White noise attenuation of seismic data by integrating variational mode decomposition and convolutional neural network

<table>
<thead>
<tr>
<th>Journal:</th>
<th>Geophysics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manuscript ID</td>
<td>GEO-2018-0635.R1</td>
</tr>
<tr>
<td>Manuscript Type</td>
<td>Technical Paper</td>
</tr>
<tr>
<td>Keywords:</td>
<td>noise, neural networks, decomposition</td>
</tr>
<tr>
<td>Area of Expertise:</td>
<td>Signal Processing</td>
</tr>
</tbody>
</table>
White noise attenuation of seismic data by integrating variational mode decomposition and convolutional neural network

ABSTRACT

Seismic noise attenuation is an important step in seismic data processing. Most of the noise attenuation algorithms are based on the analysis of time-frequency features of the seismic data and noise. We propose to attenuate the white noise of seismic data using convolutional neural network (CNN). Traditional CNN based noise attenuation algorithms need prior information (the “clean” seismic data or the noise contained in the seismic) in the training process. However, it is very difficult to obtain such prior information in practice. We assume that the white noise contained in the seismic data can be simulated by user-generated white noise through enough trials. We then propose to attenuate the seismic white noise using the modified denoising convolutional neural network (DnCNN). The modified DnCNN does not need “clean” seismic data nor the noise contained in the seismic data in the training procedure. To accurately and efficiently learn the features of seismic data and white noise at different frequency bandwidths, we propose to decompose the seismic data into several components before the training procedure of the modified DnCNN. There are four main steps in the proposed workflow. The first step is decomposing the seismic volume into different intrinsic mode function (IMFs) and a residual component by using variational mode decomposition (VMD). The second step is adding user-generated white noise to each decomposed component. The third step is building the neural network hierarchy to learn the feature of additive white noise. The last step is obtaining denoised seismic data by applying the well-trained network to the original seismic data. We use both synthetic and field data examples to illustrate the robustness and superiority of the proposed method.

Keyword: Noise attenuation, CNN, VMD
INTRODUCTION

Seismic noise attenuation is a key step to enhance the quality of seismic data. Seismic denoising not only lowers the effect of subjectivity in seismic interpretation but also improve the reliability of seismic inversion. In recent decades, numerous seismic denoising approaches have been developed and widely applied in practice. Methods for seismic denoising can be generally classified into four categories: The first category is based on building a prediction filter to remove the noise of seismic data. The commonly used algorithms within the first category includes $f$-$x$ predictive filtering (Canales, 1984), $t$-$x$ predictive filtering (Abma et al., 1995), the forward-backward prediction approach (Wang, 1999), the polynomial fitting-based approach (Liu et al. 2011), and non-stationary predictive filtering (Liu et al., 2012). The second category projects the seismic data to a transformed domain and rejects the noise by applying a bandpass filter to the transformed data. Finally, obtain the denoised seismic data by projecting the filtered data back to the time domain. The commonly used algorithms within the second category include Fourier transform (Chen et al., 2014), curvelet transform (Herrman et al., 2008), seislet transform (Fomel et al., 2010), shearlet transform (Kong et al., 2015), Radon transform (Trad et al., 2002; Xue et al., 2016), wavelet transform (Donoho et al., 1994), and dictionary learning based sparse transform (Elad et al., 2006). The third category decomposes the seismic traces into a set of components. Then, examine the time or frequency features of each decomposed component. Finally, obtain the “clean” seismic traces by rejecting the components which are regarded as “noise”. The commonly used algorithms within the third category include empirical mode decomposition (EMD) (Huang et al., 1998; Bekara et al., 2009), variational mode decomposition (VMD) (Dragomiretskiy et al., 2014; Li et al., 2017), and singular value decomposition (SVD) based approaches (Bekara et al., 2007). Yuan et al. (2018) proposed a novel inversion-based denoising method. The method has
the advantages of preserving 3-D spatial edges and low-frequency signals. The fourth category is based on the rank-reduction reconstruction of seismic data. The commonly used algorithms within the fourth category include Cadzow filtering (Trickett, 2008), and singular spectrum analysis (Vautard et al., 1992; Oropeza et al., 2011).

