velocity outside the expected range.

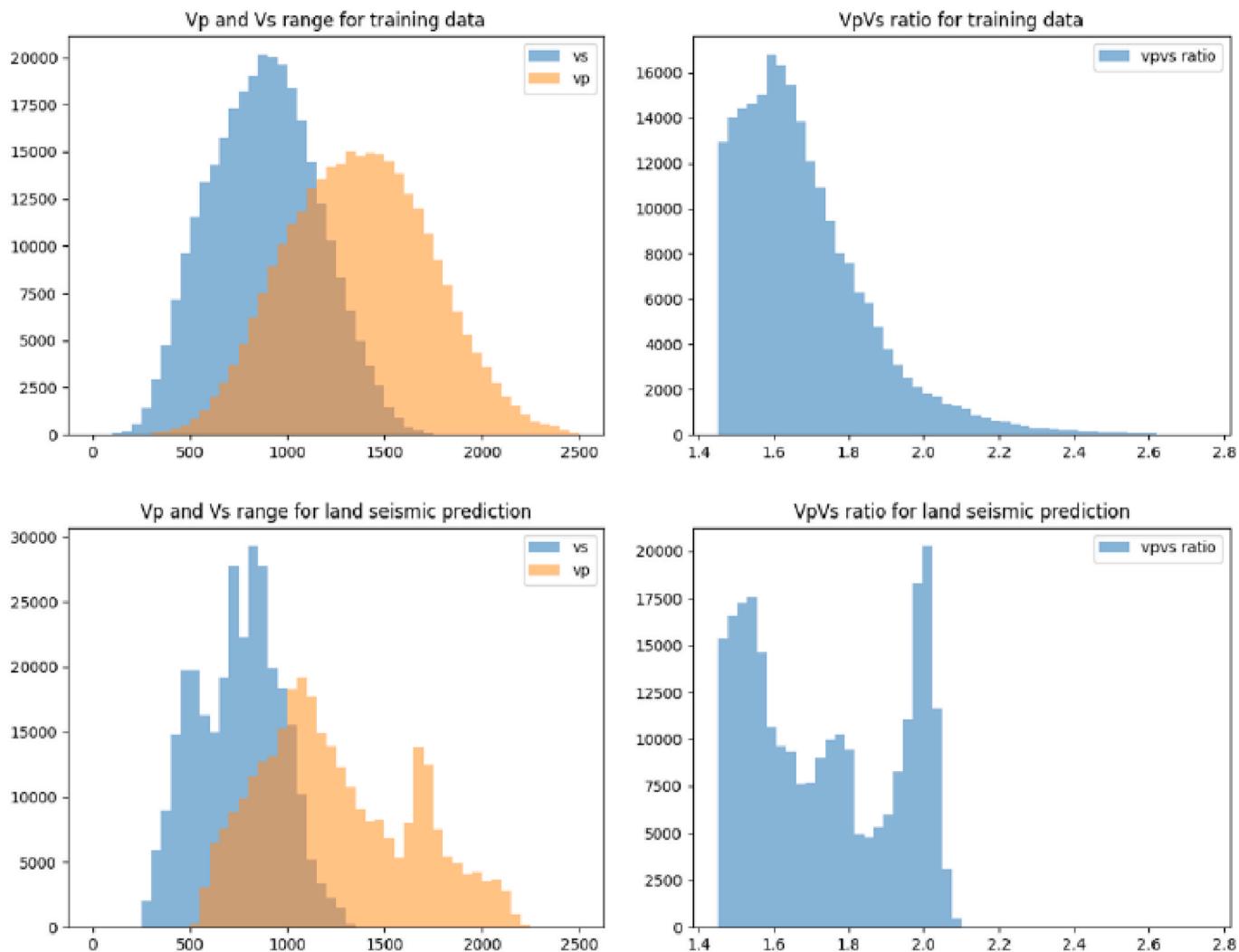


Fig. 7. Histogram of V_p and V_s velocities of the training data and the predictions for the land seismic line. The column on the right shows the V_p/V_s ratios.

