IDEA League

Master of Science in Applied Geophysics

Research Thesis

Comparing results of an AI neural network method with a sparse optimization method for frequency-band reconstruction of seismic images

Samuel L. Tuppen

August 5, 2022

Comparing results of an AI neural network method with a sparse optimization method for frequency-band reconstruction of seismic images

Master of Science Thesis

for the degree of Master of Science in Applied Geophysics at
Delft University of Technology
ETH Zürich
RWTH Aachen University
by

Samuel L. Tuppen

August 5, 2022

Department of Geoscience & Engineering · Delft University of Technology
Department of Earth Sciences · ETH Zürich
Faculty of Georesources and Material Engineering · RWTH Aachen University

Printed in Switzerland

IDEA LEAGUE
JOINT MASTER'S IN APPLIED GEOPHYSICS

TU Delft, The Netherlands
ETH Zürich, Switzerland
RWTH Aachen, Germany

Dated: *August 5, 2022*

Supervisor(s):

Dr. Dirk-Jan van Manen

Dr. Thomas Elboth

Committee Members:

Dr. Dirk-Jan van Manen

Dr. Thomas Elboth

Prof. Evert C. Slob

Abstract

Motivated by environmental concern, the industry has been developing an alternative marine seismic source, in particular the marine vibrator. By spreading the emitted energy out over time, vibrator sources are perceived to be less intrusive to marine mammals. It is also believed that vibrators have greater control of the emitted source wavelet than can be achieved with traditional airguns. With the added control, it is possible to only emit portions of the frequency spectrum, which in turn allows for many applications such as deblending and the ability to avoid masking mammal communications. To effectively implement these, two methodologies are proposed to interpolate the frequency data that are not emitted. The first is a deep learning approach utilizing a U-Net architecture, with a custom frequency loss function. The second is a sparse optimization method that approximates the reflectivity series of the subsurface using known frequency content. By assuming that the signal can be represented sparsely and that all frequencies interact with the subsurface interfaces similarly at all frequencies, the frequency spectrum can be reconstructed. Both of the presented methods are tasked to interpolate the missing frequency band(s) in North Sea shot data. It is found that both methods are able to interpolate narrow 2.5 Hz bands, but are unable to accurately reconstruct wider (ex. 10 Hz), frequency bands. Overall, the U-Net shows better results than the sparse optimization method when the frequency gaps are positioned closely.

Acknowledgements

First of all, I would like to thank everyone who has played a part in this project. I have had the opportunity to be a part of this exciting project that has been offered by the Shearwater office in Oslo, where I got to dip my feet into the world of machine learning, seismic acquisition and processing. For this opportunity, I would like to thank my company supervisors Thomas Elboth and Kambiz Iranpour who not only shared a lot of their wisdom from the industry but also encouraged me to explore the beautiful outdoors of Norway.

Next, I want to give special thanks to Dr. Dirk-Jan van Manen who guided me throughout this thesis and gave me compassionate encouragement especially when the going got tough.

I would like to thank Jing Sun giving insight and help with the coding of the neural networks.

Lastly, I want to thank my family and friends who have been my constant source of support for these last two years. Their presence gave me strength to keep going and to finally reach the finish line of my masters.

Thank you all, I am extremely grateful.

Swiss Federal Institute of Technology — Samuel L. Tuppen

August 5, 2022

Table of Contents

Acronyms

Adam	Adaptive Moment Estimation
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
DNN	Deep Neural Networks
FWI	Full Waveform Inversion
LMO	Linear Moveout
MAE	Mean Absolute Error
MSE	Mean Square Error
NN	Neural Networks
NRMS	Normalized Root Mean Squared
PSNR	Peak Signal-to-Noise Ratio
ReLU	Rectified Linear Unit
RMSE	Root Mean Square
SNR	Signal-to-Noise Ratio
SSIM	Structural Similarity Index Measure

Chapter 1

Introduction

For years the industry standard for acquiring offshore seismic lines has been with large airgun arrays towed by seismic surveying vessels. The main benefit of using such airguns is that they are both repeatable and reliable, however, concerns have been raised about their potential negative environmental impact. Motivated by this, the industry is considering a viable alternative source design, for example, the marine vibrator. Whilst an airgun emits energy in a short-intense impulse, a marine vibrator will emit the energy over time as a so-called sweep. Both wavelets can be seen in Figure 1-1, where a standard airgun wavelet is shown on the left and a synthetic vibrator sweep is shown on the right. As these vibrator sources spread the emitted energy over time, they are perceived to be less intrusive to the environment.

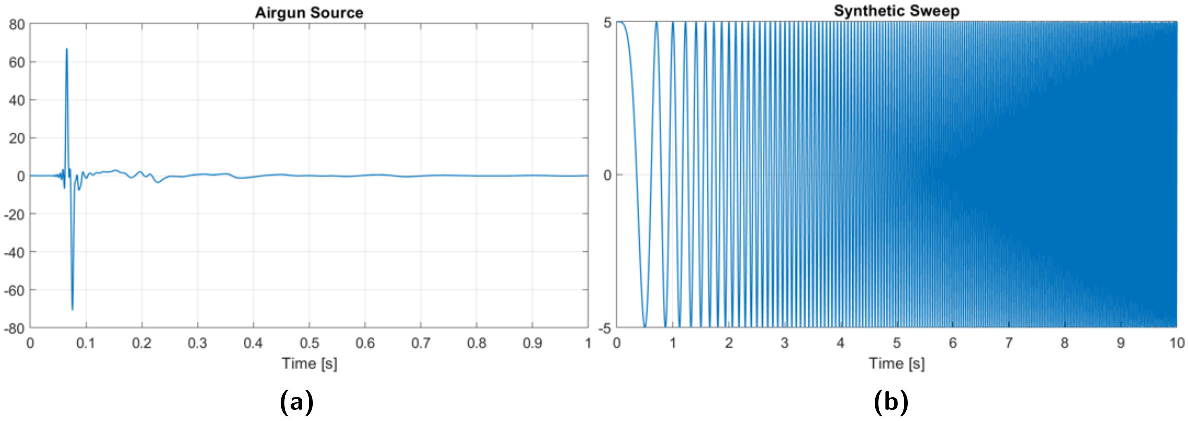


Figure 1-1: Illustrations comparing the emitted wavelet of a traditional airgun **(a)** and a vibrator source **(b)**. Notice that the amplitude is plotted on the y-axis, and is not the same in each figure.

Furthermore, it has been theorized that vibrators can achieve greater control of the emitted source (Southall et al., 2019), which leads to several interesting geophysical applications. One example of such an application is by altering the phase of the emitted waveform to remove residual noise from the data (Laws et al., 2019). This thesis, in particular, will look into the interpolation of missing frequency bands.

Possessing a tool to accurately interpolate missing frequency bands has various advantages, such as deblending and the avoidance of masking marine mammal communication, both of which are only possible with the control of the emitted waveform of a vibrator. Furthermore, interpolating missing frequency bands is also useful to potentially remove the ghost-notch and merge the bandwidth of a low and high-frequency source. A short description of the mentioned applications are described below.

- With deblending it is possible to improve the efficiency and/or sampling of seismic surveys by means of blended acquisition, whereby multiple shots/sweeps are emitted in near succession. With traditional airguns, deblending these shots is typically enabled in one of two manners. The first is by means of dithering the shot/sweep times (Tomishin and Chizhik, 1982) and the second is by means of phase encoding (Robertsson, 2016) and (Laws et al., 2019) where each source is phase encoded to allow for the sources to later be discerned/separated from one another. With vibrators we also have a third option; By taking advantage of the high-fidelity control of the emitted waveform, we can separate the data in the frequency domain by having each vibrator emit interleaved non-overlapping frequencies. Deblending is then reduced to applying a simple frequency filter. However, the missing frequency bands for each vibrator will still need to be interpolated.

- It is believed that mammals' communication does not get masked, as long as we do not emit over more than a third of an octave (Hawkins et al., 2014). However, the missing bandwidth then needs to be interpolated.

- Another situation where frequency must be interpolated is when a survey consists of several marine vibrators whereby each source focuses its energy on specific regions of the frequency spectrum. In this situation, the sources could potentially emit non-overlapping frequency bandwidth, resulting in a missing frequency band in the total bandwidth of the sources.

- Traditional marine seismic data contains a so-called ghost notch, caused by the signal reflecting on the sea surface. The removal of this notch from the recorded data has proven to be a challenge since the beginning of marine acquisition. Although this specific task is outside of the scope of this thesis, perhaps the proposed frequency interpolation methods will prove to be beneficial when addressing this problem.

In this thesis, I propose two methods to interpolate the missing frequency bands. The first is a deep learning approach whereby a network is trained on a dataset that is similar to the problem being solved. The main benefit of deep learning/Neural Networks (NN) is their ability to learn a specific problem, without explicit instructions (LeCun et al., 2015). It is for this reason that the use of deep learning has steadily increased over the last few years and seen numerous geoscientific applications (Dramsch, 2020). For example, deep learning approaches have been demonstrated for fault detection (Ma et al., 2018), earthquake magnitude prediction (Panakkat and Adeli, 2007) and land cover classification with remote sensing data (Kussul et al., 2017). Works more related to this thesis are a deep learning application wherein low-frequency data is extrapolated (Ovcharenko et al., 2019), the interpolation of missing traces from seismic data (Wang et al., 2019) and seismic interference noise removal (Sun et al., 2020).

Clearly, deep learning has shown to be a promising approach for complex problems, such as the one presented.

The second method that will be considered is a sparse optimization method that is inspired by compressed sensing approaches. The method approximates the reflectivity of the subsurface using the known frequency content. By then assuming that all frequencies interact in with the subsurface in the same way, that is, by reflecting off the same interfaces, the frequency spectrum is reconstructed. Originally, the method was first presented by Taylor et al. (1979) and later implemented to seismic data by Wang and Herrmann (2016) to extrapolate low frequency data. I then also hypothesize that with an additional τ-p transform, spatial continuity in the reconstruction can be implemented.

However, as there is unfortunately no vibrator data available for the purposes described in this thesis, one must speculate as to what can be expected in such vibrator data. Throughout this thesis, it is assumed that the emitted waveform is known and can be used to deconvolve the vibrator data and replaced with with an alternative waveform by means of a filtering algorithm, for example, Wiener filtering. The new waveform could then be one that resembles the waveform of an airgun, for example. This way, the vibrator data will essentially be pseudo airgun data, which is the data type used for the tests in this thesis.

1-1 Research Aim

Being able to interpolate missing frequency bands is beneficial from an environmental perspective, as well as for the efficiency of marine acquisition. The goal of this study is to compare the presented methods' ability to reconstruct missing frequency bands in seismic shot gathers. The hypothesis is that both the neural network and sparse solver are able to reconstruct the missing frequency bands in the spectrum, whereby the accuracy of the sparse solver is expected to be better.

The research question then becomes: "What differences are there between a deep learning approach and a sparse optimization method when tasked to interpolate missing frequency bands?" Whereby a secondary research question, "Are specific frequency ranges harder to reconstruct than others?" will also be looked into.

Being able to answer these questions will bring insight as to whether the high fidelity control of the vibrator, in particular its ability to emit particular frequency ranges, can be employed to enhance acquisition efficiency and sampling, as well as minimize negative environmental impacts.

1-2 Research outline

The structure of the thesis is as follows. First, in Chapter 2, I will describe the basic components of Artificial and Convolutional Neural Networks, such that the components of the employed network are understood. I will then also explain the data preparation and training process for the network to learn the problem. In Chapter 3 I will briefly explain the τ-p/slant stank transform, such that it can later be implemented into the sparse solver, whereafter then present the sparse solver. Chapter 4 will present the results of the research and Chapter 5 will discuss those presented results. The final chapter, Chapter 6, of this thesis sums up the main findings and concludes the thesis.

Chapter 2

Deep Learning Approach

This chapter looks into the first method for interpolating missing frequency bands in seismic data using a deep learning approach. Deep learning is a subfield of machine learning that uses Deep Neural Networks (DNN), which are a type of Artificial Neural Network (ANN) that consists of many layers and are hence referred to as 'deep'. Such ANNs take inspiration from biological neural networks, whereby a number of artificial neurons are arranged into a series of layers that can receive and send signals in the form of a real number to and from other artificial neuron layers through connections. Each neuron and connection has an associated weight and bias, which is tuned when training the network for a specific task. Through training, ANNs are able to learn how to perform certain tasks, extract information and/or understand the relationships in complex datasets, without explicit programming (LeCun et al., 2015).

In image-related problems, one shortcoming of a traditional ANN is that a large number of input parameters are necessary to feed the image into the network, and thus there are a large number of weights and biases that need to be tuned. This issue is circumvented with the help of a Convolutional Neural Network (CNN), which is a class of ANN that contains a so-called convolutional layer. Rather than connecting every neuron in each layer to those in the subsequent layers, the convolution layer uses a convolution operation to sample regions of the image. This aspect makes CNNs powerful tools as they reduce the number of parameters in the network and thus reduce the training time required to tune those parameters. CNNs are also great tools for detecting local features, which is very applicable in image-related problems.

In this thesis, I use a U-Net, which is a specific type of DNN that consists of several encoder-decoder pairs arranged into a characteristic U-shape. However, before I elaborate on this particular network architecture, I will first explain the necessary components to construct a U-Net. Thereafter, I will explain the training procedure as well as the data preparation steps.

2-1 Basic neural networks

In this section of the thesis, I will present the components of ANNs that are necessary to understand the architecture and inner workings of the final network used.

2-1-1 Forward Propagation

An ANN is comprised of several neuron layers wherein each neuron is connected to neurons in successive layers. The first layer of the network is referred to as the input layer, whilst the last layer of the network is referred to as the output layer. All layers between the input and output layers are called hidden layers. The more hidden layers that make up a network's architecture, the deeper the network is considered to be. As the name suggests, the input layer is where the network is presented with the input parameters, for example, all pixel values of an image. This information is then fed through the network following a feed-forward/forward propagation algorithm with the hopes that the network is able to produce an accurate approximation $\hat{\mathbf{y}}$ of the ground truth $\mathbf{y}$. What the ground truth is, of course, depends on the particular application. The forward propagation algorithm is the process of computing the values of each neuron in a layer, by using the output of the previous neuron layer. This process starts at the input layer and continues through to the output layer.

If we temporarily consider a single node, typically the input vector for this neuron is denoted by $\mathbf{x}$. So, if an input to this neuron has n number of features, then $\mathbf{x} = [x_1, x_2, ..., x_n]$. Furthermore, each of the input features contained in the vector $\mathbf{x}$ has a corresponding weight w_i, which are the elements of the weights vector $\mathbf{w}$. These can be viewed as the weights of the connections between two neurons. The output of the neuron, z, can then be computed by:

$$z = \mathbf{w}^T\mathbf{x} + b, \tag{2-1}$$

where $\mathbf{x}$ is a column vector, $\mathbf{w}^T$ is a row vector and b is a bias that is added to deter dead (zero value) neurons. For a layer consisting of multiple neurons, we can rewrite (2-1) as

$$\mathbf{z} = \mathbf{W}^T\mathbf{x} + \mathbf{b}, \tag{2-2}$$

where $\mathbf{W}^T$ is the weights matrix. Furthermore, with an ANN we tend to require a certain degree of complexity to solve most problems and as (2-2) is a linear operation, it has a reduced complexity compared to nonlinear operations. Hence, the above output $\mathbf{z}$ is typically passed through a so-called nonlinear activation function $g(\mathbf{z})$, which will be expanded on in Section 2-1-2, to introduce an added degree of complexity to the network. Then, the output $\mathbf{a}$ of a given layer is given by:

$$\mathbf{a} = g(\mathbf{W}^T\mathbf{x} + \mathbf{b}) = g(\mathbf{z}). \tag{2-3}$$

The above procedure is computed for every layer of the network, ending at the output layer, which also concludes the forward propagation algorithm.

2-1-2 Activation functions

ANNs are very powerful tools that are able to find relationships in complex datasets, which is why they are often employed for highly complex problems. However, without a nonlinear activation function, the network is a system of linear operations and will struggle to approximate such problems. Activation functions are a means to add additional complexity to the network that can not be obtained with solely linear operations and is hence essential in most ANN applications (Sibi et al., 2013).

There are various nonlinear activation functions and one can implement different activation functions for every layer of the network. Though, within the layers, each neuron is typically given the same activation function. In the past, the most commonplace activation functions were the logistic sigmoid and hyperbolic tangent:

$$g_{sigmoid}(z) = \frac{1}{1+e^{-z}}, \tag{2-4}$$

$$g_{tanh}(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}. \tag{2-5}$$

Each activation function has it's characteristics, for example, the sigmoid will always output a value between 0 and 1, and is therefore useful for applications where a probability is desired. The hyperbolic tangent, on the other hand, outputs values in the range of $[-1, 1]$, and for that reason is often preferred over the sigmoid (Karlik and Olgac, 2011). In modern-day applications, the most popular activation function is Rectified Linear Unit (ReLU), which returns zero for any negative input value and returns the input for any positive input value:

$$g_{ReLU}(\mathbf{z}) = \max\{0, \mathbf{z}\}. \tag{2-6}$$

The rationale behind choosing ReLU over an alternative activation function is that it and its derivative are cheap and fast to compute, which in turn greatly reduces the required training time (Goodfellow et al., 2017). For this reason, ReLU has also been implemented in all hidden layers in the network created for this thesis. Despite that, ReLU is not perfect. One shortcoming is that the derivative is undefined at $z = 0$. However, in practice this fortunately rarely impacts results. For completeness, another shortcoming is that it is possible for dead neurons to occur during training, as the gradient, at $x < 0$ is zero. In some cases, a modified ReLU, called a leaky ReLU, is implemented to circumvent such inactive neurons. It is defined as

$$g_{LeakyReLU}(\mathbf{z}) = \max\{a\mathbf{z}, \mathbf{z}\}, \tag{2-7}$$

where a is a small constant, which allows for the gradient for $x < 0$ to be non-zero.

