GEOPHYSICS®

Unsupervised deep learning with higher-order total-variation regularization for multidimensional seismic data reconstruction

Journal:	*Geophysics*
Manuscript ID	GEO-2021-0099.R2
Manuscript Type:	Technical Paper
Keywords:	4D, artificial intelligence, interpolation, inversion, machine learning
Manuscript Focus Area:	Signal Processing, Seismic Inversion

DOI:10.1190/geo2021-0099.1

DOI: 10.1190/geo2021-0099.1

Unsupervised deep learning with higher-order total-variation regularization for multidimensional seismic data reconstruction

Thomas André Larsen Greiner[*†], Jan Erik Lie[†], Odd Kolbjørnsen[†‡], Andreas Kjelsrud Evensen[†], Espen Harris Nilsen[†], Hao Zhao[§], Vasily Demyanov[¶] and Leiv-J Gelius[*]

[*] *University of Oslo, Department of Geoscience, Sem Sælands vei 1, Oslo, NO 0316.*

E-mail: tgreiner1983@gmail.com; l.j.gelius@geo.uio.no.

[†] *Lundin-Energy Norway AS, Strandveien 4, Lysaker, NO 1366.*

E-mail: jan-erik.lie@lundin-energy.com;

Andreas-Kjelsrud.Evensen@lundin-energy.com;

Espen-Harris.Nilsen@lundin-energy.no; Thomas-larsen.greiner@lundin-energy.com

[‡] *University of Oslo, Department of Mathematics, Moltke Moes vei 35, Oslo, NO 0851. E-mail: oddkol@math.uio.no.*

[§] *Listen AS, Gaustadalléen 21, Oslo, NO 0349. E-mail: zhaohao7109@gmail.com.*

[¶] *Heriot-Watt University, Institute of Petroleum Engineering, Third Gait, Edinburgh UK EH14 4AS. E-mail: v.demyanov@hw.ac.uk.*

(November 4, 2021)

GEO-2021-0099.R2

Running head: **Unsupervised deep learning**

DOI: 10.1190/geo2021-0099.1

ABSTRACT

In 3D marine seismic acquisition, the seismic wavefield is not sampled uniformly in the spatial directions. This leads to a seismic wavefield consisting of irregularly and sparsely populated traces with large gaps between consecutive sail-lines especially in the near-offsets. The problem of reconstructing the complete seismic wavefield from a subsampled and incomplete wavefield, is formulated as an underdetermined inverse problem. We investigate unsupervised deep learning based on a convolutional neural network (CNN) for multidimensional wavefield reconstruction of irregularly populated traces defined on a regular grid. The proposed network is based on an encoder-decoder architecture with an overcomplete latent representation, including appropriate regularization penalties to stabilize the solution. We proposed a combination of penalties, which consists of the ℓ_2-norm penalty on the network parameters, and a first- and second-order total-variation (TV) penalty on the model. We demonstrate the performance of the proposed method on broad-band synthetic data, and field data represented by constant-offset gathers from a source-over-cable data set from the Barents Sea. In the field data example we compare the results to a full production flow from a contractor company, which is based on a 5D Fourier interpolation approach. In this example, our approach displays improved reconstruction of the wavefield with less noise in the sparse near-offsets compared to the industry approach, which leads to improved structural definition of the near offsets in the migrated sections.

INTRODUCTION

In marine seismic acquisition, the data are not sampled uniformly in the spatial directions, having a much coarser sampling along the crossline direction. The reasons for this are: large streamer separation, narrow source distribution, and large separation between consecutive sail-lines. The data recorded closest to the source array typically exhibit irregular and sparsely populated traces with large gaps. Some modern marine seismic acquisition solutions introduce close to zero-offset data and wide-azimuths by utilizing a split-spread, source-over-cable configuration (Vinje et al., 2017). However, this further complicates the reconstruction problem because the trace distribution becomes even sparser with the smaller distance to the source array.

Interpolation and regularization of the subsampled wavefield is an important and necessary processing step since coarse and irregular sampling of the seismic wavefield affects migration quality and analysis of amplitude variations with offset and azimuth (Trad, 2009). Several techniques have been developed for tackling the interpolation problem, such as frequency-space domain methods (Spitz, 1991; Porsani, 1999; Crawley, 2000; Naghizadeh and Sacchi, 2007), minimum weighted norm interpolation (MWNI) (Liu and Sacchi, 2004), projection onto convex sets (POCS) (Abma and Kabir, 2006), shaping regularization (Fomel, 2007), rank reduction based (Trickett et al., 2010, 2013), spectral analysis (Ghaderpour et al., 2018; Ghaderpour, 2019), common reflection surface (CRS) attribute-based (Hoecht et al., 2009; Xie and Gajewski, 2016; Zhao et al., 2020), Fourier based (Xu et al., 2005; Zwartjes and Sacchi, 2007; Schonewille et al., 2009; Naghizadeh and Sacchi, 2010b), Radon based (Ibrahim et al.,

2015), Curvelet (Naghizadeh and Sacchi, 2010a; Herrmann and Hennenfent, 2008), Seislet (Gan et al., 2015) and Focal transform (Kutscha et al., 2010). Amongst the popular techniques for multidimensional interpolation, such as 5D interpolation, are (Trad, 2014): MWNI, POCS, Anti-Leakage Fourier Transform (ALFT) (Xu et al., 2005), and rank reduction of Hankel tensors (Trickett et al., 2013). Methods such as CRS have also shown promising results compared to industry-standard methods in the multidimensional case (Zhao et al., 2020).

Recently, data-driven approaches have drawn significant attention for solving the interpolation and reconstruction problem. This includes techniques such as dictionary learning (Liang et al., 2014; Yu et al., 2015; Zhu et al., 2017; Turquais et al., 2018), support vector regression (Jia and Ma, 2017; Jia et al., 2018) and CNN-based deep learning such as Residual network (Wang et al., 2019), encoder-decoder networks (Wang et al., 2020), wavelet domain CNN (Larsen Greiner et al., 2020), generative adverserial network (Oliveira et al., 2018) and hybrid CNN/POCS-based method (Zhang et al., 2020). The use of CNN's have also drawn considerable attention in other processing applications, such as interference noise attenuation (Sun et al., 2020a), deblending (Sun et al., 2020b) and inversion (Das et al., 2018). So far, existing deep learning approaches have exclusively been applied and trained in a supervised fashion in 2D (image-based). In this case, a labeled data set is available to train the CNN, and traces must be in close proximity in order to be handled adequately in 2D. For filling in large gaps between traces it is necessary for the interpolation or reconstruction method to handle multiple dimensions simultaneously (Trad, 2014). In addition, in

geophysical reconstruction problems, labeled training data are limited and needs to be provided from either interpolation using a conventional technique or trained in a representative domain where labeling is possible. By extending the reconstruction problem into 4D (inline, crossline, time, offset) —or higher dimensions—, the extra dimensions carry information in different spatial locations, thus the distance between traces becomes smaller.

A few studies have shown promising results in unsupervised learning applied to seismic data reconstruction in 2D (Hu et al., 2019; Shi et al., 2020) and 3D (Kong et al., 2020). In Shi et al. (2020) and Kong et al. (2020), the authors makes use of the deep image prior (DIP) approach (Ulyanov et al., 2018), which exploits the ability of the network to capture low-level statistics of the data without learning from a specific input pattern. At first glance, the DIP approach looks like a basic autoencoder. However, while the network of an autoencoder is trained to reconstruct an approximation of its input, the DIP approach trains the CNN as a generator network to predict the observed data from a fixed random tensor. Liu et al. (2019) further improved the DIP approach on natural images by introducing an explicit total-variation (TV) regularizer on the model. A downside of the DIP approach is that the network needs to be reparameterized (retrained) for every new prediction. Thus, DIP does not generalize to a specific input pattern —as is the case with autoencoding—, which becomes computationally demanding with large data sets.

We introduce a regularized convolutional autoencoder (RCAE), as a solution to the multidimensional seismic wavefield reconstruction problem, i.e., we solve for the

inverse problem directly using deep learning methodology. This is accomplished by extending the deep learning approach from 2D/3D to 4D, by using a 3D CNN where the fourth dimension is represented by offset-classes. Thus, the offset dimension is treated as a feature space and is non-convolutional. The network architecture used in this study consists of an encoder-decoder network with an overcomplete latent representation constrained by three different regularization penalties. Similar to Liu et al. (2019), we propose an explicit regularizer to the model, but extend it to include both the first- and second-order TV norm, to properly fit the seismic data reconstruction problem. In addition, we include an ℓ_2-norm penalty on the network parameters. We demonstrate the methodology on densely sampled broad-band synthetic data and a field data set from the Barents Sea (Vinje et al., 2017). Both the synthetic and field data sets are represented by constant-offset gathers from a source-over-cable survey design. Both data sets pose challenges at near-offsets due to a sparse distribution of traces and large gaps in the crossline direction caused by the distances between consecutive sail-lines, as illustrated from the field data in Figure 1.

First, we introduce unsupervised deep learning through the use of autoencoders for multidimensional seismic wavefield reconstruction. The differences between supervised and unsupervised learning in such settings are also discussed. The performance of the proposed methodology is then demonstrated on synthetic data, followed by the field data example. Finally, a discussion part and a set of conclusions are given.

DOI: 10.1190/geo2021-0099.1

UNSUPERVISED DEEP LEARNING FOR SEISMIC WAVEFIELD RECONSTRUCTION

Autoencoders

Traditionally, autoencoders were used for dimensionality reduction, aiming to find hidden representations of an input vector $\mathbf{d}$ on a lower dimensional Hilbert space, and are therefore referred to as undercomplete (Goodfellow et al., 2016). Dimensionality reduction with autoencoding is accomplished by first mapping a vector $\mathbf{d}$ to a lower dimensional latent representation ζ through a functional $\zeta = \phi(\mathbf{d})$ —also known as the encoder—, followed by a mapping back to an approximation $\hat{\mathbf{d}}$ of the input, through another functional $\psi(\phi(\mathbf{d}))$ —also known as the decoder. A shortcoming of undercompleteness in reconstruction applications is that valuable information may be lost. To address such limitations, some encoder-decoder architectures include by-pass connections with concatenating layers to retain details, such as the U-Net architecture (Ronneberger et al., 2015). The advantage of multi-resolution architectures is that they increase the receptive field of the network, i.e., increasing the effective kernel size due to downsampling of features, while retaining high frequencies. However, as discussed by Ye et al. (2018) the U-Net architecture leads to an over-emphasis on the low frequencies since pooling with by-pass connections does not guarantee perfect reconstruction. Ye et al. (2018) therefore proposed the use of wavelet expansions as a generalized pooling operation in order to separate low frequencies from the high frequencies.

Alternative to by-pass connections, are encoder-decoder architectures with over-complete latent representation, i.e., the latent representation lies on a higher dimensional Hilbert space relative to its input. When downsampling in overcomplete networks, strided convolutions can be employed as an alternative to pooling. This avoids the low-pass filtering from pooling, but adds additional complexity since the downsampling operations are learned instead of being fixed operators. A simplified illustration of an undercomplete and overcomplete autoencoder is presented in Figure 2. However, overcomplete autoencoders are susceptible to overfitting without proper regularization. Frequently employed regularizers in deep learning includes the ℓ_2-norm penalty to the network parameters and sparseness constraints (Ranzato et al., 2007) including penalties to derivatives such as contractive autoencoders (Rifai et al., 2011). In addition, TV regularization (Rudin et al., 1992) has recently received attention in the field of deep learning for image inpainting problems (Liu et al., 2018, 2019). The TV-norm is a traditional and popular regularizer for solving inverse problems and has a wide range of applications, such as image inpainting (Shen and Chan, 2002), image restoration (Strong and Chan, 2003), video restoration (Chan et al., 2011) and seismic inversion (Gholami, 2015; Kolbjørnsen et al., 2019; Esser et al., 2018). In image inpainting, the introduction of higher-order TV-norms has also shown the ability to fill in large gaps and to reduce stair-casing artifacts with the price of some blur (Papafitsoros et al., 2013; Papafitsoros and Schönlieb, 2014). Multidimensional wavefield reconstruction of irregularly distributed traces defined on regular grids can be considered analogous with these types of computational inverse problems for natural images.

Supervised vs. unsupervised reconstruction

In the context of wavefield reconstruction, the inverse problem is the task of recovering the complete wavefield, which we will denote by $\mathbf{m}$, from the subsampled and observable wavefield, which we will denote by $\mathbf{d}_s$, by solving an equation on the form

$$\mathbf{d}_s = \mathcal{S}\big(\mathbf{m}\big) + \boldsymbol{\varepsilon}_s, \tag{1}$$

where $\mathcal{S}(\cdot)$ is the direct operator, which describes how the complete wavefield gives rise to the observed wavefield, and where $\boldsymbol{\varepsilon}_s$ represents the observation noise. The main difference between supervised and unsupervised deep learning for solving equation 1, is what we consider known a priori. This leads to two different strategies for training the neural network.

To solve for the inverse problem in a supervised fashion, we need a labeled data set that consists of many patches of input-target pairs $\mathcal{D} = \{(\mathbf{d}_s^{[i]}, \mathbf{m}^{[i]})\}_{i=1}^{P}$ where the i^{th} example $\mathbf{d}_s^{[i]}$ and $\mathbf{m}^{[i]}$ are commonly referred to as the input label and target label, respectively. The labeled training data are then used to estimate a functional representation of the complete wavefield

$$\mathbf{m}_f = f\big(\mathbf{d}_s; \Theta\big), \tag{2}$$

where the parameters Θ are learned through solving an optimization problem. A common approach is to minimize a misfit function in a least-squares sense, by including a penalty to the network parameters. This leads to the following optimization

problem

$$\min_{\boldsymbol{\Theta}} \Big\{ \frac{1}{2P} \sum_{i=1}^{P} \big\| \mathbf{m}^{[i]} - \mathbf{m}_f^{[i]} \big\|_2^2 + \mathcal{R}_\lambda\big(\Theta\big) \Big\}, \tag{3}$$

where $\| \cdot \|_2$ denotes the ℓ_2-norm and $\lambda \geq 0$ denotes the regularization parameter that defines the trade-off between the data misfit error and the regularization penalty. In deep learning, the ℓ_2-norm is commonly used as penalty to the network parameters, which reduces overfitting by dampening the size of the parameters and spreading them more evenly over the hidden units (Hastie et al., 2009). Now, given an example of an unseen and representative wavefield $\mathbf{d}_u$, we can predict the complete wavefield from the trained function $\hat{\mathbf{m}}_f = \hat{f}\big(\mathbf{d}_u; \hat{\Theta}\big)$ where $\hat{\Theta}$ denotes the learned parameters.