Deep learning is a subset of machine learning that based on learning data representation. Convolutional neural network (CNN) (LeCun et al., 1998) is one of the most popular and widely used deep learning algorithms. The CNN based algorithms already achieve great success in the field of computer vision. CNN is extremely efficient in learning the features of the images and labeling the objectives in the images. A lot of CNN based algorithms also have been proposed to address the problem of image denoising. Jain et al. (2009) successfully applied CNN to images denoising. Burger et al. (2012) denoise the images using the multiple layer perceptron (MLP). Other popular CNN based image denoising methods include stacked sparse denoising auto encoder (Xie et al., 2012) and trainable nonlinear reduction diffusion (TNRD) model (Chen et al., 2015). Zhang et al. (2017) proposed denoise convolutional neural network (DnCNN) to learn the feature of noise contained in the images. The main disadvantage of current CNN-based denoising methods is that these methods need “clean” data and the corresponding noisy data in the training process. Unfortunately, it is unfeasible to obtain “clean” seismic data for training in practice.

In this paper, we propose a novel seismic noise attenuation method by integrating VMD with the modified DnCNN, which is named as VMD-DnCNN. We organized this paper as follow: we first describe the improvement for DnCNN. We then detailly introduce the workflow of white noise attenuation in this paper. We finally use both synthetic and field data examples to illustrate the robustness and superior performance of VMD-DnCNN over f-x deconvolution.

Theory
There are a lot of successful applications of images denoising by using CNN based algorithms (Xie et al., 2012; Zhang et al., 2017). The main advantage of CNN based denoising methods is that CNN with deep architecture (multiple hidden layers) can recognize various features of the input data and classify the recognized features into corresponding categories.

**Objective function**

The image which needs to be denoised can be defined as \( y = x + n \), where \( y \) is the noisy image, \( x \) is the corresponding clean image, and \( n \) is the additive noise. The goal of image denoising is building a model to recover the clean image \( x \) from the corresponding noisy image \( y \). According to the objective function, the image denoising methods using CNN can be classified into two categories.

The first category is modeling the “clean” image (Jain et al., 2009; Xie et al., 2012) by minimizing the following objective function \( J(\theta) \):

\[
J(\theta) = \arg \min_{\theta} \frac{1}{M} \sum_{i=1}^{M} \left\| x_i - R_\theta(y_i) \right\|^2
\]  

(1)

where \( R_\theta \) denotes the entire convolutional neural network with all the trainable parameters \( \theta \), \( R_\theta(y_i) \) is the predicted clean image by using the trained convolutional neural network \( R_\theta \).

The second category is modeling the “noise” by applying the residual learning formulation (Zhang et al., 2017):

\[
J(\theta) = \arg \min_{\theta} \frac{1}{M} \sum_{i=1}^{M} \left\| R_\theta(y_i) - (y_i - x_i) \right\|^2
\]  

(2)

where \( R_\theta(y_i) \) is the predicted noise by using the trained convolutional neural network \( R_\theta \).

In geophysics field, \( x \) and \( y \) can be regarded as the noise-free seismic trace and noise contaminated seismic trace, respectively. Both equations 1 and 2 require clean and the
corresponding noisy data in the training process. Unfortunately, it is unfeasible to obtain the purely

clean seismic data in practice. However, we assume that the white noise contained in the seismic
data can be simulated by user-generated white noise, \( n' \), through enough trials (Wu and Huang,
2005). The new “more” noisy seismic trace can be expressed as

\[
y' = x + n + n'
\]  

(3)

The new objective function of DnCNN is given by:

\[
J(\theta) = \arg\min_{\theta} \frac{1}{M} \sum_{i=1}^{M} \left[ R_{\theta}(y_i) - (y'_i - y_i) \right]^2
\]  

(4)

\[
J(\theta) = \arg\min_{\theta} \frac{1}{M} \sum_{i=1}^{M} \left[ R_{\theta}(y_i) - n'_i \right]^2
\]  