2-1-3 Loss function

The loss/cost function of an ANN is a measure that is used to quantify the network's performance. The choice of loss function is case specific and can vary greatly depending on the application. For example, if the ANN is tasked with a classification problem, then a completely different loss function is chosen than what would be optimal for a regression-type problem. The choice of loss function has a direct impact on the rate of convergence of the network. In regression-type problems, Mean Absolute Error (MAE) and Mean Square Error (MSE) are the most commonly used loss functions, due to their simplicity and high adaptability. In the case of image processing, they work by comparing each pixel value of the ground truth $\mathbf{y}$ and the output $\hat{\mathbf{y}}$ by means of an $\ell1$ and $\ell2$ norm, respectively:

$$L_{MAE} = ||\mathbf{y} - \hat{\mathbf{y}}||_1, \tag{2-8}$$

$$L_{MSE} = ||\mathbf{y} - \hat{\mathbf{y}}||_2. \tag{2-9}$$

In general, both loss functions produce satisfactory results, however in the case of interpolating missing frequency bands, the MSE loss alone was insufficient to accurately reconstruct the frequency spectrum. This stems from the fact that MSE loss is sensitive to outliers and tends to produce over-smoothed results (Li et al., 2021a). Conversely, MAE is less sensitive to outliers and penalizes high errors less. Neither of these is ideal for frequency band interpolation, so a custom loss function L consisting of two components is considered. The first is a Huber loss function, which is formulated as:

$$L_{Huber} = \begin{cases} \frac{1}{2}(\mathbf{y} - \hat{\mathbf{y}})^2 & \text{for } |\mathbf{y} - \hat{\mathbf{y}}| \leq \delta, \\ \delta \cdot (|\mathbf{y} - \hat{\mathbf{y}}| - \frac{1}{2}\delta), & \text{otherwise,} \end{cases} \tag{2-10}$$

where $\delta = 1$. As can be seen in (2-10), the Huber loss behaves quadratically for small values of $\mathbf{y} - \hat{\mathbf{y}}$ and linearly for large values of $\mathbf{y} - \hat{\mathbf{y}}$. This has the benefit of being sensitive to small changes, whilst also penalizing large errors. The second component of L is a MAE loss and evaluates the difference between $\mathbf{y}$ and $\hat{\mathbf{y}}$ in the FK domain:

$$L_{MAE} = ||\tilde{\mathbf{y}} - \tilde{\hat{\mathbf{y}}}||_2 \tag{2-11}$$

where $\tilde{\mathbf{y}}$ and $\tilde{\hat{\mathbf{y}}}$ are the FK transforms of $\mathbf{y}$ and $\hat{\mathbf{y}}$, respectively. By combining (2-10) and (2-11) into one loss function we obtain L:

$$L = L_{Huber} + L_{MAE}, \tag{2-12}$$

which is the loss function used throughout this thesis.

2-1-4 Backpropagation

Where forward propagation represents the forward flow of information to compute a loss, the backpropagation algorithm (Rumelhart et al., 1986) will use the loss function and flow backwards through the network to compute the gradient (Goodfellow et al., 2017). Finding the gradient is a necessary part of the process that enables the network to update its weights and biases in a meaningful way and is, therefore, a crucial step in the network's learning process.

Nielsen (2015) describes the backpropagation algorithm as follows. Consider a modified version of (2-2) that contains multiple layers in a network,

$$\mathbf{a}^l = g((\mathbf{W}^l)^T \mathbf{x}^l + \mathbf{b}^l) = g(\mathbf{z}^l), \tag{2-13}$$

where the superscript l represents the layer number in the ANN. Next, we compute the gradient of the loss function with respect to the weights and biases: $\frac{\partial L}{\partial w_{jk}}$ and $\frac{\partial L}{\partial b_j}$, where subscripts j and k are the node number in the layer l and the node number in the previous layer $l-1$, respectively. Furthermore, let δ_j^l be the error[1] in the j$^{\text{th}}$ neuron of the l$^{\text{th}}$ layer. Then,

$$\delta_j^l = \frac{\partial L}{\partial z_j^l} = \frac{\partial L}{\partial a_j^l}\frac{\partial a_j^l}{\partial z_j^l} = \frac{\partial L}{\partial a_j^l} g'(\partial z_j^l) \tag{2-14}$$

where ∂z_j^l is the weighted input to a node and $a_j^l = g(\partial z_j^l)$, meaning that $\frac{\partial a_j^l}{\partial z_j^l} = g'(\partial z_j^l)$. Initially, (2-14) is evaluated at $l = L$, where L is the last layer of the network and should not be confused with the loss function. In a vectorized form we then have

$$\boldsymbol{\delta}^L = \nabla_a L \odot g'(\mathbf{z}^L), \tag{2-15}$$

where $\odot$ represents an element-wise multiplication. Then, (2-15) can be rewritten into a general form representing the backward moving error from layer $l+1$ to l as

$$\delta^l = ((\mathbf{W}^{l+1})^T \delta^{l+1}) \odot g'(\mathbf{z}^l). \tag{2-16}$$

Lastly, as mentioned, when training a network we adjust the weights and biases such that the output of the ANN closely resembles the target. Fortunately, we can find the rate of change of the loss function w.r.t. every the weight and bias in the network with (2-17) and (2-18), respectively:

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l, \tag{2-17}$$

$$\frac{\partial C}{\partial w_{jk}^l} = a_k^{(l-1)} \delta_j^l. \tag{2-18}$$

As we know δ_j^l from (2-16), we can find how each weight and bias should be adjusted to improve the network's performance.

2-1-5 Optimization function

In both deep learning as well as machine learning in general, the minimization of the loss function is a key part of the learning process. The role of the optimizer is to use the information from the backpropagation algorithm to adjust the weights and biases and minimize the

[1] It will become clear shortly why we can refer to an error, even if it is not the last layer of the network.

loss. There are numerous algorithms to choose from, however, each method works under the same principle: to adjust the weights and biases in an increasingly negative direction along the gradient of the loss function. The most fundamental optimization function is gradient descent and is the basis for many minimization algorithms. It is defined as

$$\theta_t = \theta_{t-1} - \eta \cdot \nabla_{\theta_t} L(\theta_t), \tag{2-19}$$

where θ represents the weights and biases in the ANN, η the learning rate and L the loss function. One drawback of this straightforward scheme is that selecting a proper learning rate, which controls the step size of the descent, is challenging. As a result, numerous alternative optimization algorithms have been developed, each having benefits and drawbacks of their own. Examples of other optimization algorithms are: Stochastic gradient descent, Momentum (Qian, 1999), Adagrad (Duchi et al., 2011), RMSprop and Adaptive Moment Estimation (Adam) (Kingma and Ba, 2014). For further reading, Ruder (2016) shows a comprehensive list of gradient descent algorithms.

As it is the most well-established and widely used, I opted to use the Adam optimizer throughout this thesis, which as the name suggests, is an adaptive-learning algorithm. It works by computing learning rates for each parameter and storing a weighted average of the past squared gradients v_t, as well as a weighted average of past gradients m_t, such that:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \tag{2-20}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \tag{2-21}$$

where v_t and m_t are estimates of the mean and uncentered variance of the gradients, respectively (Ruder, 2016). However, it has been observed that v_t and m_t are initially biased towards zero, especially at early time steps. Kingma and Ba (2014) propose bias-corrected mean and uncentered variance estimates, which are defined as:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \tag{2-22}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}. \tag{2-23}$$

Thereafter, (2-22) and (2-23) are used to to update the parameters by

$$\theta_{t+1} = \theta_t - \frac{\xi}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t, \tag{2-24}$$

where the learning rate ξ is typically chosen to be 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$.

2-2 Convolutional Neural Networks

While DNN refers to any ANN with many hidden layers, a CNN is a class of ANN that includes at least one convolutional layer in the network's architecture (Goodfellow et al., 2017). Such convolutional layers are powerful tools that are capable of significantly reducing the amount of memory required compared to a fully connected network, particularly when working on image-related problems. In the case of seismic data, an event's fluctuation in amplitude has a relationship with the spatial variation and thus possessing a tool that learns to interpret the spatial dependencies in the data is greatly beneficial. This section of the thesis discusses layers that are specific to CNNs.

2-2-1 (Transpose) Convolutional layer

Convolutional layers are the main building blocks of the CNN and in a broad sense, allow the convolution layer to generalize the features of an image by only considering sections of a given image at a time. The motivation behind this is that, typically, the information content of one region of an image provides little to no information about another distant region of the image, so it does not make sense for these opposing sections to influence the same neuron. The convolutional layer is able to separate the regions by using a filter, often called a kernel, that iterates from the top left to the bottom right of the input image. Here, the input image refers to the input into the convolutional layer, rather than the input into the CNN. Typically, the iterating kernel has a size of 3×3 (which is what is used in this thesis) or 5×5, but could theoretically have any dimension the user specifies. An example of the workings of a convolutional layer is showcased in Figure 2-1, where it can be seen that a dot product of the kernel and the considered section of the input image is performed, which is then filled into the corresponding element of the feature map (output) until all elements of the output have been computed. Furthermore, each element in the kernel is a trainable parameter, which means that it is tuned during training to optimize the network's performance.

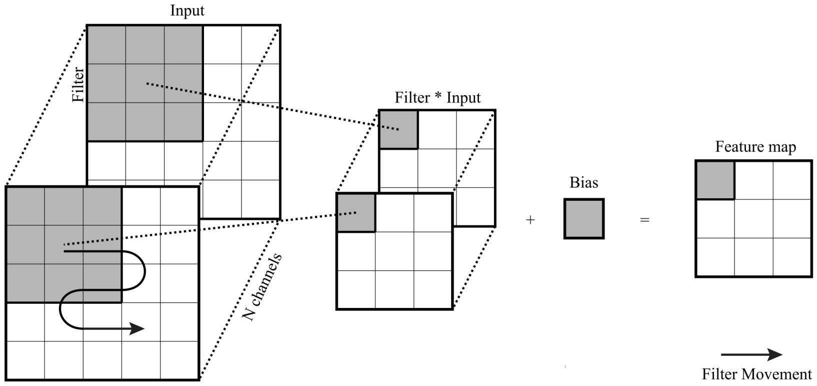


Figure 2-1: Visualization of a convolutional layer operation (Slang, 2019).

There are two other components in the convolution operation that need to be considered, namely the stride and the padding. The stride refers to the number of pixel values the kernel is moved along in each step. The padding refers to the number of zero cells that are added around the input image prior to performing the convolution operation and has been indicated in grey in Figure 2-1. Together, the kernel size, stride and padding determine the dimensions of the convolutional layer's output by $m = \frac{n+2p+k}{s} + 1$, where n is the dimension in the input, p is the padding, k is the dimension of the kernel, s is the stride and m is the dimension of the output.

The reverse operation of the convolutional layer is the transpose convolutional layer. The transpose convolution will multiply each given element of the input by the elements in the kernel to produce an up-sampled version of the input. Naturally, like the convolution kernel, the parameters of the transpose convolution layer are trainable.

Lastly, it is possible for the input and output to consist of multiple channels. With multiple channels, it is possible to further incorporate an additional degree of information. For example, one could include particular transforms of the data. Regardless, when multiple channels are present in the data, the kernel will need to be adjusted accordingly by matching the number of channels in the kernel to the number of channels in the input. Naturally, the elements of all channels of the kernel are trainable parameters.

2-2-2 Pooling layer

Pooling layers reduce the dimension of feature maps, which in turn reduces the number of parameters in the CNN and reduces training time. Additionally, by implementing a pooling layer the number of neurons and connections are reduced, which reduces training time. Furthermore, through a simplified feature map, the CNN also becomes more robust to small spatial variabilities, which in turn also reduces over-fitting Yu et al. (2014). The three most common types of pooling layers are minimum pooling, average pooling and maximum pooling. The latter of which has been employed in this thesis. Furthermore, like the convolutional layer, the pooling layer samples sections of the input and reduces those to a single number by either taking the maximum, minimum or average within that section, depending on the type of pooling layer used. However, unlike the convolutional layer, a pooling layer's parameters are not trainable and thus the exact same operation is performed in every loop.

2-2-3 U-Net

The CNN architecture used in this thesis is a U-Net. Originally, the U-Net was published as a network for biomedical image segmentation (Ronneberger et al., 2015), however, it has since shown good results in previous works relating to seismic data. For example, Fang et al. (2021) show an implementation to interpolating missing traces, Sun et al. (2020) use a U-Net to denoise a seismic image and Li et al. (2021b) show a U-Net based seismic multiple removal scheme. The main benefit of such a U-Net over a standard CNN, is that it has a reduced number of parameters making it more efficient to train while requiring less memory, without any loss in prediction accuracy (Sun, 2022).

Figure 2-2 shows the U-Net employed for this thesis. Here, it can be seen that the U-Net consists of several encoding and decoding blocks, each containing two convolutional layers and a max pooling/transpose convolutional layer. These blocks are also connected through 'skip' connects (yellow in Figure 2-2) that take the feature maps from the encoder blocks and concatenate them with the output of the decoder blocks. It is believed that through the encoding-decoding route the network is able to learn the structure of the seismic data, whilst the skip connection allows the network to preserve the high-frequency detail (Hlebnikov, 2022). The numbers above each layer represent the number of channels in that layer.

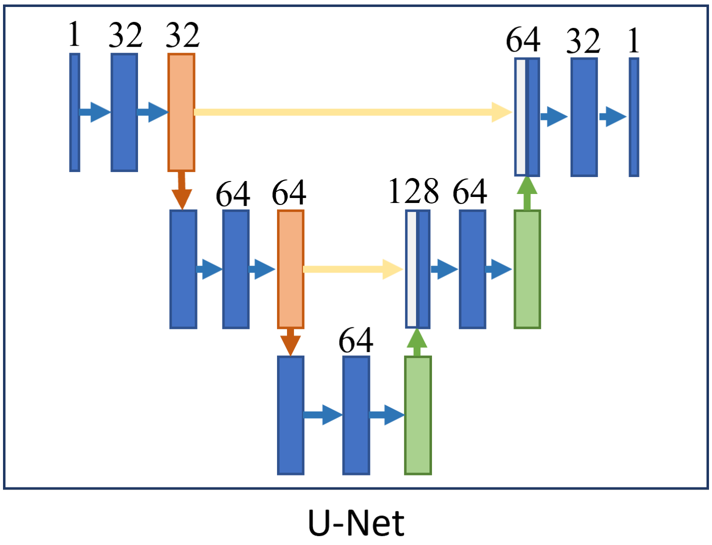


Figure 2-2: U-Net is used for frequency interpolation. The network consists of two encoding-decoding block pairs that make up the characteristic U-shape.

2-3 Preparing the data and Training the network

The main drawback of an ANN is the need for training data from which the network is able learn how to perform the given task. For our purposes, the CNN is trained on a North Sea airgun dataset. Ideally, we wish for the network to estimate the full frequency bandwidth of a gather when given a shot gather where certain frequency bands are missing. The training data is generated by taking several shots and removing the same frequency bands from each shot. This is considered to be the input for the network. Furthermore, the network is also given a 'label' or target gather, which is the same gather, but unfiltered. By giving the network a lot[2] of these training samples and by following the above-described training algorithm, the network is able to learn the specific task.

During the training, a certain percentage of the training data is used as a validation dataset, typically 20%, which is data that the CNN has not been trained on. This is used to quantify how well the network is able to generalize what it has learned and to avoid over-fitting to the training dataset. Furthermore, a network is trained for a certain number of epochs, which is the number of times the network trains on the entire training dataset. Although there is no correct number of epochs that a network should be trained for, one typically trains until no further improvement in the network's performance is observed.

In the case of the network employed in this thesis, due to the limited availability of RAM, the network is trained the network on 80 gathers, which will be shown to correspond to 4860 samples, for 20 epochs.

[2] It is preferable to give the network as much training data as possible, for the network to perform best.

2-3-1 Windowing

Naturally, prior to feeding the data into the CNN, the training data must be prepared. For example, an entire shot gather is typically too large to be fed into the CNN at once. As a result, the gather either needs to be cropped, or more computational memory needs to be made available. In this thesis, I have opted to 'window' each shot gather so that the input size into the network is 512 × 64 samples. Each window can then be combined into a single shot gather after the frequency spectrum has been interpolated for each window individually. An illustration showcasing the windowing algorithm can be seen in Figure 2-3. Here, the number below each segment corresponds to the window number, where n is the last window of each line. Notice that each window consists of two parts: an inner and outer window, where all the outer windows have a 25% overlap with the neighbouring windows. The motivation behind using two windows, rather than simply cropping the data, is to avoid edge effects that might occur during interpolation. Following interpolation, the inner windows are combined to reconstruct the original shot gather.

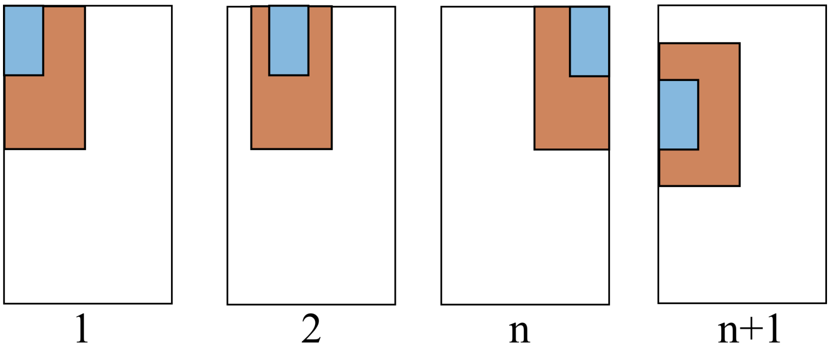


Figure 2-3: Illustration of the windowing algorithm used to reduce the required memory of the neural network.