In many circumstances, we do not have access to a set of input-target pairs, and therefore need to turn to unsupervised approaches. In this case, assuming $\mathcal{S}(\cdot)$ is known, we can use the unlabeled data $\mathbf{d}_s$ to train a functional representation of the complete wavefield by minimizing an objective function of the form

$$\min_{\boldsymbol{\Theta}} \Big\{ \frac{1}{2} \big\| \mathbf{d}_s - \mathcal{S}\big(\mathbf{m}_f\big) \big\|_2^2 + \mathcal{R}_\lambda\big(\Theta\big) + \mathcal{R}_\alpha\big(\mathbf{m}_f\big) \Big\}, \tag{4}$$

where $\mathcal{R}_\alpha\big(\mathbf{m}_f\big)$ denotes an explicit regularization penalty applied to the functional representation $\mathbf{m}_f$. From a Bayesian point of view, the formulation in expression 4 can be seen as a maximum a posteriori estimate using a prior on $\mathbf{m}_f$ (Getreuer, 2012). Thus, expression 4 includes a priori information on the likelihood of the solution $\mathbf{m}_f$. The minimization of the objective function in expression 4 leads to an approximation

of the input $\mathbf{d}_s$ by $\mathcal{S}(\mathbf{m}_f)$. This is a typical autoencoder setup, where $\mathcal{S}(\cdot)$ can be seen as a final layer with fixed weights. Both expressions 3 and 4 attempt to solve for the inverse problem in equation 1. Whereas the former considers the observed data $\mathbf{d}_s$ and the complete data $\mathbf{m}$ as known variables, the latter considers the direct operator $\mathcal{S}(\cdot)$ and observed data $\mathbf{d}_s$ as known.

Wavefield reconstruction problem

We will consider the reconstruction problem in relation to expression 4. Let the observed and incomplete wavefield be defined by the vector $\mathbf{d}_s \in \mathbb{R}^{M\times 1}$, which is modeled as the output of a sampling operator $\mathbf{S} : \mathbb{R}^{N\times 1} \rightarrow \mathbb{R}^{M\times 1}$ acting on the complete wavefield $\mathbf{m} \in \mathbb{R}^{N\times 1}$. That is, we want to solve the inverse problem

$$\mathbf{d}_s = \mathbf{S}\mathbf{m} + \boldsymbol{\varepsilon}_s. \tag{5}$$

Equation 5 defines an underdetermined system of equations with infinite number of solutions. To illustrate the action of the sampling operator, let the unknown data consist of six consecutive samples

$$\mathbf{m} = [\mathrm{m}_0, \mathrm{m}_1, \mathrm{m}_2, \mathrm{m}_3, \mathrm{m}_4, \mathrm{m}_5]^{\mathrm{T}}, \tag{6}$$

with observed samples in index position $S = \{0, 2, 4, 5\}$. In this case equation 5 becomes

$$\begin{pmatrix} \mathrm{d}_{s0} \\ \mathrm{d}_{s1} \\ \mathrm{d}_{s2} \\ \mathrm{d}_{s3} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathrm{m}_0 \\ \mathrm{m}_1 \\ \mathrm{m}_2 \\ \mathrm{m}_3 \\ \mathrm{m}_4 \\ \mathrm{m}_5 \end{pmatrix} + \begin{pmatrix} \varepsilon_{s0} \\ \varepsilon_{s1} \\ \varepsilon_{s2} \\ \varepsilon_{s3} \end{pmatrix}. \tag{7}$$

Because of the zero column, the sampling operator in equation 7 is singular and has no inverse. Since we will consider the reconstruction problem of irregularly distributed traces defined on a regular grid, it implies the introduction of another type of operator, also known as a masking operator. The masking operator can be defined from the sampling operator as $\mathbf{M} = \mathbf{S}^T\mathbf{S}$, where $\mathbf{M} : \mathbb{R}^{N\times 1} \rightarrow \mathbb{R}^{N\times 1}$ is represented by ones at sampled traces and zeros everywhere else. After introducing the masking operator, equation 5 becomes

$$\mathbf{d} = \mathbf{M}\mathbf{m} + \boldsymbol{\varepsilon}, \tag{8}$$

where $\mathbf{d} = \mathbf{S}^T\mathbf{d}_s$ and $\boldsymbol{\varepsilon} = \mathbf{S}^T\boldsymbol{\varepsilon}_s$. Using the example introduced above, equation 7 correspondingly transforms to

$$\begin{pmatrix} \mathrm{d}_0 \\ 0 \\ \mathrm{d}_2 \\ 0 \\ \mathrm{d}_4 \\ \mathrm{d}_5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathrm{m}_0 \\ \mathrm{m}_1 \\ \mathrm{m}_2 \\ \mathrm{m}_3 \\ \mathrm{m}_4 \\ \mathrm{m}_5 \end{pmatrix} + \begin{pmatrix} \varepsilon_0 \\ 0 \\ \varepsilon_2 \\ 0 \\ \varepsilon_4 \\ \varepsilon_5 \end{pmatrix}. \tag{9}$$

A simplified illustration of the inverse problem from a masked wavefield is illustrated in Figure 3. Since the operator $\mathbf{M}$ is known and well-defined it is easily extracted from the observed data and can therefore be used to solve for the inverse problem. However, the matrix $\mathbf{M}$ is only used for illustration of the problem. In practice, the diagonal structure of $\mathbf{M}$ implies that it can be represented by its diagonal elements for computational considerations.

Convolutional autoencoder for multidimensional reconstruction

In the case of 4D wavefield reconstruction, consider the observed wavefield $\mathbf{d}$ as a 4D binned wavefield. The observed wavefield can then be represented by a four-dimensional function $d(x, y, t, h)$ where (x, y) denotes the spatial directions (crossline and inline), t denotes the temporal direction and h denotes a scalar representation of the offset-class. Suppose each offset-gather has N_x crosslines, N_y inlines, N_t tem-

poral samples and N_h offset-classes, then the discrete samples of $d(x, y, t, h)$ form a 4D tensor. To ease the discussion of the multidimensional wavefield reconstruction problem we will, as before, represent the observed wavefield as a vector according to its lexicographic order of size $N \times 1$, where $N = N_x N_y N_t N_h$. Now, let the relation between the observed- and reconstructed wavefield have the form of a multi-layer CNN, which can be expressed by the functional relationship

$$\mathbf{m}_f = f(\mathbf{d}; \mathbf{W}, \mathbf{b}), \tag{10}$$

where $\mathbf{W}$ and $\mathbf{b}$ represent the convolutional kernels and biases, respectively. Here, the function in equation 10 is a 3D CNN where offset-classes are represented as different features in the input space. In this case, the convolutional kernels are represented by the 5D tensors $\mathbf{W}_{l+1} \in \mathbb{R}^{\mathrm{f}\times\mathrm{f}\times\mathrm{f}\times c_l \times c_{l+1}}$ and biases by the vectors $\mathbf{b}_{l+1} \in \mathbb{R}^{c_{l+1}}$ where $\mathrm{f} \times \mathrm{f} \times \mathrm{f}$ denotes the kernel size and c_l denotes the number of features in layer $l \in \{0, 1, ..., L-1\}$. A deep CNN consists of L feature layers —including input and output layer— with $L-1$ linear convolutions and non-linear transforms. The computational transition from an arbitrary layer to the next one consists of computing a set of convolution operations on the input values $\mathbf{a}_l$ to give the linear output $\mathbf{z}_{l+1}$, where $\mathbf{a}_l$ and $\mathbf{z}_{l+1}$ can be considered as 4D objects consisting of 3D features. The linear transition from one layer to the next is defined as

$$\mathbf{z}_{l+1} = \mathbf{W}_{l+1}\mathbf{a}_l + \mathbf{b}_{l+1}, \tag{11}$$

where $\mathbf{a}_l$ represents the non-linear output from the previous layer

$$\mathbf{a}_l = \sigma(\mathbf{z}_l) = \sigma(\mathbf{W}_l \mathbf{a}_{l-1} + \mathbf{b}_l). \tag{12}$$

where $\mathbf{a}_0 = \mathbf{d}$ for the input layer $l = 0$. In equation 12, $\sigma(\cdot)$ represents a component wise non-linear activation function. Here, we employ the rectifier linear unit (ReLU), which is defined as

$$\sigma(\mathbf{z}_l) = \max(0, \mathbf{z}_l), \tag{13}$$

where the output layer is in our case not bounded by a non-linearity; it is given by the linear transition from the last hidden layer

$$\mathbf{m}_f = \mathbf{z}_{L-1} = \mathbf{W}_{L-1} \mathbf{a}_{L-2} + \mathbf{b}_{L-1}. \tag{14}$$

Since the training of the CNN is done in an unsupervised fashion, it is commonly referred to as a convolutional autoencoder (CAE). As an example, consider a CAE with a single hidden layer, i.e., $L = 3$ layers. The CAE then consists of two functional transitions: the encoder $\phi : \mathbf{d} \rightarrow \zeta$, and the decoder $\psi : \zeta \rightarrow \mathbf{m}_f$, where $\zeta \in \mathbb{R}^{K \times 1}$ defines the K-dimensional latent representation of the input. The CAE, parameterized by $\{\mathbf{W}_l, \mathbf{b}_l\}_{l=1}^{2}$, gives the following encoder

$$\phi(\mathbf{d}) = \sigma\big(\mathbf{W}_1 \mathbf{d} + \mathbf{b}_1\big), \tag{15}$$

and decoder

$$\mathbf{m}_f = \psi(\phi(\mathbf{d})) = \mathbf{W}_2\phi(\mathbf{d}) + \mathbf{b}_2, \tag{16}$$

where $\mathbf{W}_1$ represents the input-to-hidden $K \times N$ dimensional weight matrix, $\mathbf{W}_2$ represents the hidden-to-output $N \times K$ dimensional weight matrix, and $\mathbf{b}_1$ and $\mathbf{b}_2$ are the biases. Multiplying the masking operator with the output from the decoder gives an approximation of the input $\mathbf{M}\mathbf{m}_f$. In our case, we seek an approximate function $\hat{f}(\cdot) : \mathbf{d} \in \mathbb{R}^{N\times 1} \rightarrow \hat{\mathbf{m}}_f \in \mathbb{R}^{N\times 1}$ where $\hat{\mathbf{m}}_f$ denotes the predicted wavefield from the learned function $\hat{f}(\cdot)$ represented by the 3D CNN.

Before we explain the network used in this study we want to emphasize on the role of downsampling of features and at the same time retaining overcompleteness. The main goal of downsampling is that the effective kernel size increases, i.e., smaller kernels captures larger regions in the lower levels of the network. The ability of the network to correlate over larger regions is especially important for diffracted energy due to large Fresnel zones. On the other hand, the main goal of overcompleteness is to ensure that the network manages to preserve most of the relevant signal structure. Overcomplete autoencoders also promotes sparse feature representations (Goodfellow et al., 2016), and as shown by Papyan et al. (2017) if the ReLU non-linearity is employed, the neural network can be interpreted as a sparse coding algorithm. The role of overcompleteness in relation to the perfect reconstruction condition is further discussed in Ye et al. (2018).

In our experiments, we have found that by employing large kernels in the first layers gives improved reconstruction performance. Increased performance by employing

large kernels, typically in the first layer(s), has also been reported in both single-image super-resolution (Dong et al., 2015) and in super-resolution application within in seismic reconstruction (Greiner et al., 2019). For downsampling of features we employed 3D strides of size $k \times k \times k$, which implies a downsampling ratio of k^3. To ensure overcompleteness in this case, it is necessary to increase the number features in the next layer by a factor greater than k^3. This implies that the number of kernels increases by k^3, which increases the computational cost significantly and therefore limits the number of downsampling layers in practice.

In this study, the 3D CNN, summarized in Table 1, has nine layers in total — input, output and seven hidden layers — and consists of a basic feed-forward architecture, employing strided convolutions and strided transposed convolutions to downsample and upsample the features, respectively. The first hidden layer is computed by strided convolutions with kernels of size $7 \times 7 \times 7$ with N_h number of input features and 90 output features and with a stride of $2 \times 2 \times 2$, which gives a downsampling ratio of 8 in the first layer. We have used $N_h = 10$ which ensures redundancy in this case. The second hidden layer is computed by strided convolutions with kernel sizes of $5 \times 5 \times 5$ and stride of $2 \times 2 \times 2$, giving features of $1/64^{th}$ the size of the input features. This layer is followed by a regular convolution with kernel size of $3 \times 3 \times 3$. To upsample back to the original input size, we employed transposed convolutions with strides of $2 \times 2 \times 2$, using kernel sizes of $3 \times 3 \times 3$, followed by two layers of regular convolution with kernel sizes of $3 \times 3 \times 3$.