(5)

According to the statistical properties of white noise, the distribution of the original white noise \( n \)
and the additive white noise \( n' \) are given by:

\[
n_i \sim \mathcal{N}(\mu_1, \sigma_1^2), n'_i \sim \mathcal{N}(\mu_2, \sigma_2^2)
\]  

(6)

where \( \mathcal{N} \) denotes the normal distribution, \( \mu_1 \) is the expectation and \( \sigma_1 \) is the standard deviation of
the noise contained in the seismic data, \( \mu_2 \) is the expected value and \( \sigma_2 \) is the standard deviation
of the additive white noise. Equation 6 illustrates that we can simulate the noise contained in the
seismic data if we have sufficient trials. To accurately simulate the noise contained in the seismic
data, the noise level of additive white noise should be close to the noise level of original white
noise. We employ the peak signal noise ratio method to compute the signal to noise ratio (SNR)
for the input seismic data:

\[
SNR \approx PSNR = \frac{\max(y^2)}{MSE}
\]  

(7)
\[ MSE = \frac{1}{M} \sum_{i} (y_i - \hat{y})^2, \]  

where \( \hat{y} \) denotes the mean value of the original seismic data and MSE is the mean squared error.

After sufficient trials of adding additive white noise, the expectation \( \mu_2 \) and standard deviation \( \sigma_2 \) of certain realization of simulated noise should approximately equal to the expectation \( \mu_1 \) and standard deviation \( \sigma_1 \) of noise contained in the seismic data:

\[ \mu_2 \approx \mu_1, \sigma_2 \approx \sigma_1 \]  

Then, we obtain:

\[ J(\theta) = \arg \min_{\theta} \frac{1}{M} \sum_{i=1}^{M} \left\| R_{\theta}(y_i) - n_i \right\|^2 \approx \arg \min_{\theta} \frac{1}{M} \sum_{i=1}^{M} \left\| R_{\theta}(\hat{y}_i) - n_i \right\|^2, \]

Equation 10 indicates that the proposed method does not require clean seismic data in the training process.

**Architecture**

The architecture of the proposed neural network is a sequence of nonlinear processing layers and followed by a sigmoid classifier layer that based on the architecture of DnCNN (Figure 1). The input of the network is the original seismic data \( y \) and the “more” noisy data \( y' = y + n' \).

The network in our paper contains 17 layers in total. The first layer contains 64 convolution filters of size 3×3 and 64 rectified linear units (ReLU) activation operator. The objective of the convolution filter is to generate feature maps of the input seismic data. The objective of the ReLU is to activate the main features contained in the feature map. Different from the first layer, a batch-normalization (BN) (Ioffe and Szegedy, 2015) is added between the convolution filter and ReLU for the following 2~16 layers. Batch-normalization is a re-parametrization aimed to stabilize the parameters updating and improve the learning process. The last layer only contains 64 convolution
filters to reconstruct the output and the size of each filter is 3×3. Then the built neural network transforms the seismic noise attenuation procedure into an optimization problem by solving a sequence of nonlinear functions. A gradient-based optimization algorithm of adaptive moment estimation (Adam) (Kingma et al., 2015) is employed to minimize the proposed objective function through iterative updating the parameters of the network.

**VMD-DnCNN**

Figure 2 shows a real seismic section of F3-block that acquired from the North Sea, Netherlands. Figures 3a and 3b show the denoised result and the rejected noise by using the modified DnCNN. Note the noise indicated by the yellow arrows in Figure 3a and rejected visible reflections indicated by the red arrows in Figure 3b. The main reason for this phenomenon is that seismic data are band-limited and the noise to signal ratio (SNR) is varying with bandwidths. As a result, it is very difficult for the modified DnCNN to learn the feature of original white noise at full bandwidth. To better learn the noise feature, we propose to first decompose the seismic data into different components and then apply the modified DnCNN to each decomposed component. In this paper, we apply the variational mode decomposition (VMD) to decompose the seismic traces. We named the proposed seismic noise attenuation procedure as VMD-DnCNN.

