

Extrapolated full waveform inversion with deep learning

Hongyu Sun and Laurent Demanet

April, 2019

Abstract

The lack of low frequency information and a good initial model can seriously affect the success of full waveform inversion (FWI), due to the inherent cycle skipping problem. Computational low frequency extrapolation is in principle the most direct way to address this issue. By considering bandwidth extension as a regression problem in machine learning, we propose an architecture of convolutional neural network (CNN) to automatically extrapolate the missing low frequencies without preprocessing and post-processing steps. The bandlimited recordings are the inputs of the CNN and, in our numerical experiments, a neural network trained from enough samples can predict a reasonable approximation to the seismograms in the unobserved low frequency band, both in phase and in amplitude. The numerical experiments considered are set up on simulated P-wave data. In extrapolated FWI (EFWI), the low-wavenumber components of the model are determined from the extrapolated low frequencies, before proceeding with a frequency sweep of the bandlimited data. The proposed deep-learning method of low-frequency extrapolation shows adequate generalizability for the initialization step of EFWI. Numerical examples show that the neural network trained on several submodels of the Marmousi model is able to predict the low frequencies for the BP 2004 benchmark model. Additionally, the neural network can robustly process seismic data with uncertainties due to the existence of noise, poorly-known source wavelet, and different finite-difference scheme in the forward modeling operator. Finally, this approach is not subject to the structural limitations of other methods for bandwidth extension, and seems to offer a tantalizing solution to the problem of properly initializing FWI.

1 Acknowledgments

The authors thank Total SA for support. LD is also supported by AFOSR grant FA9550-17-1-0316. Tensorflow and Keras are used for deep learning. The Python Seismic Inversion Toolbox (PySIT) (Hewett et al., 2013) is used for FWI in this paper.

2 Introduction

FWI requires low frequency data to avoid convergence to a local minimum in the case where the initial models miss a reasonable representation of the complex structure. However,

because of the acquisition limitation in seismic processing, the input data for seismic inversion are typically limited to a band above 3Hz. With assumptions and approximations to make inferences from tractable but simplified models, geophysicists have started reconstructing the reflectivity spectrum from the bandlimited records by signal processing methods. L_1 -norm minimization (Levy and Fullagar, 1981; Oldenburg et al., 1983), autoregressive modelling (Walker and Ulrych, 1983) and minimum entropy reconstruction (Sacchi et al., 1994) have been developed to recover the isolated spikes of seismic recordings. Recently, bandwidth extension to the low frequency band has attracted the attention of many people in terms of FWI. For example, they recover the low frequencies by the envelope of the signal (Wu et al., 2014; Hu et al., 2017) or the inversion of the reflectivity series and convolution with the broadband source wavelet (Wang and Herrmann, 2016; Zhang et al., 2017). However, the low frequencies recovered by these methods are still far from the true low frequency data. Li and Demanet (2016) attempt to extrapolate the true low frequency data based on the phase tracking method (Li and Demanet, 2015). Unlike the explicit parameterization of phases and amplitudes of atomic events, here we propose an approach that can automatically process the raw bandlimited records. The deep neural network (DNN) is trained to automatically recover the missing low frequencies from the input bandlimited data.

Because of the state-of-the-art performance of machine learning in many fields, geophysicists have begun adapting such ideas in seismic processing and interpretation (Chen et al., 2017; Guitton et al., 2017; Xiong et al., 2018). By learning the probability of salt geobodies being present at any location in a seismic image, Lewis and Vigh (2017) investigate CNN to incorporate the long wavelength features of the model in the regularization term. Richardson (2018) constructs FWI as recurrent neural networks. Araya-Polo et al. (2018); Wu et al. (2018); Li et al. (2019) produce layered velocity models from shot gathers with DNN.

Like these authors and many others, we have selected DNN for low frequency extrapolation due to the increasing community agreement in favor of this method as a reasonable surrogate for a physics-based process (Grzeszczuk et al., 1998; De et al., 2011; Araya-Polo et al., 2017). The universal approximation theorem also indicates that the neural networks can be used to replicate any function up to our desired accuracy if the DNN has sufficient hidden layers and nodes (Hornik et al., 1989). Although training is therefore expected to succeed arbitrarily well, only empirical evidence currently exists for the often-favorable performance of testing a network out of sample. Furthermore, we choose to focus on DNN with a convolutional structure, i.e., CNN. The idea behind CNN is to mine the hidden correlations among different frequency components.

In the case of bandwidth extension, the relevant data are the amplitudes and phases of seismic waves, which are dictated by the physics of wave propagation. For training, large volumes of synthetic shot gathers are generated from different models, in a wide band that includes the low frequencies, and the network’s parameters are fit to regress the low frequencies of those data from the high frequencies. The frequency division is defined from soft wavelet indicators in a standard fashion. A soft wavelet indicator means that it is at least continuous, or better, in the frequency domain. For testing, bandlimited (and not otherwise processed) data from a new geophysical scenario are used as input of the network, and the network generates a prediction of the low frequencies. In the synthetic case, validation of the testing step is possible by computing those low frequencies directly from the wave solver.

By now, neural networks have shown their ability to fulfill the task of low frequency

extrapolation. Ovcharenko et al. (2018) train neural network on data generated for random velocity models to predict single low frequency from multiple high frequency data. However, the prediction accuracy is barely sufficient to be directly usable in FWI. Jin et al. (2018) use a deep inception based convolutional networks to synthesize data at multiple low frequencies. The input of their neural network contains the phase information of the true low frequency by leveraging the beat tone data (Hu, 2014). In contrast, we design an architecture of CNN to directly deal with the bandlimited data in the time domain. The proposed architecture can predict the data in a low frequency band with high enough accuracy that it can be used for FWI.

The limitations of neural networks for such signal processing tasks, however, are (1) the unreliability of the prediction when the training set is insufficient, and (2) the absence of a physical interpretation for the operations performed by the network. In addition, no theory can currently explain the generalizability of a deep network, i.e., the ability to perform nearly as well on testing as on training in a broad range of cases. Even so, the numerical examples indicate that the proposed architecture of CNN enjoys sufficient generalizability to extrapolate the low frequencies of unknown subsurface structures, in a range of numerical experiments.

We demonstrate the reliability of the extrapolated low frequencies to seed frequency-sweep FWI on the Marmousi model and the BP 2004 benchmark model. Two precautions are taken to ensure that trivial deconvolution of a noiseless record (by division by the high frequency (HF) wavelet in the frequency domain) is not an option: (1) add noise to the testing records, and (2) for testing, choose a hard bandpass HF wavelet taken to be zero in the low frequency (LF) band. In one numerical experiment involving bandlimited data above 0.6Hz from the BP 2004 model, the inversion results indicate that the predicted low frequencies are adequate to initialize conventional FWI from an uninformative initial model, so that it does not suffer from the otherwise-inherent cycle-skipping at 0.6Hz. Additionally, the proposed neural network has acceptable robustness to uncertainties due to the existence of noise, poorly-known source wavelet, and different finite-difference schemes in the forward modeling operator.

This paper is organized as follows. We start by formulating bandwidth extension as a regression problem in machine learning. Next, we introduce the general workflow to predict the low frequency recordings with CNN. We then study the generalizability and the stability of the proposed architecture in more complex situations. Last, we illustrate the reliability of the extrapolated low frequencies to initialize FWI, and analyze the limitations of this method.

3 Deep Learning

A neural network defines a mapping $\mathbf{y} = f(\mathbf{x}, \mathbf{w})$ and learns the value of the parameters \mathbf{w} that result in a good fit between \mathbf{x} and \mathbf{y} . DNNs are typically represented by composing together many different functions to find complex nonlinear relationships. The chain structures are the most common structures in DNNs (Goodfellow et al., 2016):

$$\mathbf{y} = f(\mathbf{x}, \mathbf{w}) = f_L(\dots f_2(f_1(\mathbf{x}))), \tag{1}$$

where f_1, f_2 and f_L are the first, the second and the L^{th} layer of the network (with their own parameters omitted in this notation). Each f_j consists of three operations taken in succession: an affine (linear plus constant) transformation, a batch normalization (multiplication by a scalar chosen adaptively), and the componentwise application of a nonlinear activation function. It is the nonlinearity of the activation function that enables the neural network to be a universal function approximator. The overall length L of the chain gives the depth of the deep learning model. The final layer is the output layer, which defines the size and type of the output data. The training sets specify directly what the output layer must do at each point \mathbf{x} , and constrain but do not specify the behavior of the other hidden layers. Rectified activation units are essential for the recent success of DNNs because they can accelerate convergence of the training procedure. Our numerical experiments show that, for bandwidth extension, Parametric Rectified Linear Unit (PReLU)(He et al., 2015) works better than the Rectified Linear Unit (ReLU). The formula of PReLU is

$$g(\alpha, \mathbf{y}) = \begin{cases} \alpha \mathbf{y}, & \text{if } \mathbf{y} < 0 \\ \mathbf{y}, & \text{if } \mathbf{y} \geq 0 \end{cases}, \quad (2)$$

where α is also a learned parameter and would be adaptively updated for each rectifier during training.

Unlike the classification problem that trains the DNNs to produce discrete labels, the regression problem trains the DNNs for the prediction of continuous-valued outputs. It evaluates the performance of the model by means of the mean-squared error (MSE) of the predicted outputs $f(\mathbf{x}_i; \mathbf{w})$ vs. actual outputs \mathbf{y}_i :

$$J(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^m L(\mathbf{y}_i, f(\mathbf{x}_i, \mathbf{w})), \quad (3)$$

where the loss L is the squared error between the true low frequencies and the estimated outputs of the neural networks. The cost function J is here minimized over \mathbf{w} by a stochastic gradient descent (SGD) algorithm, where each gradient is computed from a mini-batch, i.e., a subset in a disjoint randomized partition of the training set. Each gradient evaluation is called an iteration, while the full pass of the training algorithm over the entire training set using mini-batches is an epoch. The learning rate η (step size) is a key parameter for deep learning and must be finetuned. The gradients $\frac{\partial J(\mathbf{w}^t)}{\partial \mathbf{w}}$ of the neural networks are calculated by the backpropagation method (Goodfellow et al., 2016).

CNN is an overwhelmingly popular architecture of DNN to extract spatial features in image processing, and it is the choice that we make in this paper. In this case, the matrix-vector multiplication in each of the f_j is a convolution. In addition, imposing local connections and weight sharing can exploit both the local correlation and global features of the input image. CNNs are normally designed to deal with the image classification problem. For bandwidth extension, the data to be learned are the time-domain seismic signals, so we directly consider the amplitude at each sampling point as the pixel value of the image to be used as input of the CNN.

Recall that CNN involve stacks of: a convolutional layer, followed by a PReLU layer, and a batch normalization layer. The filter number in each convolutional layer determines the dimensionality of the feature map or the channel of its output. Each output channel

of the convolutional layer is obtained by convolving the channel of the previous layer with one filter, summing and adding a bias term. The batch normalization layer can speed up training of CNNs and reduce the sensitivity to network initialization by normalizing each input channel across a mini-batch. Although a pooling layer is typically used in the conventional architecture of CNNs, we leave it out because both the input and output signals have the same length, so since downsampling of feature maps is unhelpful for bandwidth extension in our experiments.

An essential hyperparameter for low frequency extrapolation with deep learning is the receptive field of a neuron. It is the local region of the input volume that affects the response of this neuron – otherwise known as the domain of dependence. The spatial extent of this connectivity is related to the filter size. Unlike the small filter size commonly used in the image classification problem, we directly use a large filter in the convolutional layer to increase the receptive field of the CNN quickly with depth. The large filter size gives the neural network enough freedom to learn the low-pass filters and to reconstruct the long-wavelength information

The architecture of our neural network (Figure 1) is a feed-forward stack of five sequential combinations of the convolution, PReLU and batch normalization layers, followed by one fully connected layer that outputs continuous-valued amplitude of the time-domain signal in the low frequency band. The first convolutional layer filters the $nt \times 1$ input time series with 128 kernels of size $200 \times 1 \times 1$ where nt is the number of time steps. The second convolutional layer has 64 kernels of size $200 \times 1 \times 128$ connected to the normalized outputs of the first convolutional layer. The third convolutional layer has 128 kernels of size $200 \times 1 \times 64$. The fourth convolutional layer has 64 kernels of size $200 \times 1 \times 128$, and the fifth convolutional layer has 32 kernels of size $200 \times 1 \times 64$. The last layer is fully connected, taking features from the last convolutional layer as input in a vector form of length $nt \times 32$. The stride of the convolution is one, and zero-padding is used to make the output length of each convolution layer the same as its input. Additionally, a dropout layer (Srivastava et al., 2014) with a probability of 50% is added after the first convolution layer to reduce the generalization error.