Pre-interpolation of input features

Some supervised CNN-based techniques introduce pre-interpolation of the input, such as bicubic interpolation (Wang et al., 2019), to upsample the input to equal size as the output. The CNN is then trained to remove the unwanted effects from the sub-optimal interpolation. Binning of the data into a regular grid using zero-padding could be seen as a special case of interpolation where data are interpolated —and extrapolated— with zero values. However, a particular challenge for the CAE approach arise when zero-padding introduces large gaps in all spatial directions; crossline, inline as well as offset, as displayed in Figure 1**b-e**. For the less sparse locations, i.e., close to the source array, this can be compensated by using large kernels in the first layers. However, using kernel sizes appropriate for the largest spatial gaps —up to 20 grid points in some locations—, increases the computational complexity significantly. Alternatively, one can compensate by adding more offsets to the input layer, but this comes at a cost of increasing the number of features within the hidden layers to ensure overcompleteness. Another alternative is to increase the regularization strength until the gaps in the near-offsets are filled adequately. However, in our experience this yields overly smooth results in the prediction. To overcome the near-offset challenge, we found that a simple nearest neighbor interpolation by interpolating smaller offsets using higher offset-classes balances the amplitudes and improves signal continuity in the large gaps. In our case, we used only a single observation as the nearest neighbor. Thus, we find the observation closest to the offset h in the input space to interpolate the smaller offsets. Since we are using only the input space as observation, this is

easily implemented as part of the forward pass within the network and does not add any significant additional complexity to the method or computational cost. Also, the pre-interpolated data is not used as the observed data in the data misfit within the objective function. It serves only as information about the wavefield pattern instead of zero values.

Regularization strategy

To stabilize the solution towards a desired solution, appropriate regularization is therefore introduced. Here, we propose a combination of regularization penalties, which includes the ℓ_2-norm on the CNN kernels and a first- and second-order TV-norm on the model $\mathbf{m}_f$. Thus, we want to minimize an objective function of the form

$$\min_{\mathbf{W},\mathbf{b}}\Big\{\frac{1}{2}\big\|\mathbf{d}-\mathbf{M}\mathbf{m}_f\big\|_2^2+\frac{\lambda}{2}\big\|\mathbf{W}\big\|_2^2+\mathrm{TV}(\mathbf{m}_f) +\mathrm{TV}^2(\mathbf{m}_f)\Big\}, \tag{17}$$

where $\mathrm{TV}(\mathbf{m}_f)$ and $\mathrm{TV}^2(\mathbf{m}_f)$ denote the first- and second-order TV-norm, respectively. The regularization parameters for the TV-norms are implicitly defined within the two terms. These norms can be defined from discrete approximations of derivatives. Thus, the first-order approximation is given by the gradient of the functional, which gives the following ensemble of first-order derivatives

$$\nabla \mathbf{m}_f = \left(\mathbf{D}_x \mathbf{m}_f, \mathbf{D}_y \mathbf{m}_f, \mathbf{D}_t \mathbf{m}_f, \mathbf{D}_h \mathbf{m}_f\right). \tag{18}$$

The second-order approximation can be defined from the gradient of the model gradients, which gives the symmetric Hessian distribution of second-order derivatives

$$\nabla^2 \mathbf{m}_f = \begin{pmatrix} \mathbf{D}_{xx}\mathbf{m}_f & \mathbf{D}_{xy}\mathbf{m}_f & \mathbf{D}_{xt}\mathbf{m}_f & \mathbf{D}_{xh}\mathbf{m}_f \\ \mathbf{D}_{yx}\mathbf{m}_f & \mathbf{D}_{yy}\mathbf{m}_f & \mathbf{D}_{yt}\mathbf{m}_f & \mathbf{D}_{yh}\mathbf{m}_f \\ \mathbf{D}_{tx}\mathbf{m}_f & \mathbf{D}_{ty}\mathbf{m}_f & \mathbf{D}_{tt}\mathbf{m}_f & \mathbf{D}_{th}\mathbf{m}_f \\ \mathbf{D}_{hx}\mathbf{m}_f & \mathbf{D}_{hy}\mathbf{m}_f & \mathbf{D}_{ht}\mathbf{m}_f & \mathbf{D}_{hh}\mathbf{m}_f \end{pmatrix}. \tag{19}$$

where $\mathbf{D}_i$ and $\mathbf{D}_{ij}$ are finite difference operators along the spatial and temporal directions. We employ the anisotropic version of the norms, which is defined as the ℓ_1-norm on the individual first- and second-order gradients. The definitions of the two norms then become

$$\begin{aligned} \mathrm{TV}(\mathbf{m}_f) = \frac{1}{N}\sum_{i=1}^{N} \big(&\alpha_1 |[\mathbf{D}_x \mathbf{m}_f]_i| + \alpha_2 |[\mathbf{D}_y \mathbf{m}_f]_i| \\ &+ \alpha_3 |[\mathbf{D}_t \mathbf{m}_f]_i| + \alpha_4 |[\mathbf{D}_h \mathbf{m}_f]_i|\big) \end{aligned} \tag{20}$$

for the first-order, and

$$\begin{aligned}\mathrm{TV}^2(\mathbf{m}_f) = \frac{1}{N}\sum_{i=1}^{N}\big(&\beta_1|[\mathbf{D}_{xx}\mathbf{m}_f]_i| + \beta_2|[\mathbf{D}_{yy}\mathbf{m}_f]_i| \\ &+\beta_3|[\mathbf{D}_{tt}\mathbf{m}_f]_i| + \beta_4|[\mathbf{D}_{hh}\mathbf{m}_f]_i| \\ &+\beta_5|[\mathbf{D}_{xy}\mathbf{m}_f]_i| + \beta_6|[\mathbf{D}_{xt}\mathbf{m}_f]_i| \\ &+\beta_7|[\mathbf{D}_{xh}\mathbf{m}_f]_i| + \beta_8|[\mathbf{D}_{yt}\mathbf{m}_f]_i| \\ &+\beta_9|[\mathbf{D}_{yh}\mathbf{m}_f]_i| + \beta_{10}|[\mathbf{D}_{th}\mathbf{m}_f]_i|\big)\end{aligned} \tag{21}$$

for the second-order. The scaling factors $(\alpha_j, j = 1, \ldots, 4)$ and $(\beta_j, j = 1, \ldots, 10)$ are introduced to add more flexibility to the impact of each individual operator on the regularization penalty. Consider data sorted in common-offset 3D cubes. From such a data cube, subsets of data can be formed by extracting either horizontal (time) slices or vertical slices. The former type of data are typically characterized by stronger structural features and with weaker texture as fill in. In case of vertical data, geological structures will dominate but with superimposed wave patterns associated with diffracted energy. The use of a first-order TV-norm will promote sparse gradients, i.e., it penalizes oscillations and promotes discontinuities. This fits well with geological or structural features present in the horizontal slices. However, in order to preserve the weaker texture as well as the vertical wave patterns, an additional use of a second-order TV-norm is essential. This norm promotes sparse second-order gradients, thus it puts emphasis on patterns with higher curvature, such as waves.

Introducing the TV-norms adds 14 additional hyperparameters. From our experiments, we found that it is sufficient to set $\alpha_{j\geq 2}$ and $\beta_{j\geq 1}$ values relative to α_1, which reduces the complexity to only one hyperparameter. In our case we have

used $\alpha_2 = \alpha_1$, $\alpha_3 = 0.1\alpha_1$, $\alpha_4 = 10\alpha_1$, $\beta_1 = \beta_2 = \beta_3 = \beta_4 = 0.001\alpha_1$ and $\beta_5 = \beta_6 = \beta_7 = \beta_8 = \beta_9 = \beta_{10} = 0.0001\alpha_1$.

Training and implementation details

Minimizing expression 17 and learning the parameters $\Theta \in \{\mathbf{W}_l, \mathbf{b}_l\}_{l=1}^{L-1}$, is achieved by solving an optimization problem. First, the observed data are divided into smaller overlapping patches, with P as the number of training examples $\mathcal{D} = \{\mathbf{d}^{[i]}\}_{i=1}^{P}$. These patches are then grouped into batches of size B, such that the input to the optimization algorithm consists of randomized mini batches of training examples. During training, we apply a stochastic corruption of the input. This is done by forcing a number of traces to zero, such that the model learns to fill in these corrupted patterns. Stochastic corruption of the input is also a known approach in denoising autoencoders for manifold learning (Vincent et al., 2008). For the minimization of the objective function, we employed a first-order stochastic gradient descent algorithm known as decoupled weight decay (Loshchilov and Hutter, 2017). In this case, let $\mathcal{L}\big(\mathbf{d}^{[i]};\Theta\big)$ denote the data misfit including the two TV-norms and let $\mathcal{R}\big(\mathbf{W}\big)$ denote the ℓ_2 penalty on the kernels. The decoupled weight decay algorithm for update of Θ for one mini batch is then defined as

$$\Theta_{l,n} = \Theta_{l,n-1} - \eta\Big(\frac{1}{B}\sum_{i=1}^{B}\nabla_{\Theta_l}\mathcal{L}\big(\mathbf{d}^{[i]};\Theta\big)_n + \nabla_{\mathbf{W}_l}\mathcal{R}\big(\mathbf{W}\big)_{n-1}\Big), \tag{22}$$

where η is the fixed learning rate, n denotes the iteration number and $\nabla_{\boldsymbol{\Theta}_l}\mathcal{L}\big(\mathbf{d}^{[i]};\Theta\big)$ are the partial derivatives with respect to the parameters in the l'th layer using the Adam update rule (Kingma and Ba, 2014) and $\nabla_{\mathbf{W}_l}\mathcal{R}\big(\mathbf{W}\big)$ are the partial derivatives of the ℓ_2 penalty on the kernels. Initialization of the weights was done using He-initialization (He et al., 2015), which is particularly designed for handling ReLU non-linearity, and the biases were initialized to zero values. For model selection, we experimented by trial and error for a range of values for the learning rate, ℓ_2 regularization and TV-norms. We found $\eta = 0.0001$, $\lambda = 10^{-7}$, $\alpha_1 = 0.0001$ to be a good choice of hyperparameters. Neural network implementation and training was performed in Python using the NumPy (Harris et al., 2020) and Tensorflow (Abadi et al., 2016) libraries.

SYNTHETIC DATA EXAMPLE

To evaluate the 4D reconstruction approach using unsupervised deep learning, we test the proposed approach on densely sampled synthetic broadband data, employing a 2D survey, modeled on a 6.25×6.25 m^2 grid and with an offset spacing of 12.5 m. The temporal sample rate was set to 4 ms. The 3D model, displayed in Figure 4, is represented by a complex water bottom of bathymetry data from the Barents Sea, and a series of equally spaced diffraction lines beneath. The synthetic data were modeled by the diffraction modeling method (Jaramillo and Bleistein, 1999) with a constant background velocity of 1480 m/s. A subset of the modeled data was employed for training, which consists of 400 inlines, 400 crosslines and 200 temporal

samples, and a total of 10 offset classes decimated to a constant offset interval of 50 m (from 12.5-462.5 m offset). The observed data was constructed by first applying white noise to the fully sampled offset gathers, and then by removing traces with a masking operator extracted from the field data employed in the field data example in the next section. In this case, we have a noise-free and densely sampled data set, which we will refer to as the ground truth. This data set serves as testing data which we only use for comparison and estimation of model performance. The masked and noisy data, i.e., the observed data, is used to train the model.

To fit the model to the observed data, the offset-gathers were randomly split into overlapping patches of size $44 \times 44 \times 44 \times 10$ (crosslines, inlines, temporal samples and offsets) and grouped into batches of size 4. With this setup, training the model took approximately 4 min for each epoch on a fairly modern GPU (NVIDIA Tesla V100 SXM2 32GB), where we used a total of 250 epochs. The patches defines the observed data examples $\mathcal{D} = \{\mathbf{d}^{[i]}\}_{i=1}^{P}$ for $P = 3456$ number of examples.

The 12.5 m offset from the diffraction modeled data, the prediction and difference plot, along with their respective $f - k$ plots are displayed in Figure 5. Figure 5**a** shows that the geological model from Figure 4 sets up a complex wavefield with numerous diffraction events. Figure 5**b** displays the gather in 5**a** with white noise and the masking operator applied. Figure 5**c** and Figure 5**d** display the predicted wavefield from the RCAE, and the difference between the ground truth and predicted, respectively. Figure 5**e**-**h** display the ground truth, observed, predicted and difference plot in the $f - k$ domain, respectively.

The strong events from the waterbottom reflections in the reconstructed wavefield are relatively accurate compared to the ground truth. These events display less reconstruction error compared to the weaker diffraction events beneath. In the wiggle plot in Figure 6, we observe that the RCAE approach struggles with conflicting dips from weaker events and where the amplitude of the events display different amplitude strengths. This is prominent at the weaker diffraction events annotated by the red arrows in Figure 6**c**. The weaker diffraction events from the reconstruction show in general slightly weaker amplitude compared to the ground truth. Figure 7 show examples from the 12.5 m offset on time-slices through the water bottom reflection, where the RCAE reconstruction manages to fill in the with similar patterns as the ground truth, as implied by the low difference displayed in Figure 7**d**.

FIELD DATA EXAMPLE

The field data in this study consists of a 3D source-over-cable seismic acquisition, with 14 streamers and 3 source arrays, from the Barents Sea. The available data for this study consists of 255 inlines, 3172 crosslines and 1000 temporal samples with a sampling rate of 4 ms. The size of the 4D binning grid was set to 6.25×8.33 m^2 (inline$\times$crossline), and a scalar offset interval of 50 m with 25 m as the nearest absolute offset. We extracted a subset of the data consisting of 10 offset-classes, from 25-475 m, for training and demonstration. Binning of traces was done by keeping only the trace nearest to the bin-center. To fit the model to the observed data we used the same number of batches and the same patch size as in the synthetic study.

The patches define the observed data examples $\mathcal{D} = \{\mathbf{d}^{[i]}\}_{i=1}^{P}$ for $P = 5184$ number of examples. With this setup, training took approximately 7 minutes for each epoch for a total of 150 epochs. The computed mean squared error (MSE) of the data misfit during training is displayed in Figure 8. After training, reconstruction of the input data of size $255 \times 2000 \times 400 \times 10$ took approximately 2 min.