It should be noted that some windows purely contain noise, particularly the windows prior to the first arrival. These have been removed from the training dataset, as it was found that without these noisy training samples, the network is able to learn the dataset in fewer epochs. In summary, each gather is split into 80 training windows/samples, so in total for 80 gathers there are 6400 training samples. However, as we remove those with insufficient Signal-to-Noise Ratio (SNR) there are only 4860 training samples in total.

Chapter 3

Sparse Optimization Approach

The second method used to interpolate the missing frequency bands is a scheme related to compressed sensing and convex optimization. The underlying optimization approach was first suggested by Taylor et al. (1979), however, the first application to seismic data was presented by Wang and Herrmann (2016), whereby a comparison is shown between the originally presented scheme, as well as a modified version that accounts for spatial similarity to extrapolate low-frequency data. Although having knowledge of the low frequencies is important for performing modern processing techniques, e.g. stabilizing Full Waveform Inversion (FWI) and avoiding cycle skipping, this goes beyond the scope of the thesis and I will restrict myself to the interpolation of arbitrary missing frequency bands. Moreover, I will investigate the possibility of using the presented sparse optimization approach as a scheme for interpolating these missing frequency bands and later compare it to the previously discussed CNN approach. The structure of this chapter is as follows: first, the slant-stack transform will be introduced for later use. Then, the sparse solver will be presented in a general form and a reconstruction of a single trace will be showcased. Thereafter, it will be shown that with the additional slant stack transform, spatial continuity of the reconstruction can be reassured.

3-1 Slant stack transform

It is possible to describe a gather in the *tau*-p domain using a slant stack/*tau*-p transform. The benefit of using this domain is that the linear events are collapsed into points, whilst hyperbolic events become ellipsoidal, making them very effective at removing certain types of noise in the data. In particular, slant stack transforms are very powerful tools and have become a standard part of a processing flow, for example when removing seismic interference noise generated by nearby seismic surveys. In this work, however, I take advantage of the transform's ability to spread the measured energy over multiple p values and enhance the trace-to-trace continuity of a 2D reconstruction. However, conceptually, an alternative 2D or potentially 3D transform could have been applied, but τ-p was chosen because it was easily available and well understood in a seismic setting.

In its essence, a slant stack transform is performed by summing up the energy of the gather at various slanted paths. As explained by Yilmaz (2001), the data is Linear Moveout (LMO) corrected using a coordinate transformation (Claerbout, 1978) defined as:

$$\tau = t - px, \tag{3-1}$$

where t is the two-way traveltime, p is the ray parameter, x is the offset, and τ is the intercept time at $x = 0$. Thereafter, the energy is summed along the offset axis by:

$$S(p, \tau) = \sum_x P(x, \tau + px), \tag{3-2}$$

where $S(p, \tau)$ is a plane wave with ray parameter $p = sin\frac{\theta}{v}$. It should be noted that a slant-stack transform is not a true plane-wave decomposition, but such ideas/processes can be made more precise by considering linear Radon transforms and true 3D sampling. This process is then repeated for a number of p values until all dip components within the desired range have been computed and a complete slant stack gather is obtained.

The reverse process, or inverse slant stack transform is obtained by rewriting (3-1) and (3-2) to obtain (3-3) and (3-4), respectively:

$$t = \tau + px, \tag{3-3}$$

$$P(x, t) = \sum_p S(p, t - px). \tag{3-4}$$

To illustrate the procedure, Figure 3-1 shows three events in the time domain and the equivalent event in the τ-p domain. Here it is illustrated how the energies of the events are summed, according to (3-2). First, consider the horizontal line representing p = 0, which corresponds to a single point in the τ-p domain. The hyperbolic event will intersect with several slopes, which will result in the energy being spread over several p values. Finally, the upper-most linear event again corresponds to a single point, as all the energy along it can be summed up with along one plane wave. Note that in practice, the τ-p transform is often computed through a least-squares formulation (Foster and Mosher, 1992).

3-2 1D Sparse Interpolation

Wang and Herrmann (2016) demonstrate the ℓ1-minimization scheme as follows. Assuming a situation with no dispersion or attenuation, it can be said that a single trace d_j can be written as

$$d_j(t) = q(t) * G_j(t), \quad j = 1, ..., n \tag{3-5}$$

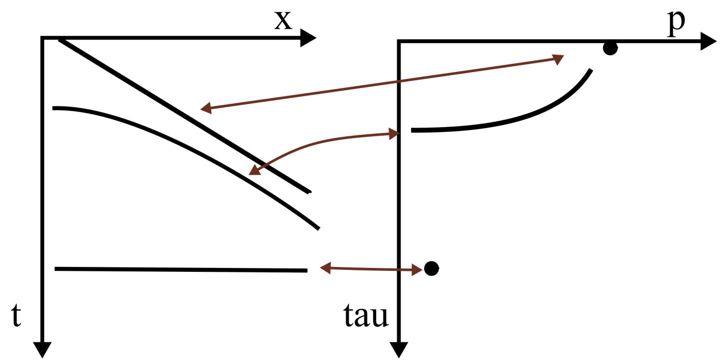


Figure 3-1: Examples of slant stack/τ-p transformations of simple events. Left shows the gather in the time domain and right shows the gather in the τ-p domain.

$$G_j(t) = \sum_{i=1}^{1} a_i^j \chi(t - t_i^j), \quad i = 1, ..., m \tag{3-6}$$

where $q(t)$ is the time signature of the source wavelet, $G_j(t)$ is the reflectivity series at the j-th receiver, n is the number of receivers, $\chi(t - t_i)$ is a spike corresponding to a seismic event at time t_i and a_i^j is the reflection coefficient, or in other words each spike's amplitude. Furthermore, $q(t)$, $G_j(t)$ and d_j are assumed to be discretised, and are thus vectors in $\mathbf{R}^N$.

Moreover, we can define frequency regions Ω_f that contain all frequencies except for the missing frequency band(s). The frequencies for which information must be interpolated can be written as Ω_b. Then, $\Omega_f + \Omega_b = \Omega$ represents the full spectrum. By taking the temporal Fourier transform, we can rewrite (3-5) in the frequency domain as

$$\hat{d}_j(\omega) = \hat{q}(\omega)\hat{G}_j(\omega), \ \omega \in \Omega_f. \tag{3-7}$$

Next, with d_j and q it is possible to estimate the reflection series using an $\ell 1$-minimization scheme:

$$G_j^{est} = \arg\min_G ||G||_1 \text{ subject to } \hat{G}(\omega) = \frac{\hat{d}_j(\omega)}{\hat{q}(\omega)}, \ \omega \in \Omega_f, \tag{3-8}$$

where G_j^{est} is the estimated reflection series. Using the estimated G_j^{est}, we can find $\hat{d}_j^{est}(\Omega)$ by

$$\hat{d}_j^{est}(\omega) = \hat{q}(\omega)\hat{G}_j^{est}(\omega), \ \omega \in \Omega. \tag{3-9}$$

It is preferable to add the reconstructed signal in the frequency region Ω_b to the original signal in the frequency region Ω_f rather than using $\hat{d}_j^{est}$ in expression (3-9) as our final estimate. This is done to avoid any unwanted side-effects in $\hat{d}_j^{est}$ inside the frequency band Ω_f. For this, we can apply a zero-phase bandpass filter $W_b(\omega)$ that is ideally 1 inside Ω_b and 0 outside, although, in reality, a filter will have some roll-off. Then we have:

$$\hat{d}_j^{recon}(\omega) = \hat{d}_j(\omega) + \hat{d}_j^{est}(\omega)W_b(\omega) \tag{3-10}$$

In a test example, we can start with a seismic gather with no frequency band gap and as a benchmark filter it with a stopband filter $1 - W_b(\omega)$. Additionally, we add noise to the frequencies where the stopband is located and then use the bandpass zero phase filter $W_b(\omega)$ in our reconstruction and reconstruct the filtered frequency band. This would allow us to compare the reconstructed signal with the benchmark. Clearly $1 - W_b(\omega) + W_b(\omega) = 1$, meaning that we are trying to preserve the amplitudes in our benchmark test. In the benchmark test we will have:

$$\hat{d}_j^{recon}(\omega) = (\hat{d}_j(\omega) + n(\omega))(1 - W_b(\omega)) + \hat{d}_j^{est}(\omega)W_b(\omega). \tag{3-11}$$

The standard deviation of the added noise $n(\omega)$, and its type, can be discussed as a parameter. One method to apply a zero-phase filter is to apply a nonzero-phase filter using forward-backward filtering. MATLAB function 'filtfilt' can be used for this purpose. In the latter, the phase of the filter is introduced in the forward filtering and then is subtracted in the backward filtering step, resulting in a zero-phase filter, but squared magnitude response.

As in compressed sensing, one of the driving assumptions is that the data can be represented sparsely, which in this case is accomplished by means of the reflectivity G_j. Another assumption is that the reflectivity at each frequency is similar, i.e. the interface in the subsurface reflects all seismic frequencies of interest. This likely holds for most earthly geology.

Furthermore, an intuitive interpretation of the procedure is that by knowing the reflectivity series, the system is able to 'add' the required energy at the correct times, as opposed to spreading the energy randomly. This in turn preserves the phase, as it is related to time by $\frac{\omega}{2f} = T$ relative to every reflection. This is how the scheme is able to estimate the phase of the missing frequency bands. In speech recognition, data is divided into smaller time windows using a similar approach inside each time window (Abdelmalek et al., 2022). This is interesting to consider since speech data consists of single tones and their harmonics in small time windows, which is more similar to uncorrelated marine vibrator data than broadband airgun data.

It should be mentioned that one of the main drawbacks of this scheme is that there is a potentially reduced resolution. For example, two spikes that are located closer than the smallest wavelength will not be resolved if they have opposing polarity. Fortunately, this issue is local to where two spikes are too close (Donoho et al., 1992; Wang and Herrmann, 2016).

In principle, the above $\ell 1$-minimization scheme can be used successively along each trace in a gather to interpolate the missing frequency content of the whole gather. However, in this case the reconstruction along each trace will lack information from neighbouring traces and as a consequence, the final interpolated gather can/may show reduced trace-to-trace continuity. In practice, the arrivals will not always smoothly align as they are normally expected to behave in a shot-gather.

3-3 2D Sparse Interpolation

As mentioned, attempting to interpolate the missing frequency content of each individual trace of a shot-gather separately may result in an image with a reduced horizontal continuity.

Therefore, a method that considers multiple traces is necessary. Wang and Herrmann (2016) add trace continuity through a minimization scheme that takes spatial similarity into account. Although this scheme has shown to accurately approximate low frequencies when working on extrapolation problems on a simple dataset, I opted to use an alternative approach, inspired by conventional seismic processing, to account for trace continuity: by implementing a τ-p (slant stack) transform prior to interpolating the missing frequencies. In this manner, the spatial information will be spread over multiple p values, thus causing the traces to be spatially dependent and possibly more continuous on inverse transform. Another benefit of the τ-p transform is that it does not alter the frequency content of the data, however, it does assume that each trace is missing the same frequency data. Figure 3-2 shows the frequency interpolation workflow of a shot-gather with the τ-p transform.

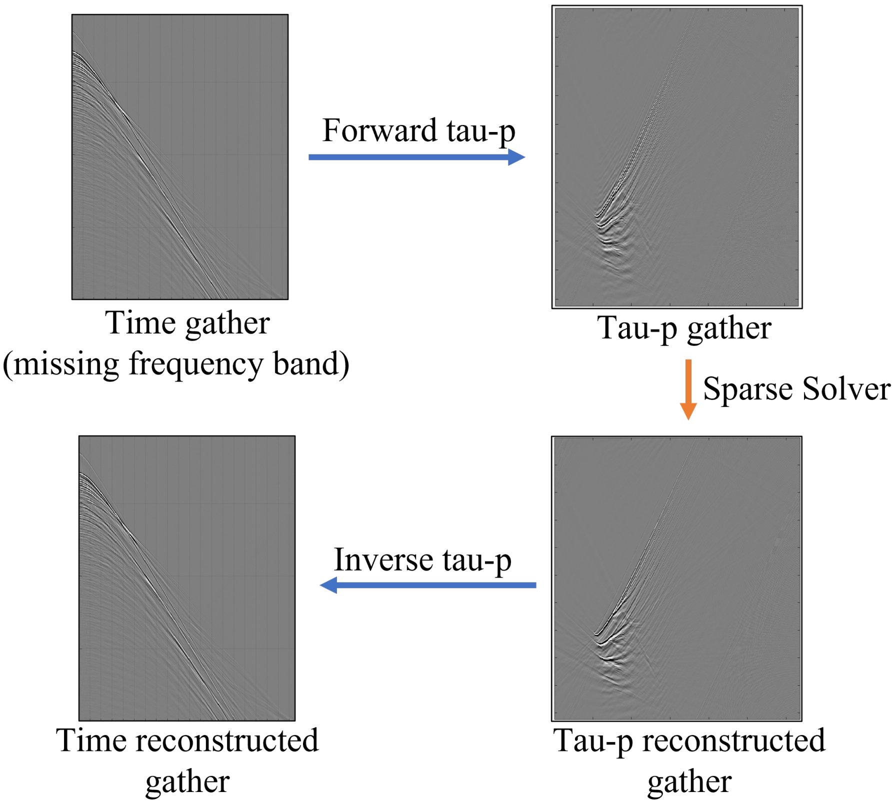


Figure 3-2: Workflow of the 2D frequency interpolation. Initially, the shot-gather is transformed into the τ-p domain, where the missing frequency bands are interpolated for each p value. Once the frequencies are interpolated, the shot-gather is transformed back into the time domain and the interpolation procedure is completed.

Once the input shot-gather with missing frequency content is transformed into the τ-p domain, we successively interpolate the missing frequency content for each p value, similar to the interpolation procedure described in Section 3-2. Thereafter the interpolated data is transformed back into the time domain.

Chapter 4

Results

This section of the thesis presents both the CNN and sparse solver interpolation results for various missing frequency bands. Naturally, we wish for the interpolated band(s) to closely match the original signal (target) in that/those band(s). However, as perfectly predicting the energy in the band is practically very difficult (if not impossible), there will be an error that must be quantified. For this, I consider several different metrics to quantify the error such as: relative difference/error in amplitude, average difference in phase and Normalized Root Mean Squared (NRMS). The relative difference in amplitude spectrum is computed by $|\frac{\tilde{D}_{target}-\tilde{D}_{recon}}{\tilde{D}_{target}}|$, where $\tilde{D}_{target}$ and $\tilde{D}_{recon}$ are the FK transforms of the target and reconstructed gathers. Furthermore, the NRMS is chosen as it is a well-established error measure in a 4D seismic setting. For example, in marine seismics, different surveys will encounter changes in the surroundings, such as a shift in temperature, salinity, water flow, as well as changes in receiver/source locations, etc. Depending on how severe the changes are, the seismic data can be (un)suitable for a time-lapse. The NRMS is the most commonly used measure to quantify whether these changes are within acceptable boundaries (Kazemi et al., 2011). The NRMS is formulated as:

$$NMRS = 2\frac{RMS(A-B)}{RMS(A)+RMS(B)}, \tag{4-1}$$

where RMS is the root mean square operator, A and B are the target and reconstructed gathers, respectively. The output of NRMS will be between 0 and 2, where 0 for our purposes means a perfect reconstruction. In this thesis, the NRMS has been used to quantify the reconstruction accuracy in both the time domain as well as the frequency domain.

Two other commonly used measures are the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM). The PSNR is a measure to quantify the image quality and although it is typically used to compare a compressed image to the original, it is also used in seismic trace interpolation. PSNR is expressed in dB, where a higher dB is more favourable. The SSIM is also used to determine the loss due to compression and is a measure that quantifies how similar one image is to another. The output is a number on a scale from 0 to 1, where 1 is a perfect match.

For the sake of clarity, the same shot-gather (Figure 4-1) is used to illustrate the effects of the missing band gaps and different interpolations. As is usual with North Sea data, most shots are fairly similar due to the very flat subsurface bedding and thus using a single shot is deemed representative of the data set as a whole. Since the interest of this research is the interpolation of frequencies specifically, Figure 4-1 also includes the FK transform of the target gather, which can be used to compare the reconstructions in the remainder of the chapter.

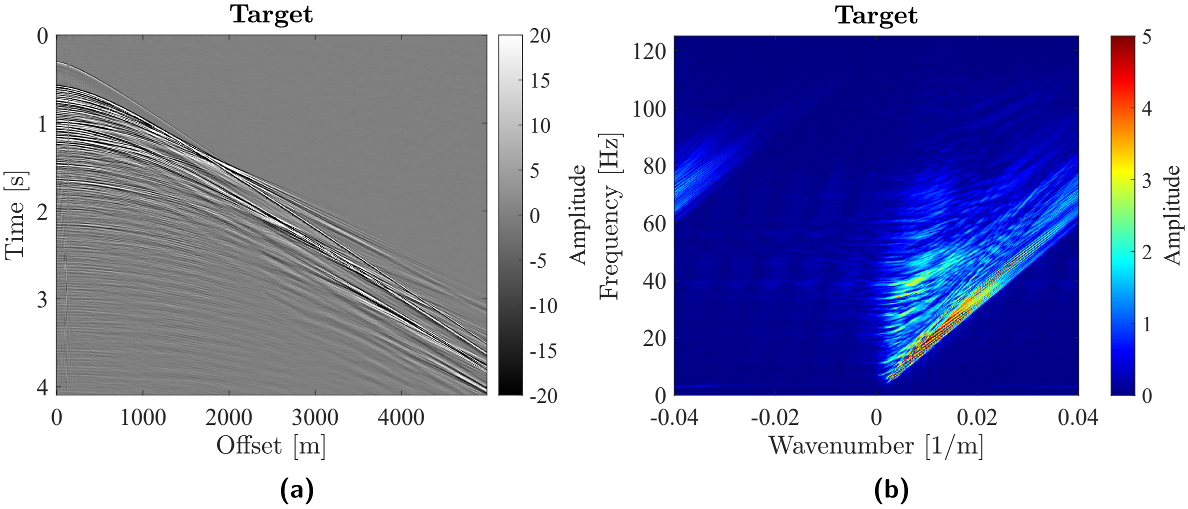


Figure 4-1: Target shot-gather for all further interpolation results in the time-domain **(a)** and FK domain **(b)**.