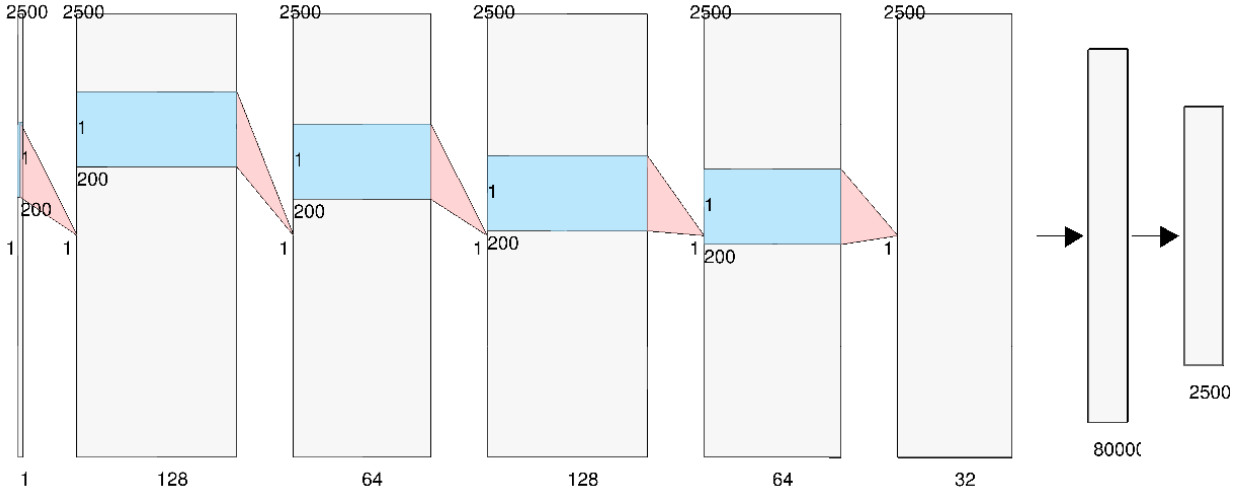


Figure 1: An illustration of the architecture of our CNN to extrapolate the low frequency data from the bandlimited data in the time domain trace by trace. The architecture is a feed-forward stack of five sequential combinations of the convolution, PReLU and batch normalization layers, followed by one fully connected layer that outputs continuous-valued amplitude of the time-domain signal in the low frequency band. The networks input is a one-dimensional bandlimited recording of length nt where $nt = 2500$ is the number of time steps. The number of neurons in the networks remaining convolutional layers is given by $nt \times 128$, $nt \times 64$, $nt \times 128$, $nt \times 64$ and $nt \times 32$.

We use CNN in the context of supervised learning, i.e., inference of \mathbf{y}_i from \mathbf{x}_i . We need to first train the CNN from a large number of samples $(\mathbf{x}_i, \mathbf{y}_i)$ to determine the coefficients of the network, and then use the network for testing on new \mathbf{x}_i . In statistical learning theory, the generalization error is the difference between the expected and empirical error, where the expectation runs over a continuous probability distribution on the \mathbf{x}_i . This generalization error can be approximated by the difference between the errors on the training and on the test sets.

The object of this paper is that the \mathbf{x}_i can be taken to be seismograms bandlimited to the high frequencies, and \mathbf{y}_i can be the same seismograms in the low frequency band. Generating training samples means collecting, or synthesizing seismogram data from a variety of geophysical models, which enter as space-varying elastic coefficients in a wave equation. For the purpose of good generalization (small generalization error), the models used to create the large training sets should be able to represent many subsurface structures, including different types of reflectors and diffractors, so we can find a representative set of parameters to handle data from different scenarios or regions. The performance of the neural network is sensitive to the architecture and the hyperparameters, so we must design them carefully. Next, we illustrate the specific choice of hyperparameters for bandwidth extension, along with numerical examples involving synthetic data from community models.

4 Numerical Examples

In this section, we demonstrate the reliability of extrapolated FWI with CNN (EFWI-CNN) in three parts. In the first part, we show CNN’s ability to extrapolate low frequency data (0.1 – 5Hz) from bandlimited data (5 – 20Hz) on the Marmousi model (Figure 2). In the second part, we verify the robustness of the method with uncertainties in the seismic data due to the existence of noise, different finite difference scheme, and poorly-known source wavelet. In the last part, we perform EFWI-CNN on both the Marmousi model and the BP 2004 benchmark model (Billette and Brandsberg-Dahl, 2005), by firstly using the extrapolated low frequencies to synthesize the low-wavenumber background velocity model. Then, we compare the inversion results with the bandlimited data in three cases which respectively start FWI from an uninformed initial model, the low-wavenumber background model created from the extrapolated low frequencies, and the low-wavenumber background model created from the true low frequencies.

4.1 Low frequency extrapolation

In the paper, the true unknown velocity model for FWI is referred to as the test model, since it is used to collect the test data set in deep learning. To collect the training data set, we create training models by randomly selecting nine parts of the Marmousi model (Figure 2) with different structure but the same number of grid points 166×461 . We also downsample the original model to 166×461 pixels as the test model. We find that the randomized models produced in this manner are realistic enough to demonstrate the generalization of the neural network if the structures of the submodels are diversified enough.

In this example, we have the following processing steps to collect each sample (i.e., shot record) in both the training and test data sets.

- The acquisition geometry of forward modeling on each model is the same. It consists of 30 sources and 461 receivers evenly spaced at the surface. We consider each time series or trace as one sample in the data set, so have 124,470 training samples and 13,830 test samples for each test model in total.
- We use a fourth order in space and second order in time finite-difference modeling method with PML to solve the 2D acoustic wave equation in the time domain, to generate the synthetic shot gathers of both the training and test data sets. The sampling interval and the total recording time are 2ms and 5s, respectively.
- We use a Ricker wavelet with dominant frequency 7Hz to synthesize the full-band seismic recordings. Then the data below 5Hz and above 5Hz are split to synthesize the output and input of the neural networks, respectively (Figure 3). Both the low and high frequency data are obtained by a sharp windowing of the same trace.

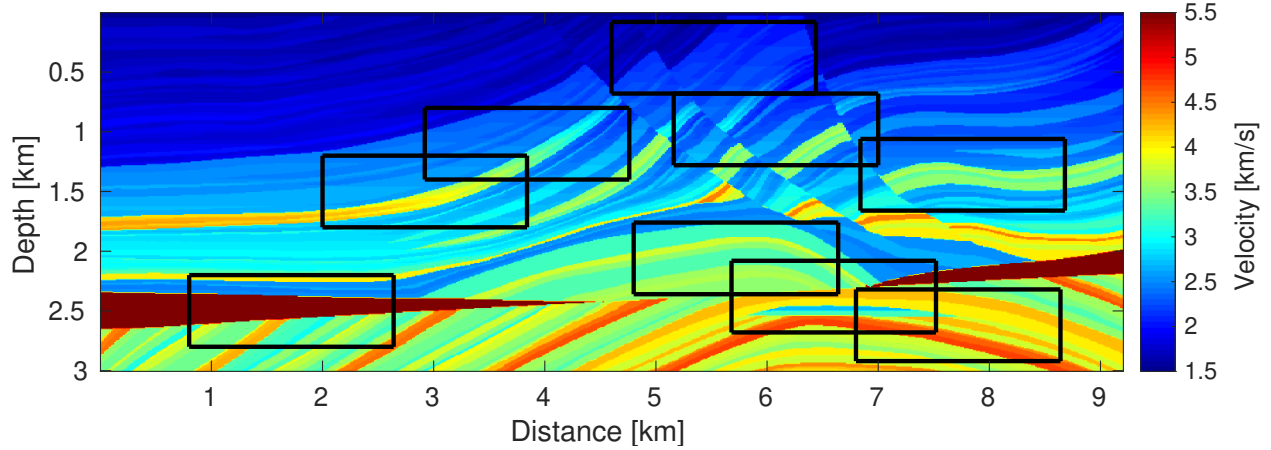


Figure 2: The nine training models randomly extracted from the Marmousi velocity model to collect the training data set. The test models are the Marmousi model and the BP 2004 benchmark model. A water layer with 300m depth is added to the top of these training models and Marmousi model. We use the same training models to extrapolate the low frequencies on both test models.

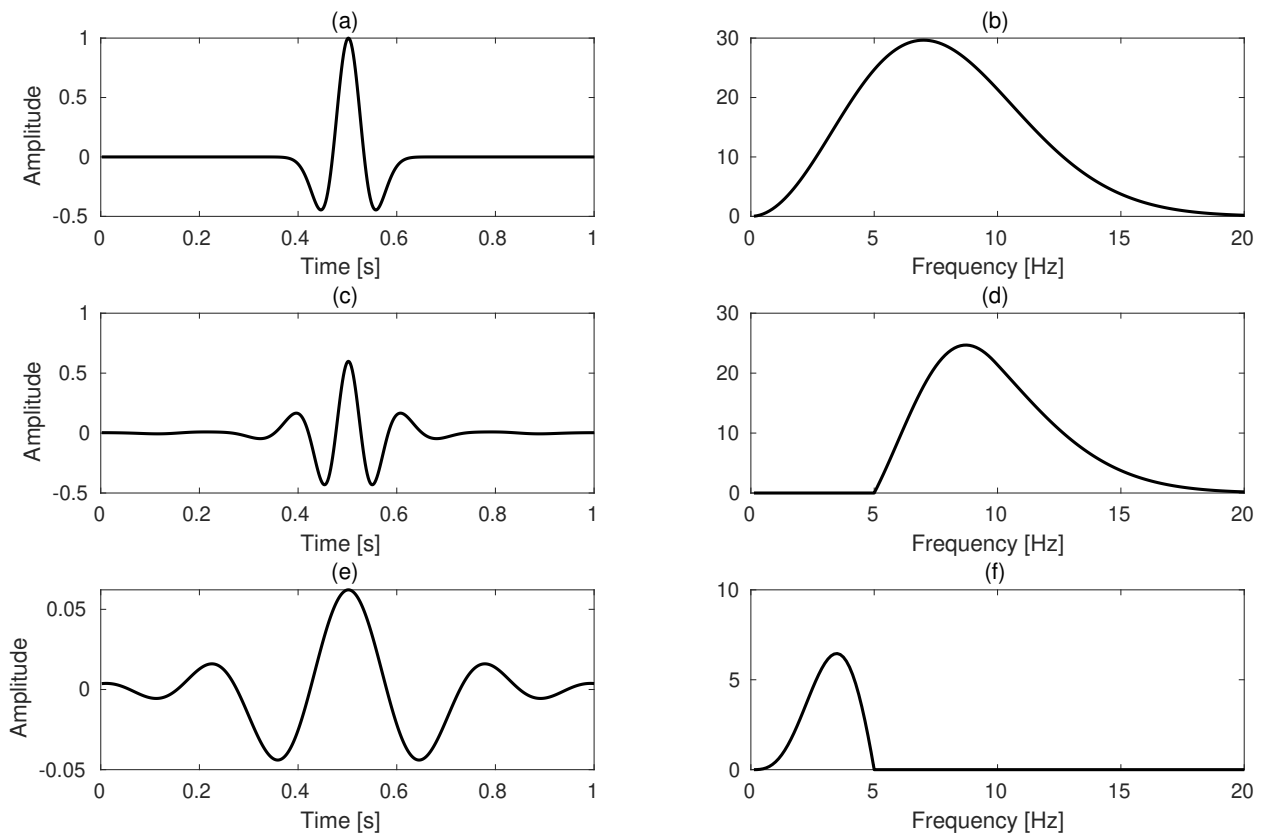


Figure 3: (a) The Ricker wavelet with 7Hz dominant and its amplitude spectrum in (b). (c) The high frequency wavelet bandpassed from (a) and its amplitude spectrum in (d). (e) The low frequency wavelet bandpassed from (a) and its amplitude spectrum in (f).

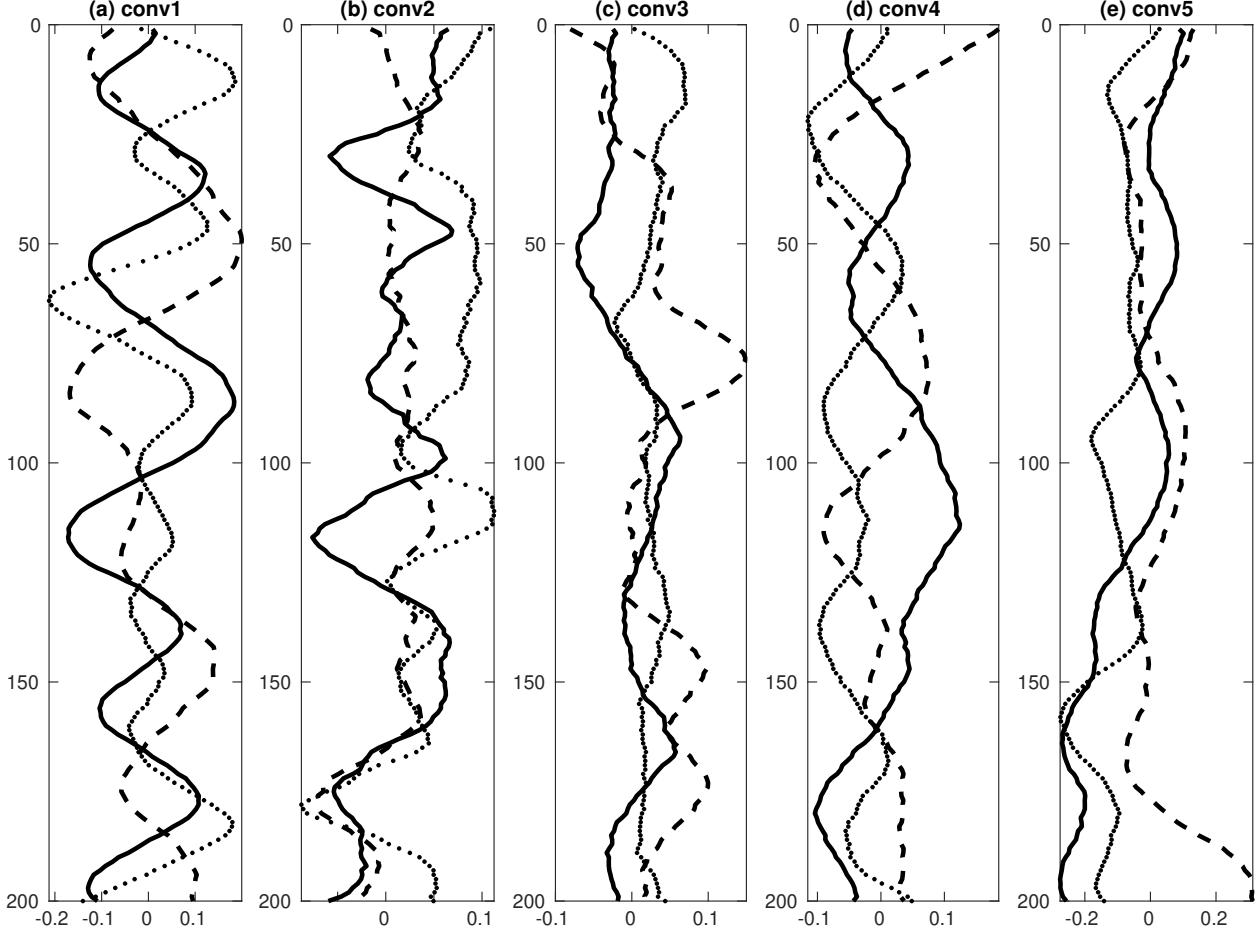


Figure 4: Three kernels of the first channel in the (a) first (b) second (c) third (d) fourth and (e) last convolutional layer learned by training with 20 epochs to predict the low frequency data below 5Hz on the Marmousi model.