We compare the proposed RCAE to a full production flow from a contractor company. Their approach is based on a 5D interpolation approach employing the ALFT algorithm (Xu et al., 2005). We will refer to this approach as the industry-standard processing (ISP). We will in this example, visually assess the quality of the two approaches. We also asses the results after migration to compare the structural definition of the geology after imaging. Both approaches, ISP and the RCAE, include normal-moveout (NMO) correction in their workflows.

Figure 9 shows examples from the 25 m offset on time-slices, where the observed data (Figure 9**a**-**b**) display a challenging interpolation setting. We observe that the RCAE reconstruction (Figure 9**c**-**d**) and the ISP interpolation (Figure 9**e**-**f**) manage to fill in data with slightly different characteristics. The ISP is noisier with horizontally aligned striped artifacts. In the center of the zoomed sections Figure 9**d** and **f** we observe a feature dipping at 45deg downwards from left to right in the RCAE reconstruction. This feature is not present in the ISP. However, the feature is observable in the 125 m offset-class in both ISP and RCAE, as displayed in Figure 10**d** and **f**. This implies that the feature does not represent a false structure in the reconstructed 25 m offset from RCAE. On the other hand, we observe slightly more

dipping structures in the ISP, which is prominent in the crossline direction displayed in Figure 11**a**, **b** and **c**. From the $f-k$ domain displayed in Figure 12 it is difficult to compare the overall difference between the two approaches.

To compare the ability of the two approaches to preserve the structural definition of the geology, we migrate the results from the RCAE reconstruction and the ISP interpolation using 3D pre-stack Kirchhoff time migration. The migrated sections from the 25 m offset are displayed as time-slices in Figure 13 and as stacked offsets from 25-475 m in Figure 14. We see that the RCAE reconstruction (Figure 13**a**-**b**) shows fewer migration artifacts and clearer structural definition in the near-offset compared to the ISP (Figure 13**c**-**d**). We also observe that the general structural definition in the migrated 25 m offset-class from the RCAE (Figure 13**a**-**b**) is closer to the stacked offset section (Figure 14**a**-**b**), than what is displayed by the ISP (Figure 13**c**-**d**) when compared to its corresponding stacked section (Figure 14**c**-**d**). This implies improved reconstruction of wavefield patterns using RCAE in the sparse near-offsets. This is also prominent on the crossline sections displayed in Figure 15, where the RCAE (Figure 15**a**) display a more similar character to the stacked offset (Figure 15**b**) and more prominent fault patterns and fewer migration artifacts, than what is displayed by ISP in the migrated 25 m (Figure 15**c**). However, we observe that the RCAE and ISP display similar characteristics in the offset stacked migrated sections.

DISCUSSION

The multidimensional wavefield reconstruction by inversion is complex by nature due to the natural ill-posedness of the problem. To solve the inverse problem directly using a deep learning-based methodology, we extend the typical 2D approaches into 4D. This has advantages over the aforementioned image-based approaches due to the additional dimensions that allow for more trace contribution for learning local correlations. In addition, in the absence of labeled training data we have shown that the RCAE offers a promising alternative to the supervised deep learning techniques and compares with state-of-the-art processing from the industry.

The proposed method may also be considered as a more general approach for inverse problems, where the masking operator implies that we solve a special case of noise pattern, i.e., multiplicative noise by forcing parts of the data to zero. Therefore, the proposed method offers greater range of applications over pure interpolation algorithms. For instance, if the masking operator in expression 17 is equal to the identity operator this would be a deep learning-based TV regularized optimization problem for additive noise suppression. In addition, the TV-norms implemented here are also well established in super-resolution theory, which implies an even broader use of the proposed methodology.

The bottleneck for inversion by deep learning, is the training stage. Training the model can take a few hours or up to days, which depends on the network architecture, available computational power and convergence rate of the optimization algorithm. The main advantage of the unsupervised compared to the supervised approach is

that there is no need for a labeled data set. However, as shown in the synthetic example, the current methodology struggles with conflicting dips. A possible solution to this problem would be to use a semi-supervised approach, by augmenting the data set with a few labeled input-target pairs with these patterns, such that the model learns to reconstruct conflicting dips more accurately. In addition, if the trained model does not generalize well to different data sets, i.e., different wavefield character, frequency content and binning setup, then a transfer learning approach could be employed. In this case, the initial model corresponds to the learned parameters from this data set, rather than model parameters initialized to small random values. This may save computational time in the optimization stage due to the model being closer to an optimal local minimum. Comparing the DIP approach to the RCAE, DIP actually benefits from having more uniform energy in the input compared to irregularly sampled data with many zero-traces. As we experienced, using the sparse data as input caused problems and was susceptible to overfitting with overly smooth results in the interpolated traces. In our case, this was simply fixed by introducing a nearest-neighbor like interpolation, i.e., a non-convolutional layer, within the forward pass of the network. Compared to traditional techniques, the RCAE automatically tries to learn an optimal set of kernel functions, which are representative of the wavefield patterns in the observable data. Once the RCAE is learned, reconstruction of the 4D wavefield is achieved automatically.

Another issue that complicates the multidimensional reconstruction using CNNs, is that binning is necessary because the convolutions are applied to regular grids.

First, large bin sizes cause many traces to overlap within bins, which implies that traces need to be excluded or averaged. On the other hand, small bin sizes leads to sparser offset cubes and makes the reconstruction problem more challenging. Second, jittering effects —or saw tooth patterns— appears and increases with offset, which causes problems for learning an accurate wavefield pattern in the shallow part of the record. Since jittering increases with offset, this limits the number of offsets as input to the RCAE. Thus, there is a trade-off between bin size and the jittering introduced by increasing the number of offsets as input to the CNN. This changes from data set to data set and depends on the data- and regularization setup. Also, since the methodology proposed here is a 4D approach, we give up azimuth information. One possible extension to 5D, would be to bin the data to separate azimuth volumes. However, this would further complicate the problem, since the number of offset cubes increases and with a sparser distribution of traces. Many of the issues pointed out here have not been explored in the synthetic study, and is therefore something that should be enlightened in future studies using CNNs for the multidimensional reconstruction problem.

In the case of CNN architecture, the focus of this study is not finding an optimal architecture for the multidimensional reconstruction problem. As mentioned in the section "Convolutional autoencoder for multidimensional reconstruction", downsampling of features is advantageous since smaller kernels captures larger regions, but the exponential increase in the number of features puts a limit on the number of downsampling layers. Adding additional complexity to the network is not trivial, since

this may increase the difficulty of training a robust model, and because the 3D CAE approach is by itself computationally demanding.

CONCLUSION

In this paper, we investigated the potential of unsupervised deep learning as a tool for multidimensional wavefield reconstruction. The proposed method is based on a 4D approach employing a 3D CNN with offset-classes as the input feature space. The network architecture is based on an encoder-decoder network with an overcomplete latent representation. To stabilize the solution, we proposed a combination of three explicit regularization penalties including ℓ_2-norm penalty on the network parameters and a first- and second-order TV-norm on the model. We tested the proposed method on synthetic broadband data and a field data set from the Barents Sea represented by a 3D source-over-cable acquisition. We visually assessed the results and compared them with an industry-standard Fourier based processing flow, both after reconstruction and in the migrated domain. In the near-offsets, our approach showed in general a less noisy character, improved structural definition compared to the industry-standard approach. In the migrated sections, our approach showed improved structural definition and fewer migration artifacts in the near-offsets compared to the industry-standard approach. However, the results from the two methods display similar structural definition after stack of migrated offsets. The unsupervised deep-learning approach proposed in this paper has shown to be an effective tool for multidimensional wavefield reconstruction. For further improvements and robustness

of the model more inline sections should be included. A more sophisticated binning strategy along with other type of network designs should also be considered in future research.

ACKNOWLEDGMENTS

The authors would like to thank Lundin for allowing us to use the TopSeis, and Volodya Hlebnikov at University of Oslo for preparing the synthetic data and for valuable discussions. This research is financially supported by the Research Council of Norway, project number 287664.

REFERENCES

Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al., 2016, Tensorflow: Large-scale machine learning on heterogeneous distributed systems: arXiv preprint arXiv:1603.04467.

Abma, R., and N. Kabir, 2006, 3d interpolation of irregular data with a pocs algorithm: Geophysics, **71**, E91–E97.

Chan, S. H., R. Khoshabeh, K. B. Gibson, P. E. Gill, and T. Q. Nguyen, 2011, An augmented lagrangian method for total variation video restoration: IEEE Transactions on Image Processing, **20**, 3097–3111.

Crawley, S., 2000, Seismic trace interpolation with nonstationary prediction-error filters: PhD thesis, Stanford University.

Das, V., A. Pollack, U. Wollner, and T. Mukerji, 2018, Convolutional neural network for seismic impedance inversion, *in* SEG Technical Program Expanded Abstracts 2018: Society of Exploration Geophysicists, 2071–2075.

Dong, C., C. C. Loy, K. He, and X. Tang, 2015, Image super-resolution using deep convolutional networks: IEEE transactions on pattern analysis and machine intelligence, **38**, 295–307.

Esser, E., L. Guasch, T. van Leeuwen, A. Y. Aravkin, and F. J. Herrmann, 2018, Total variation regularization strategies in full-waveform inversion: SIAM Journal on Imaging Sciences, **11**, 376–406.

Fomel, S., 2007, Shaping regularization in geophysical-estimation problems: Geophysics, **72**, R29–R36.

Gan, S., S. Wang, Y. Chen, Y. Zhang, and Z. Jin, 2015, Dealiased seismic data interpolation using seislet transform with low-frequency constraint: IEEE Geoscience and remote sensing letters, **12**, 2150–2154.

Getreuer, P., 2012, Rudin-osher-fatemi total variation denoising using split bregman: Image Processing On Line, **2**, 74–95.

Ghaderpour, E., 2019, Multichannel antileakage least-squares spectral analysis for seismic data regularization beyond aliasing: Acta Geophysica, **67**, 1349–1363.

Ghaderpour, E., W. Liao, and M. P. Lamoureux, 2018, Antileakage least-squares spectral analysis for seismic data regularization and random noise attenuation: Geophysics, **83**, V157–V170.

Gholami, A., 2015, Nonlinear multichannel impedance inversion by total-variation regularization: Geophysics, **80**, R217–R224.

Goodfellow, I., Y. Bengio, and A. Courville, 2016, Deep learning: MIT Press. (http://www.deeplearningbook.org).

Greiner, T. L., O. Kolbjørnsen, J. E. Lie, E. H. Nilsen, A. K. Evensen, and L. Gelius, 2019, Cross-streamer wavefield interpolation using deep convolutional neural network, *in* SEG Technical Program Expanded Abstracts 2019: Society of Exploration Geophysicists, 2207–2211.

Harris, C. R., K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, 2020, Array programming with NumPy: Nature,

585, 357–362.

Hastie, T., R. Tibshirani, and J. Friedman, 2009, The elements of statistical learning: data mining, inference, and prediction: Springer Science & Business Media.

He, K., X. Zhang, S. Ren, and J. Sun, 2015, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification: Proceedings of the IEEE international conference on computer vision, 1026–1034.

Herrmann, F. J., and G. Hennenfent, 2008, Non-parametric seismic data recovery with curvelet frames: Geophysical Journal International, **173**, 233–248.

Hoecht, G., P. Ricarte, S. Bergler, and E. Landa, 2009, Operator-oriented crs interpolation: Geophysical Prospecting, **57**, 957–979.

Hu, L., X. Zheng, Y. Duan, and X. Yan, 2019, Unsupervised seismic data interpolation via deep convolutional autoencoder: 81st EAGE Conference and Exhibition 2019, European Association of Geoscientists & Engineers, 1–5.

Ibrahim, A., M. D. Sacchi, and P. Terenghi, 2015, Wavefield reconstruction using a stolt-based asymptote and apex shifted hyperbolic radon transform, *in* SEG Technical Program Expanded Abstracts 2015: Society of Exploration Geophysicists, 3836–3841.

Jaramillo, H. H., and N. Bleistein, 1999, The link of kirchhoff migration and demigration to kirchhoff and born modeling: Geophysics, **64**, 1793–1805.

Jia, Y., and J. Ma, 2017, What can machine learning do for seismic data processing? an interpolation application: Geophysics, **82**, V163–V177.

Jia, Y., S. Yu, and J. Ma, 2018, Intelligent interpolation by monte carlo machine learning: Geophysics, **83**, V83–V97.

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

Kolbjørnsen, O., A. Kjelsrud Evensen, E. H. Nilsen, and J. E. Lie, 2019, Digital superresolution in seismic avo inversion: The Leading Edge, **38**, 791–799.

Kong, F., F. Picetti, V. Lipari, P. Bestagini, X. Tang, and S. Tubaro, 2020, Deep prior-based unsupervised reconstruction of irregularly sampled seismic data: IEEE Geoscience and Remote Sensing Letters.

Kutscha, H., D. Verschuur, and A. Berkhout, 2010, High resolution double focal transformation and its application to data reconstruction, *in* SEG Technical Program Expanded Abstracts 2010: Society of Exploration Geophysicists, 3589–3593.

Larsen Greiner, T. A., V. Hlebnikov, J. E. Lie, O. Kolbjørnsen, A. Kjelsrud Evensen, E. Harris Nilsen, V. Vinje, and L.-J. Gelius, 2020, Cross-streamer wavefield reconstruction through wavelet domain learning: Geophysics, **85**, 1–84.

Liang, J., J. Ma, and X. Zhang, 2014, Seismic data restoration via data-driven tight frame: Geophysics, **79**, V65–V74.

Liu, B., and M. D. Sacchi, 2004, Minimum weighted norm interpolation of seismic records: Geophysics, **69**, 1560–1568.

Liu, G., F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, 2018, Image inpainting for irregular holes using partial convolutions: Proceedings of the European Conference on Computer Vision (ECCV), 85–100.

Liu, J., Y. Sun, X. Xu, and U. S. Kamilov, 2019, Image restoration using total variation regularized deep image prior: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 7715–7719.

Loshchilov, I., and F. Hutter, 2017, Decoupled weight decay regularization: arXiv preprint arXiv:1711.05101.