The input/test data that each method is tasked to reconstruct is generated by filtering the target gather. For this, the well-known Butterworth filter is used with an order $n = 7$. Furthermore, each result presented below contains the same added random noise for a fair comparison and naturally, the CNN is trained on data with different noise, so that it does not learn the behaviour of the noise. To illustrate the filtering effect, it is helpful to show the Fourier transform of a single trace, together with the Fourier transform of the same trace that has been filtered. This is shown in Figure 4-2 where amplitude and phase spectrum of a trace located at 1250 m from the source can be seen. Here, the target is plotted in black, together with the same trace that has been band-pass filtered in orange.

The results have been divided into several sections. The first section looks into the reconstruction of a single frequency band, whereby a visual comparison between the different methods is performed and the frequency dependency of the reconstructions is quantified. The second section compares the reconstruction results of two missing frequency bands. The third section presents the reconstruction of six missing frequency bands, as a more realistic scenario and a stress test for each method. In the fourth section, the results from two additional interpolation schemes are shown. The first is a modification to the neural network whereby the input data consists of 3 channels to provide the network with more spatial information. The second is the τ-p transformation described in Section 3-3.

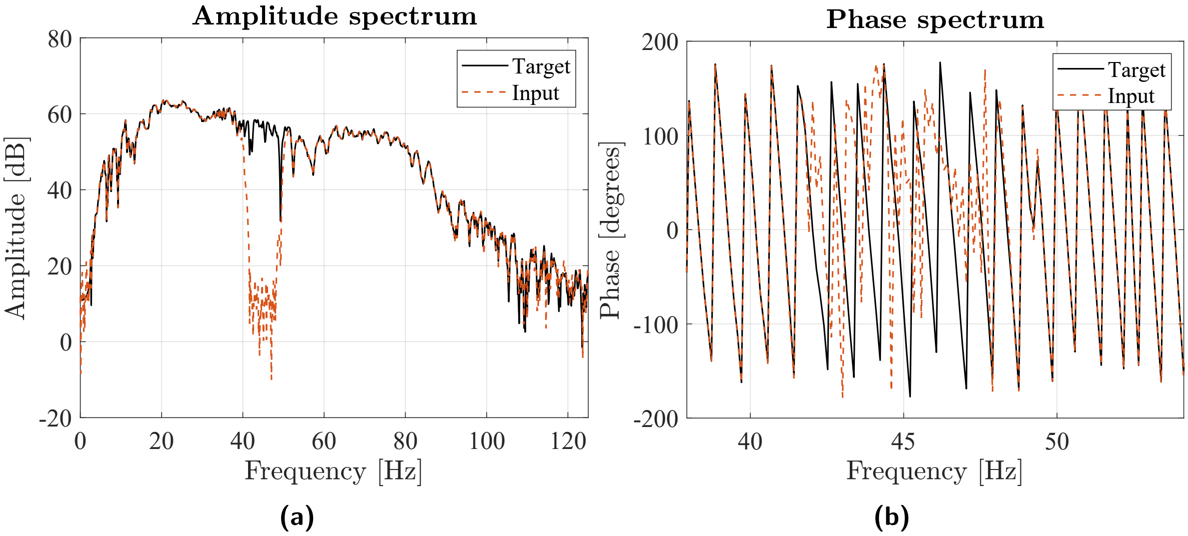
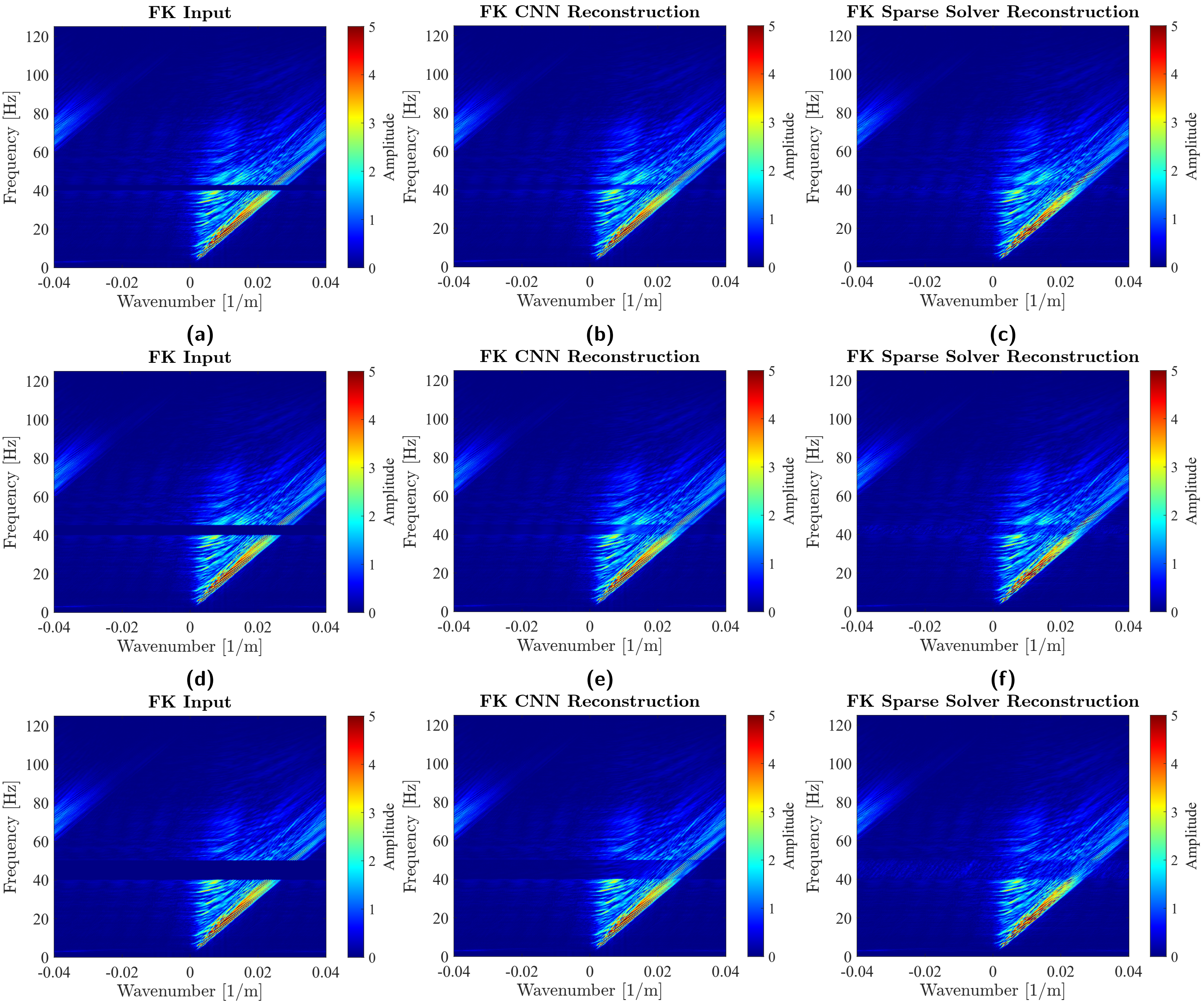


Figure 4-2: Amplitude **(a)** and phase **(b)** spectrum of a trace located at an offset of 1250 m from the source, to illustrate the missing frequency band. Notice that **(b)** is only plotted for frequencies 38-60 Hz.

4-1 One missing frequency band

We first consider a situation where a single frequency band must be interpolated, which may occur when merging two non-overlapping frequency bandwidths. Typically, one can expect such a gap to be fairly narrow, spanning for example 2.5 Hz. However, as this problem is not commonplace, certain applications could require the gaps to span more frequencies. In particular, I will consider the interpolation of band gaps spanning 2.5 Hz, 5 Hz and 10 Hz. Each gap is created using a Butterworth filter, whereupon 10% normally distributed noise is added, to dissuade the interpolation schemes from merely amplifying the dampened, but not completely zero, frequencies. Figure 4-3 shows the results where frequency bands have been filtered from the data, together with each method's corresponding interpolation result. As we are interested in the reconstruction of the frequency domain components, it is most straightforward to observe differences in the FK spectra. The first column in Figure 4-3 shows the input data, the second shows the CNN reconstruction and the third column shows the sparse solver reconstruction. Furthermore, it should also be mentioned that tests were performed where the frequency gaps were completely zeroed and similar results were achieved as those shown below. In the experiments below, the CNN was trained for 20 epochs with 80 shot-gathers, totalling 4640 training samples. The training curves of these networks can be referenced in Appendix A-1 of this thesis.

When comparing the two approaches in Figure 4-3, we can see that in all cases the reconstructions slightly underestimate the amplitude. We can also see that the CNN tends to produce an over-smoothed result, whereas the sparse solver's interpolation respects the behaviour of the spectrum. However, in regions of the FK spectrum with no signal, the sparse solver adds sharp, low amplitude noise, which is absent in the CNN's reconstruction. Furthermore, both methods appear to most accurately find the high amplitudes of the FK spectra. Visually, the best interpolation is the 2.5 Hz band gap produced by the sparse solver, which will also be confirmed by the statistics shorty.



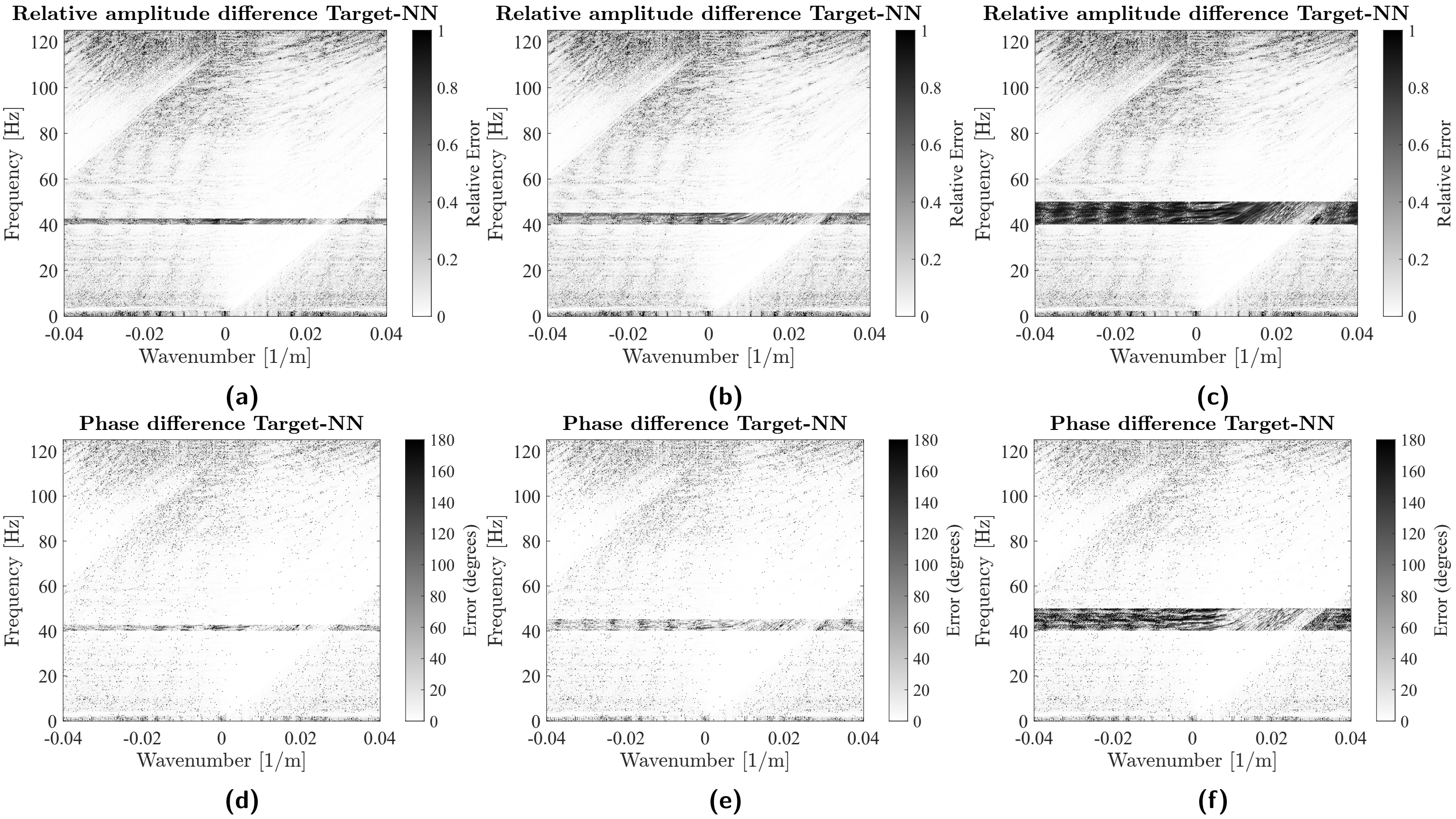
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4-3: FK transforms of the reconstructed gathers. The first column shows the FK plot of the input data where one band has been filtered. The second column shows the CNN reconstruction and the sparse solver reconstruction. From top to bottom 2.5 Hz, 5 Hz and 10 Hz missing frequency bands are considered, each of which starts at 40 Hz.

The next step is to examine the reconstruction of the phase, which is equally important as the estimation of the amplitude spectrum. However, due to the wrapped nature of the phase (Figure 4-2b), it is difficult to illustrate the phase error of the reconstruction in a meaningful manner. This is because the phase of the phase spectrum lacks structure and typically appears random. As a compromise, I have opted to show the absolute difference between the phase of the target gather and the phase of the reconstructed gather. This yields the phase error of the reconstruction in degrees. For each method, a total of six error plots are shown to illustrate the amplitude and phase error for each different frequency band gap. Figure 4-4 and Figure 4-5 show the error plots for the CNN and sparse solver, respectively. The first row of each figure shows the relative difference in amplitude spectrum and the second row is the difference in phase. The columns represent the three different filtered bands: 2.5 Hz, 5 Hz and 10 Hz.



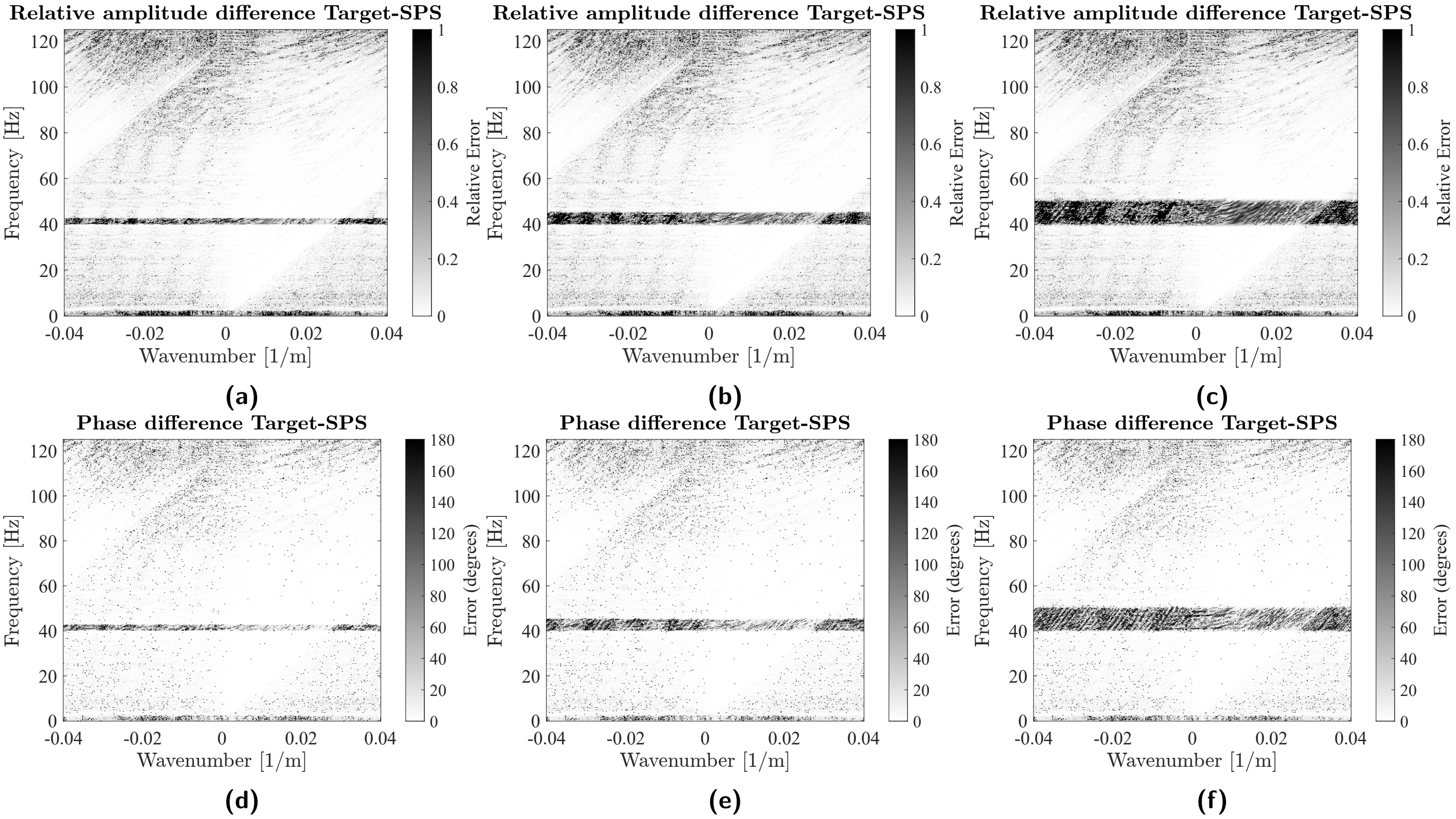
(a) (b) (c)

(d) (e) (f)

Figure 4-4: Error plots showing the relative error in amplitude and phase error of the CNN's reconstruction. These plots correspond to the reconstruction from Figure 4-3, where from left to right each band is 2.5 Hz, 5 Hz and 10 Hz wide.