Variational Mode Decomposition (VMD) is an adaptive and non-recursive signal decomposition method (Dragomiretskiy and Zosso, 2013). VMD decompose a signal into a series of intrinsic mode function (IMF) where IMFs have the sparsity properties in the frequency domain. The frequency spectrum of every IMF is compacted around the center frequency \( \omega_i \) and the sparsity of every IMF is constrained by its bandwidth in the frequency domain. In other words, VMD decomposes the signal into different IMFs and the amplitude spectrum of each mode
exponential tuned around the center frequency \( \omega_i \). We obtain each IMF by recursively solving the following optimization problem:

\[
\min_{\{u_i, \omega_i\}} \left\{ \frac{1}{N} \sum_i \left( \left| \hat{\phi}_i \left[ \delta(t) + \frac{j}{\pi t} \right] * u_i(t) \right| - j \omega_i t \right)^2 \right\},
\]

(11)

\[
\sum_i u_k = s(t)
\]

Where \( u_i \) and \( \omega_i \) are modes and their center frequencies, respectively, \( \delta(t) \) is a Dirac impulse, \( s(t) \) is the signal to be decomposed, the constraint condition is that the summation over all modes should be the input signal, the term \( (\delta(t) + \frac{j}{\pi t}) * u_i(t) \) is the Hilbert transform of \( u_i \), and the parameters \( N \) is the user defined decomposed number of IMFs.

Then the denoising objective function, \( J(\theta)^{(j)} \), for the \( j \)th decomposed seismic component is given as:

\[
J(\theta)^{(j)} = \arg \min_{\theta} \frac{1}{M} \sum_{i=1}^{M} \left\| \begin{bmatrix} y_i^{(j)} \\ y_i^{(j)} \end{bmatrix} - \begin{bmatrix} y_i \\ y_i \end{bmatrix} \right\|^2,
\]

(12)

where \( y^{(j)} \) and \( y^{(j)} \) represent the \( j \)th decomposed component and the components with additive white noise, respectively.

Figure 4 shows the proposed denoising workflow using VMD-DnCNN. We first decompose the seismic data (the 3D seismic data have been reshaped to 2D arrays) into several decomposed components and compute the SNR for each component. We produce a “noisier” IMF by adding additive white noise to each decomposed IMF and residual component. The energy of the additive white noise approximately equals to the energy of the white noise estimated within each decomposed component. We next learn the feature of white noise by minimizing the
difference between additive noise and learned white noise from the “noisier” decomposed components (Equation 12). We produce the denoised components by subtracting the learned noises from the corresponding decomposed components. We finally obtain the denoised seismic data by integrating the denoised components. Figures 5a and 5b show the denoised results and rejected noise by using VMD-DnCNN. Note that our method successfully rejects the noise indicated by the yellow arrows in Figure 3a and preserves the seismic reflections indicated by the red arrows in Figure 3b.

**Synthetic Example**

To demonstrate the performance of VMD-DnCNN, we first test our method using synthetic seismic data (Figure 6a) generated using the Marmousi model. We use zero-phase Ricker wavelet to generate our synthetic seismic data. The dominant frequency of the Ricker wavelet is 30 Hz. The synthetic seismic data contains 128 traces. Each trace has 128 time samples and the time sample interval is 4ms. Figure 6b shows the noisy synthetic seismic data. The additive noise is Gaussian noise and the SNR is 2. To ensure that the additive noise has the same frequency bandwidth with the seismic data, we applied a band pass Butterworth filter (5-10-95-100Hz) to the Gaussian noise before we add the noise to the noise-free synthetic seismic data. Figures 7a and 7b show the denoised seismic data using f-x deconvolution and VMD-DnCNN, respectively. Figures 8a and 8b show the rejected noise using f-x deconvolution and VMD-DnCNN, respectively. Note that the VMD-DnCNN not only rejects the white noise (the yellow arrows in Figures 7a and 7b) but also preserves the visible reflections rejected by f-x deconvolution (the red arrows in Figure 8a and 8b).