Naghizadeh, M., and M. Sacchi, 2010a, Hierarchical scale curvelet interpolation of aliased seismic data, *in* SEG Technical Program Expanded Abstracts 2010: Society of Exploration Geophysicists, 3656–3661.

Naghizadeh, M., and M. D. Sacchi, 2007, Multistep autoregressive reconstruction of seismic records: Geophysics, **72**, V111–V118.

———, 2010b, On sampling functions and fourier reconstruction methods: Geophysics, **75**, WB137–WB151.

Oliveira, D. A., R. S. Ferreira, R. Silva, and E. V. Brazil, 2018, Interpolating seismic data with conditional generative adversarial networks: IEEE Geoscience and Remote Sensing Letters, **15**, 1952–1956.

Papafitsoros, K., C. B. Schoenlieb, and B. Sengul, 2013, Combined first and second order total variation inpainting using split bregman: Image Processing On Line, **3**, 112–136.

Papafitsoros, K., and C.-B. Schönlieb, 2014, A combined first and second order variational approach for image reconstruction: Journal of mathematical imaging and vision, **48**, 308–338.

Papyan, V., Y. Romano, and M. Elad, 2017, Convolutional neural networks analyzed via convolutional sparse coding: The Journal of Machine Learning Research, **18**, 2887–2938.

Porsani, M. J., 1999, Seismic trace interpolation using half-step prediction filters: Geophysics, **64**, 1461–1467.

Ranzato, M., C. Poultney, S. Chopra, and Y. L. Cun, 2007, Efficient learning of sparse representations with an energy-based model: Advances in neural information processing systems, 1137–1144.

Rifai, S., P. Vincent, X. Muller, X. Glorot, and Y. Bengio, 2011, Contractive auto-encoders: Explicit invariance during feature extraction: Presented at the Icml.

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

Rudin, L. I., S. Osher, and E. Fatemi, 1992, Nonlinear total variation based noise removal algorithms: Physica D: nonlinear phenomena, **60**, 259–268.

Schonewille, M., A. Klaedtke, A. Vigner, J. Brittan, and T. Martin, 2009, Seismic data regularization with the anti-alias anti-leakage fourier transform: First Break, **27**.

Shen, J., and T. F. Chan, 2002, Mathematical models for local nontexture inpaintings: SIAM Journal on Applied Mathematics, **62**, 1019–1043.

Shi, Y., X. Wu, and S. Fomel, 2020, Deep learning parameterization for geophysical inverse problems: SEG 2019 Workshop: Mathematical Geophysics: Traditional vs Learning, Beijing, China, 5-7 November 2019, Society of Exploration Geophysicists, 36–40.

Spitz, S., 1991, Seismic trace interpolation in the fx domain: Geophysics, **56**, 785–794.

Strong, D., and T. Chan, 2003, Edge-preserving and scale-dependent properties of total variation regularization: Inverse problems, **19**, S165.

Sun, J., S. Slang, T. Elboth, T. Larsen Greiner, S. McDonald, and L.-J. Gelius,

2020a, Attenuation of marine seismic interference noise employing a customized u-net: Geophysical Prospecting, **68**, 845–871.

———, 2020b, A convolutional neural network approach to deblending seismic data: Geophysics, **85**, WA13–WA26.

Trad, D., 2009, Five-dimensional interpolation: Recovering from acquisition constraints: Geophysics, **74**, V123–V132.

———, 2014, Five-dimensional interpolation: New directions and challenges: CSEG Recorder, **39**, 40–46.

Trickett, S., L. Burroughs, and A. Milton, 2013, Interpolation using hankel tensor completion, *in* SEG Technical Program Expanded Abstracts 2013: Society of Exploration Geophysicists, 3634–3638.

Trickett, S., L. Burroughs, A. Milton, L. Walton, and R. Dack, 2010, Rank-reduction-based trace interpolation, *in* SEG Technical Program Expanded Abstracts 2010: Society of Exploration Geophysicists, 3829–3833.

Turquais, P., E. G. Asgedom, W. Söllner, and L. Gelius, 2018, Parabolic dictionary learning for seismic wavefield reconstruction across the streamers: Geophysics, **83**, V263–V282.

Ulyanov, D., A. Vedaldi, and V. Lempitsky, 2018, Deep image prior: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 9446–9454.

Vincent, P., H. Larochelle, Y. Bengio, and P.-A. Manzagol, 2008, Extracting and composing robust features with denoising autoencoders: Proceedings of the 25th international conference on Machine learning, 1096–1103.

Vinje, V., J. E. Lie, V. Danielsen, P. E. Dhelle, R. Silliqi, C.-I. Nilsen, E. Hicks, and

A. Camerer, 2017, Shooting over the seismic spread: First Break, **35**.

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

Wang, Y., B. Wang, N. Tu, and J. Geng, 2020, Seismic trace interpolation for irregularly spatial sampled data using convolutional autoencoder: Geophysics, **85**, V119–V130.

Xie, Y., and D. Gajewski, 2016, Interpolation and regularization with the 3d crs operator, *in* SEG Technical Program Expanded Abstracts 2016: Society of Exploration Geophysicists, 4092–4096.

Xu, S., Y. Zhang, D. Pham, and G. Lambaré, 2005, Antileakage fourier transform for seismic data regularization: Geophysics, **70**, V87–V95.

Ye, J. C., Y. Han, and E. Cha, 2018, Deep convolutional framelets: A general deep learning framework for inverse problems: SIAM Journal on Imaging Sciences, **11**, 991–1048.

Yu, S., J. Ma, X. Zhang, and M. D. Sacchi, 2015, Interpolation and denoising of high-dimensional seismic data by learning a tight frame: Geophysics, **80**, V119–V132.

Zhang, H., X. Yang, and J. Ma, 2020, Can learning from natural image denoising be used for seismic data interpolation?: Geophysics, **85**, WA115–WA136.

Zhao, H., V. Hlebnikov, E. Harris Nilsen, A. Kjelsrud Evensen, T. Greiner, J. Erik Lie, and L.-J. Gelius, 2020, Kinematic wavefield attributes based multidimensional prestack data reconstruction, *in* SEG Technical Program Expanded Abstracts 2020: Society of Exploration Geophysicists, 3304–3308.

Zhu, L., E. Liu, and J. H. McClellan, 2017, Joint seismic data denoising and in-

terpolation with double-sparsity dictionary learning: Journal of Geophysics and Engineering, **14**, 802–810.

Zwartjes, P., and M. Sacchi, 2007, Fourier reconstruction of nonuniformly sampled, aliased seismic data: Geophysics, **72**, V21–V32.

List of figures

Figure 1: This figure shows an illustration of a **a** source-over-cable survey employing both a streamer vessel and source vessel, and time-slices of a 4D binned wavefield from this type of survey with bin centers at **b** 25 m and **c** 125 m and their zoomed sections **d**-**e**, respectively. Adding near-offsets increases the complexity of the interpolation problem due to an even sparser distribution of traces, where the largest distances are ≈160 m in the 25 m offset section.

Figure 2: An autoencoder consists of two functional transitions from input $\mathbf{d} \in \mathbb{R}^N$ to the output $\hat{\mathbf{d}} \in \mathbb{R}^N$. The encoder transition $\phi : \mathbf{d} \rightarrow \zeta$ takes the input and transforms it to a latent feature space $\zeta \in \mathbb{R}^K$, followed by the decoder transition $\psi : \zeta \rightarrow \hat{\mathbf{d}}$, where the output represents an approximation of the input. In case of **a** undercompleteness we have $N > K$ where the input data is represented on a lower-dimensional space. For **b** overcompleteness we have $N \leq K$ and thus represents the data on a higher-dimensional space.

Figure 3: In the direct problem, the transformation $\mathcal{S}(\cdot)$ is applied to a complete wavefield $\mathbf{m}$. The inverse problem aims to obtain an approximation $\hat{\mathbf{m}}_f$ of the complete wavefield $\mathbf{m}$ from the observed wavefield $\mathbf{d}$. Here, we show the wavefield reconstruction case where $\mathcal{S}(\cdot)$ is represented by a masking operator $\mathbf{M}$.

Figure 4: Water bottom model of bathymetry data from side-scan sonar, in addition to a set of diffraction lines beneath. The model is defined on a grid size of 6.25×6.25 m^2. The red rectangle depicts the modelling area.

Figure 5: A crossline section from **a** the ground truth, **b** the observed wavefield, **c** the predicted wavefield, **d** the difference between **a** and **c**, and **e**-**h** shows the ground truth, observed, predicted and difference in the $f - k$ domain, respectively. The red and yellow boxes indicates the zoomed sections in the wiggle plots shown in Figure 6.

Figure 6: Zoomed section from the **a**,**d** ground truth, **b**,**e** predicted wavefield and **c**,**f** difference from the crossline section in Figure 5 marked by the red and yellow boxes.

Figure 7: Time-slices from the 12.5 m offset-class, with **a** the ground truth, **b** the observed wavefield, **c** predicted wavefield and **d** the difference between **a** and **c**.

Figure 8: The computed MSE from the data misfit function $\mathcal{L}(\mathbf{d}, \mathbf{L}\mathbf{m}_f)$ during training.

Figure 9: Time-slices from the 25 m offset-class, where **a**-**b** the observed data show a challenging interpolation setting with large gaps between traces. From **c**-**d** the RCAE reconstruction and **e**-**f** the ISP interpolation we observe in general a noisier character within the ISP. In addition, the feature observed within **d** the zoomed RCAE section annotated by the red arrow, is not present in **f** the zoomed ISP section.

Figure 10: Time-slices from the 125 m offset, where **a**-**b** displays the observed data. From **c**-**d** the RCAE is less noisy compared to **e**-**f** the ISP. However, the ISP interpolation display parts of the feature that was not present in the 25 m offset. The RCAE display this feature in both the 25 m offset and in the 125 m offset.

Figure 11: A crossline section from **a** the observed data, **b** the RCAE reconstruction and **c** the ISP. Here, we observe slightly more dipping energy in the ISP compared to the RCAE.

Figure 12: The crossline sections from Figure 7 displayed in the $f - k$ domain where **a** is the observed data, **b** the RCAE reconstruction and **c** the ISP.

Figure 13: Time-slices after migration of the 25 m offset, where **a** show the RCAE reconstruction and **c** show the ISP, with **b** and **d** as their corresponding zoomed sections, respectively.

Figure 14: Time-slices of the offset stacked migrated from 25-475 m, where **a** show the RCAE reconstruction and **b** show the ISP, with **c** and **d** as their corresponding zoomed sections, respectively.

Figure 15: Crossline sections after migration from the 25 m offset and offset stacked from 25-475 m. The migrated 25 m offset from **a** the RCAE reconstruction and **b** from the ISP. The offset stack from 25-475 m are shown in **c** and **d** from the RCAE and ISP, respectively.

List of tables

Table 1: The example CNN used in this study. The number of input features, i.e., offset-classes is given by N_h, where we have used $N_h = 10$.

Table

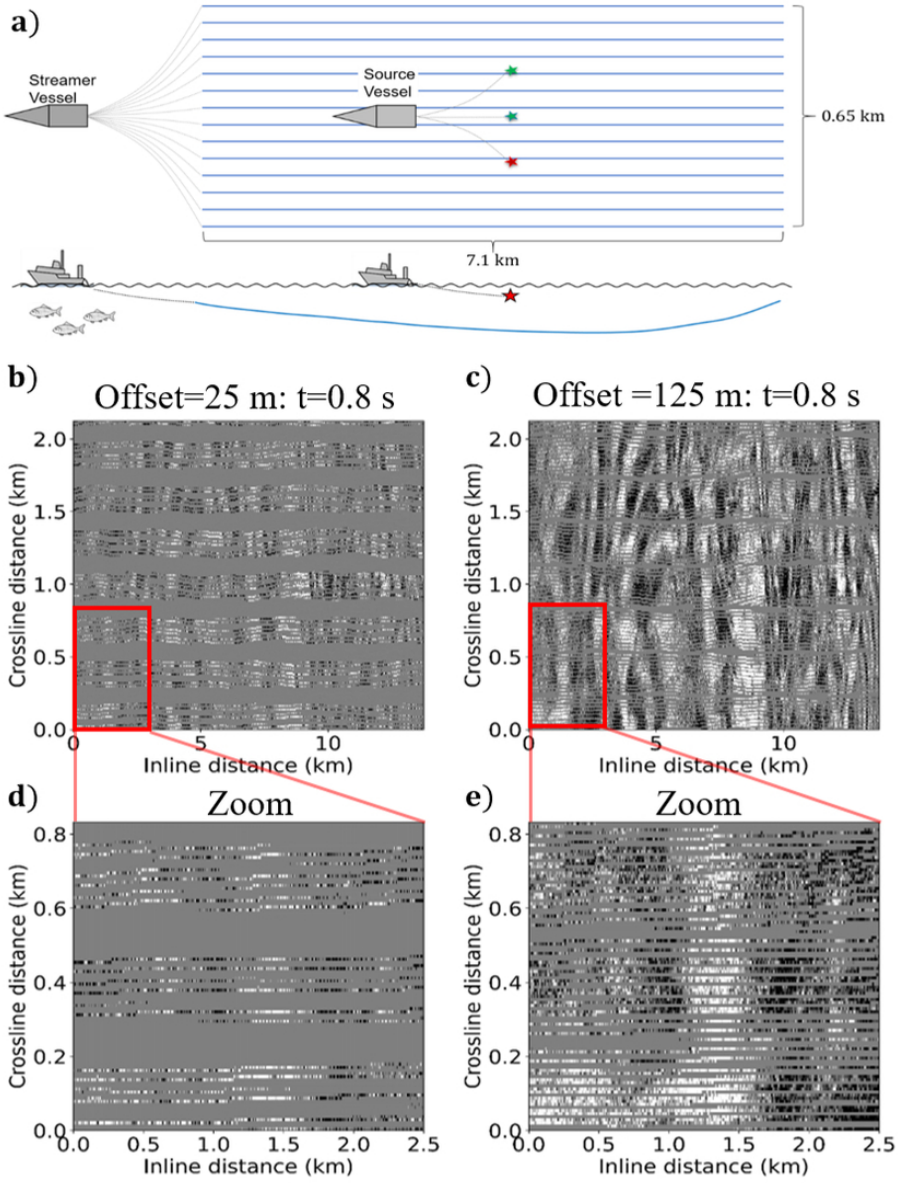

Figure 1: This figure shows an illustration of a **a** source-over-cable survey employing both a streamer vessel and source vessel, and time-slices of a 4D binned wavefield from this type of survey with bin centers at **b** 25 m and **c** 125 m and their zoomed sections **d**-**e**, respectively. Adding near-offsets increases the complexity of the interpolation problem due to an even sparser distribution of traces, where the largest distances are ~160 m in the 25 m offset section.