Little difference is observable in the reconstructions of the 2.5 Hz and 5 Hz bands. For the 10 Hz gap, the error is significantly higher compared to the 2.5 Hz and 5 Hz missing bands. Similarly to what was observed in Figure 4-3, regions of the FK spectra with little to no signal are more erroneous in the sparse solver's interpolation than the CNN's interpolation.

Table 4-1 shows statistics corresponding to the reconstructions of Figure 4-3. Here, the average NRMS of the traces, the NRMS of the amplitude spectrum, the average phase error, PSNR



(a) (b) (c)

(d) (e) (f)

Figure 4-5: Error plots showing the relative error in amplitude and phase error of the sparse solver's reconstruction. These plots correspond to the reconstruction from Figure 4-3, where from left to right each band is 2.5 Hz, 5 Hz and 10 Hz wide.

and SSIM are shown. In general we observe that the NRMS and phase error increase for wider band gaps and that the PSNR and SSIM decrease for wider band gaps. Furthermore, we observe that the results are fairly similar for both interpolation methods. It can also be confirmed that the 2.5 Hz sparse solver reconstruction is the most similar to the target of the reconstructions shown in Figure 4-3, based on the PSNR and SSIM.

Finally, to observe the frequency dependency of the reconstruction schemes, frequency bands are sequentially filtered and reconstructed to gauge whether certain frequency bands result in a more accurate reconstruction. Figure 4-6 shows a bar graph of the NRMS of each interpolated band gap. In this figure, for example, the 25-30 Hz bin corresponds to the interpolation error of a 25-30 Hz band gap in the spectrum. The first bar graph shows the average NRMS of the traces in the time domain, the second graph shows the average NRMS of the amplitude spectrum of each trace and the third plot shows the error in phase. In each plot, the CNN results have been plotted in blue and the sparse solver in brown.

In Figure 4-6 we can see that for almost all 5 Hz frequency gaps, the sparse solver's approximation is more accurate than the reconstruction of the CNN. We can also see, that although it is faint, the reconstruction error decreases with increasing frequency, especially for the sparse solver. The CNN, initially increases in error until around NRMS gives similar results for time and amplitude.

Table 4-1: Table showing the accuracy of the reconstructions of both methods when reconstructing three different band gaps.

	CNN 2.5 Hz	CNN 5 Hz	CNN 10 Hz	Sparse Solver 2.5 Hz	Sparse Solver 5 Hz	Sparse Solver 10 Hz
NRMS Time	0.0753	0.1005	0.2148	0.0789	0.1381	0.2816
NRMS Frequency	0.0583	0.0751	0.1691	0.0556	0.0971	0.1966
Average Phase Error (Degrees)	21.0631	22.0279	33.744	25.3978	32.170300	46.5060
PSNR (dB)	57.4427	55.3932	48.5277	58.0281	53.3782	47.9270
SSIM	0.9988	0.9983	0.9947	0.9989	0.9973	0.9935

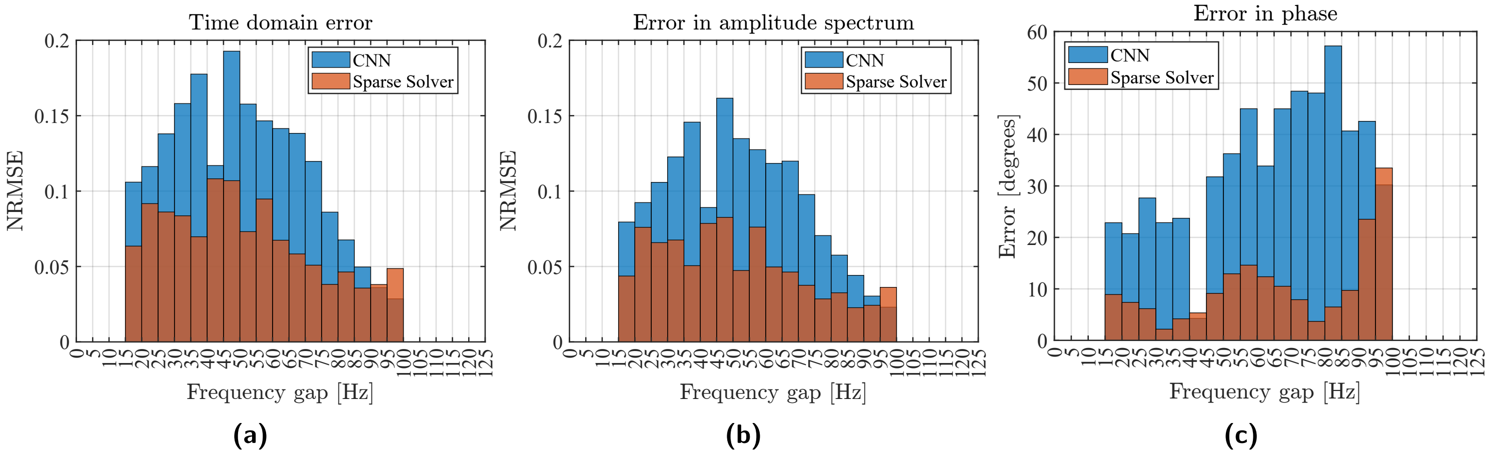


Figure 4-6: Bar plots showing the average NRMS for each trace **(a)**, average NRMS **(b)** and average error phase **(c)** of the frequency band interpolation for various 5 Hz gaps. Here the CNN approach has been shown in blue and has been trained for 20 epochs. The sparse solver has been shown in brown. For a fair comparison, the same normally distributed noise is added for all interpolations, which corresponds to an additional 1% error.

4-2 Two missing frequency bands

In the previous section, we investigated how the width of a missing frequency band influences the reconstruction. Here, we analyse the influence of multiple missing frequency bands on the reconstruction. As was done previously, three different bandwidths of the gaps are considered: 2.5 Hz, 5 Hz and 10 Hz. Furthermore, as shown in Figure 4-7, the first gap starts at 20 Hz, the second at 40 Hz. The first column of Figure 4-7 shows the FK transform of the input data for the three cases, the second column shows the CNN reconstruction and the third column shows the sparse solver reconstruction. For each test, the network was trained for 20 epochs on 80 individual shots, of 4640 training samples.

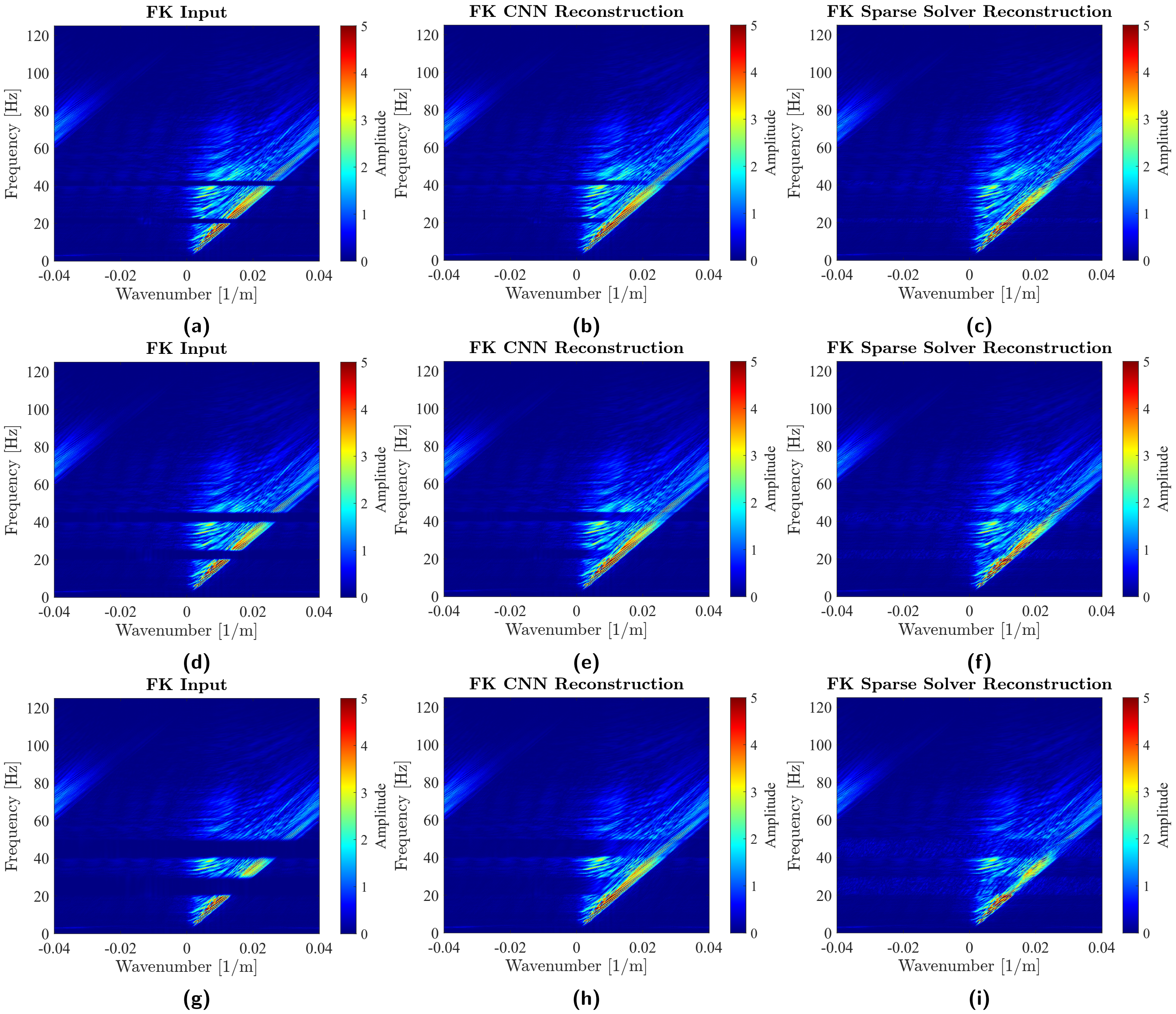


Figure 4-7: FK transforms of the reconstructed gathers. The first column shows the FK plot of the input data where two bands have been filtered. The second column shows the CNN reconstruction and the sparse solver reconstruction. From top to bottom a 2.5 Hz, 5 Hz and 10 Hz missing frequency bands are considered, each of which starts at 20 and 40 Hz.

Focusing on the second column of Figure 4-7, i.e. the CNN reconstruction, it is evident that much of the energy removed by the filter is not fully recovered after reconstruction, particularly in the low amplitude regions. For the 2.5 Hz gap (first row), little to no frequencies are recovered. In contrast, the high amplitude energy is successfully interpolated, even for the 10 Hz band gap. If we also consider the sparse solver reconstruction, we see that the estimated amplitudes bare a closer resemblance to the target data than the CNN approach, most notably when reconstructing the 2.5 Hz band gap. In the 5 Hz and 10 Hz band gaps however, the high amplitudes are reasonably well reconstructed. However, in the low amplitude regions, the reconstructed spectrum roughly has the same amplitude, but does not accurately estimate the structure of the target spectrum.

Figure 4-8 and Figure 4-9 show error plots of the CNN and sparse solver, respectively. The first row shows the relative errors in the amplitude spectra and the second row shows the error in the phase spectra. In Figure 4-8 specifically, we again clearly see that the large amplitudes are recovered in both amplitude and phase. However, low amplitudes are relatively less accurately reconstructed by the network but do find a close estimate of the phase. In Figure 4-9, we again see that the reconstructions of the high amplitudes are successful, while the low amplitudes in turn are relatively less accurate.

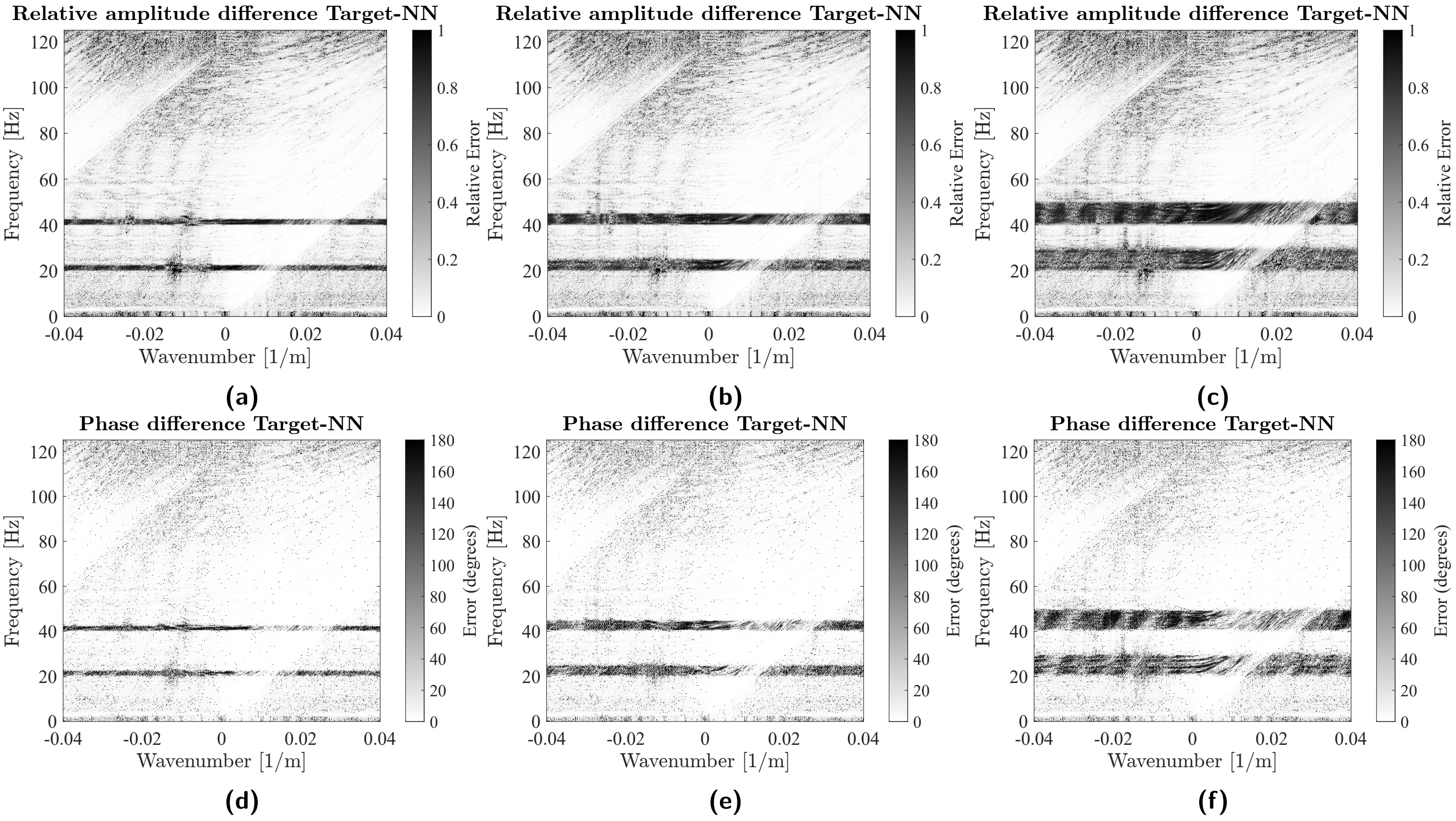


Figure 4-8: Error plots showing the relative error in amplitude and phase error of the CNN's reconstruction. These plots correspond to the reconstruction from Figure 4-7, where from left to right each band is 2.5 Hz, 5 Hz and 10 Hz wide.

Table 4-2 shows various interpolation statistics to quantify the accuracy of the reconstruction.