In this example, we train the network with the Adam optimizer and use a mini-batch of 64 samples at each iteration. The initial learning rate and forgetting rate of the Adam are the same as the original paper (Kingma and Ba, 2014). The initial value of the bias is zero. The weight initialization is via the Glorot uniform initializer (Glorot and Bengio, 2010). It randomly initializes the weights from a truncated normal distribution centered on zero with the standard deviation $\sqrt{2/n_1 + n_2}$ where n_1 are n_2 are the numbers of input and output units in the weight tensor, respectively.

Figure 4 shows the kernels of CNN learned by training with 20 epochs. Three filters of the first channel are plotted in each of the convolutional layers. We test the performance of the neural networks by feeding the bandlimited data in the test set into the pretrained neural networks and obtain the extrapolated low frequencies on the Marmousi model. Figure 5 compares the shot gather between the bandlimited data (5 – 20Hz), extrapolated and true low frequencies (0.1 – 5Hz) where the source is located at the horizontal distance $x = 2.94km$ on the Marmousi model. The extrapolated results in Figure 5(c) show that the proposed neural network can accurately predict the recordings in the low frequency band, which are totally missing before the test. Figure 6 compares two individual seismograms in Figure 5(c)

where the receivers are located at the horizontal distance $x = 2.82km$ and $x = 2.92km$, respectively. The extrapolated low frequency data match the true recordings well. Then we combine the extrapolated low frequencies with the bandlimited data and compare the amplitude spectrum of the full-band data with the extrapolated and true low frequencies. The amplitude and phase spectrum comparison of the single trace where the receiver is located at $x = 2.92km$ (Figure 7) clearly shows that the neural networks can capture the relationship between low and high frequency components constrained by the wave equation.

Figure 8 shows the low frequency extrapolation without direct waves. The direct waves are muted from the full-band shot gathers with a smooth time window before splitting into the bandlimited recordings and the low frequencies. The low frequencies of reflections are recovered without the existence of the direct waves. Therefore, the neural network is robust with the presence of muting.

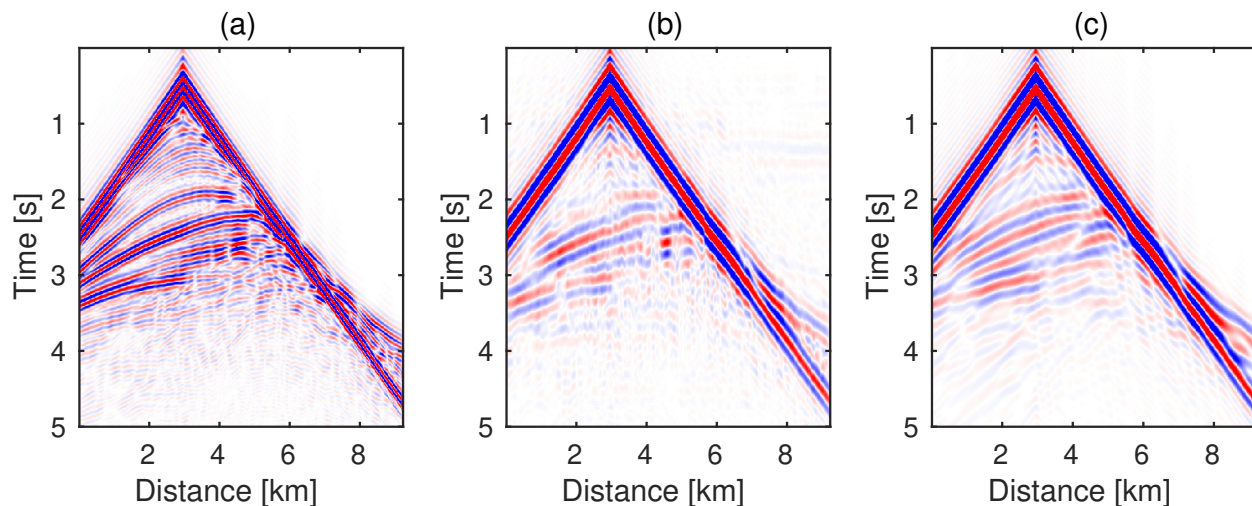


Figure 5: The extrapolation result on the Marmousi model: comparison among the (a) bandlimited recordings (5 – 20Hz), (b) predicted and (c) true low frequency recordings (0.1–5Hz). The bandlimited data in (a) are the inputs of CNNs to predict the low frequencies in (b).

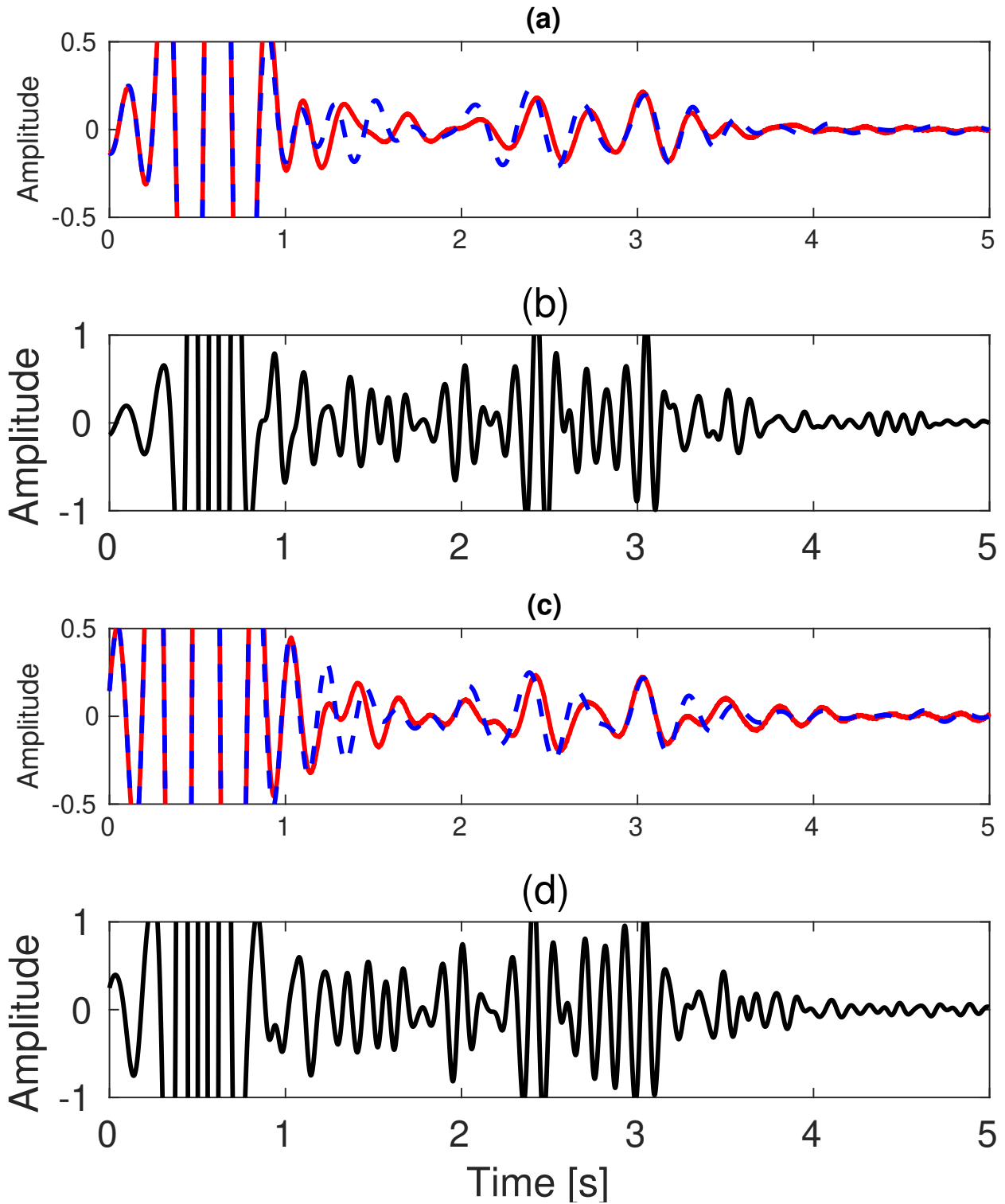


Figure 6: The extrapolation result on the Marmousi model: comparison among the predicted (red line), the true (blue dash line) recording in the low frequency band (0.1 – 5Hz) and the bandlimited recording (black line) (5 – 20Hz) at the horizontal distance (a) (b) $x = 2.82\text{km}$ and (c) (d) $x = 2.92\text{km}$.

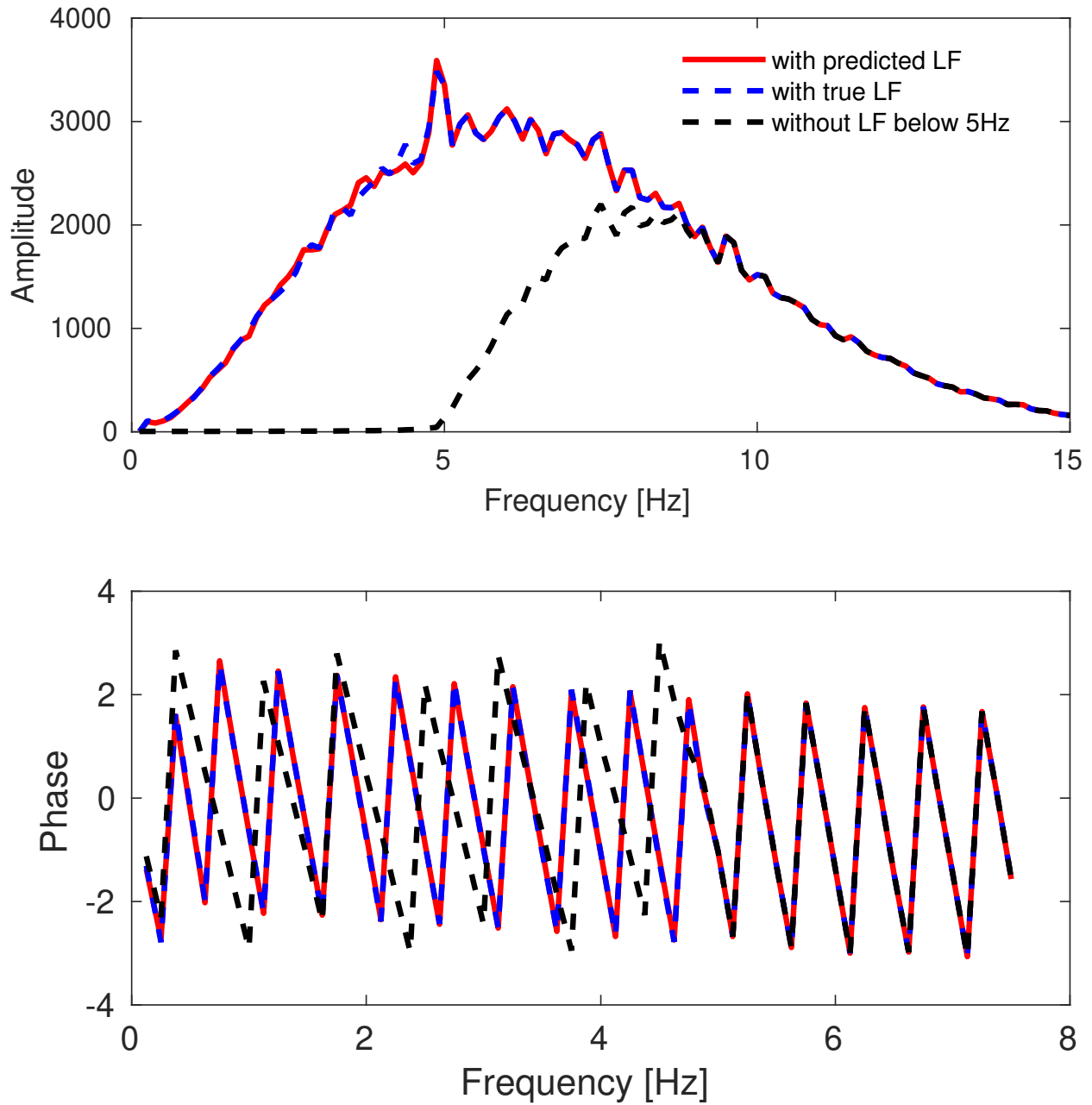


Figure 7: The extrapolation result on the Marmousi model: comparison of (a) the amplitude spectrum and (b) the phase spectrum at $x = 2.92km$ among the bandlimited recording (5 – 20Hz), the recording (0.1 – 20Hz) with true and predicted low frequencies (0.1 – 5Hz).