The filter length is 12 and the cutoff frequency range is 5-100Hz for the f-x deconvolution. The IMFs number is determined by the performance of denoising and computation cost. In this
paper, we found that there is no obvious difference between the denoised results both for the synthetic and real seismic data if the IMFs number is equal or greater than 2. The computation cost increases with the increasing of IMFs number. We set the IMFs number as 2 in our synthetic testing. The moderate bandwidth constraint and the tolerance of convergence criterion are 100 and 0.01 for VMD decomposition, respectively. The input for our VMD-DnCNN are the first decomposed IMF plus additive noise, the second decomposed IMF plus additive noise, and the residual component plus additive noise. Figure 9 shows the training and validation loss varying with optimization epochs (Considering that the data size used in deep learning is huge, we usually divide the learning data set into several small subsets (batch). The optimization procedure is implemented batch by batch and one epoch means one optimization pass of the full batches.). We obtain the training and validation loss by applying the objective function shown in Equation 12 to the training and validation data set, respectively. To overcome the overfitting problem in the training procedure, the training and validation seismic traces are randomly selected during each optimization epoch. The percentage of the training and validation seismic traces in this paper is 70% and 30% both for the synthetic and real seismic data, respectively. A certain seismic trace may belong to training seismic traces set in current optimization epoch but may belong to validation seismic traces set in the next optimization epoch. Figure 9 illustrates that we obtain a stable neural network hierarchy after 50 epochs in the training procedure.

Figure 10 shows the average amplitude spectrum of the original seismic (red), the denoised result using $f$-$x$ deconvolution (blue), and the denoised result using VMD-DnCNN (black), respectively. Note that the amplitude spectrum of denoised result using VMD-DnCNN has a very good match with that of original seismic data. Unfortunately the denoised result using $f$-$x$ deconvolution lost certain high frequency content when compared that of original seismic data.
Field data example

We further apply VMD-DnCNN to a public seismic survey (Penobscot) to illustrate the effectiveness of our proposed method. The Penobscot seismic survey was acquired over Scotian shelf, oversea Canada. The seismic survey contains 601 inlines and 482 crosslines. The time increment of the seismic survey is 4ms. We observe both residual noise and possible migration artifacts indicated by the yellow arrows in Figure 11.

We first use VMD to decompose the original seismic data into two IMFs and a residual volume. Based on the testing of real data application, we found that we can successfully simulate the noise contained in the seismic data if the trial number of white noise is above 2000 times. In this paper, we choose 2000 as the trail number of white noise simulation. The simulated noise is then added to the decomposed two IMFs and a residual component. Figure 12 illustrates that we obtain a stable neural network hierarchy after 60 epochs in the training procedure.

Figures 13a and 13b show the denoised results using f-x deconvolution and VMD-DnCNN, respectively. Figures 14a and 14b show the difference between the original seismic data and denoised results using f-x deconvolution and VMD-DnCNN, respectively. Note that the VMD-DnCNN not only reject the white noise and migration artifact indicated by yellow arrows in Figure 13a but also preserves the visible reflections rejected by the f-x deconvolution indicated by red arrows in Figure 14a. Figures 15, 16a and 16b show the 3D cube of original seismic data, denoised seismic data using f-x deconvolution and denoised seismic data using VMD-DnCNN, respectively. Figures 17a and 17b show the 3D rejected noise using f-x deconvolution and VMD-DnCNN, respectively. Note that the VMD-DnCNN have successfully rejected most of the white noise and migration artifacts indicated by yellow arrows in Figure 15. However, the denoised results using f-x deconvolution still contain white noise and migration artifacts indicated by yellow arrows in
Figure 16a. Figures 17a and 17b indicates that our proposed method not only attenuate the white noise but also preserve most of the useful seismic amplitude indicated by red arrows.