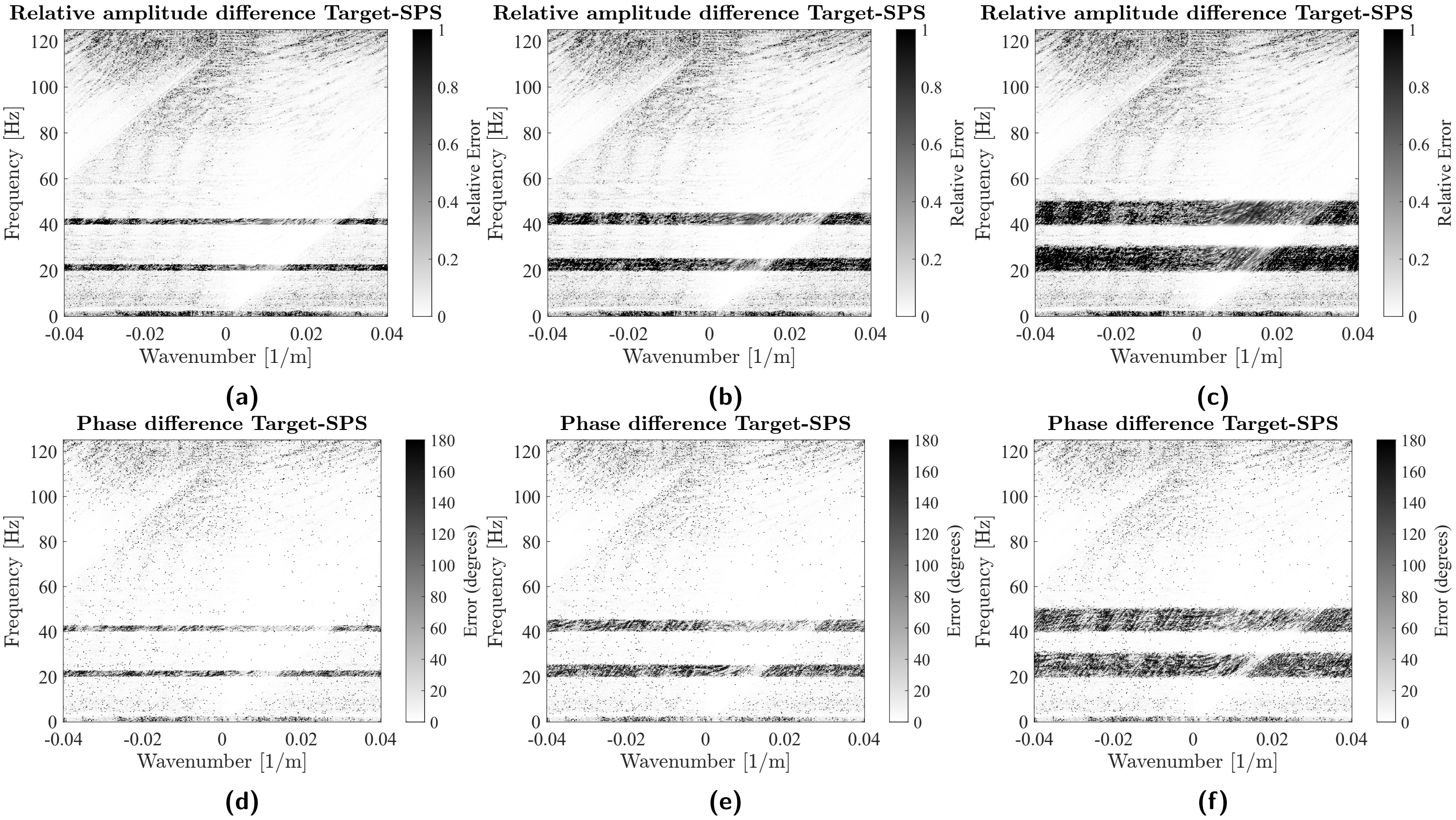


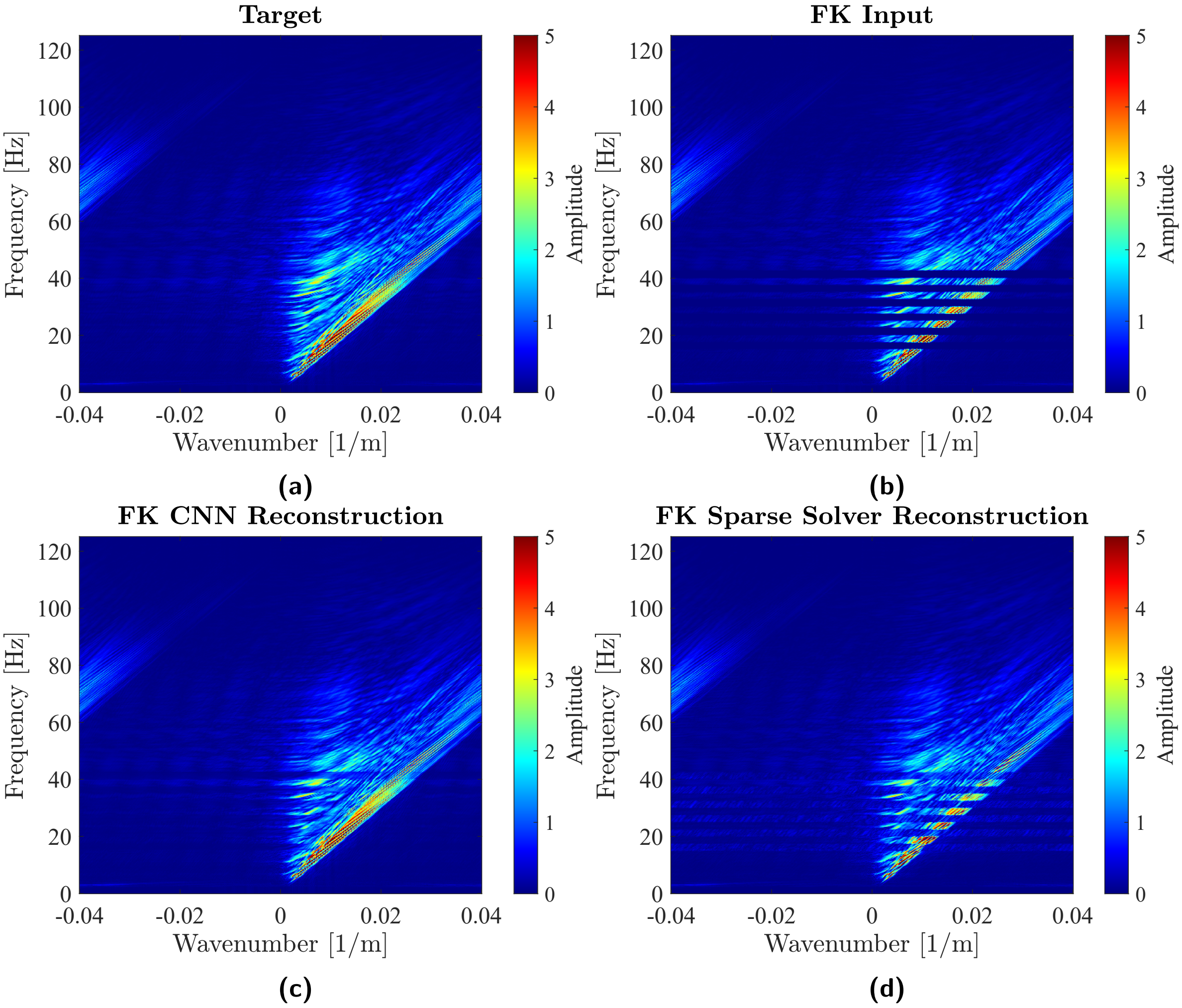
Figure 4-9: Error plots showing the relative error in amplitude and phase error of the sparse solver's reconstruction. These plots correspond to the reconstruction from Figure 4-7, where from left to right each band is 2.5 Hz, 5 Hz and 10 Hz wide.

Here, it can be seen that overall, both methods are able to produce an accurate reconstruction for narrow bands, but become less accurate with larger missing frequency bands as was also previously observed. It can also be seen that for missing bands with the same width, a slight increase in error occurs when the number of bands increases. However, the increase is fairly small.

4-3 Six missing frequency bands

To illustrate a more realistic deblending scenario, six frequency bands that are 2.5 Hz wide are filtered from the data, where each band's centre frequency is spaced 5 Hz from the previous band's centre frequency. This scenario, together with the reconstructions is shown in Figure 4-10. The first and last missing bands span 15-17.5 Hz and 40-42.5 Hz, respectively.

Unlike previous examples, however, the CNN was trained for 80 epochs rather than 20 epochs, which was the case in previous tests. This change is due to the problem at hand being more complex and needing more training to be solved accurately. The training curves for this network are shown in Appendix A-3. Furthermore, the relative amplitude error w.r.t.



(a)

(b)

(c)

(d)

Figure 4-10: FK spectra of the interpolation of six missing frequency bands. **(a)** shows the target gather, **(b)** shows the input gather with six missing frequency band, **(c)** shows the CNN reconstruction and **(d)** shows the sparse solver reconstruction.

the target gather and phase error are shown in Figure 4-11 and absolute error plots of the amplitude spectrum can be referred to in Appendix B-3.

From Figure 4-10 and Figure 4-11 we see that the reconstruction obtained through the network is more accurate in both amplitude and phase and it is able to recover the high amplitudes is most accurately. The sparse solver's reconstruction in contrast lacks much of the energy in all regions of the FK plots.

Table 4-3 shows a table with the various interpolation statistics to quantify the accuracy of the reconstruction. Here, it can be seen that the CNN is able to produce an accurate reconstruction, whereas the sparse solver does not.

Table 4-2: Table showing the accuracy of the reconstructions of both methods when reconstructing two missing frequency bands with varying width.

	CNN 2.5 Hz	CNN 5 Hz	CNN 10 Hz	Sparse Solver 2.5 Hz	Sparse Solver 5 Hz	Sparse Solver 10 Hz
NRMS Time	0.1277	0.1760	0.2911	0.1270	0.2211	0.4490
NRMS Frequency	0.1092	0.1417	0.2209	0.0927	0.1588	0.3238
Average Phase Error (Degrees)	20.4940	23.3963	28.6676	24.5458	32.4613	45.6851
PSNR (dB)	53.4121	50.6793	46.9006	54.9750	50.2043	44.6652
SSIM	0.9973	0.9959	0.9921	0.9980	0.9950	0.9867

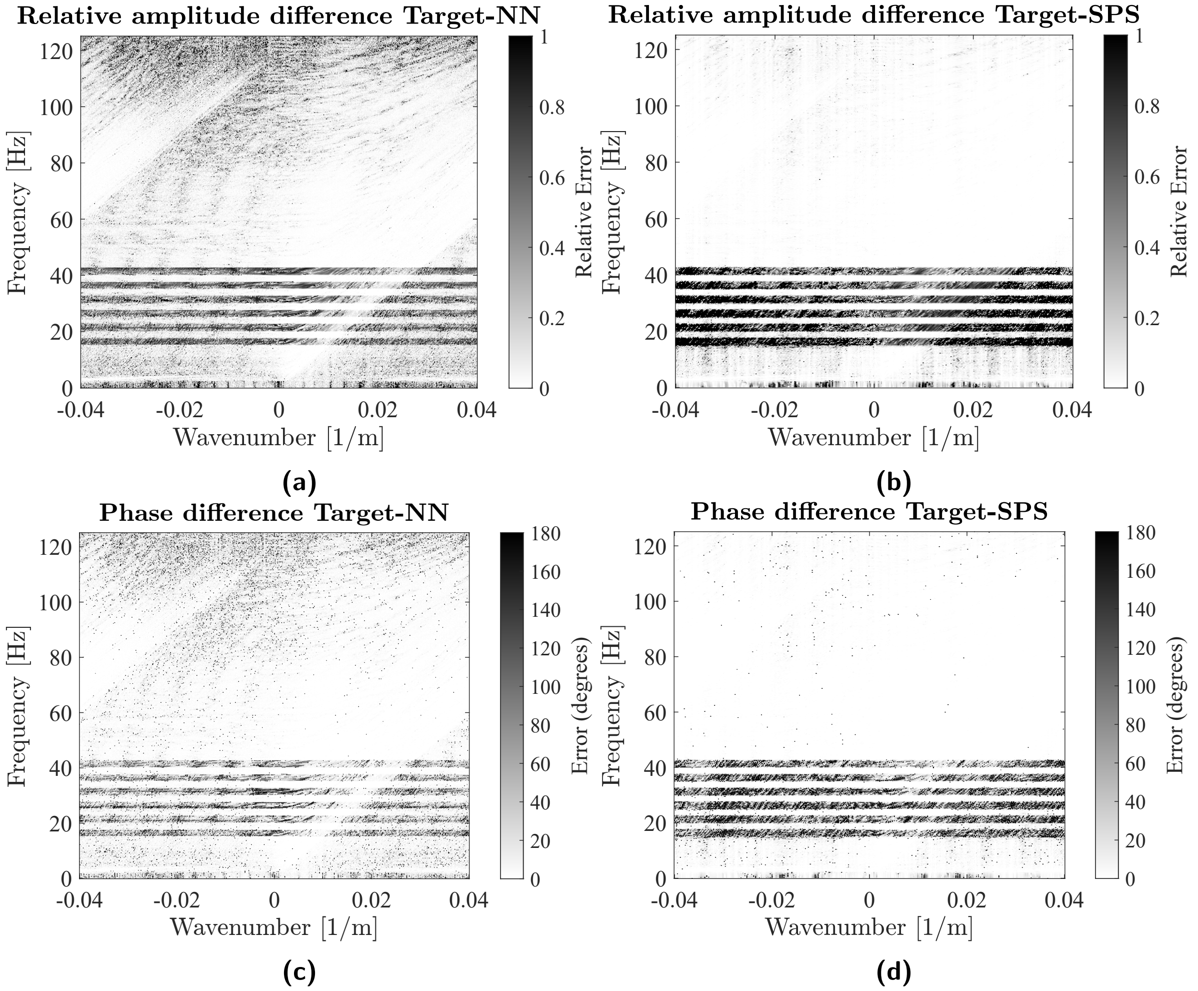


Figure 4-11: Error plots showing the relative error in amplitude and phase error of the sparse solver's reconstruction. These plots correspond to the reconstructions from Figure 4-10, where six 2.5 Hz frequency bands have been filtered and reconstructed.

Table 4-3: Table showing the accuracy of the reconstructions of both methods when reconstructing three different band gaps.

	CNN	Sparse Solver
NRMS Time	0.1722	0.4981
NRMS Frequency	0.1311	0.3683
Average Phase Error (Degrees)	17.7303	51.6092
PSNR (dB)	51.4819	44.1452
SSIM	0.9960	0.9832

4-4 $\tau - p$ Interpolation

In this section we will review the τ-p interpolation scheme that was considered in the thesis. The principle is that energy will be spread out over multiple p-values and introduce spatial dependency into the reconstruction. Figure 4-12 shows the interpolation of a single 5 Hz frequency band located at 40-45 Hz. The reconstruction shows that although the energy is recovered, a lot of the energy over the entire FK spectrum is amplified. Furthermore, in Table 4-4, we can see that there is a large error, as the NRMS is close to 2. Moreover, the PSNR is low and the SSIM is also low.

Table 4-4: Table showing the accuracy of the reconstruction using the sparse solver accompanied by a τ-p transform.

	τ-p Sparse Solver
NRMS Time	1.9993
NRMS Frequency	1.9992
Average Phase Error (Degrees)	56.5274
PSNR (dB)	33.4479
SSIM	0.9363

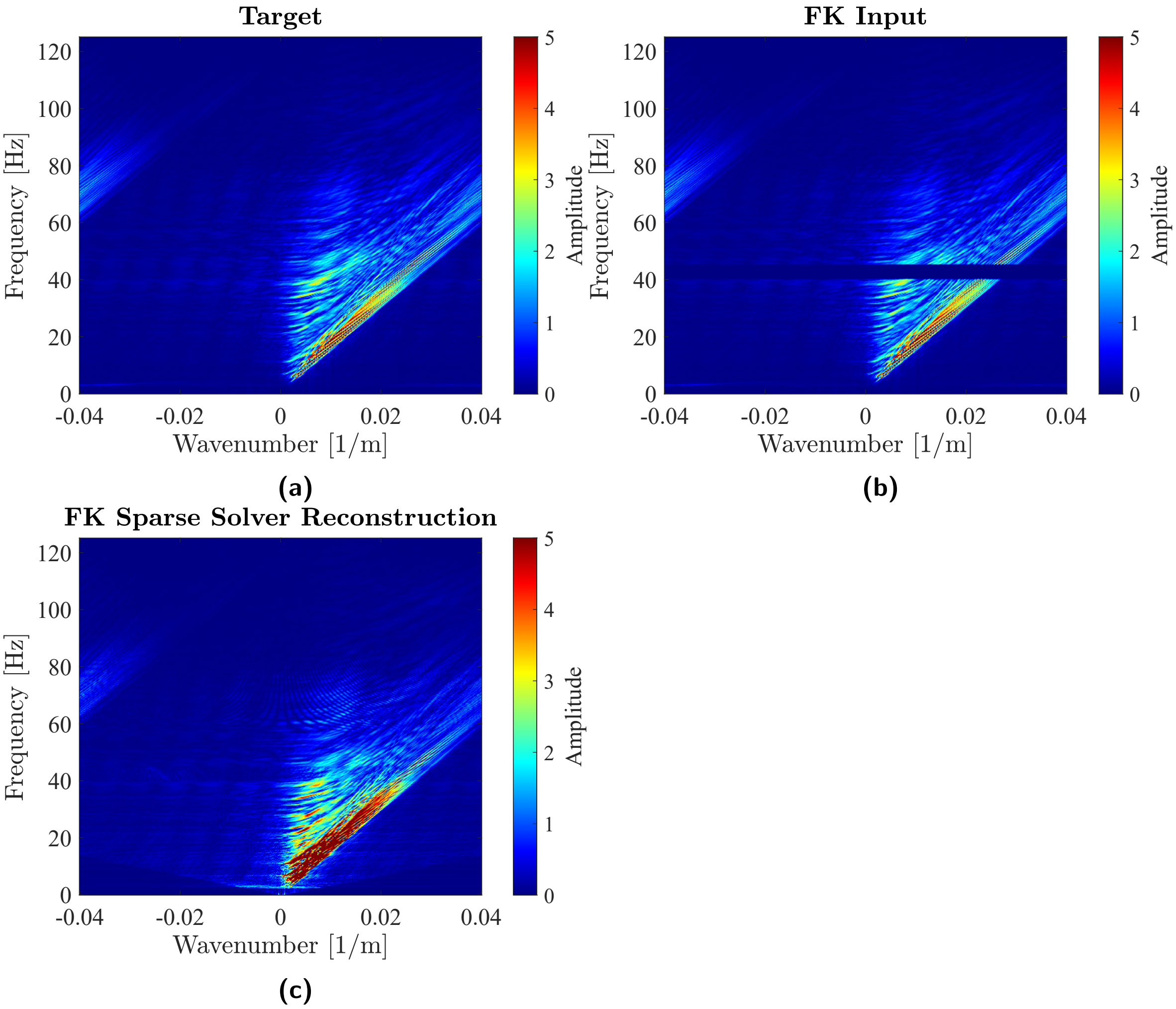


Figure 4-12: FK plots of the τ-p reconstruction. Notice that above 60 Hz, there is noticeable noise spreading across the data.