activation function. Let, y_{i-1} is input to this residual block, and the output of the i th block is defined as

$$y_i = f_i(y_{i-1}) + y_{i-1} \quad (\text{A-1})$$

Addition of y_{i-1} to the output of i th block, enable to gradients of $(i-1)^{\text{th}}$ block to skip over the i th block solving the vanishing gradient problem. ResNet18. It is the 18-layer variant of the residual network (ResNet) as described in He et al. (2016). Network architecture is described in Fig. A-1. The ResNet18 architecture can be broken down in 4 blocks of 3×3 convolutions. In each block the width of convolutional layers is fixed i.e., 64, 128, 256, 512 respectively. In Fig. A-1, the bold skip connections refer to addition of features, whereas dotted skip connections are convolutional layers which are used to lower the features' height and width and increase number of channels. Further, each 3×3 convolutional layer is followed by batchnorm layer and ReLU activation function.

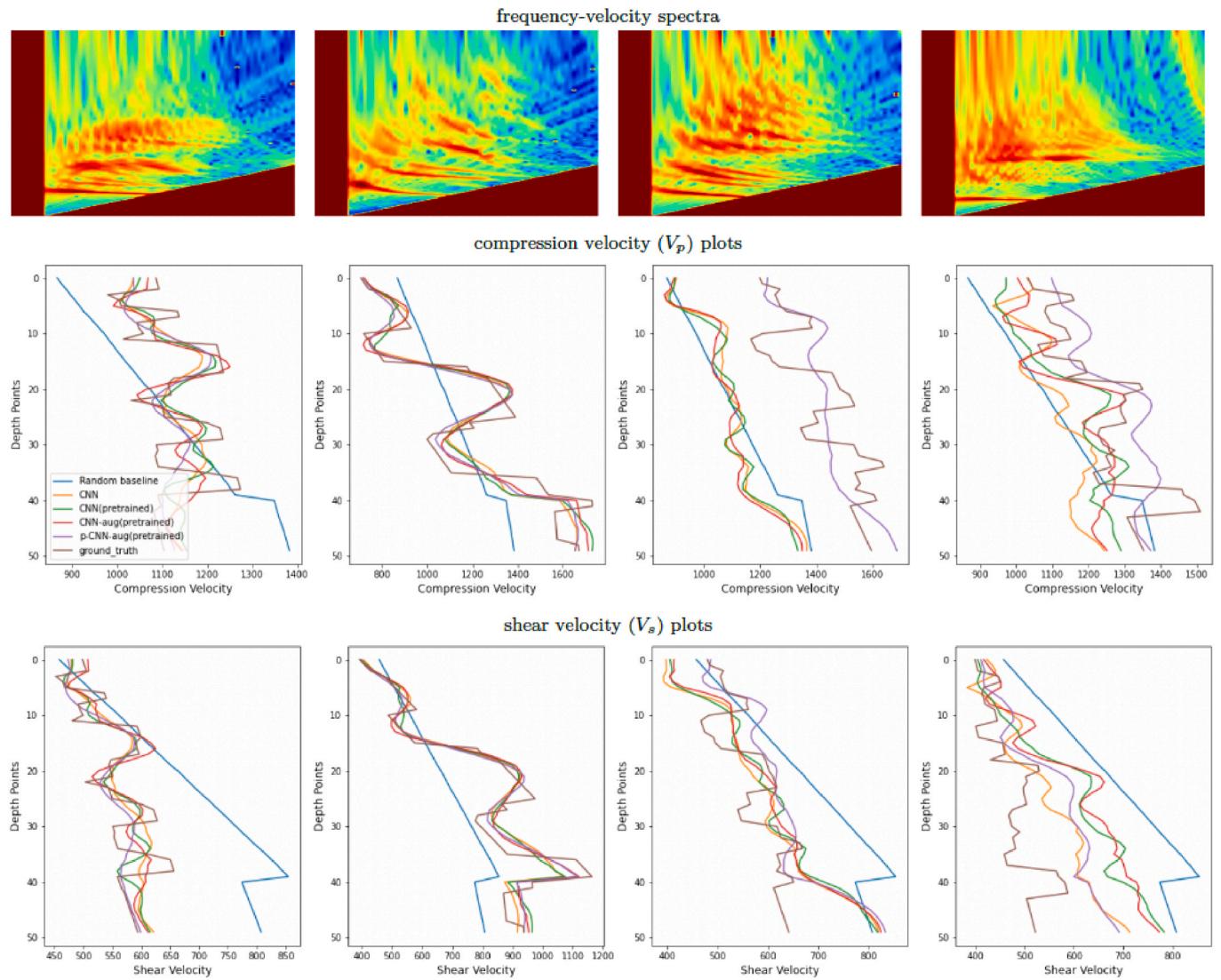


Fig. 8. Example input f-v spectra and the corresponding output V_p and V_s for two samples from in-distribution (first two columns from left) and two samples from out-of-distribution of the validation set (third and fourth columns from left) showing ground-truth as well as the velocities obtained with different variations of the proposed deep learning approach.