72x92mm (300 x 300 DPI)

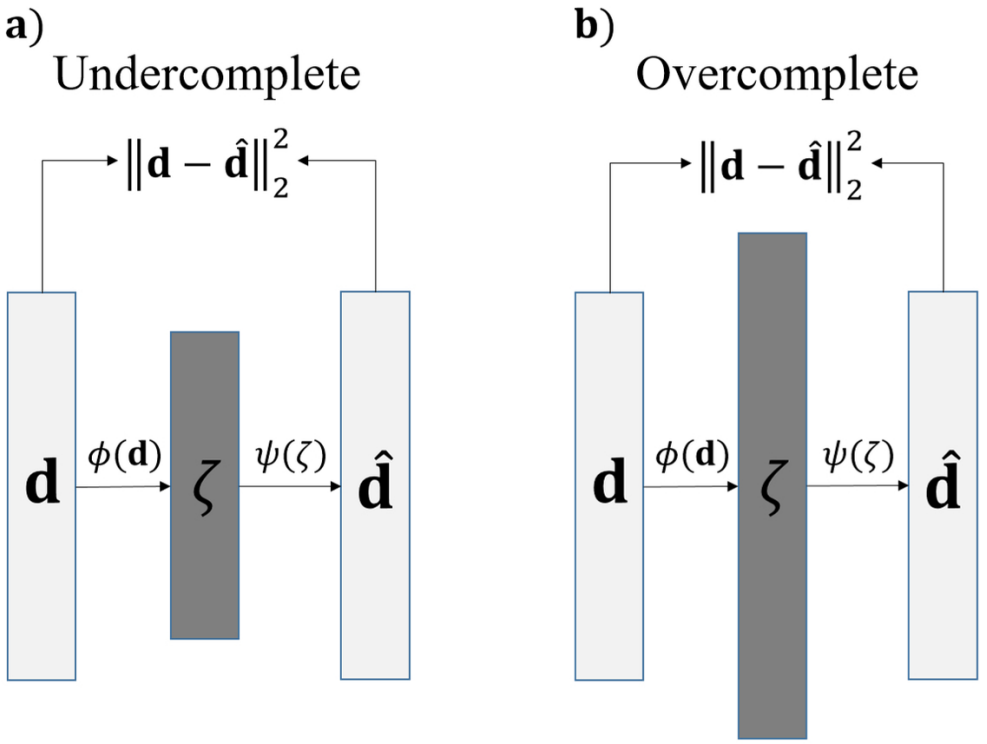

Figure 2: An autoencoder consists of two functional transitions from input $\mathbf{d}\in R^N$ to the output $\mathbf{d}\ \in R^N$. The encoder transition Φ:$\mathbf{d} \rightarrow \zeta$ takes the input and transforms it to a latent feature space $\zeta\in R^K$, followed by the decoder transition Ψ:$\zeta \rightarrow \mathbf{d}$, where the output represents an approximation of the input. In case of **a** undercompleteness we have N > K where the input data is represented on a lower-dimensional space. For **b** overcompleteness we have N ≤ K and thus represents the data on a higher-dimensional space.

97x73mm (300 x 300 DPI)

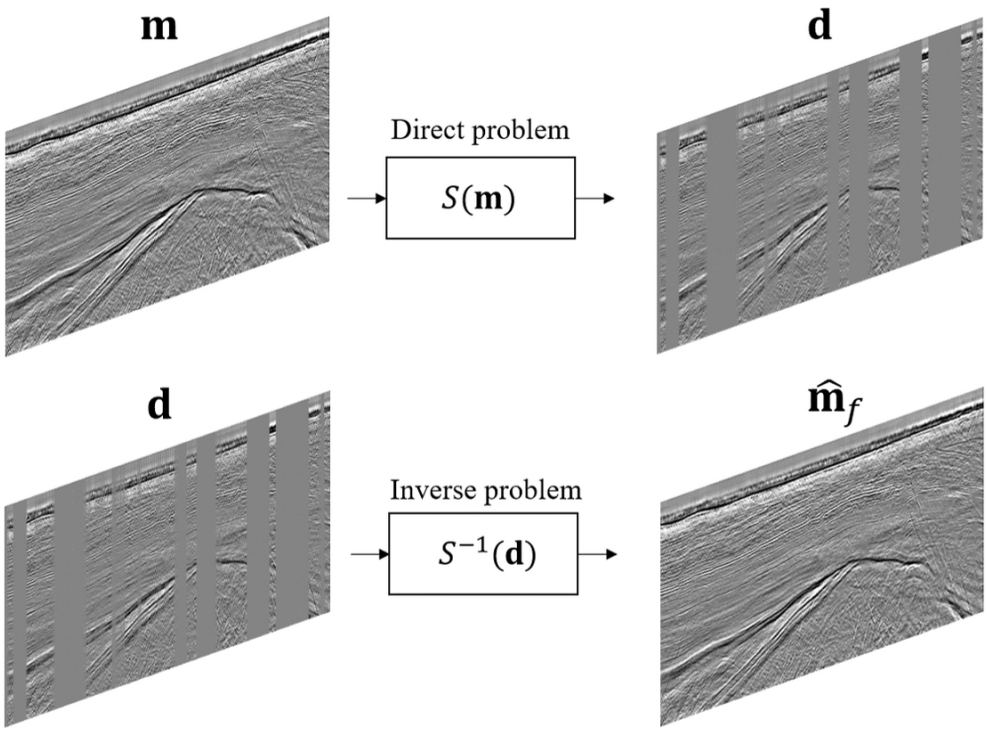

Figure 3: In the direct problem, the transformation S(·) is applied to a complete wavefield **m**. The inverse problem aims to obtain an approximation **m** of the complete wavefield **m** from the observed wavefield **d**. Here, we show the wavefield reconstruction case where S(·) is represented by a masking operator **M**.

95x70mm (300 x 300 DPI)

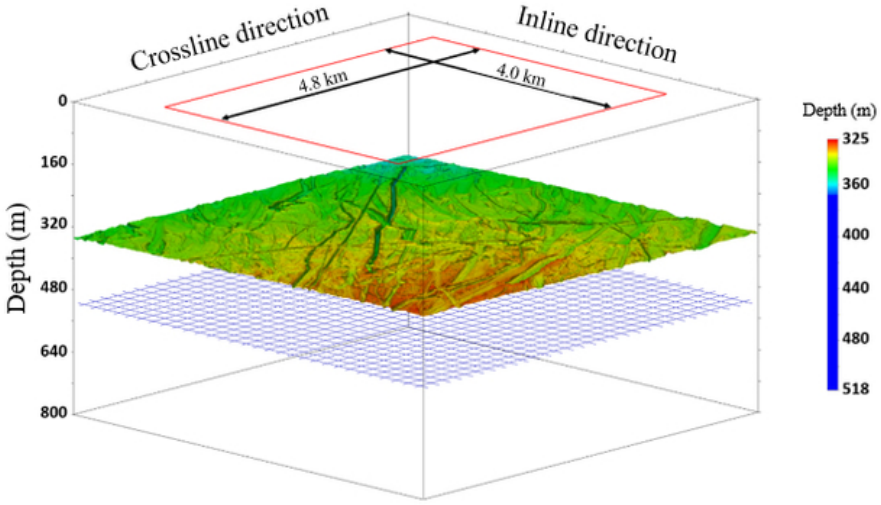

Figure 4: Water bottom model of bathymetry data from side-scan sonar, in addition to a set of diffraction lines beneath. The model is defined on a grid size of 6.25×6.25 m^2. The red rectangle depicts the modelling area.

58x36mm (300 x 300 DPI)

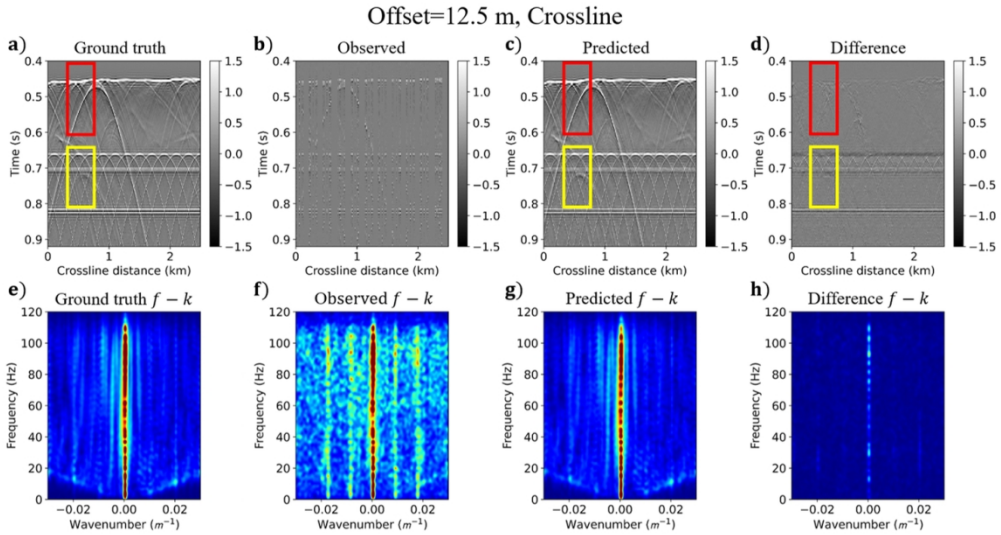

Figure 5: A crossline section from **a** the ground truth, **b** the observed wavefield, **c** the predicted wavefield, **d** the difference between **a** and **c**, and **e**-**h** shows the ground truth, observed, predicted and difference in the *f-k* domain, respectively. The red and yellow boxes indicates the zoomed sections in the wiggle plots shown in Figure 6.

124x66mm (300 x 300 DPI)

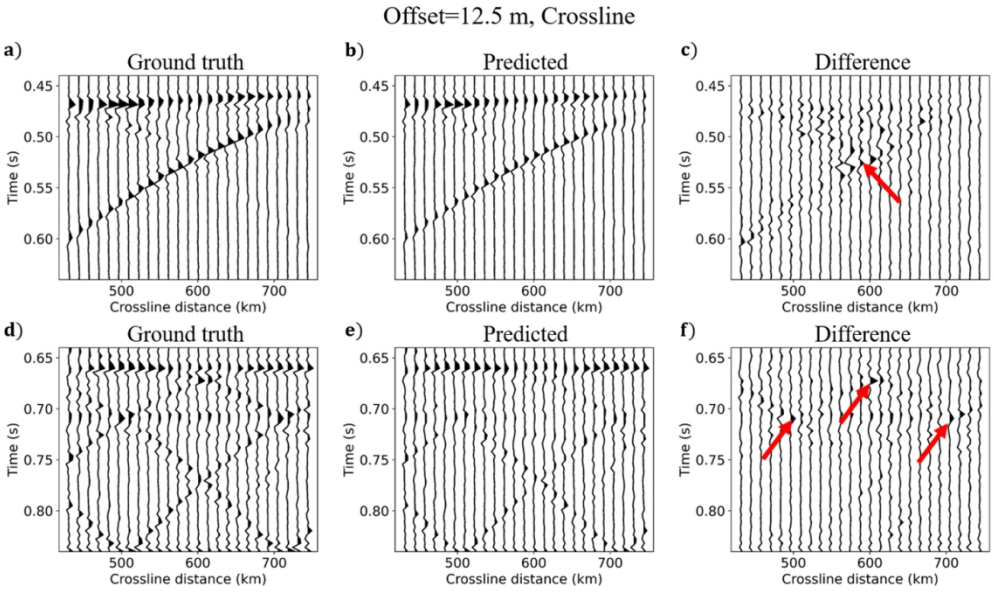

Figure 6: Zoomed section from the **a**,**d** ground truth, **b**,**e** predicted wavefield and **c**,**f** difference from the crossline section in Figure 5 marked by the red and yellow boxes.