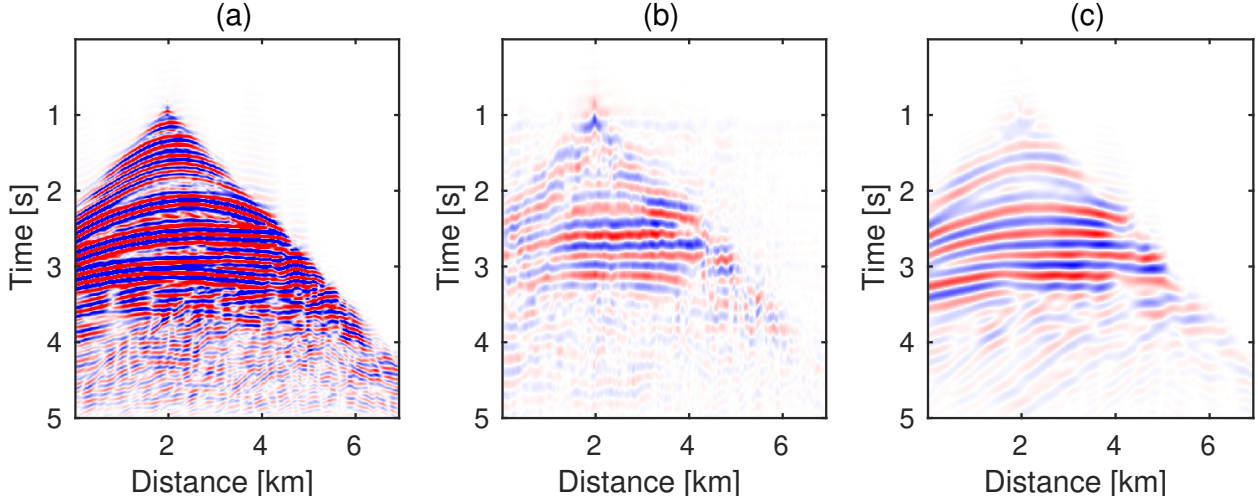


Figure 8: Low frequency extrapolation without direct waves: comparison among the (a) bandlimited recordings (5 – 20Hz), (b) predicted and (c) true low frequency recordings (0.1 – 5Hz) on the Marmousi model. The direct waves are muted from the full-band shot gathers with a smooth time window before splitting into the bandlimited recordings and the low frequencies. The extrapolation is robust with the presence of muting.

4.2 Uncertainty analysis

With a view toward dealing with complex field data, we investigate the stability of the neural network’s predictive performance under three kinds of discrepancies, or uncertainties, between training and test: additive noise; different finite difference operator in the forward modeling; and different source wavelet. In every case, we compare the extrapolation accuracy with the reference in Figure 5, where training and testing are set up the same way (noiseless bandlimited data, finite difference operator with second order in time and fourth order in space, and the Ricker wavelet with 7Hz dominant frequency). The RMSE between the extrapolated and true low frequency data in Figure 5 is 1.9813×10^{-4} .

In the first case, the neural network is expected to extrapolate the low frequencies from the noisy bandlimited data. We add additive 25% Gaussian noise to the bandlimited data in the test data set and 30% additive Gaussian noise to the bandlimited data in the training data set. The low frequencies in the training set are noiseless as before. Even though noise will disturb the neural network to find the correct mapping between the bandlimited data with their low frequencies, Figure 9 shows that the proposed neural network can still successfully extrapolate the low frequencies of the main reflections. The RMSE between the extrapolated and true low frequency data in Figure 9 is 1.9048×10^{-4} . The neural network is able to perform extrapolation as well as denoising. Incidentally, we make the (unsurprising) observation that CNN has potential for the denoising of seismic data.

Another challenge of FWI is that the observed and calculated data can come from different wave propagation schemes. For example, under the control of different numerical dispersion curves, the phase velocity would have different behaviors if we used different finite difference (FD) operators to simulate the observed and calculated data. Therefore, it is necessary to study the influence of different discretization, or other details of the simulation,

on the accuracy of low frequency extrapolation. In our case, the shot gathers in the test data set are simulated with a sixth order spatial FD operator, but the neural network is trained on the samples simulated with a fourth order spatial FD operator. The extrapolation result in Figure 10(b) shows that the neural network trained on the fourth order operator is able to extrapolate the low frequencies of the bandlimited data collected with the sixth order operator. In this case, the RMSE between the extrapolated and true low frequency data in Figure 10 is 2.1114×10^{-4} . The neural network appears to be stable with respect to mild modifications to the forward modeling operator, at least in the examples tried.

Another uncertainty is the unknown source wavelet. To check the extrapolation capability of the neural network in the context of data excited by an unknown source wavelet, we train the neural network with a 7Hz Ricker wavelet, but test it with an Ormsby wavelet. The four corner frequencies of the Ormsby wavelet are 0.2Hz, 1.5Hz, 8Hz and 14Hz, respectively. Figure 11 shows that the neural network trained on the data from the 7Hz Ricker wavelet source wavelet is able to extrapolate the data synthesized with Ormsby source wavelet. However, the recover of the amplitude is much poorer than the phase. The RMSE between the extrapolated and true low frequency data in Figure 11 is increased to 1.4×10^{-3} . The commonplace issue of the source wavelet being unknown or poorly known in FWI has seemingly little effect on the performance of the proposed neural network to extrapolate the phase of low frequency data, at least in the examples tried.

Even though all of the uncertain factors hurt the accuracy of extrapolated low frequencies to some extent, the CNN’s prediction has a degree of robustness that surprised us. All of the extrapolation results in the above numerical examples can be further improved by increasing the diversity of the training data set, because subjecting the network to a broader range of scenarios can fundamentally reduce the generalization error of the deep learning predictor. (For example, we can simulate the training data set with multiple kinds of source wavelets and finite difference operators)

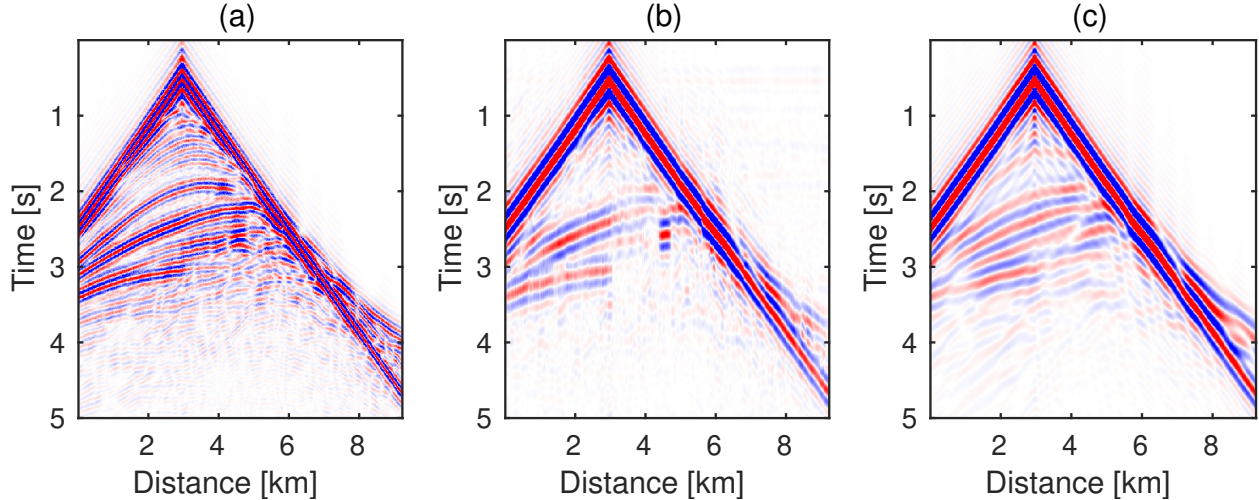


Figure 9: Noise robustness: comparison among the (a) bandlimited recordings (5 – 20Hz), (b) predicted and (c) true low frequency recordings (0.1 – 5Hz) on the Marmousi model. We add 25% additive Gaussian noise to the bandlimited data in the test data set and 30% additive Gaussian noise to the bandlimited data in the training data set. Even though noise will disturb the neural network find the correct mapping between the bandlimited data with their low frequencies, the proposed neural network can still extrapolate the low frequencies of the main reflections. The neural network is able to perform extrapolation as well as denoising.

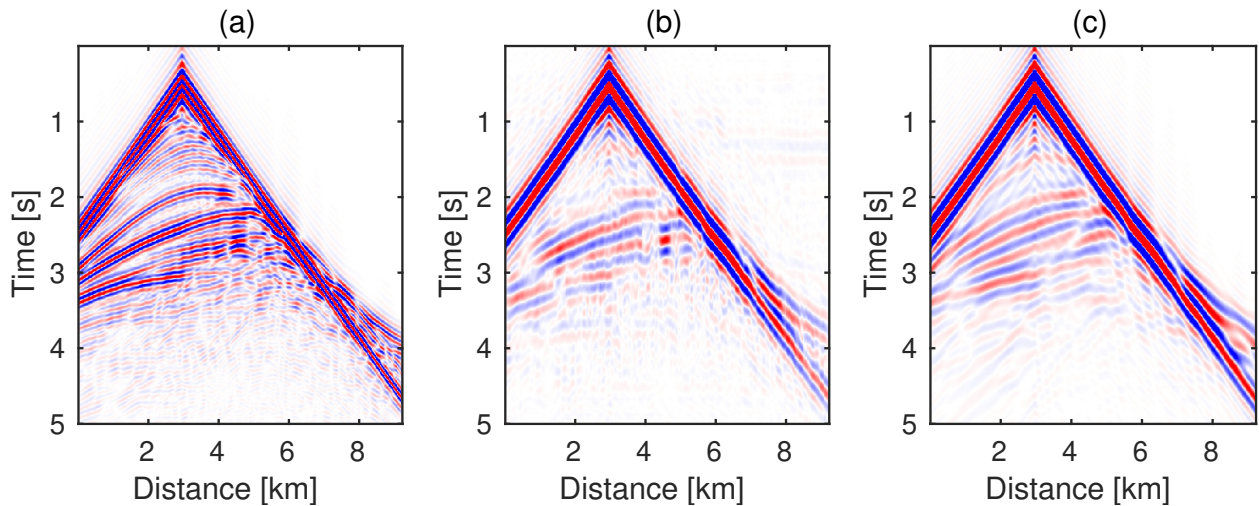


Figure 10: Forward modeling operator robustness: comparison among the (a) bandlimited recordings (5 – 20Hz), (b) predicted and (c) true low frequency recordings (0.1 – 5Hz) on the Marmousi model. The shot gather in the test data set is simulated with the sixth order operator, while the neural network is trained with the samples simulated with the fourth order operator. The extrapolation result in (b) shows that the neural network trained on the fourth order finite difference operator can extrapolate the low frequencies of the bandlimited data coming from the sixth order operator. The neural network is stable with the variance of the forward modeling operator.

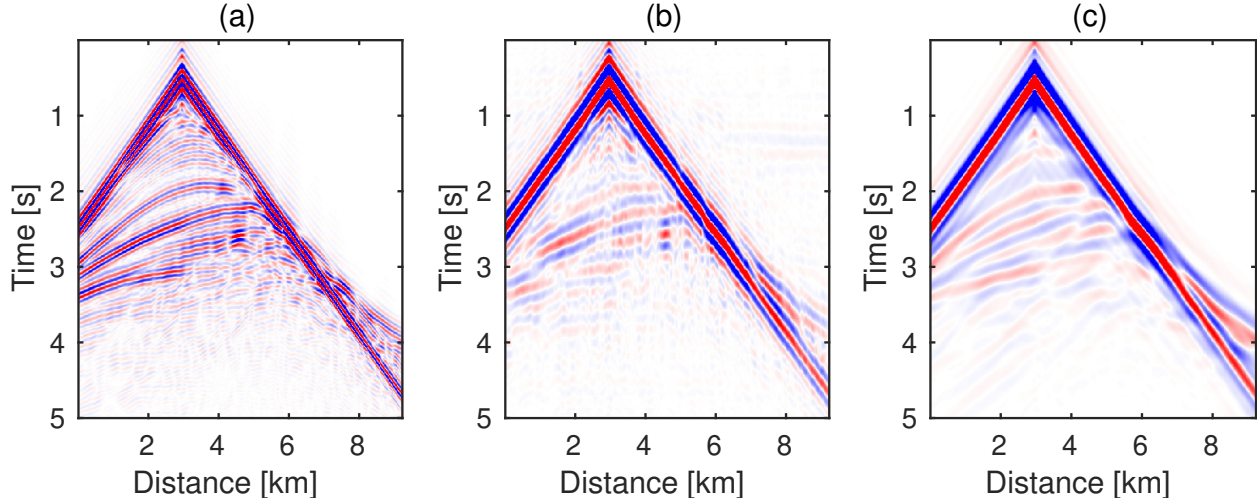


Figure 11: Unknown source wavelet robustness: comparison among the (a) bandlimited recordings (5 – 20Hz), (b) predicted and (c) true low frequency recordings (0.1 – 5Hz) on the Marmousi model. In this case, we use an Ormsby wavelet with the four corner frequencies 0.2Hz, 1.5Hz, 8Hz and 14Hz to synthesize the output and input of neural network for samples in the test data set. The result in (b) shows that the neural network trained on the data from 7Hz Ricker wavelet is able to extrapolate the data synthesize with an Ormsby wavelet. However, the recover of the amplitude is poorer than the phase.