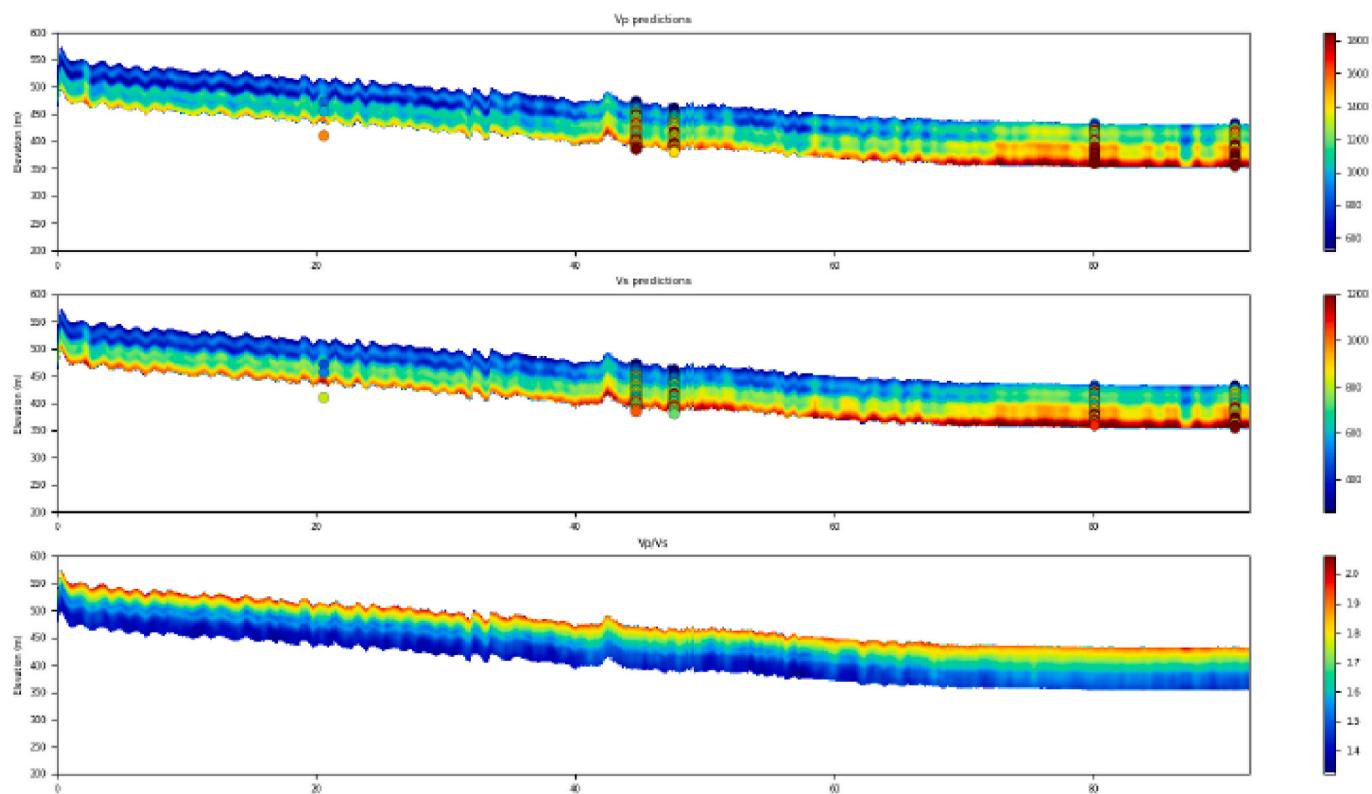


Fig. 9. Predicted V_p (top row) and V_s (middle row) interval velocities for the land seismic line. The bottom row shows the V_p/V_s ratio along the line.