123x73mm (300 x 300 DPI)

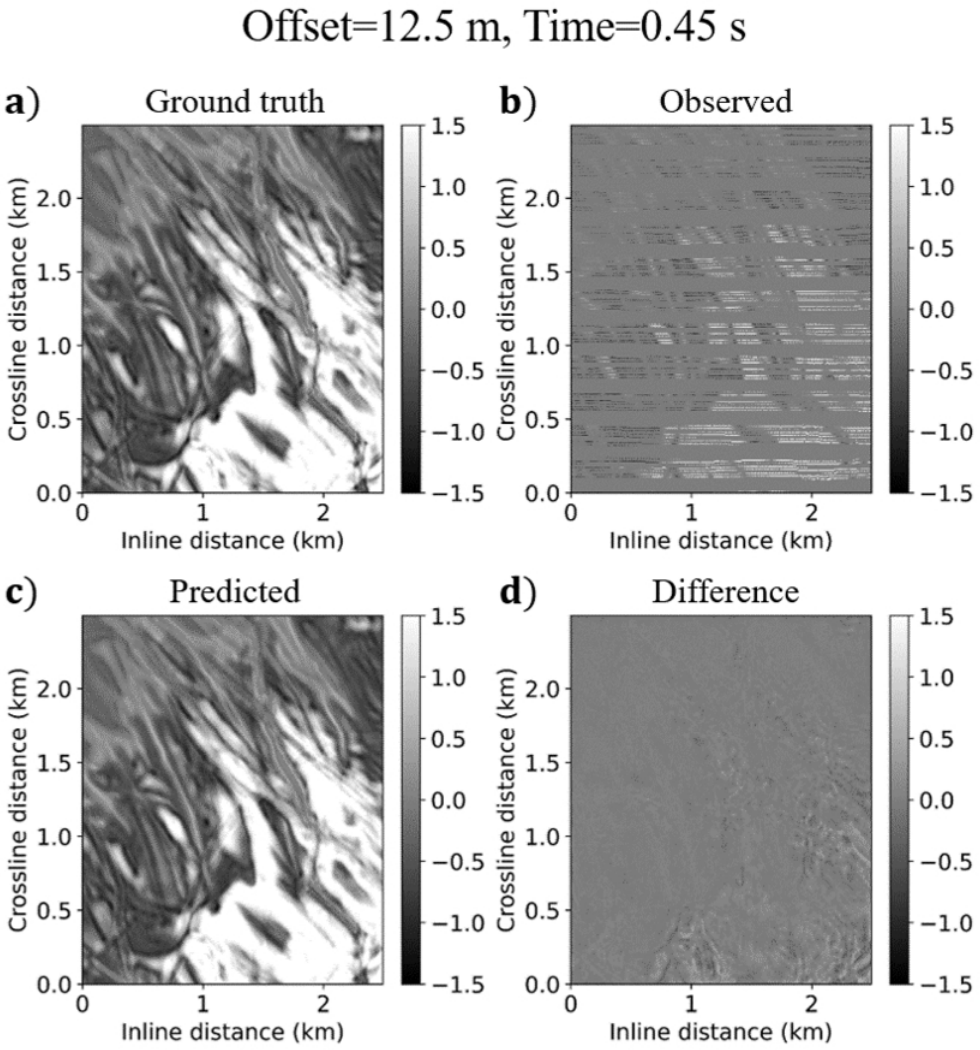

Figure 7: Time-slices from the 12.5 m offset-class, with **a** the ground truth, **b** the observed wavefield, **c** predicted wavefield and **d** the difference between **a** and **c**.

70x74mm (300 x 300 DPI)

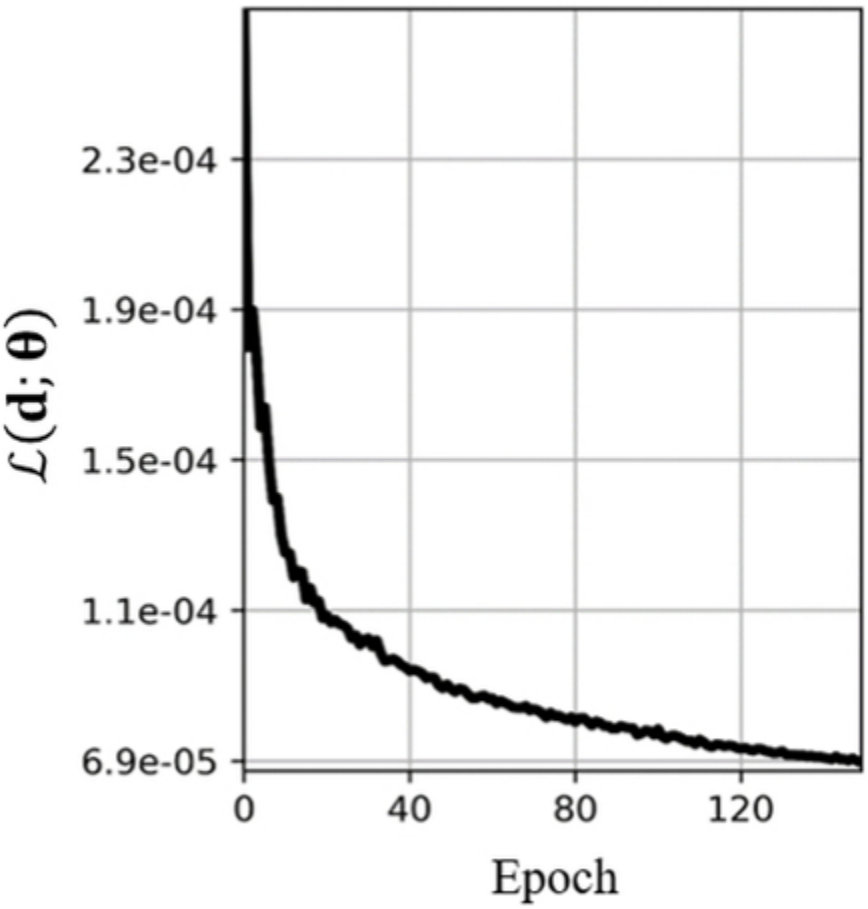

Figure 8: The computed MSE from the data misfit function $L(\mathbf{d};\boldsymbol{\theta})$ during training.

37x38mm (300 x 300 DPI)

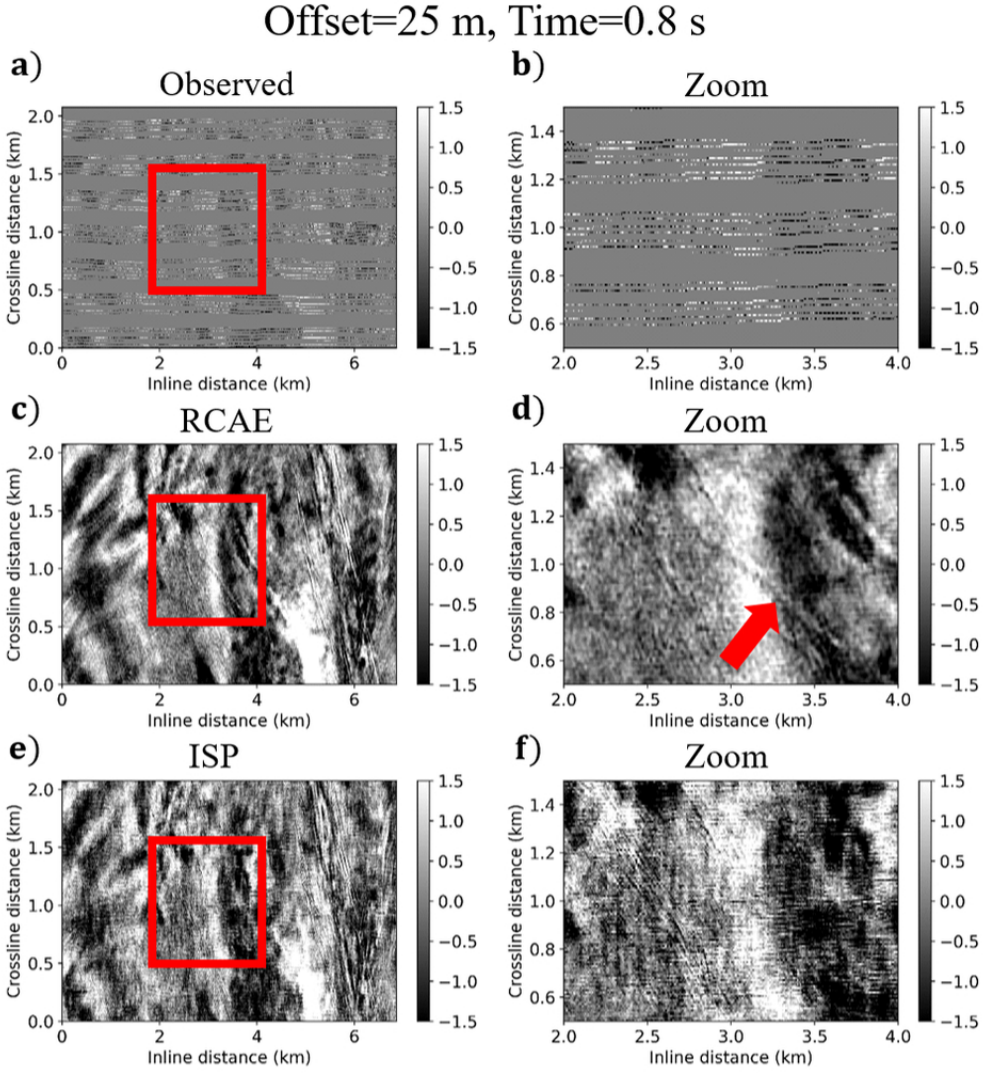

Figure 9: Time-slices from the 25 m offset-class, where **a**-**b** the observed data show a challenging interpolation setting with large gaps between traces. From **c**-**d** the RCAE reconstruction and **e**-**f** the ISP interpolation we observe in general a noisier character within the ISP. In addition, the feature observed within **d** the zoomed RCAE section annotated by the red arrow, is not present in **f** the zoomed ISP section.

79x85mm (300 x 300 DPI)

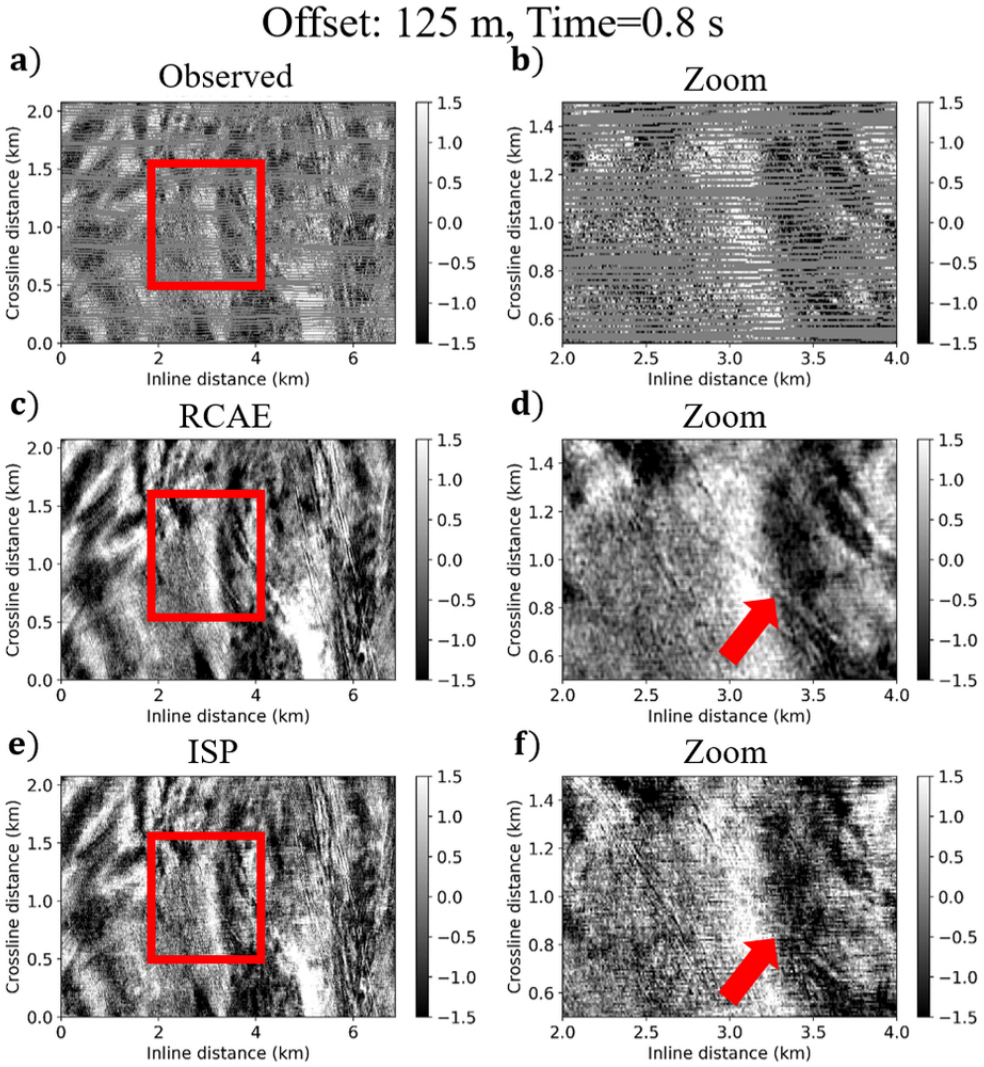

Figure 10: Time-slices from the 125 m offset, where **a**-**b** displays the observed data. From **c**-**d** the RCAE is less noisy compared to **e**-**f** the ISP. However, the ISP interpolation display parts of the feature that was not present in the 25 m offset. The RCAE display this feature in both the 25 m offset and in the 125 m offset.

79x84mm (300 x 300 DPI)

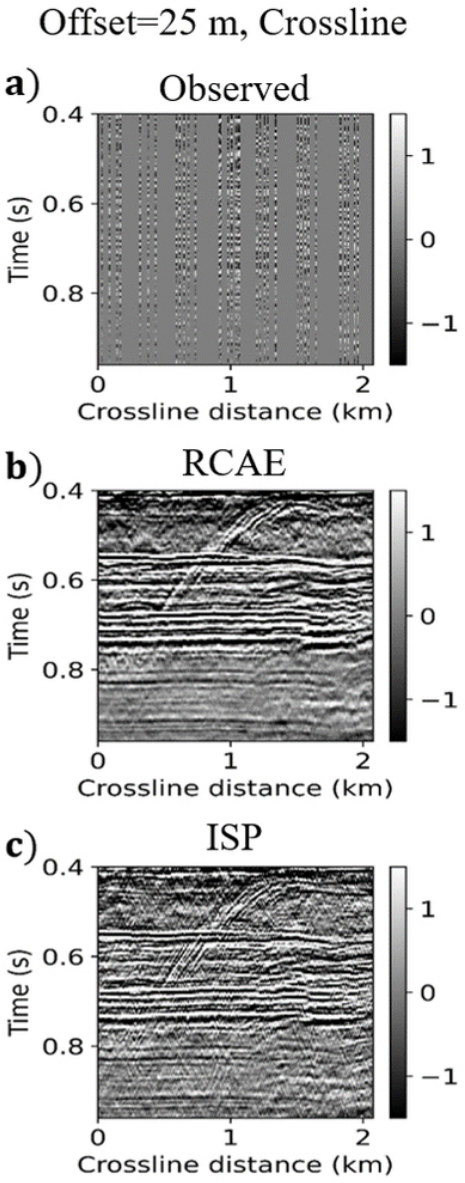

Figure 11: A crossline section from **a** the observed data, **b** the RCAE reconstruction and **c** the ISP. Here, we observe slightly more dipping energy in the ISP compared to the RCAE.