4.3 Extrapolated FWI: Marmousi model

In this example, we construct the low-wavenumber velocity model for the Marmousi model, by leveraging the extrapolated low frequency data (Figure 5(b)) to solve the cycle-skipping problem for FWI on the bandlimited data. The objective function of the inversion is formulated as the least-squares misfit between the observed and calculated data in the time domain. Starting from an initial model in Figure 12(b), we use the LBFGS (Nocedal and Wright, 2006) optimization method to update the model gradually. To help the gradient-based iterative inversion method avoid falling into local minima, we also perform the inversion from the lowest frequency that the data contained, to successively higher frequencies.

With this inversion scheme, we test the reliability of the extrapolated low frequencies (Figure 5(b)) on the Marmousi model (Figure 12(a)). The velocity structure of the initial model is far from the true model. The true model was not used in the training stage. The acquisition geometry and source wavelet are the same as in the example in the previous section. The observed data below 5.0Hz are totally missing. Therefore, we firstly use the bandlimited data in 5 – 20Hz to recover the low frequencies in (0.1 – 5.0Hz) and then use the low frequencies to invert the low-wavenumber velocity model for the bandlimited FWI. Figure 13 compares the inverted models from FWI using the true and extrapolated 0.5 – 3Hz low frequency data. Since the low frequency extrapolation accuracy of reflections after four seconds is limited (as seen in Figure 5), the low-wavenumber model constructed from the extrapolated low frequencies has lower resolution in the deeper section compared with that from the true low frequencies. However, both models capture the low wavenumber information of the Marmousi model. These models are used as the starting models for FWI

on the bandlimited data.

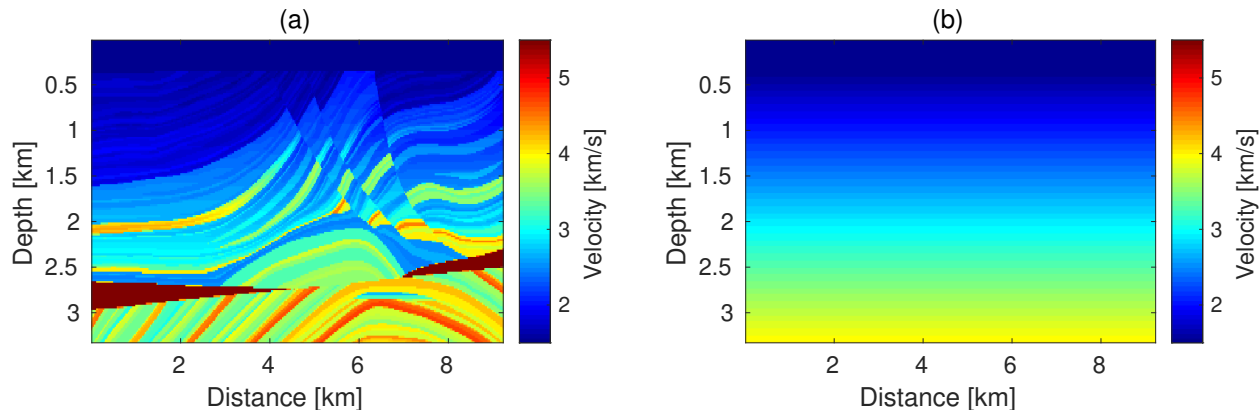


Figure 12: (a) The Marmousi model (the true model in FWI and the test model in deep learning) and (b) the initial model for FWI.

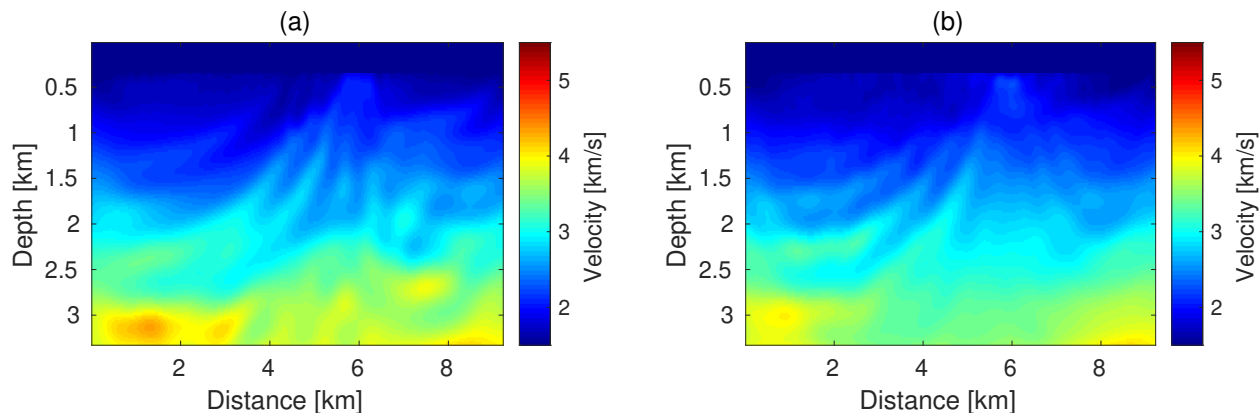


Figure 13: Comparison between the inverted low-wavenumber models using the (a) true and (b) extrapolated 0.5-3Hz low frequencies. The model constructed from the extrapolated low frequencies has lower resolution in the deeper section compared with the model from the true low frequencies because the extrapolation accuracy of deeper reflections is poor. However, both models capture the low wavenumber information of the Marmousi model.

Figure 14 compares the inverted models from FWI using the bandlimited data (5-15Hz) with different starting models. The resulting model in Figure 14(b) starts from the low-wavenumber model constructed from the extrapolated low frequencies (Figure 13(b)), which is almost the same as the one from the true low frequencies (Figure 14(a)). Since the highest frequency component in the low frequency band is 3Hz when we invert the starting model, both inversion results have slight cycle skipping phenomenon. However, Figure 14(c) performs the bandlimited FWI with the linear initial model, and shows a much more pronounced effect of cycle skipping. We cannot find much meaningful information about the subsurface structure if the bandlimited inversion starts at 5Hz from a linear initial model (Figure 12(b)).

Figure 15 compares the velocity profile among the resulting models in Figure 14 (the initial and true velocity models) at the horizontal location of $x = 3km$, $x = 5km$ and

$x = 7\text{km}$. The final inversion result started from the extrapolated low frequencies gives us almost the same model as the true low frequencies, which illustrates that the extrapolated low frequency data are reliable enough to provide an adequate low-wavenumber velocity model. However, both inversion workflows have difficulty in the recovery of velocity structure below 2km. The inversion results can be further improved by involving higher frequency components and adding more iterations. In contrast, since the velocity model in Figure 14(c) has fallen into a local minimum, the inversion cannot converge to the true model in the subsequent iterations.

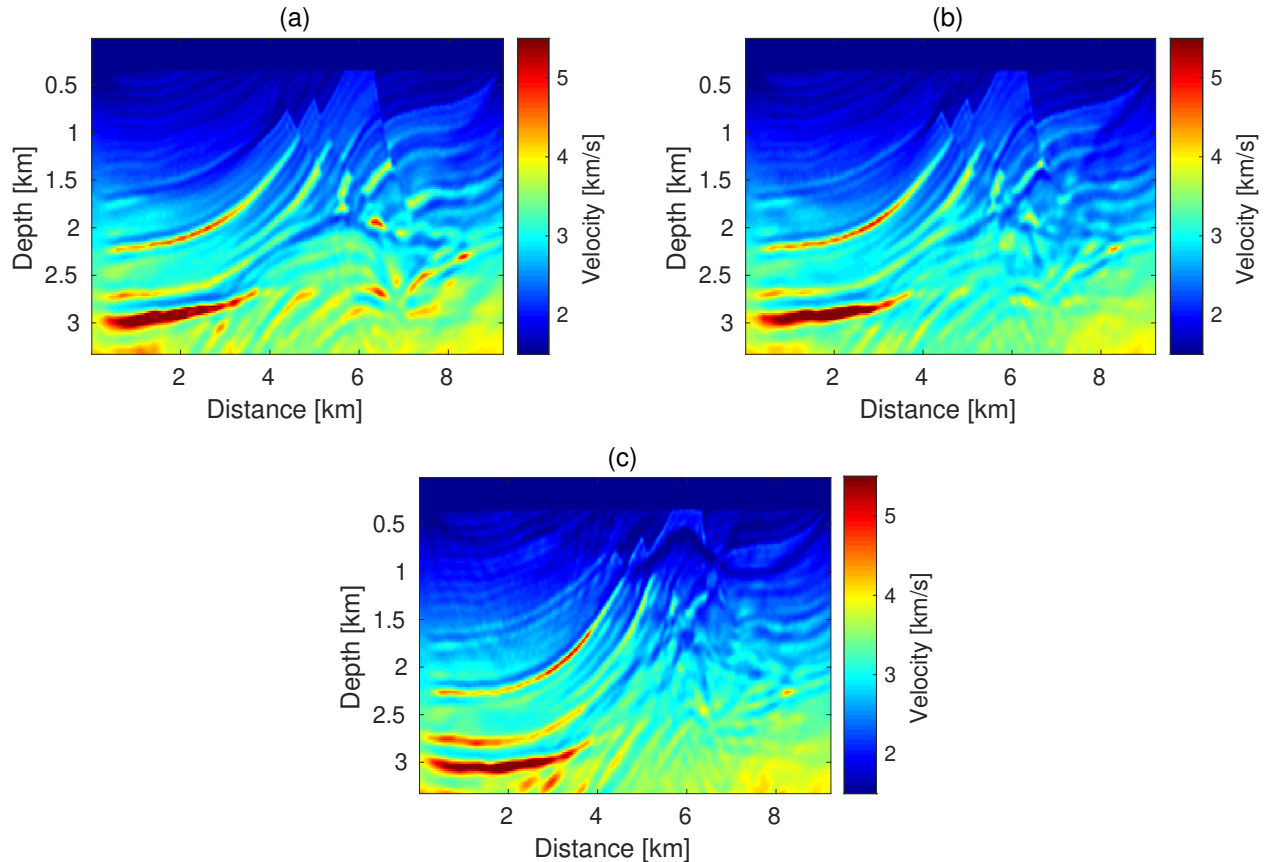


Figure 14: Comparison among the inverted models from FWI using the bandlimited data (5-15Hz). In (a), resulting model is started from the low-wavenumber velocity model constructed with the true low frequencies in Figure 13(a). In (b), resulting model is started from the low-wavenumber velocity model constructed with the extrapolated low frequencies in Figure 13(b). In (c), resulting model is started from the initial model in Figure 12(b).

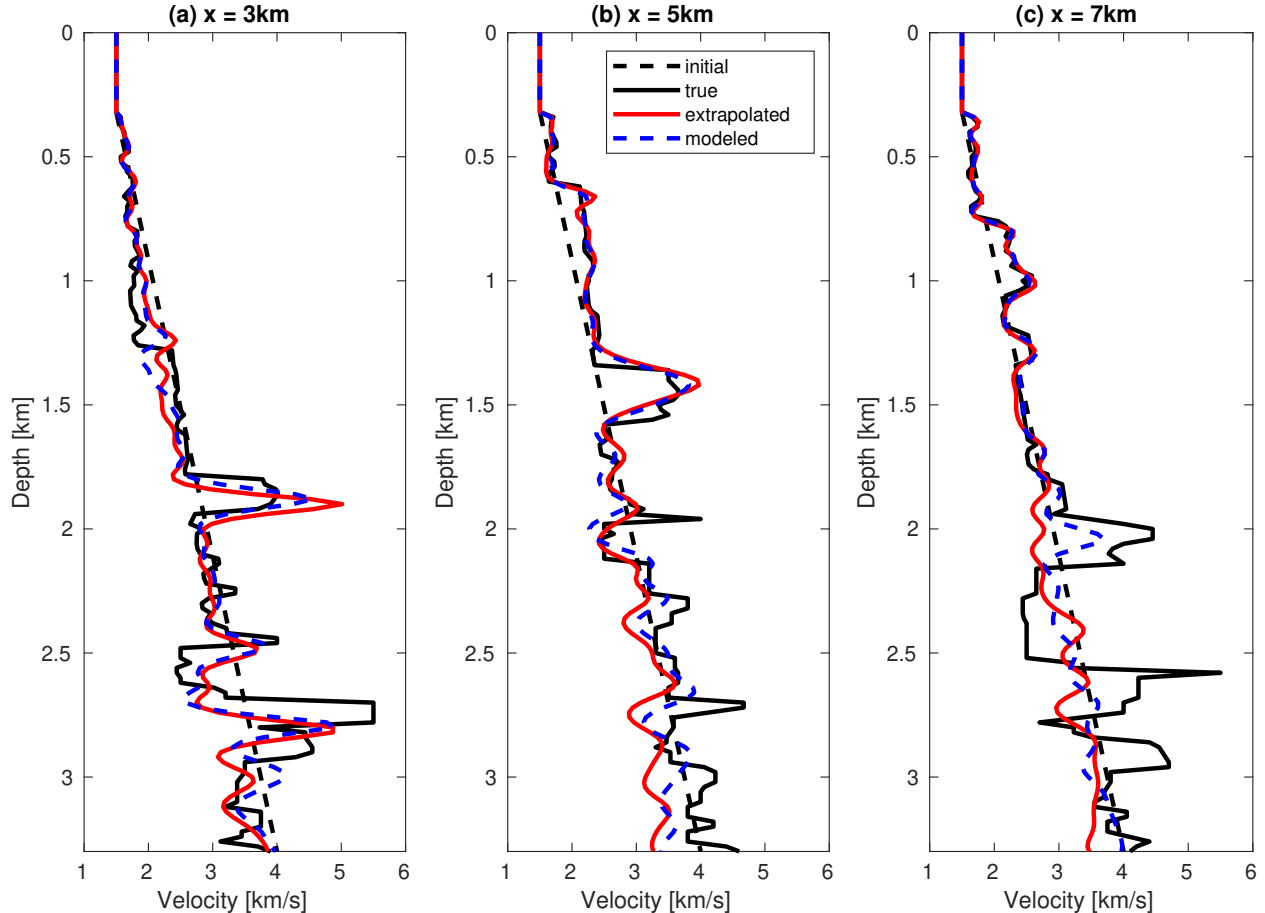


Figure 15: Comparison of velocity profiles among initial model (black dash line), true model (black line) and the resulting models started from the low-wavenumber models constructed with extrapolated (red line) and true (blue dash line) low frequencies at the horizontal locations of (a) $x=3\text{km}$, (b) $x=5\text{km}$ and (c) $x=7\text{km}$.