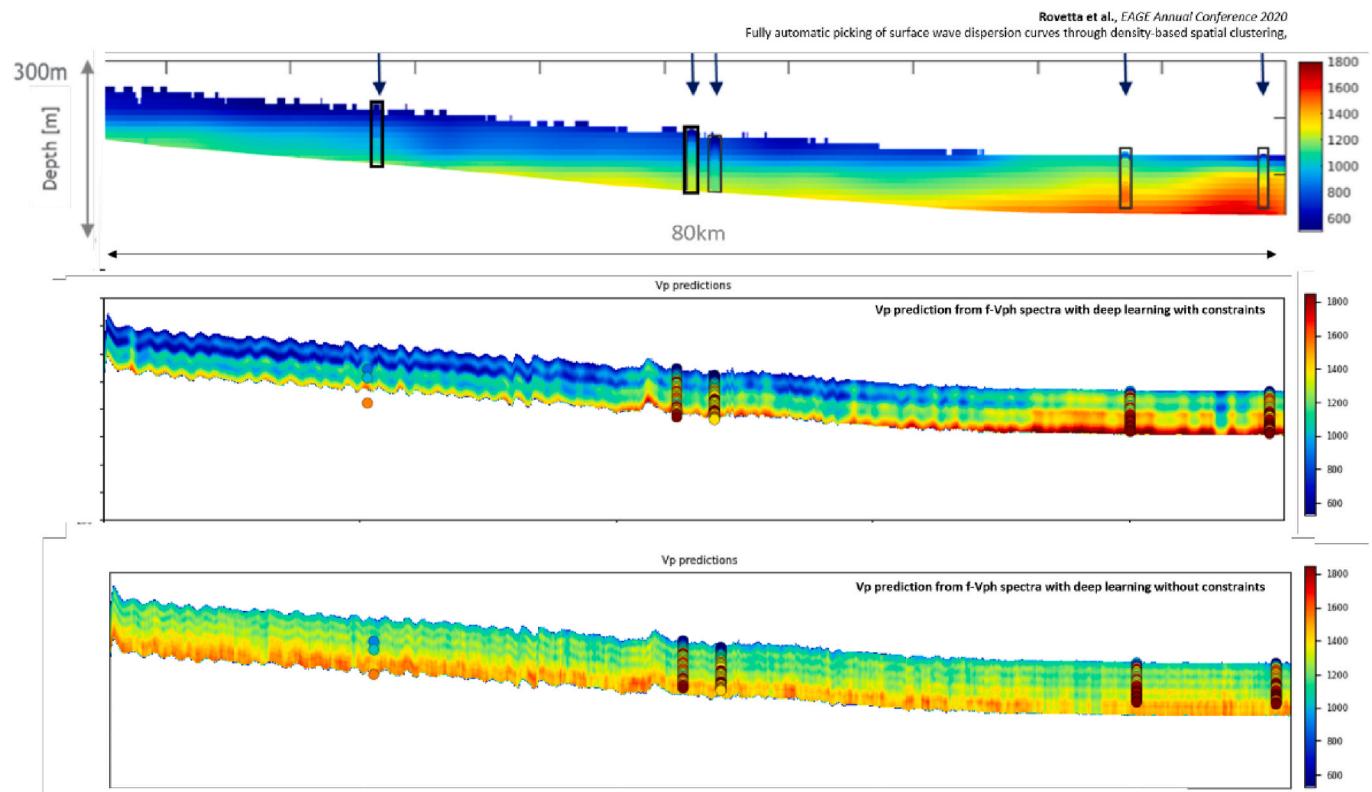


Fig. 10. Comparison of (top) the velocity profile obtained via dispersion curve inversion by Rovetta et al. (2020) vs. (middle) the result obtained using our deep learning model. The bottom figure shows the result with obtained with the model trained with velocity constraints.

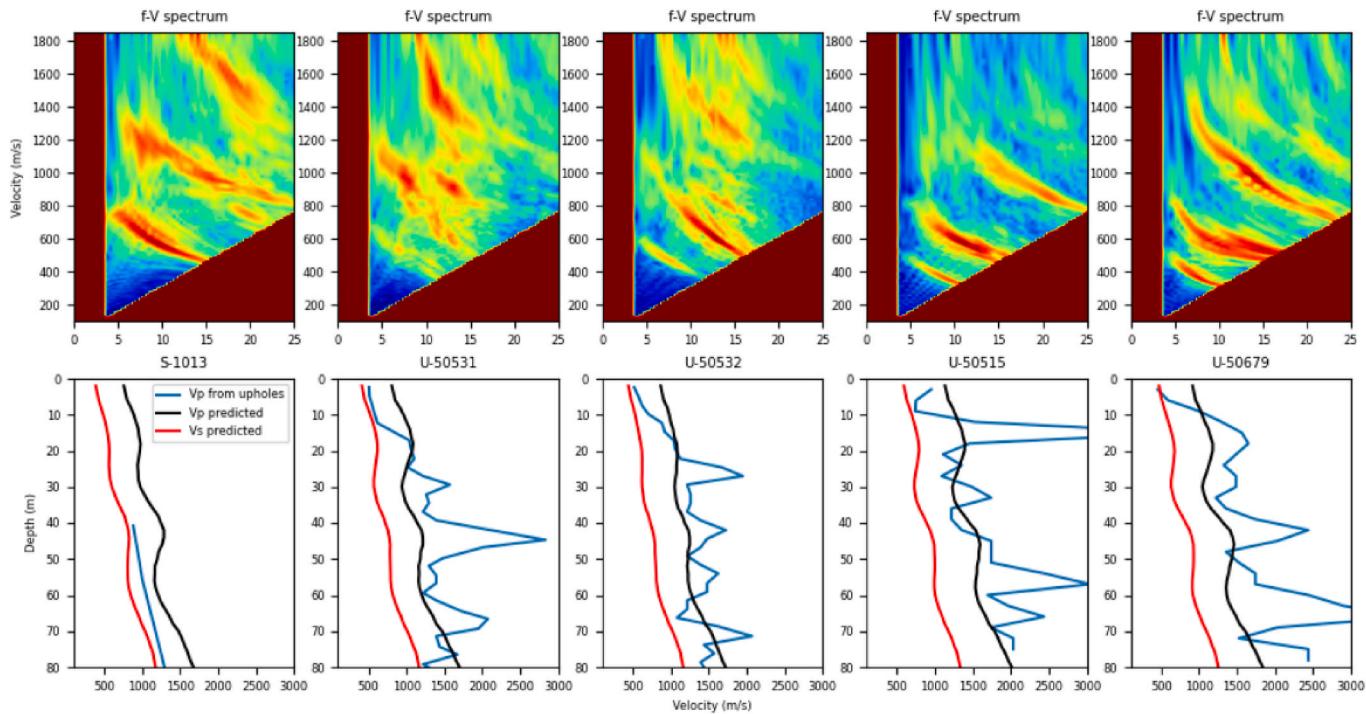


Fig. 11. The top row shows the phase velocity vs. frequency spectra at 5 locations along the land seismic line where uphole velocities were measured. Middle row shows interval velocity vs. depth at the locations of the uphole measurements. Bottom row shows average velocity recorded by the uphole velocity converted to an interval velocity.