Again we further compare the spectrum of the original and denoised seismic data to show the effectiveness of our method. Figure 18 shows the amplitude spectrum of the original noisy data (red), denoised data using $f$-$x$ deconvolution (blue) and denoised data using VMD-DnCNN (black), respectively. Note that the average amplitude spectrum of denoised data using VMD-DnCNN has a very good match with that of original seismic data. Unfortunately the denoised result using $f$-$x$ deconvolution lost certain high frequency content when compared that of original seismic data.

**Conclusion**

We propose a novel seismic noise attenuation method (VMD-DnCNN) by integrating our modified DnCNN with VMD. The current CNN based denoising methods either require the label of clean seismic data or the label of noise contained in the seismic data. However, our method does not require clean seismic label nor purely noise label. The applications demonstrate that the white noise contained in the seismic can be simulated by user-generated white noise if we have enough trials. In addition, the applications demonstrate that the modified DnCNN can obtain a more accurate estimation of the noise feature from the decomposed bandlimited seismic data. Both synthetic and real seismic data applications illustrate that our method is superior to the denosing method using $f$-$x$ deconvolution. The applications also demonstrate that our method not only effectively reject the white noise but also the migration artifacts contained in the seismic data.
List of Figures

Figure 1. The overall architecture of DnCNN.

Figure 2. One section example of F3-block seismic survey.

Figure 3. Illustration of the test on F3-block using modified DnCNN. (a) the denoised result using modified DnCNN. (b) the rejected noise using modified DnCNN.

Figure 4. The workflow of VMD-DnCNN for seismic noise attenuation.

Figure 5. Illustration of the test on F3-block using VMD-DnCNN. (a) the denoised result using VMD-DnCNN. (b) the rejected noise using VMD-DnCNN.

Figure 6. (a) The noise-free synthetic data of marmousi model. (b) The noisy synthetic data of marmousi model.

Figure 7. Illustration of the test on synthetic example. (a) The denoised result using $f$-$x$ deconvolution. (b) The denoised result using VMD-DnCNN.

Figure 8. Rejected noise through different denoising methods on synthetic example. (a) The rejected noise using $f$-$x$ deconvolution. (b) The rejected noise using VMD-DnCNN.

Figure 9. Train loss and valid loss varying with different epoch for synthetic example.

Figure 10. The frequency spectrum of the original data (red curve), VMD-DnCNN result (black curve) and $f$-$x$ deconvolution result (blue curve) for the synthetic test.

Figure 11. One representative inline section example of Penobscot.

Figure 12. Train loss and valid loss varying with different epoch for the seismic survey of Penobscot.
Figure 13. Illustration of the seismic noise attenuation test on the seismic survey of Penobscot. (a) The denoised result using f-x deconvolution. (b) The denoised result using VMD-DnCNN.

Figure 14. Rejected noises through different denoising methods on the seismic survey of Penobscot. (a) The rejected noise using f-x deconvolution. (b) The rejected noise using VMD-DnCNN.

Figure 15. The original 3D seismic data of Penobscot.

Figure 16. Illustration of the 3D volume denoised result on the seismic survey of Penobscot. (a) The denoised result using f-x deconvolution. (b) The denoised result using VMD-DnCNN.

Figure 17. Rejected noises on 3D volume through different denoising methods on the seismic survey of Penobscot. (a) The rejected noise using f-x deconvolution. (b) The rejected noise using VMD-DnCNN.

Figure 18. The amplitude spectrum of the original data (red curve), VMD-DnCNN result (black curve) and f-x deconvolution result (blue curve) for the seismic survey of Penobscot.
Reference


Figure 1. The overall architecture of DnCNN.

254x190mm (300 x 300 DPI)
Figure 2. One section example of F3-block seismic survey.

254x190mm (300 x 300 DPI)
Figure 3. Illustration of the test on F3-block using modified DnCNN. (a) the denoised result using modified DnCNN. (b) the rejected noise using modified DnCNN.
Figure 3. Illustration of the test on F3-block using modified DnCNN. (a) the denoised result using modified DnCNN. (b) the rejected noise using modified DnCNN.
Figure 4. The workflow of VMD-DnCNN for seismic noise attenuation.
Figure 5. Illustration of the test on F3-block using VMD-DnCNN. (a) the denoised result using VMD-DnCNN. (b) the rejected noise using VMD-DnCNN.