4.4 Extrapolated FWI: BP model

In deep learning, it is essential to estimate the generalization error of the proposed neural network for understanding its performance. Clearly, the intent is not to compute the generalization error exactly, since it involves an expectation over an unspecified probability distribution. Nevertheless, we can access the test error in the framework of synthetic shot gathers, hence we can use test error minus training error as a good proxy for the generalization error. For the purpose of assessing whether the network can truly generalize “out of sample” (when the training and testing geophysical models are very different) we train it with the samples collected from the submodels of Marmousi, but test it on the BP 2004 benchmark model (Figure 16). With the extrapolated low frequency data predicted by the neural network trained on the submodels of Marmousi, we perform the EFWI-CNN on the BP 2004 benchmark model (Figure 16).

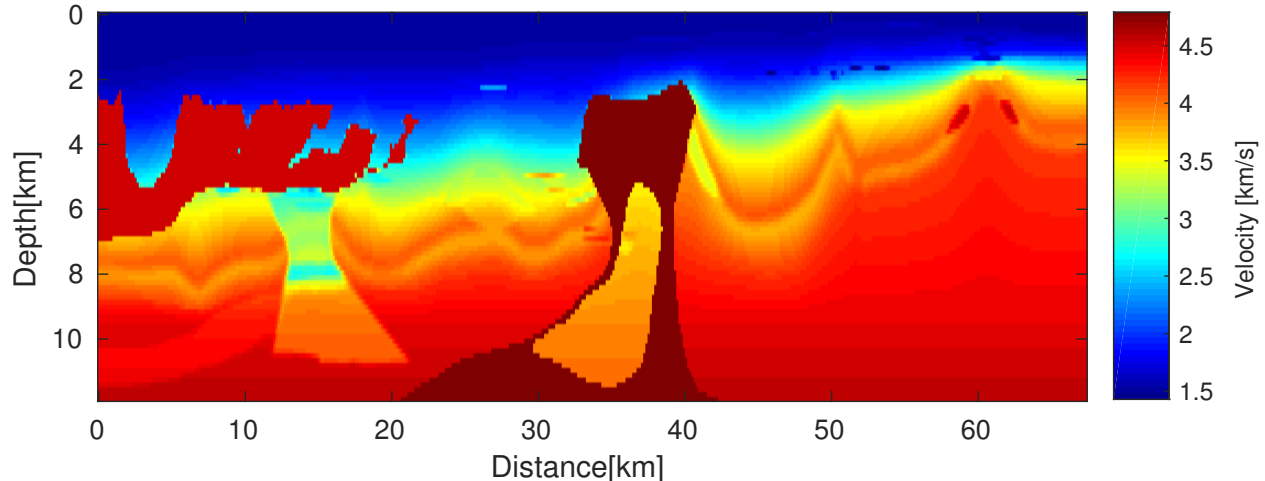


Figure 16: The 2004 BP benchmark velocity model used to collect the test data set for studying the generalizability of the proposed neural network. This model is the true model in extrapolated FWI.

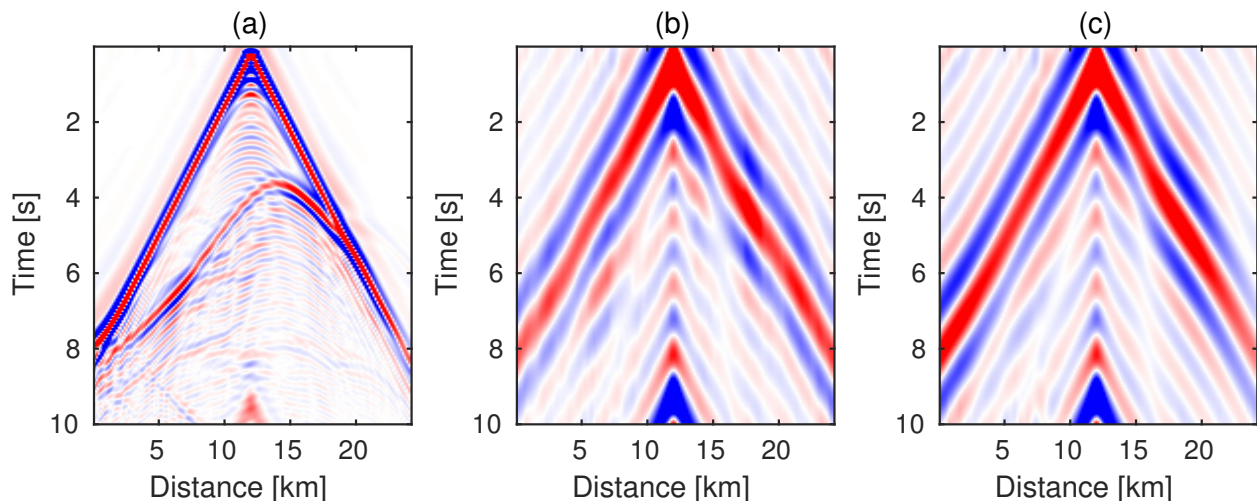


Figure 17: The extrapolation result on the BP model: comparison among the (a) bandlimited recordings (0.6 – 20Hz), (b) predicted and (c) true low frequency recordings (0.1 – 0.5Hz). The neural network trained on the Marmousi2 submodels can recover the low frequencies synthesized from the BP model, which illustrates that the proposed neural network can generalize well.

To reduce the computation burden, we downsample the BP 2004 benchmark model to 80×450 grid points with a grid interval of 150m. It is challenging for FWI to use only the bandlimited data to invert the shallow salt overhangs and the salt body with steeply dipping flanks in the BP model. Numerical examples show that, starting from the bad initial model (Figure 18(a)), the highest starting frequency to avoid cycle-skipping on this model is 0.3Hz. Therefore, we should extrapolate the bandlimited data to at least 0.3Hz to invert the BP model successfully.

In this example, we use a 7Hz Ricker wavelet as the source to simulate the fullband seismic records on both the training models (submodels of Marmousi) and the test model (BP model). The sampling interval and the total recording time are 5ms and 10s, respectively. To collect the input of the CNN, a highpass filter where the low frequency band (0.1 – 0.5Hz) is exactly zero is applied to the fullband seismic data. The bandlimited data (0.6 – 20Hz) are fed into the proposed CNN model to extrapolate the low frequency data in 0.1 – 0.5Hz trace by trace.

After training with 20 epoches, Figure 17 compares the extrapolation result of one shot gather on the BP model where the shot is located at 31.95km. The neural network can recover the low frequencies of reflections with some degree of accuracy. Even though the information contained in the data collected on Marmousi2 is physically unlike that of the salt dome model, the pretrained neural network can successfully find an approximation of their low frequencies based on the bandlimited inputs.

The extrapolated low frequency data are used to invert the low-wavenumber velocity model with the conventional FWI method. We observe that the accuracy of extrapolated low frequency decreases as the offset increases, so we limit the maximum offset to 12km. Starting from the initial model (Figure 18(a)), Figure 18(b) and Figure 18(c) show the low-wavenumber inverted models using 0.3Hz extrapolated data and 0.3Hz true data, respectively. Compared to the initial model, the resulting model using the 0.3Hz extrapolated data reveals the positions of the high and low velocity anomalies, which is almost the same as that of true data. The low-wavenumber background velocity models can still initialize the frequency-sweep FWI in the right basin of attraction.

Figure 19 compares the inverted models from FWI using 0.6-2.4Hz bandlimited data, starting from the respective low-wavenumber models in the previous figure. In (a), the resulting model starts from the original initial model. In (b), the resulting model starts from the inverted low-wavenumber velocity model using 0.3Hz extrapolated data. In (c), the resulting model starts from the inverted low-wavenumber velocity model using 0.3Hz true data. With the low-wavenumber velocity model, FWI can find the accurate velocity boundaries by exploring higher frequency data. However, the inversion settles in a wrong basin with only the higher frequency components. The low frequencies extrapolated with deep learning are reliable enough to overcome the cycle-skipping problem on the BP model, even though the training data set is ignorant of the particular subsurface structure of BP – salt bodies. Therefore, the neural network approach has the potential to deal favorably with real field data.

So far, the experiments on BP 2004 have assumed that data are available in a band starting at 0.6Hz. We now study the performance of EFWI-CNN when this band starts at a frequency higher than 0.6Hz. We still start the frequency-sweep FWI with 0.3Hz extrapolated data, and the highest frequency is still fixed at 2.4Hz. Figure 20 and Figure 21 compare the conventional FWI and EFWI-CNN results with data bandlimited above 0.9Hz and 1.2Hz, respectively. With the increase of the lowest frequency of bandlimited data, Figure 22 compares the quality of the inverted models at each iteration for FWI using fullband data, EFWI-CNN, and FWI using only the bandlimited data. The norm of the relative model error is used to evaluate the model quality, as (Brossier et al., 2009)

$$mq = \frac{1}{N} \left\| \frac{\mathbf{m}_{inv} - \mathbf{m}_{true}}{\mathbf{m}_{true}} \right\|_2 \quad (4)$$

where \mathbf{m}_{inv} and \mathbf{m}_{true} are the inverted model and the true model, respectively. N denotes the number of grid point in the computational domain. The performance of EFWI-CNN of course decreases with the increase of the lowest frequency of the bandlimited data, because this leads to more extrapolated data involved in the frequency-sweep FWI. The more iterations of FWI with the extrapolated data, the more errors the inverted model will have before exploring the true bandlimited data. Overfitting of the unfavorable extrapolated data makes the inversion worse after several iterations with the extrapolated data. However, EFWI-CNN is still superior to using FWI with only bandlimited data. We observe that EFWI-CNN with the current architecture still helps to reduce the inverted model error on the BP model when the lowest available frequency is as high as 1.2Hz.

Finally, we encountered a puzzling numerical phenomenon: the accuracy of the extrapolated data at the single frequency 0.3Hz depends very weakly on the band in which data is available, whether it be $[0.6, 20]$ Hz or $[1.2, 20]$ Hz for instance. As mentioned earlier, extrapolating data from 0.3Hz to 1.2Hz, so as to be useful for EFWI starting at 1.5Hz, is the much tougher task.

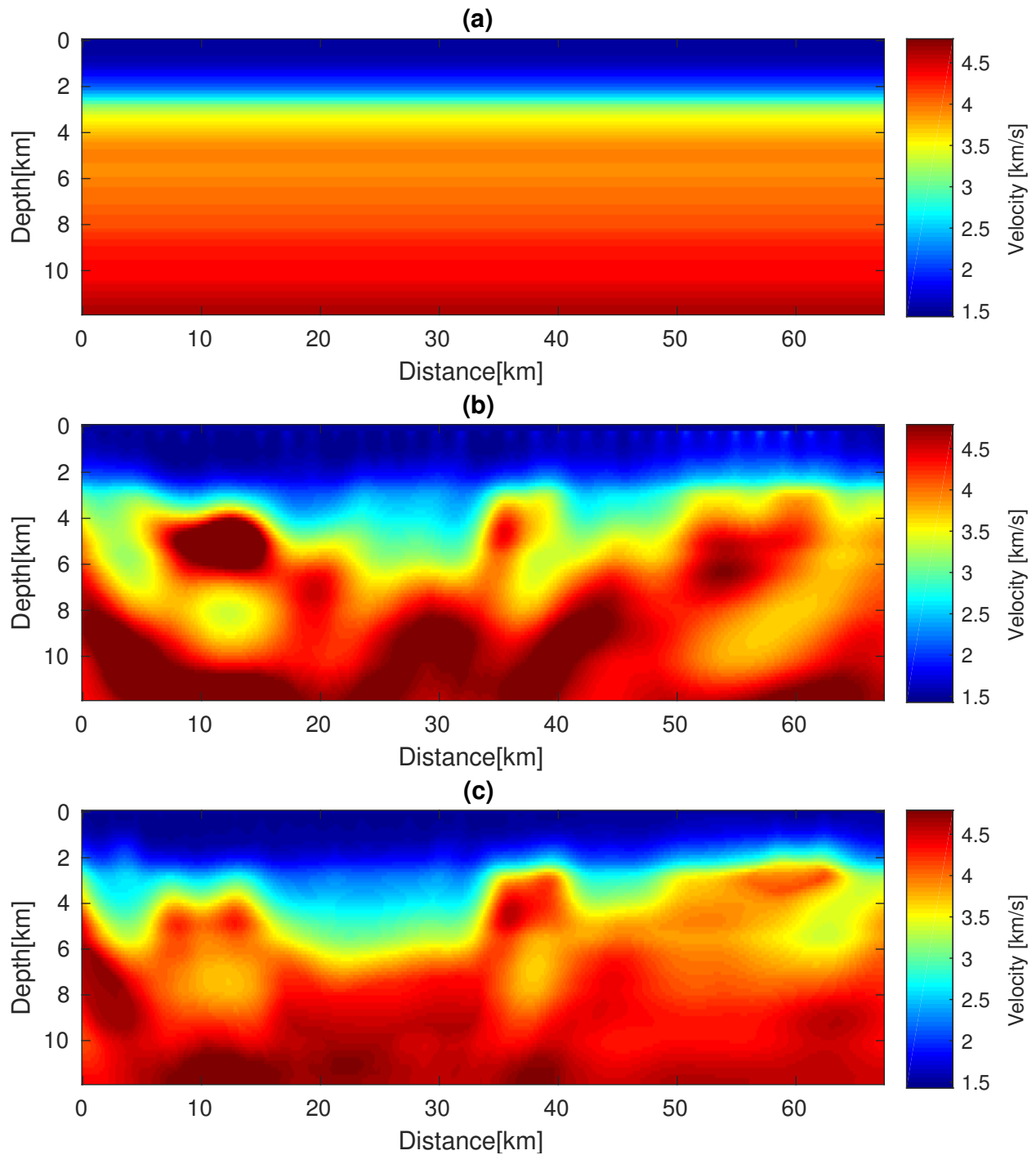


Figure 18: Comparison among (a) the initial model for FWI on the BP model, the inverted low-wavenumber velocity models using (b) 0.3Hz extrapolated data and (c) 0.3Hz true data. The inversion results in (b) and (c) are started from the initial model in (a).