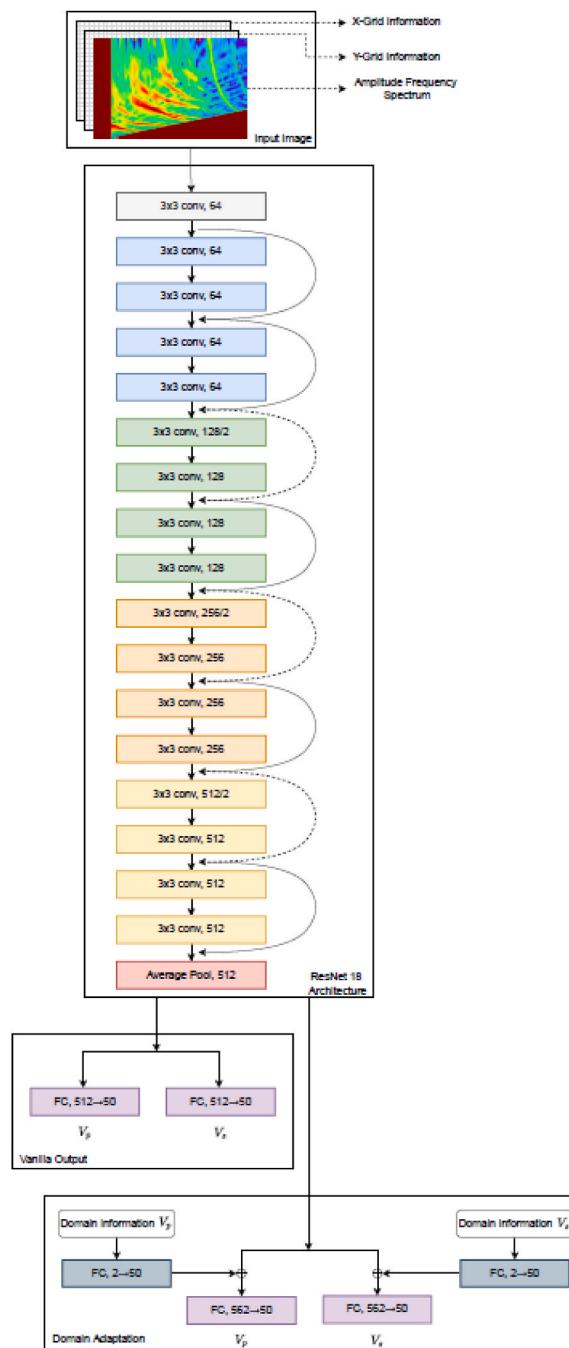


Figure A-1. Schematic representation of our deep learning framework used for the estimation and compressional and shear velocities from frequency-velocity spectrum. Our network comprises a ResNet18 backbone with two choices of the head network: a) a simple fully-connected network, and b) a fully connected network with additional input of domain information.