Chapter 5

Discussion

In this thesis, we have considered two methods for frequency band reconstruction. The first is a deep learning approach that uses a particular neural network architecture called a U-Net to learn how to interpolate frequency bands based on training data. The second method is a sparse optimization approach that approximates the reflectivity series of the subsurface such that the approximation matches the known frequency content. In this chapter, we will review the interpolation results of each method, starting with the deep learning approach and then considering the results of the sparse solver. Thereafter, we present a comparison of both methods, highlighting each method's strengths and shortcomings. The following section will then emphasize the limitations of this research, whereafter potential concepts for further research are suggested.

5-1 Deep Learning results

The initial experiment is the reconstruction of a single missing frequency band with three different bandwidths (2.5 Hz, 5 Hz and 10 Hz) and is shown in Figure 4-3, where the results of the CNN interpolation are those in the second column. This, in combination with the results from Table 4-1, show that for the CNN it can be said that larger band gaps are more difficult to interpolate, which makes sense as there is a greater degree of uncertainty. Large gaps, i.e. 10 Hz, still manage to find the sea-bottom reflections (highest amplitude event in the data) but struggle to find the lower amplitude reflection information.

When interpolating two frequency bands, the CNN is again able to accurately reconstruct the sea-bottom reflection, but the energy of the lower amplitude reflections is lacking. The best interpolation here, naturally, is the narrow 2.5 Hz band gap. Based on the NRMS, the error is slightly too high in a 4D seismic setting (which is ideally below 0.1), however, the remaining statistics, such as the PSNR and SSIM are promising. Furthermore, as is the case when a single frequency band is removed from the data, wider frequency bands result in less accurate interpolations.

So far, one may assume that the above simply indicates that more missing frequency data results in a less accurate reconstruction, which is true when only increasing the width of a single frequency gap. However, when looking at the reconstruction of six narrow frequency bands (Figure 4-10 and Table 4-3), we see that the reconstruction is more accurate than a single 10 Hz band reconstruction, even though more frequency information is removed from the data. This is in turn shows that it is possible to remove more frequency data from the gather, so long as there is sufficient information within the missing band gaps.

When considering the frequency dependency of the CNN's interpolation (Figure 4-6), we observe that the reconstruction becomes less accurate between 25 and 75 Hz, but still within reasonable boundaries. It should be noted that in the interest of time, the networks used to estimate the error for each frequency were only trained for 20 epochs each and could have likely been improved with more training. For this reason, the values presented in Figure 4-6 should be taken as an indicator of the frequency dependency of the reconstruction, rather than fact.

In each result, the most accurately reconstructed parts of the spectrum are the highest amplitude events in the data. It is likely because of their high amplitude that these events will also have the largest error after being filtered. In turn, the loss function of the network will focus on correcting these errors until they are in the same order as the error in the remaining data, whereafter the network will focus on both regions. Perhaps it would be interesting to see how the network is able to perform when the lower amplitudes have been boosted with a reversible gain.

5-2 Sparse optimization results

From the results presented Table 4-1 and Table 4-2, it can be observed that the sparse solver is capable of producing good results when interpolating single frequency bands that are narrow, as can be seen in Figure 4-3 and Table 4-1. However, it is sensitive to the width of the band gaps. When the band gap has a width of 10 Hz, interpolation becomes unsatisfactory and the solution is both inaccurate in amplitude and phase.

Furthermore, the interpolation of six missing frequency bands is inaccurate even though the band gaps are narrow, presumably because the missing energy is too closely packed for the solver to accurately fit a curve. To illustrate, during tests, it was observed that the optimizer designed for the sparse solver prioritizes the clusters of known data, over sparsely sampled data, which as consequence impedes the solver from reconstructing the high amplitudes for a majority of the data. Figure 5-1 shows the amplitude spectrum of a trace where this effect is particularly noticeable. The yellow markers show the data points that are provided to the optimizer for interpolation. The black curve is the target amplitude spectrum and the red curve is the solver's approximation. Note that this approximation is not the same as the reconstruction. Instead, this curve corresponds to $\hat{d}_j^{est}(\omega)$ from (3-9). Clearly, the approximation is able to perfectly fit the densely sampled portion of the spectrum but fails to fit through the sparsely sampled frequencies between the missing frequency bands. Since this is where the amplitude is high, much of the high amplitude information is lost in the reconstruction.

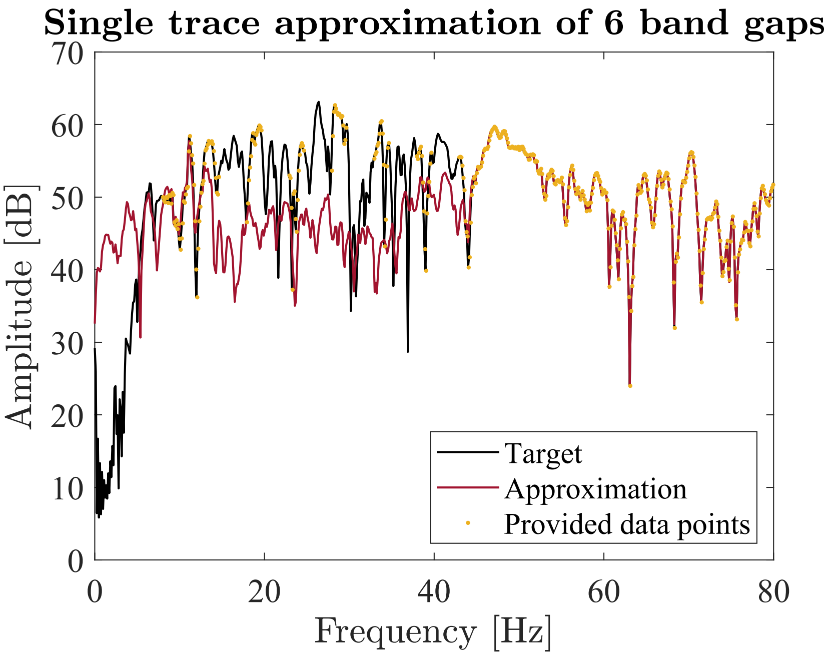


Figure 5-1: Example where the sparse solver is unable to estimate the sparsely sampled region of the spectrum. In this example the sparse solver is tasked to interpolate six missing frequency bands (2.5 Hz each). Note that the 'Approximation' curve corresponds to $\hat{d}_j^{est}(\omega)$ from (3-9) and not the full reconstruction.

Two potential methods to mitigating this issue and consequently improving multi-band interpolation are:

1. Reducing the number of samples in the densely sampled frequency ranges such that their perceived importance is equal to the sparsely sampled ranges. The downside of this approach is that useful frequency information is discarded.
2. Alternatively, the samples can be weighted such that the sparsely sampled regions carry equal weight as those in the densely sampled regions. Presumably, this can be achieved with standard inverse distance weighting algorithms.

Lastly, the τ-p reconstruction is shown to be inaccurate, however, I believe this is due to an error in the implementation of the transform, rather than an erroneous methodology. It was observed that even without interpolating frequency data, an error would exist in the gather that has been transformed back and forth to the τ-p domain. In an attempt to overcome the interpolation error I tried interpolating the data, transforming and then removing the interpolated points, however, this did not remove the error from the data. In the FK spectrum, part of the error from the transform can be observed above 60 Hz in Figure 4-12.

5-3 Comparison of interpolation methods

From the figures and interpolations shown, it is believed that the CNN is capable of providing a good estimate of the missing frequency information, given the missing frequency band is sufficiently narrow. The same is true for the sparse solver. Both methods produce comparable results for each test, except for the six-band interpolation. Here, modifications to the sparse solver's optimizer are required for the interpolation to yield an accurate result.

It is interesting to notice that each method uses a different principle to reconstruct the missing frequency gaps. The neural network will try to create a degree of prior knowledge of the problem through the learning process. Another way to interpret the neural network's method is that it tries to approximate the inverse process. The sparse solver on the other hand, only uses the knowledge it has available and instead by using assumptions of how the wave behaves in the subsurface is it able to find a reconstruction.

From there, the main disadvantage of a CNN interpolation scheme is clear. Like with all ANN implementations, there is a need for a large amount of training data. Thus, when selecting either of the discussed methods, it is important to consider the availability of training data. Although there is potential to use the sparse solver to generate the training data for the ANN, this would not typically be beneficial, as the interpolation results would deteriorate due to the network being trained on an already 'faulty' dataset. In contrast, the sparse solver is, in the case of seismic data, always applicable.

5-4 Limitations

One of the unfortunate compromises is that the network is trained on a fairly small dataset. Ideally, we would wish to train with as much data as possible in order to achieve greater accuracy and improve the network's ability to generalize in a broad array of situations. However, due to a limitation in hardware, I opted to pre-train the network on one set of portion of the data and followed by training it a second time on a different set of the data. However, I found that this approach typically did not lead to a greater interpolation accuracy. Perhaps an alternative approach to dealing with limited computational resources is possible.

The main assumption where this study is built upon is that it is possible to accurately deconvolve the vibrator sweep followed by convolving with an airgun source signature. Although it is possible if the sweep has been accurately sampled, in practice the process is not perfect and errors are likely to occur.

Limited time was spent tuning the hyperparamters of the CNN, which would likely improve the performance of the network. Parameters such as the starting values of the neurons and learning rate often play a strong role in the network's ability to minimize the loss function.

The frequency bands are removed by means of a butterworth filter, which is likely to be inconsistent with real observations, but does sufficiently represent a situation where there is a low SNR. It is possible that real-life observations will, however, measure energy in the 'missing' frequency band, perhaps from complex subsurface wave phenomena or particular noise sources. In this situation the assumption that no signal is present in this band would be invalid. Fortunately, in this situation, both methods are still able to function correctly without adjustments to their underlying principles. For the CNN, so long that training data is available that also inherits this property, little further action is required. For the sparse solver, the energy in the frequency band can be excluded from fitting. The remaining question is whether one would want to keep this additional data in the missing frequency gap.

5-5 Further research

It should be noted that the work presented in this thesis is by no measure fully functioning and ready to be implemented and should not be interpreted as such. Naturally, there are many potential improvements to the CNN and sparse solver that would reduce the interpolation error. Unfortunately, most of such improvements could not be implemented for this thesis, but would still be interesting to see in future works.

As this exact problem has not often been undertaken in previous literature, this thesis can serve as a type of benchmark and case study for future works. For example, it is possible to compare alternative network architectures to the U-Net presented in this thesis. Recent publications have shown positive results with so-called coordinate aware/attention networks for interpolation seismic data (Li et al., 2021a) or a nested U-Net (Zhou et al., 2018). It would be interesting to see how such networks perform on the problem at hand.

It is also possible to consider alternative algorithm all together. For example, in speech recognition there are a number of studies showing that the advantages of using a Short-Time Fourier Transform, which makes it possible to estimate the phase delay of each short-time increment. Thereafter, the phase delays are plotted against frequency to show the frequency dependency. The result will also show a band gap, which is then interpolated.

Furthermore, the problem can also be considered at a different stages in the processing workflow. One could perhaps consider the interpolation after CMP sorting, or even at an imaging level.

Finally, real-life tests rarely produce exactly what was predicted in synthetic models and for that reason this research would greatly benefit from real-life vibrator data, to observe any unexpected features of the data, if present.

Chapter 6

Conclusion

The goal of this research is to determine a methodology for the interpolation of missing frequency bands. For this, two methodologies are proposed. The first is a deep learning approach that implements a U-Net architecture. This is network is trained on 80 shot gathers for 20-40 epochs with a custom frequency loss function. The second interpolation method is a sparse solver that is inspired by compressed sensing approaches. The method approximates the reflectivity of the subsurface using the known frequency content, which is then used to add energy at the missing band(s).

The results found that both methods are capable of interpolating narrow frequency bands of 2.5 Hz with good accuracy (NRMS of <0.15). As the width of the frequency increases, the reconstruction of the missing frequency band becomes more difficult and the reconstruction error increases. When interpolating two missing frequency bands, the error also increases compared to a single missing frequency band, but the reconstruction is still accurate, so long as the band is narrow. For the interpolation of six narrow and closely spaced missing frequency bands, the sparse solver is unable to reconstruct the frequency spectrum. The network on the other hand is able to accurately reconstruct the spectrum in amplitude and phase. After adding an additional τ-p transform to the sparse solver methodology, spatially continuity is imposed, but the transformation proved to be unstable.

From the results, it can also be said that the sparse solver is more capable of determining high frequencies than low frequencies. For the CNN, interpolation error is smallest on the low and high frequencies but increases for 25 Hz to 75 Hz.

Bibliography

Abdelmalek, R., Mnasri, Z., and Benzarti, F. (2022). Audio signal reconstruction using phase retrieval: Implementation and evaluation. *Multimedia Tools and Applications*, 81(11):15919–15946.

Claerbout, J. (1978). Snell waves: Stanford exploration project report no. 15, 57–72.

Donoho, D. L., Johnstone, I. M., Hoch, J. C., and Stern, A. S. (1992). Maximum entropy and the nearly black object. *Journal of the Royal Statistical Society: Series B (Methodological)*, 54(1):41–67.

Dramsch, J. S. (2020). 70 years of machine learning in geoscience in review. *Advances in geophysics*, 61:1–55.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).

Fang, W., Fu, L., Zhang, M., and Li, Z. (2021). Seismic data interpolation based on u-net with texture loss. *Geophysics*, 86(1):V41–V54.

Foster, D. J. and Mosher, C. C. (1992). Suppression of multiple reflections using the radon transform. *Geophysics*, 57(3):386–395.

Goodfellow, I., Bengio, Y., and Courville, A. (2017). Deep learning (adaptive computation and machine learning series). *Cambridge Massachusetts*, pages 321–359.

Hawkins, A., Popper, A., Fay, R., Mann, D., Bartol, S., Carlson, T., Coombs, S., Ellison, W., Gentry, R., Halvorsen, M., et al. (2014). Sound exposure guidelines for fishes and sea turtles: A technical report.

Hlebnikov, V. (2022). Deep learning as a tool for seismic data interpolation.

Karlik, B. and Olgac, A. V. (2011). Performance analysis of various activation functions in generalized mlp architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems*, 1(4):111–122.

Kazemi, A., Stephen, K. D., and Shams, A. (2011). Improved normalization of time-lapse seismic data using nrms repeatabity data to improve automatic production and seismic history matching in the nelson field. In *SPE EUROPEC/EAGE Annual Conference and Exhibition*. OnePetro.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kussul, N., Lavreniuk, M., Skakun, S., and Shelestov, A. (2017). Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 14(5):778–782.

Laws, R., Halliday, D., Hopperstad, J.-F., Gerez, D., Supawala, M., Özbek, A., Murray, T., and Kragh, E. (2019). Marine vibrators: The new phase of seismic exploration. *Geophysical Prospecting*, 67(6-Geophysical Instrumentation and Acquisition):1443–1471.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

Li, X., Wu, B., Zhu, X., and Yang, H. (2021a). Consecutively missing seismic data interpolation based on coordinate attention unet. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5.

Li, Z., Sun, N., Gao, H., Qin, N., and Li, Z. (2021b). Adaptive subtraction based on u-net for removing seismic multiples. *IEEE Transactions on Geoscience and Remote Sensing*, 59(11):9796–9812.

Ma, Y., Ji, X., BenHassan, N. M., and Luo, Y. (2018). A deep learning method for automatic fault detection. In *2018 SEG International Exposition and Annual Meeting*. OnePetro.

Nielsen, M. A. (2015). *Neural networks and deep learning*, volume 25. Determination press San Francisco, CA, USA.

Ovcharenko, O., Kazei, V., Kalita, M., Peter, D., and Alkhalifah, T. (2019). Deep learning for low-frequency extrapolation from multioffset seismic data. *Geophysics*, 84(6):R989–R1001.

Panakkat, A. and Adeli, H. (2007). Neural network models for earthquake magnitude prediction using multiple seismicity indicators. *International journal of neural systems*, 17(01):13–33.

Qian, N. (1999). On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151.

Robertsson, J. (2016). Source separation method. *UK Patent: GB2547965B*.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

Sibi, P., Jones, S. A., and Siddarth, P. (2013). Analysis of different activation functions using back propagation neural networks. *Journal of theoretical and applied information technology*, 47(3):1264–1268.

Slang, S. (2019). *Attenuation of Seismic Interference Noise with Convolutional Neural Networks*. PhD thesis.

Southall, B. L., Finneran, J. J., Reichmuth, C., Nachtigall, P. E., Ketten, D. R., Bowles, A. E., Ellison, W. T., Nowacek, D. P., and Tyack, P. L. (2019). Marine mammal noise exposure criteria: Updated scientific recommendations for residual hearing effects. *Aquatic Mammals*, 45(2):125–232.

Sun, J. (2022). *Deep learning-based seismic data processing for attenuation of interference noise and deblending in the shot domain*. PhD thesis, University of Oslo.

Sun, J., Slang, S., Elboth, T., Greiner, T. L., McDonald, S., and Gelius, L.-J. (2020). Attenuation of marine seismic interference noise employing a customized u-net. *Geophysical Prospecting*, 68(3):845–871.

Taylor, H. L., Banks, S. C., and McCoy, J. F. (1979). Deconvolution with the 1 norm. *Geophysics*, 44(1):39–52.

Tomishin, Y. and Chizhik, A. (1982). *Soviet Union Patent SU1543357A1*.

Wang, B., Zhang, N., Lu, W., and Wang, J. (2019). Deep-learning-based seismic data interpolation: A preliminary result. *Geophysics*, 84(1):V11–V20.

Wang, R. and Herrmann, F. (2016). Frequency down-extrapolation with tv norm minimization. *Geophysics*, 86:1380–1384.

Yilmaz, Ö. (2001). *Seismic data analysis: Processing, inversion, and interpretation of seismic data*. Society of exploration geophysicists.