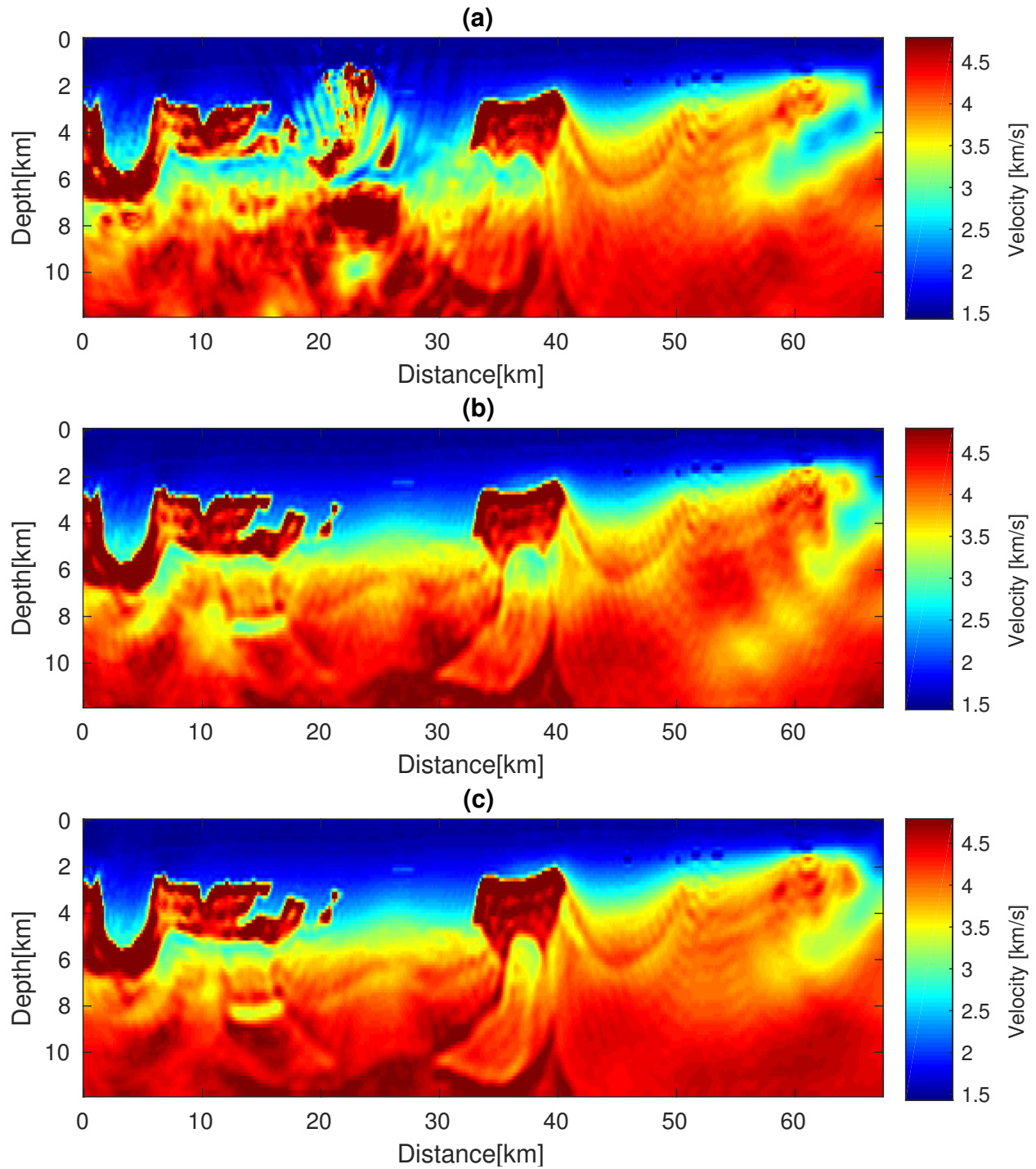


Figure 19: Comparison of the inverted models from FWI using 0.6-2.4Hz bandlimited data. In (a), resulting model starts from the original initial model. In (b), resulting model starts from the inverted low-wavenumber velocity model using 0.3Hz extrapolated data. In (c), resulting model starts from the inverted low-wavenumber velocity model using 0.3Hz true data.

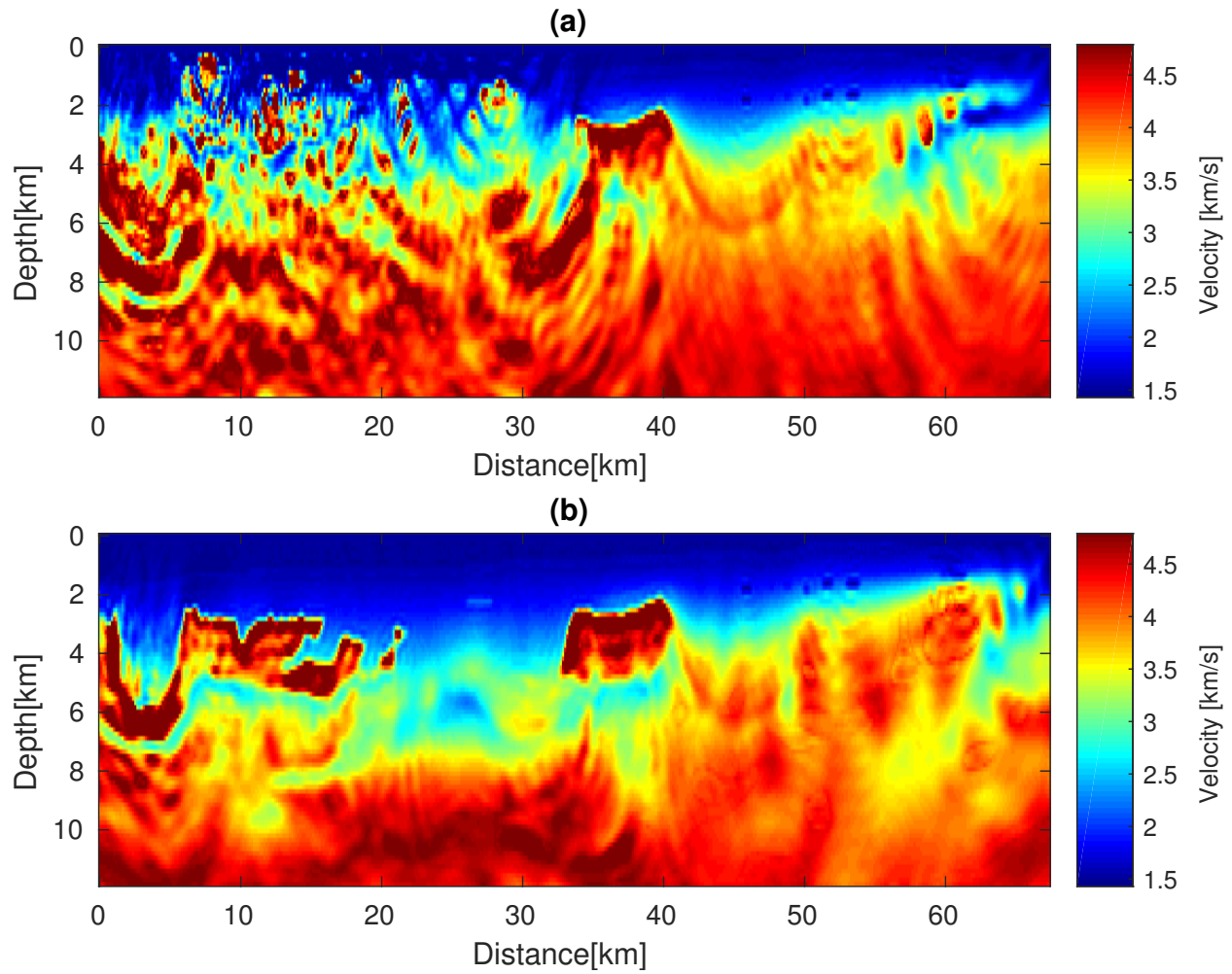


Figure 20: Comparison of the inverted models from FWI using 0.9-2.4Hz bandlimited data. In (a), resulting model starts from the original initial model. In (b), resulting model starts from the inverted low-wavenumber velocity model using 0.3Hz and 0.6Hz extrapolated data. The extrapolated data below 0.9Hz are recovered by 0.9-20Hz bandlimited data.

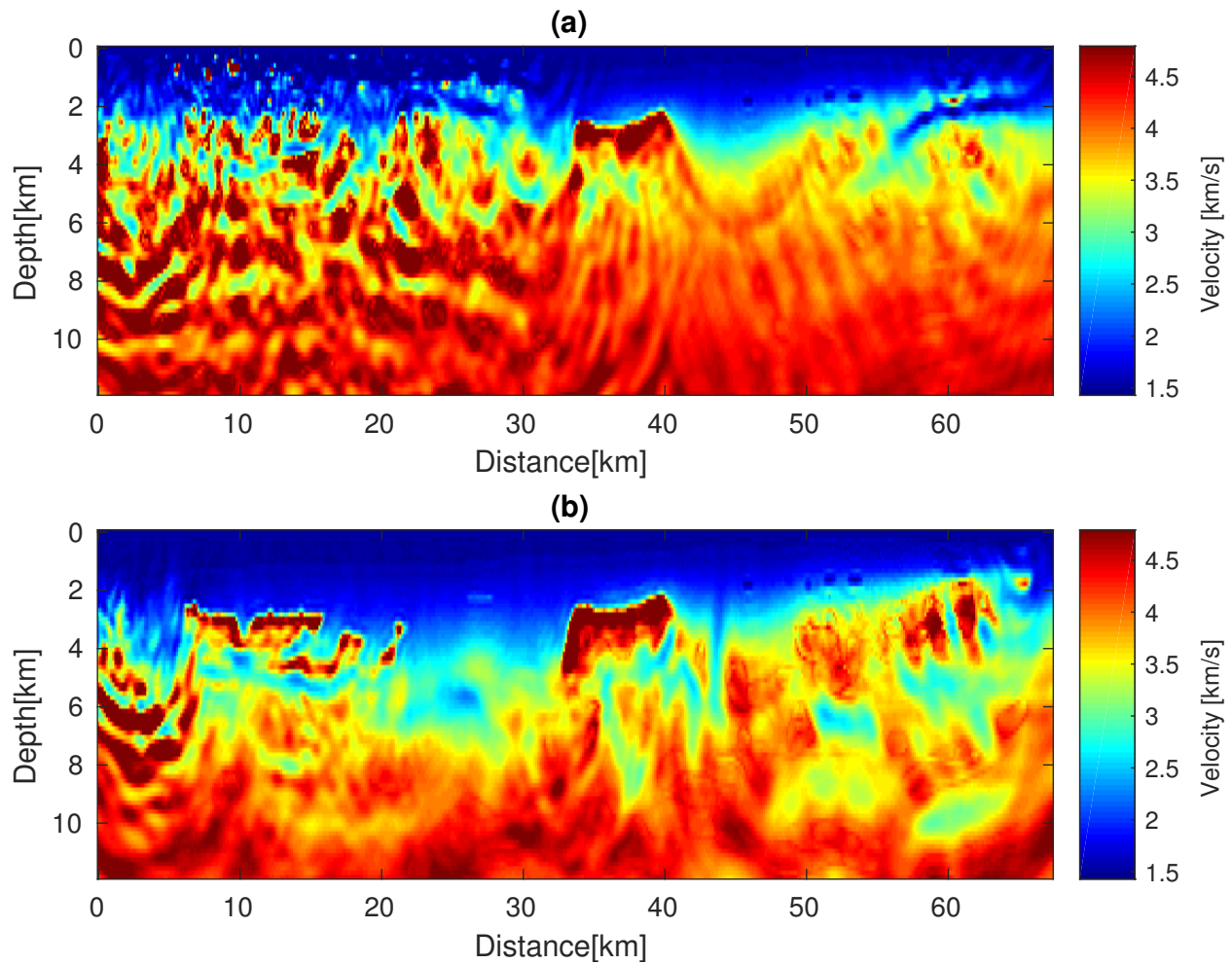


Figure 21: Comparison of the inverted models from FWI using 1.2-2.4Hz bandlimited data. In (a), resulting model starts from the original initial model. In (b), resulting model starts from the inverted low-wavenumber velocity model using 0.3Hz, 0.6Hz and 0.9Hz extrapolated data. The extrapolated data below 1.2Hz are recovered by 1.2-20Hz bandlimited data.