254x190mm (300 x 300 DPI)
Figure 5. Illustration of the test on F3-block using VMD-DnCNN. (a) the denoised result using VMD-DnCNN. (b) the rejected noise using VMD-DnCNN.

254x190mm (300 x 300 DPI)
Figure 6. (a) The noise-free synthetic data of marmousi model. (b) The noisy synthetic data of marmousi model.

254x190mm (300 x 300 DPI)
Figure 6. (a) The noise-free synthetic data of marmousi model. (b) The noisy synthetic data of marmousi model.

254x190mm (300 x 300 DPI)
Figure 7. Illustration of the test on synthetic example. (a) The denoised result using f-x deconvolution. (b) The denoised result using VMD-DnCNN.

254x190mm (300 x 300 DPI)
Figure 7. Illustration of the test on synthetic example. (a) The denoised result using f-x deconvolution. (b) The denoised result using VMD-DnCNN.

254x190mm (300 x 300 DPI)
Figure 8. Rejected noise through different denoising methods on synthetic example. (a) The rejected noise using f-x deconvolution. (b) The rejected noise using VMD-DnCNN.
Figure 8. Rejected noise through different denoising methods on synthetic example. (a) The rejected noise using f-x deconvolution. (b) The rejected noise using VMD-DnCNN.
Figure 9. Train loss and valid loss varying with different epoch for synthetic example.
Figure 10. The frequency spectrum of the original data (red curve), VMD-DnCNN result (black curve) and f-x deconvolution result (blue curve) for the synthetic test.
Figure 11. One representative inline section example of Penobscot.

254x190mm (300 x 300 DPI)
Figure 12. Train loss and valid loss varying with different epoch for the seismic survey of Penobscot.

254x190mm (300 x 300 DPI)
Figure 13. Illustration of the seismic noise attenuation test on the seismic survey of Penobscot. (a) The denoised result using f-x deconvolution. (b) The denoised result using VMD-DnCNN.

254x190mm (300 x 300 DPI)
Figure 13. Illustration of the seismic noise attenuation test on the seismic survey of Penobscot. (a) The denoised result using f-x deconvolution. (b) The denoised result using VMD-DnCNN.

254x190mm (300 x 300 DPI)
Figure 14. Rejected noises through different denoising methods on the seismic survey of Penobscot. (a) The rejected noise using f-x deconvolution. (b) The rejected noise using VMD-DnCNN.

254x190mm (300 x 300 DPI)
Figure 14. Rejected noises through different denoising methods on the seismic survey of Penobscot. (a) The rejected noise using f-x deconvolution. (b) The rejected noise using VMD-DnCNN.
Figure 15. The original 3D seismic data of Penobscot.

254x190mm (300 x 300 DPI)
Figure 16. Illustration of the 3D volume denoised result on the seismic survey of Penobscot. (a) The denoised result using f-x deconvolution. (b) The denoised result using VMD-DnCNN.

254x190mm (300 x 300 DPI)
Figure 16. Illustration of the 3D volume denoised result on the seismic survey of Penobscot. (a) The denoised result using f-x deconvolution. (b) The denoised result using VMD-DnCNN.
Figure 17. Rejected noises on 3D volume through different denoising methods on the seismic survey of Penobscot. (a) The rejected noise using f-x deconvolution. (b) The rejected noise using VMD-DnCNN.

254x190mm (300 x 300 DPI)
Figure 17. Rejected noises on 3D volume through different denoising methods on the seismic survey of Penobscot. (a) The rejected noise using f-x deconvolution. (b) The rejected noise using VMD-DnCNN.

254x190mm (300 x 300 DPI)
Figure 18. The amplitude spectrum of the original data (red curve), VMD-DnCNN result (black curve) and f-x deconvolution result (blue curve) for the seismic survey of Penobscot.

254x190mm (300 x 300 DPI)