33x85mm (300 x 300 DPI)

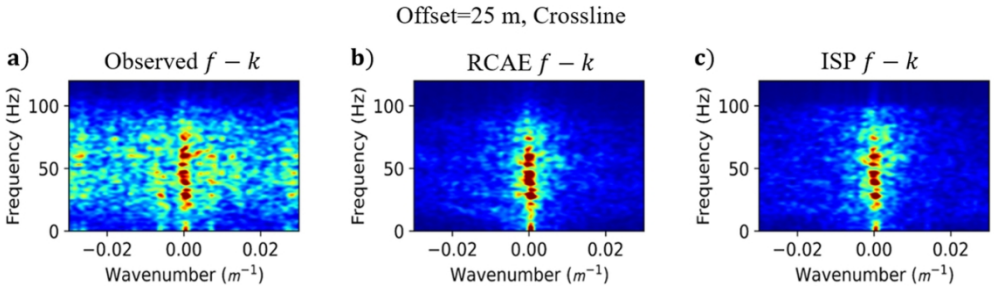

Figure 12: The crossline sections from Figure 7 displayed in the *f-k* domain where **a** is the observed data, **b** the RCAE reconstruction and **c** the ISP.

113x33mm (300 x 300 DPI)

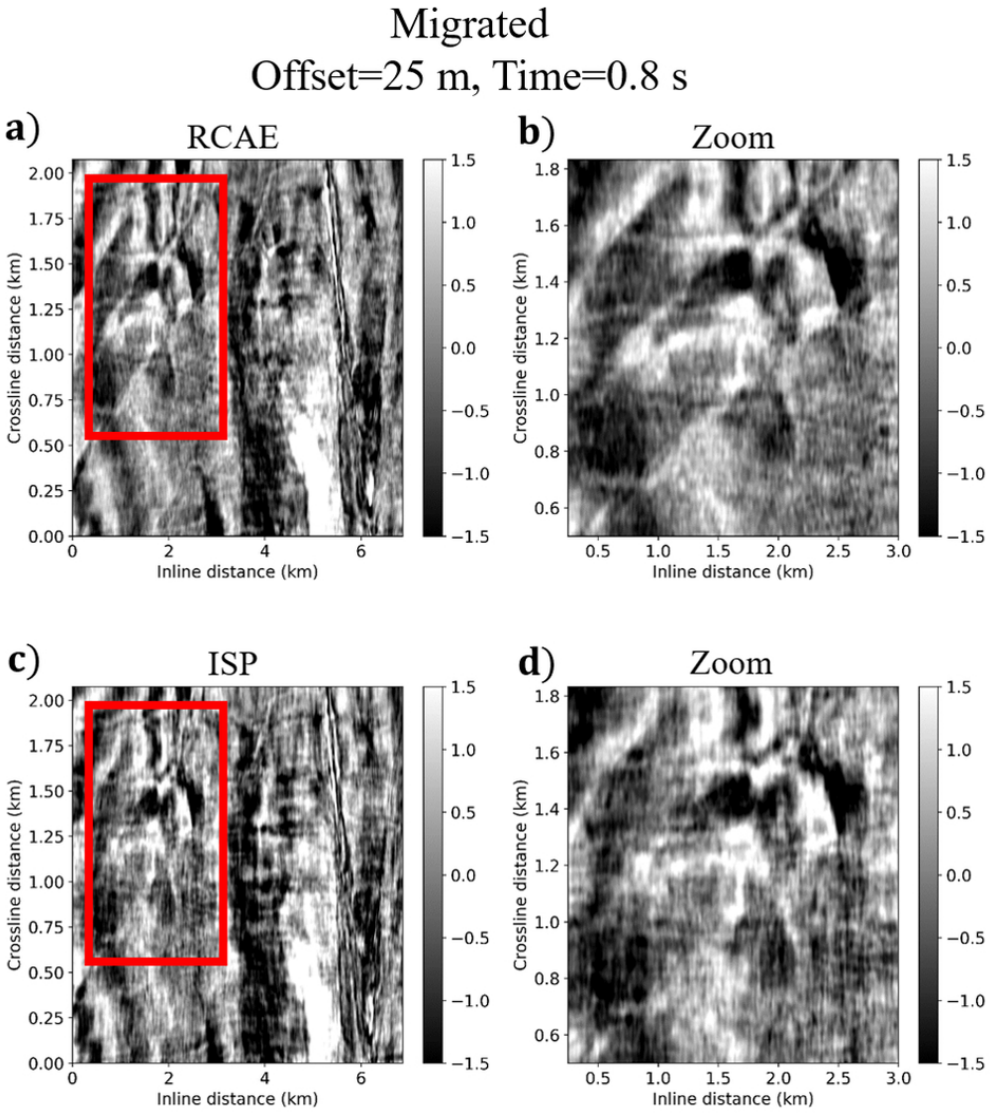

Figure 13: Time-slices after migration of the 25 m offset, where **a** show the RCAE reconstruction and **c** show the ISP, with **b** and **d** as their corresponding zoomed sections, respectively.

80x89mm (300 x 300 DPI)

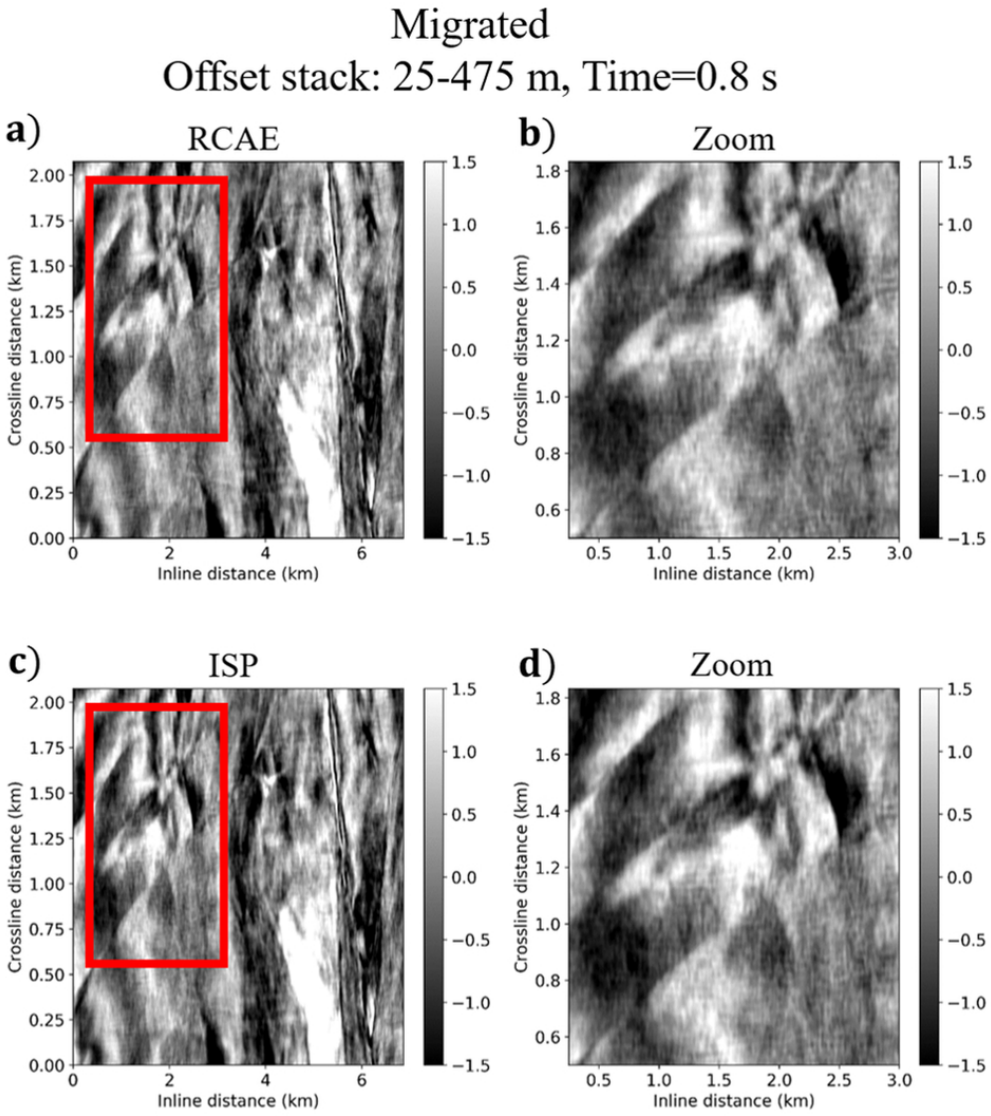

Figure 14: Time-slices of the offset stacked migrated from 25-475 m, where **a** show the RCAE reconstruction and **c** show the ISP, with **b** and **d** as their corresponding zoomed sections, respectively.

80x89mm (300 x 300 DPI)

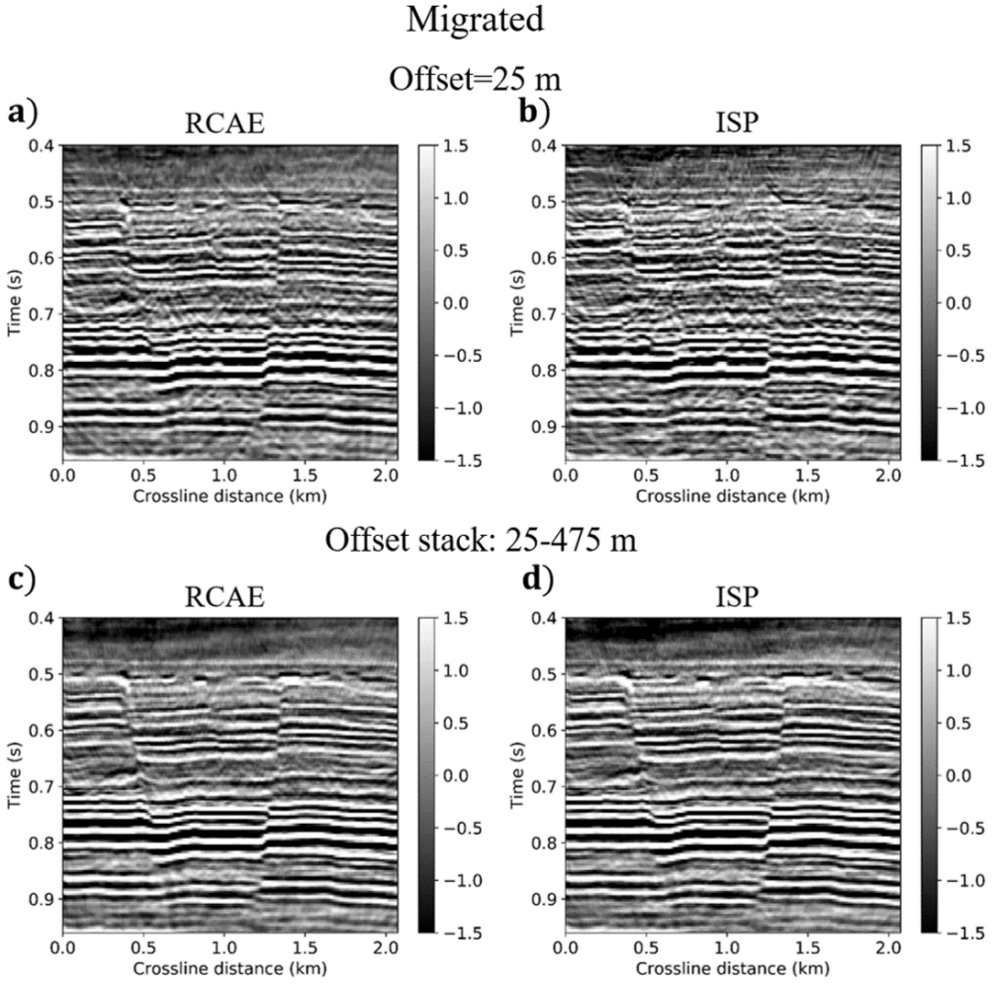

Figure 15: Crossline sections after migration from the 25 m offset and offset stacked from 25-475 m. The migrated 25 m offset from **a** the RCAE reconstruction and **b** from the ISP. The offset stack from 25-475 m are shown in **c** and **d** from the RCAE and ISP, respectively.

92x91mm (300 x 300 DPI)

Table 1: The example CNN used in this study. The number of input features, i.e. offset-classes is given by N_h, where we have used $N_h = 10$.

Layer	Kernel	Convolution	Stride
l=1	$7 \times 7 \times 7 \times N_h \times 90$	Regular	$2 \times 2 \times 2$
l=2	$5 \times 5 \times 5 \times 90 \times 810$	Regular	$2 \times 2 \times 2$
l=3	$3 \times 3 \times 3 \times 810 \times 810$	Regular	$1 \times 1 \times 1$
l=4	$3 \times 3 \times 3 \times 810 \times 90$	Transposed	$2 \times 2 \times 2$
l=5	$3 \times 3 \times 3 \times 90 \times 64$	Transposed	$2 \times 2 \times 2$
l=6	$3 \times 3 \times 3 \times 64 \times 32$	Regular	$1 \times 1 \times 1$
l=7	$3 \times 3 \times 3 \times 32 \times N_h$	Regular	$1 \times 1 \times 1$

DATA AND MATERIALS AVAILABILITY

Data associated with this research are confidential and cannot be released.

DOI:10.1190/geo2021-0099.1