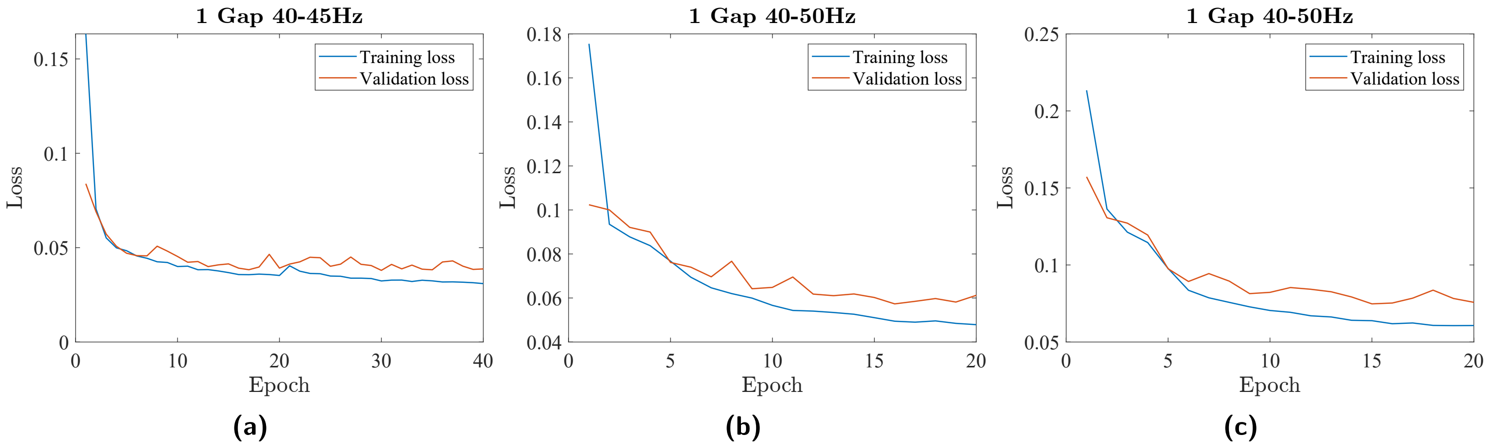
Yu, D., Wang, H., Chen, P., and Wei, Z. (2014). Mixed pooling for convolutional neural networks. In *International conference on rough sets and knowledge technology*, pages 364–375. Springer.

Zhou, Z., Rahman Siddiquee, M. M., Tajbakhsh, N., and Liang, J. (2018). Unet++: A nested u-net architecture for medical image segmentation. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 3–11. Springer.

Appendix A

Training curves of CNN reconstructions

A-1 Training curves one missing band



(a) **(b)** **(c)**

Figure A-1: Training curves of a single missing frequency band. **(a)** is 40-42.5 Hz, **(b)** is 40-45 Hz, **(c)** is 40-50 Hz.

A-2 Training curves two missing bands

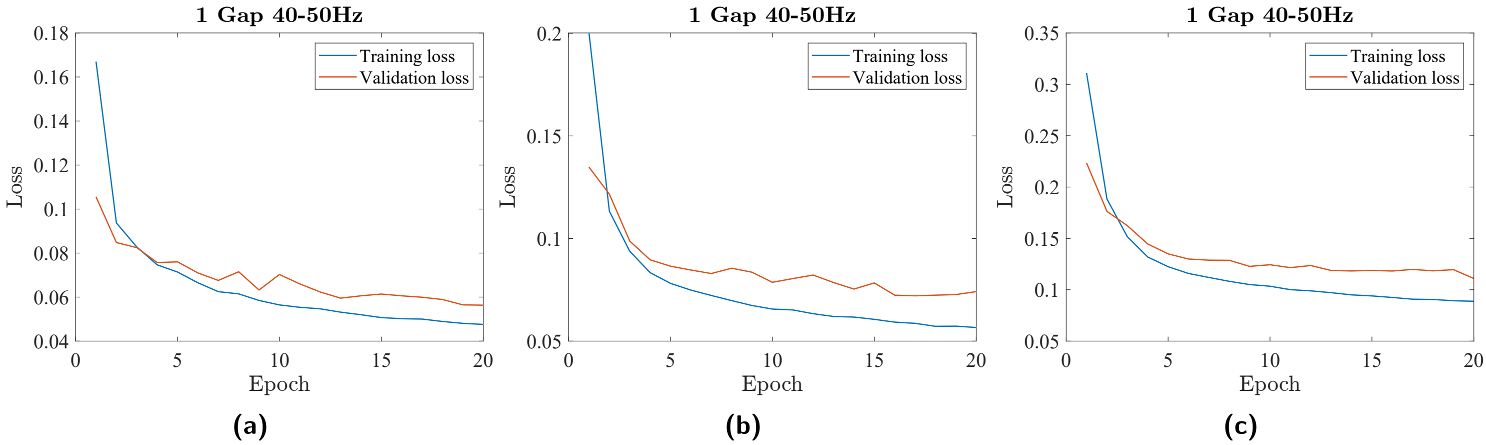


Figure A-2: Training curves of a single missing frequency band. **(a)** is 20-22.5 Hz and 40-42.5 Hz, **(b)** is 20-25 Hz and 40-45 Hz, **(c)** is 20-30 Hz and 40-50 Hz.

A-3 Training curve six missing bands

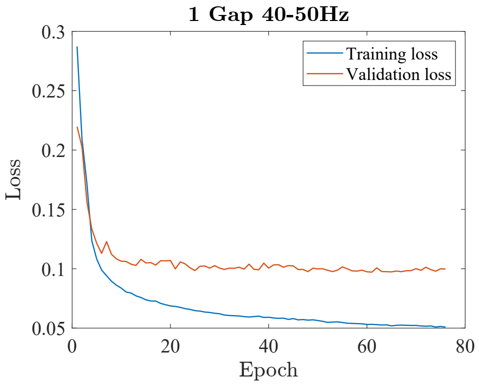
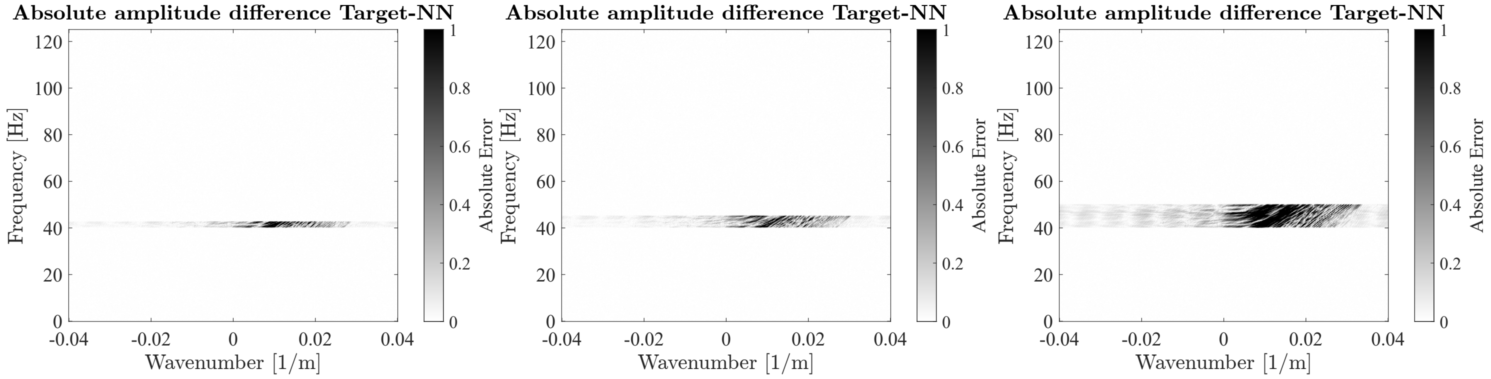


Figure A-3: Loss curves ofCNN that has been trained for six missing frequency bands

Appendix B

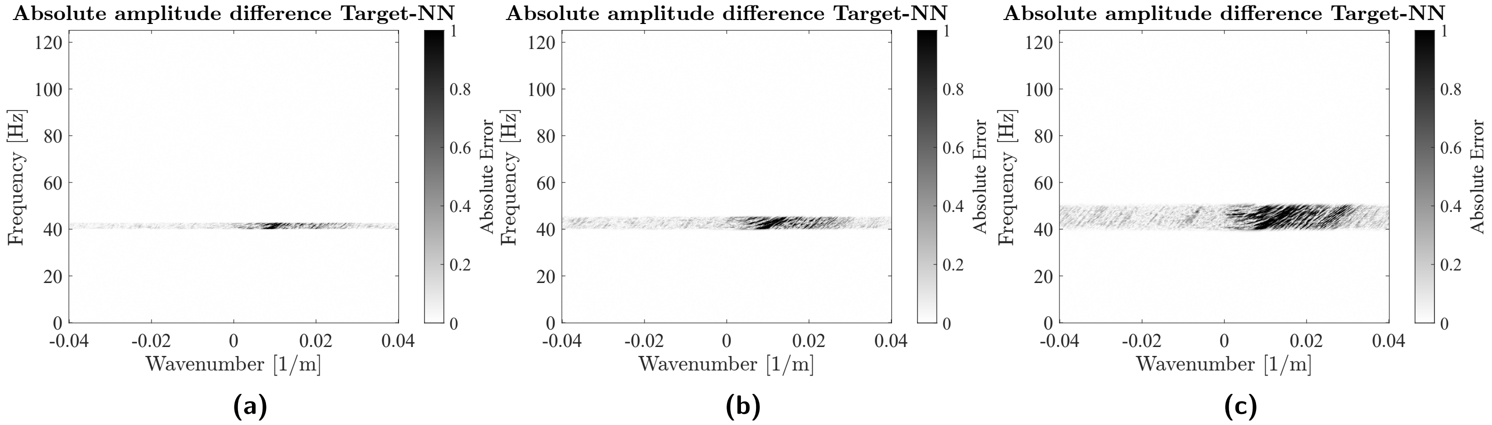
Absolute error plots of frequency reconstruction

B-1 One missing band



(a) **(b)** **(c)**

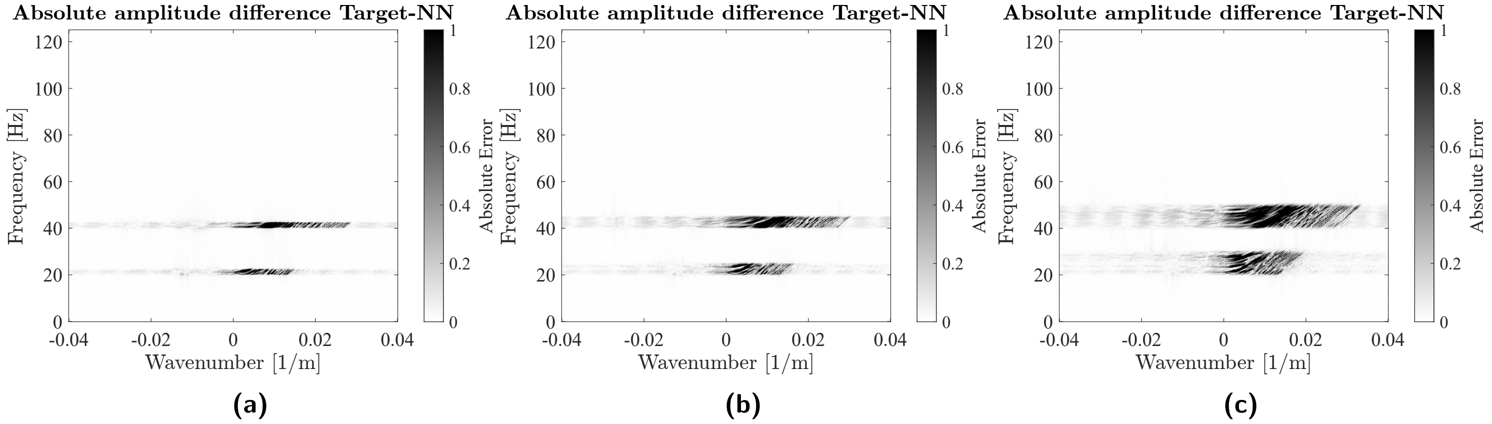
Figure B-1: Absolute error of amplitude spectra for the CNN's single band reconstruction. **(a)** is the a 2.5 Hz band, **(b)** is a 5 Hz gap and **(c)** is a 10 Hz gap.



(a) (b) (c)

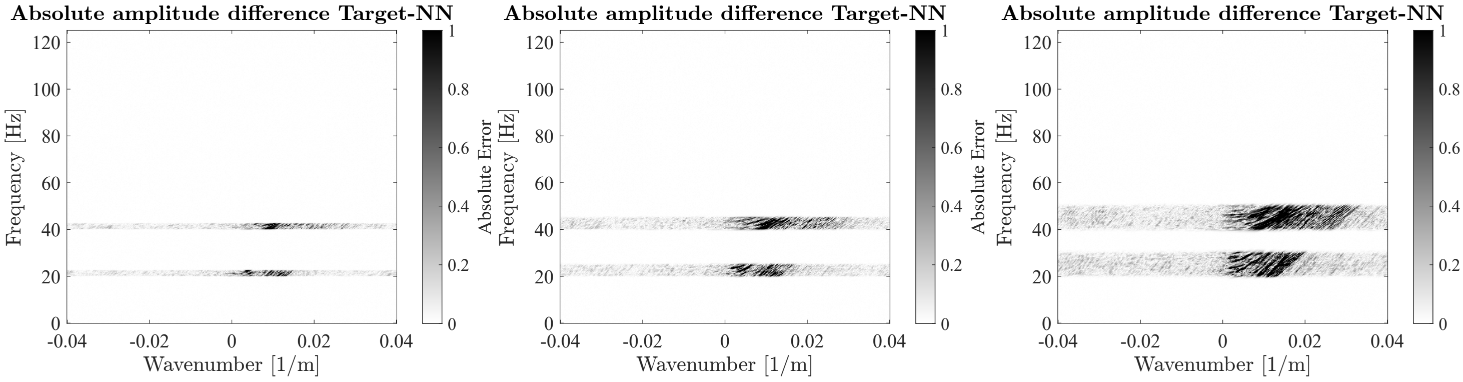
Figure B-2: Absolute error of amplitude spectra for the sparse solver's single band reconstruction. **(a)** is the a 2.5 Hz band, **(b)** is a 5 Hz gap and **(c)** is a 10 Hz gap.

B-2 Two missing bands



(a) (b) (c)

Figure B-3: Absolute error of amplitude spectra for the CNN's reconstruction of two missing frequency bands. **(a)** is the a 2.5 Hz band, **(b)** is a 5 Hz gap and **(c)** is a 10 Hz gap.



(a)

(b)

(c)

Figure B-4: Absolute error of amplitude spectra for the sparse solver's reconstruction of two missing frequency bands. **(a)** is the a 2.5 Hz band, **(b)** is a 5 Hz gap and **(c)** is a 10 Hz gap.

B-3 Six missing bands

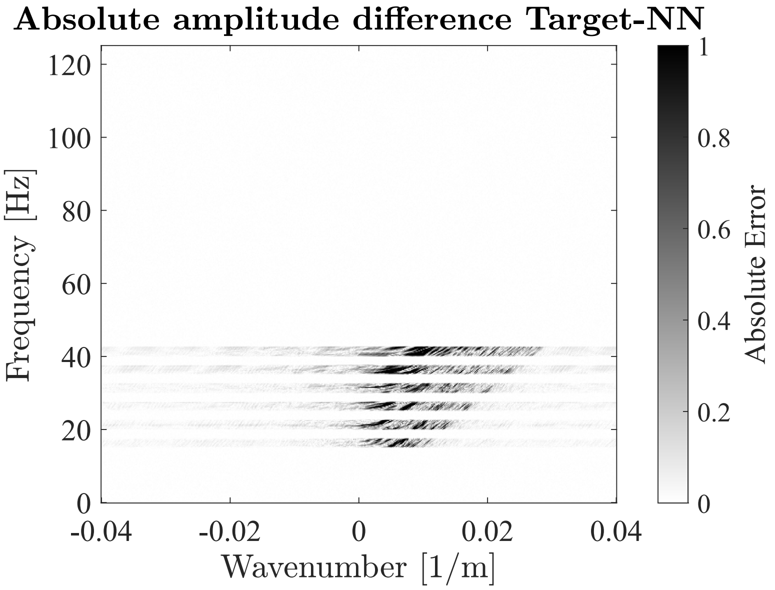


Figure B-5: Absolute error of amplitude spectra for the CNN's reconstruction of 6 missing frequency bands, each of which is 2.5 Hz wide.

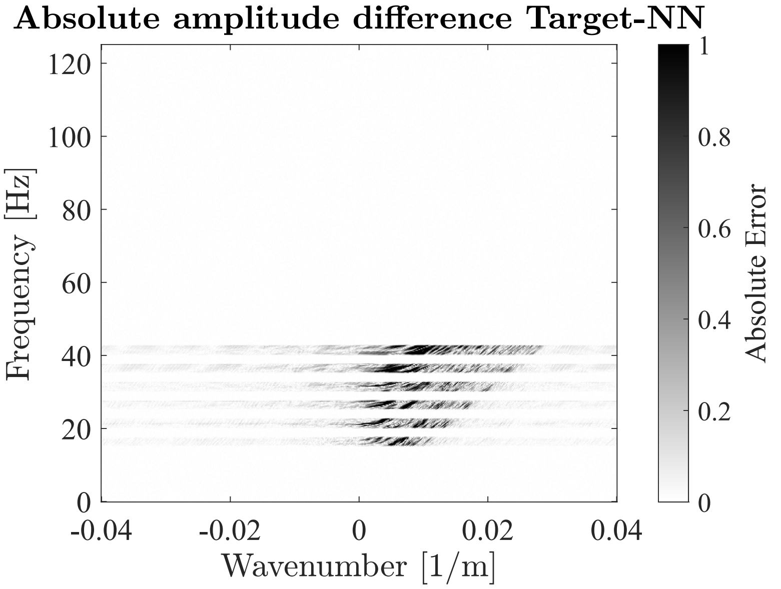


Figure B-6: Absolute error of amplitude spectra for the sparse solver's reconstruction of 6 missing frequency bands, each of which is 2.5 Hz wide.