References

- Alsinan, S., Nivlet, P., Alghenaim, H., 2021. Automated Facies Prediction in Pro-glacial Channel System Using Deep Learning. SEG Technical Program Expanded Abstracts, pp. 1551–1555.
- Araya-Polo, M., Jennings, J., Adler, A., Dahlke, T., 2018. Deep-learning tomography. Lead. Edge 37, 58–66.
- Baardman, R.H., Hegge, R.F., 2020. Machine learning approaches for use in deblending. Lead. Edge 39, 188–194.
- Bai, T., Luo, J., Zhao, J., Wen, B., Wang, Q., 2021. Recent advances in adversarial training for adversarial robustness: proceedings of the thirtieth international joint conference on artificial intelligence. IJCAI-21, 4312–4321.
- Bhowmick, D., Gupta, D.K., Maiti, S., Shankar, U., 2018. Deep Auto-Associative Neural Networks for Noise Reduction in Seismic Data, 00291 arXiv preprint :1805.
- Biswas, R., Sen, M.K., Das, V., Mukerji, T., 2019. Prestack and post-stack inversion using a physics-guided convolutional neural network. Interpretation 7, SE161–SE174.
- Chamorro, D., Zhao, J., Birnie, C., Staring, M., Fliedner, M., Ravasi, M., 2022. Extracting Surface Wave Dispersion Curves with Deep Learning. Second EAGE Subsurface Intelligence Workshop. <https://doi.org/10.3997/2214-4609.2022616016>.
- Dai, T., Xia, J., Ning, L., Chaoqiang, X., Liu, Y., Xing, H., 2021. Deep learning for extracting dispersion curves. Surv. Geophys. 42, 69–95. <https://doi.org/10.1007/s10712-020-09615-3>.
- Das, V., Mukerji, T., 2020. Petrophysical properties prediction from prestack seismic data using convolutional neural networks. Geophysics 85, N41–N55.
- Das, V., Pollack, A., Wollner, U., Mukerji, T., 2019. Convolutional neural network for seismic impedance inversion. Geophysics 84, R869–R880.

- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. ImageNet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255.
- Duque, L., Gutiérrez, G., Arias, G., Rüger, A., Jaramillo, H., 2019. Automated Velocity Estimation by Deep-Learning Based Seismic-To-Velocity Mapping: 81st EAGE Conference and Exhibition 2019.
- Fabien-Ouellet, G., Sarkar, R., 2020. Seismic velocity estimation: a deep recurrent neural-network approach. *Geophysics* 85, U21–U29.
- Gatys, L.A., Ecker, A.S., Bethge, M., 2016. Image style transfer using convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2414–2423.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning for Image Recognition: IEEE Conference on Computer Vision and Pattern Recognition. CVPR), pp. 770–778.
- Huang, X., Alkhalifah, T., 2021. Pinnup: Robust Neural Network Wavefield Solutions Using Frequency Upscaling and Neuron Splitting. *CoRR*, 14536 abs/2109.
- Kazei, V., Ovcharenko, O., Plotnitskii, P., Peter, D., Zhang, X., Alkhalifah, T., 2021. Mapping full seismic waveforms to vertical velocity profiles by deep learning. *Geophysics* 86, R711–R721.
- Keho, T.H., Kelamis, P.G., 2012. Focus on land seismic technology: the near-surface challenge. *Lead. Edge* 31, 62–68.
- Kochkov, D., Smith, J.A., Alieva, A., Wang, Q., Brenner, M.P., Hoyer, S., 2021. Machine learning-accelerated computational fluid dynamics. *Proc. Natl. Acad. Sci. USA* 118.
- Liu, S., Deng, W., 2015. Very Deep Convolutional Neural Network-Based Image Classification Using Small Training Sample Size: 3rd IAPR Asian Conference on Pattern Recognition. ACPR), pp. 730–734.
- Liu, R., Lehman, J., Molino, P., Petroski Such, F., Frank, E., Sergeev, A., Yosinski, J., 2018. An intriguing failing of convolutional neural networks and the CoordConv solution. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18). Curran Associates Inc., Red Hook, NY, USA, pp. 9628–9639.
- Luo, Y., Huang, Y., Yang, Y., Zhao, K., Yang, X., Xu, H., 2022. Constructing shear velocity models from surface wave dispersion curves using deep learning. *J. Appl. Geophys.* 196 <https://doi.org/10.1016/j.jappgeo.2021.104524>.
- Nolet, G., Panza, G.F., 1976. Array analysis of seismic surface waves: limits and possibilities. *Pure Appl. Geophys.* 114, 775–790.
- Oye, O., Dahl, E., 2019. Velocity Model Building from Raw Shot Gathers Using Machine Learning: 81st EAGE Conference and Exhibition.
- Park, C.B., Miller, R.D., Xia, J., Ivanov, J., 2007. Multichannel analysis of surface waves (masw) – active and passive methods. *Lead. Edge* 26, 60–64.
- Pham, N., Li, W., 2022. Physics-constrained deep learning for ground roll attenuation. *Geophysics* 87, V15–V27.
- Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S., 2014. CNN features off-the-shelf: an astounding baseline for recognition. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 512–519.
- Rovetta, D., Kontakis, A., Colombo, D., 2020. Fully automatic picking of surface wave dispersion curves through density-based spatial clustering. In: EAGE Annual Conference Exhibition.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Fei-Fei, L., 2015. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* 115, 211–252, 2015.
- Shtivelman, V., 1999. Using surface waves for estimating shear wave velocities in the shallow subsurface onshore and offshore Israel. In: EAGE Conference Proceedings, 5th EEGS-ES Meeting.
- Socco, L., Strobbia, C., 2004. Surface-wave method for near-surface characterization: a tutorial. *Near Surf. Geophys.* 2, 165–185.
- Socco, L.V., Foti, S., Boiero, D., 2010. Surface-wave analysis for building near-surface velocity models - established approaches and new perspectives. *Geophysics* 75, 75A83–75A102.
- Sosnovik, I., Oseledets, I., 2019. Neural networks for topology optimization. *Russ. J. Numer. Anal. Math. Model.* 34, 215–223.
- Thorbecke, J.W., Draganova, D., 2011. Finite-difference modeling experiments for seismic interferometry. *Geophysics* 76, H1–H18.
- Tunyasuvunakool, K., Adler, J., Wu, Z., et al., 2021. Highly accurate protein structure prediction for the human proteome. *Nature* 596, 590–596.
- Wang, F.Y., Song, X.D., Li, M.K., 2022. A deep-learning-based approach for seismic surface-wave dispersion inversion (sfnet) with application to the mainland of China. *Earthq. Sci.* 36.
- Xia, J., Miller, R.D., Park, C.B., 1999. Estimation of near-surface shear-wave velocity by inversion of Rayleigh waves. *Geophysics* 64, 691–700.
- Yang, D., Cai, Y., Hu, G., Yao, X., Zou, W., 2020. Seismic Fault Detection Based on 3D Unet++ Model. *SEG Technical Program Expanded Abstracts*, pp. 1631–1635.
- Yang, F., Ma, J., 2019. Deep-learning inversion: a next-generation seismic velocity model building method. *Geophysics* 84, R583–R599.
- Yosinski, J., Clune, J., Bengio, Y., Lipson, H., 2014. How transferable are features in deep neural networks? *Adv. Neural Inf. Process. Syst.* 27. ISBN: 9781510800410.
- Yu, S., Ma, J., 2021. Deep learning for geophysics: current and future trends. *Rev. Geophys.* 59 (3).
- Yuan, S.-Y., Zhao, Y., Xie, T., Qi, J., Wang, S.-X., 2022. Segnet-based first-break picking via seismic waveform classification directly from shot gathers with sparsely distributed traces. *Petrol. Sci.* 19, 162–179.
- Zheng, D., Miao, X., 2014. Multimodal Rayleigh Wave Dispersion Curve Picking and Inversion to Build Near-Surface Shear Wave Velocity Models.
- Zheng, Y., Zhang, Q., Yusifov, A., Shi, Y., 2019. Applications of supervised deep learning for seismic interpretation and inversion. *Lead. Edge* 38, 526–533.
- Zwartjes, P., 2020. Near surface velocity estimation from phase velocity-frequency panels with deep learning. In: EAGE 2020 Annual Conference & Exhibition Online, vol. 2020. <https://doi.org/10.3997/2214-4609.202010253>.