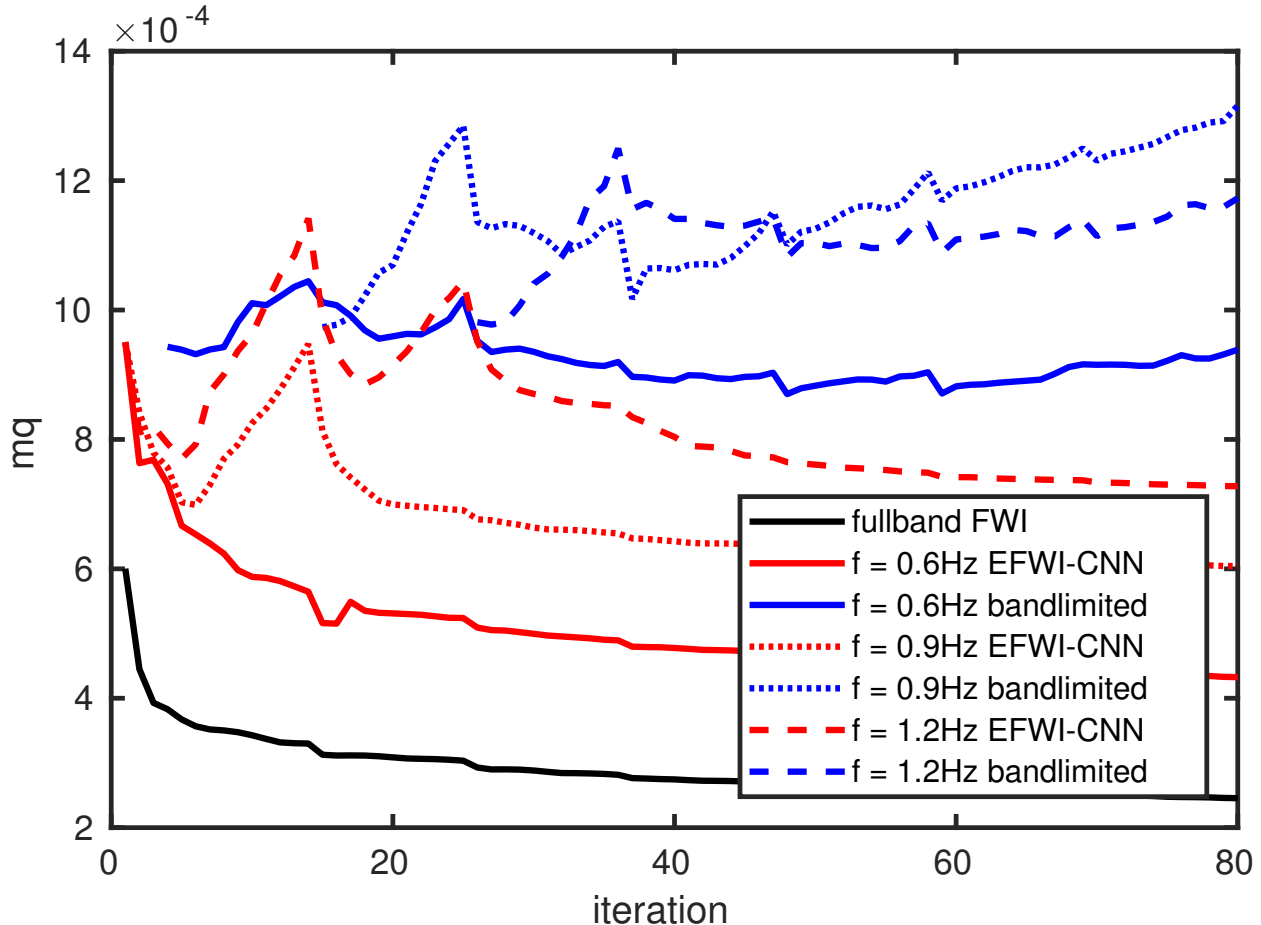


Figure 22: Comparison of the quality of the inverted models at each iteration for FWI using fullband data (black line), EFWI-CNN (red line) and FWI using only bandlimited data (blue line). f donates the lowest frequency of the bandlimited data. The highest frequency of inversion is fixed at 2.4Hz. The performance of EFWI-CNN decreases with the increase of the lowest frequency of the bandlimited data. However, compared with FWI using only bandlimited data, EFWI-CNN improves the quality of the inverted model very well.

5 Discussions and Limitations

The most important limitation of CNN for bandwidth extension is the possibly large generalization error that can result from an incomplete training set, or an architecture unable to predict well out of sample. As a data-driven statistical optimization method, deep learning requires a large number of samples (usually millions) to become an effective predictor. Since the training data set in this example is small but the model capacity (trainable parameters) is large, it is very easy for the neural network to overfit, which seriously deteriorates the extrapolation accuracy. Therefore, in practice, it is standard to use regularization or dropout, with only empirical evidence that this addresses the overfitting problem.

In addition, the training time for deep learning is highly related to the size of the dataset and the model capacity, and thus is very demanding. To speed up the training by reducing

the number of weights of neural networks, we can downsample both the inputs and outputs, and then use bandlimited interpolation method to recover the signal after extrapolation.

Another limitation of deep learning is due to the unbalanced data. The energy of the direct wave is very strong compared with that of the reflected waves, which biases the neural networks towards fitting the direct wave and contributing less to the reflected waves. Therefore, the extrapolation accuracy of the reflected waves is not as good as that of the primary wave in this example. Moreover, if we perform bandwidth extension trace by trace, the accumulation of the predicted errors reduces the coherence of the event across traces. Hence, multi-trace extrapolation can alleviate this problem to a certain degree.

Even though we are encouraged by the ability of a CNN to generate $[0.1, 0.5]$ Hz data for the BP 2004 model, much work remains to be done to be able to find the right architecture that will generate data in larger frequency bands, for instance in the $[0.1, 1.4]$ Hz band. Finding a suitable network architecture, hyperparameters, and training schedule for such cases remains an important open problem. Other community models, and more realistic physics such as elastic waves, are also left to be explored.

6 Conclusions

In this paper, deep learning is applied to the challenging bandwidth extension problem that is essential for FWI. We formulate bandwidth extension as a regression problem in machine learning and propose an end-to-end trainable model for low frequency extrapolation. Without preprocessing on the input (the bandlimited data) and post-processing on the output (the extrapolated low frequencies), CNN sometimes have the ability to recover the low frequencies of unknown subsurface structure that are completely missing at the training stage. The extrapolated low frequency data can be reliable to invert the low-wavenumber velocity model for initializing FWI on the bandlimited data without cycle-skipping. Even though there is freedom in choosing the architectural parameters of the deep neural network, making the CNN have a large receptive field is necessary for low frequency extrapolation. The extrapolation accuracy can be further modified by adjusting the architecture and hyperparameters of the neural networks depending on the characteristics of the bandlimited data.

References

- Araya-Polo, M., T. Dahlke, C. Frogner, C. Zhang, T. Poggio, and D. Hohl, 2017, Automated fault detection without seismic processing: *The Leading Edge*, **36**, 208–214.
- Araya-Polo, M., J. Jennings, A. Adler, and T. Dahlke, 2018, Deep-learning tomography: *The Leading Edge*, **37**, 58–66.
- Billette, F., and S. Brandsberg-Dahl, 2005, The 2004 bp velocity benchmark: Presented at the 67th EAGE Conference & Exhibition.
- Brossier, R., S. Operto, and J. Virieux, 2009, Seismic imaging of complex onshore structures by 2d elastic frequency-domain full-waveform inversion: *Geophysics*, **74**, WCC105–WCC118.
- Chen, Y., J. Hill, W. Lei, M. Lefebvre, J. Tromp, E. Bozdogan, and D. Komatitsch, 2017, Automated time-window selection based on machine learning for full-waveform inversion: *Society of Exploration Geophysicists*.
- De, S., D. Deo, G. Sankaranarayanan, and V. S. Arikatla, 2011, A physics-driven neural networks-based simulation system (phynness) for multimodal interactive virtual environments involving nonlinear deformable objects: *Presence*, **20**, 289–308.
- Glorot, X., and Y. Bengio, 2010, Understanding the difficulty of training deep feedforward neural networks: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 249–256.
- Goodfellow, I., Y. Bengio, and A. Courville, 2016, *Deep learning*, **1**.
- Grzeszczuk, R., D. Terzopoulos, and G. Hinton, 1998, Neuroanimator: Fast neural network emulation and control of physics-based models: *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ACM, 9–20.
- Guitton, A., H. Wang, and W. Trainor-Guitton, 2017, Statistical imaging of faults in 3d seismic volumes using a machine learning approach: *Society of Exploration Geophysicists*.
- He, K., X. Zhang, S. Ren, and J. Sun, 2015, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification: *Proceedings of the IEEE international conference on computer vision*, 1026–1034.
- Hewett, R., L. Demanet, and the PySIT Team, 2013, PySIT: Python seismic imaging toolbox v0.5. (Release 0.5).
- Hornik, K., M. Stinchcombe, and H. White, 1989, Multilayer feedforward networks are universal approximators: *Elsevier*, **2**.
- Hu, W., 2014, Fwi without low frequency data-beat tone inversion, *in* SEG Technical Program Expanded Abstracts 2014: *Society of Exploration Geophysicists*, 1116–1120.
- Hu, Y., L. Han, Z. Xu, F. Zhang, and J. Zeng, 2017, Adaptive multi-step full waveform inversion based on waveform mode decomposition: *Elsevier*, **139**.
- Jin, Y., W. Hu, X. Wu, and J. Chen, 2018, Learn low wavenumber information in fwi via deep inception based convolutional networks, *in* SEG Technical Program Expanded Abstracts 2018: *Society of Exploration Geophysicists*, 2091–2095.
- Kingma, D. P., and J. Ba, 2014, Adam: A method for stochastic optimization: *arXiv preprint arXiv:1412.6980*.
- Levy, S., and P. K. Fullagar, 1981, Reconstruction of a sparse spike train from a portion of its spectrum and application to high-resolution deconvolution: *Geophysics*, **46**, 1235–1243.
- Lewis, W., and D. Vigh, 2017, Deep learning prior models from seismic images for full-

- waveform inversion: Society of Exploration Geophysicists.
- Li, S., B. Liu, Y. Ren, Y. Chen, S. Yang, Y. Wang, and P. Jiang, 2019, Deep learning inversion of seismic data: arXiv preprint arXiv:1901.07733.
- Li, Y. E., and L. Demanet, 2015, Phase and amplitude tracking for seismic event separation: Society of Exploration Geophysicists, **80**.
- , 2016, Full-waveform inversion with extrapolated low-frequency data: Society of Exploration Geophysicists, **81**.
- Nocedal, J., and S. J. Wright, 2006, Numerical optimization 2nd.
- Oldenburg, D., T. Scheuer, and S. Levy, 1983, Recovery of the acoustic impedance from reflection seismograms: Geophysics, **48**, 1318–1337.
- Ovcharenko, O., V. Kazei, D. Peter, X. Zhang, and T. Alkhalifah, 2018, Low-frequency data extrapolation using a feed-forward ann: Presented at the 80th EAGE Conference and Exhibition 2018.
- Richardson, A., 2018, Seismic full-waveform inversion using deep learning tools and techniques: arXiv preprint arXiv:1801.07232.
- Sacchi, M. D., D. R. Velis, and A. H. Cominguez, 1994, Minimum entropy deconvolution with frequency-domain constraints: Geophysics, **59**, 938–945.
- Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, 2014, Dropout: a simple way to prevent neural networks from overfitting: The Journal of Machine Learning Research, **15**, 1929–1958.
- Walker, C., and T. J. Ulrych, 1983, Autoregressive recovery of the acoustic impedance: Geophysics, **48**, 1338–1350.
- Wang, R., and F. Herrmann, 2016, Frequency down extrapolation with tv norm minimization: Society of Exploration Geophysicists.
- Wu, R.-S., J. Luo, and B. Wu, 2014, Seismic envelope inversion and modulation signal model: Society of Exploration Geophysicists, **79**.
- Wu, Y., Y. Lin, and Z. Zhou, 2018, Inversionnet: Accurate and efficient seismic waveform inversion with convolutional neural networks, *in* SEG Technical Program Expanded Abstracts 2018: Society of Exploration Geophysicists, 2096–2100.
- Xiong, W., X. Ji, Y. Ma, Y. Wang, N. M. BenHassan, M. N. Ali, and Y. Luo, 2018, Seismic fault detection with convolutional neural network: Geophysics, **83**, 1–28.
- Zhang, P., L. Han, Z. Xu, F. Zhang, and Y. Wei, 2017, Sparse blind deconvolution based low-frequency seismic data reconstruction for multiscale full waveform inversion: Elsevier, **139**.