

Deep Learning Denoising Applied to Regional Distance Seismic Data in Utah

Rigobert Tibi^{*1}, Patrick Hammond¹, Ronald Brogan², Christopher J. Young¹, and Keith Koper³

ABSTRACT

Seismic waveform data are generally contaminated by noise from various sources. Suppressing this noise effectively so that the remaining signal of interest can be successfully exploited remains a fundamental problem for the seismological community. To date, the most common noise suppression methods have been based on frequency filtering. These methods, however, are less effective when the signal of interest and noise share similar frequency bands. Inspired by source separation studies in the field of music information retrieval (Jansson et al., 2017) and a recent study in seismology (Zhu et al., 2019), we implemented a seismic denoising method that uses a trained deep convolutional neural network (CNN) model to decompose an input waveform into a signal of interest and noise. In our approach, the CNN provides a signal mask and a noise mask for an input signal. The short-time Fourier transform (STFT) of the estimated signal is obtained by multiplying the signal mask with the STFT of the input signal. To build and test the denoiser, we used carefully compiled signal and noise datasets of seismograms recorded by the University of Utah Seismograph Stations network. Results of test runs involving more than 9000 constructed waveforms suggest that on average the denoiser improves the signal-to-noise ratios (SNRs) by ~5 dB, and that most of the recovered signal waveforms have high similarity with respect to the target waveforms (average correlation coefficient of ~0.80) and suffer little distortion. Application to real data suggests that our denoiser achieves on average a factor of up to ~2–5 improvement in SNR over band-pass filtering and can suppress many types of noise that band-pass filtering cannot. For individual waveforms, the improvement can be as high as ~15 dB.

KEY POINTS

- We implement a seismic convolutional neural network model to decompose a waveform into signal and noise.
- The denoiser achieves up to a factor of 2–5 average improvement in signal-to-noise over band-pass filtering.
- The denoiser can work on continuous data, possibly enabling use in routine processing of network data.

Supplemental Material

INTRODUCTION

Seismic waveform data are generally contaminated by noise from various sources, which interfere with the signals of interest. Thus, the efficiency of the noise suppression approach used early in a processing pipeline affects the quality of the downstream products. Seismic noise suppression methods include, among others, techniques based on independent component analysis (Bharadwaj et al., 2017) or empirical mode decomposition (Han and van der Baan, 2015) and frequency filtering. Among these methods, frequency filtering is very popular partly due to its convenience. Frequency filtering works by

retaining content within a predefined frequency band while suppressing anything that lies outside that band. As such, its application may not result in the desired outcome when the signal of interest and noise share the same frequency band. In addition, this method is specific to the type of noise, and must be tuned on a case-by-case basis. Furthermore, frequency filtering is known to distort the signal, in some cases, making phase onsets and polarities difficult to determine despite improved signal-to-noise ratios (SNRs). For that reason, Douglas (1997) argued against using band-pass filtering and instead proposed customized Wiener filters that are based on signal and noise models.

A musical mixture is a superposition of two or more sources such as singing voice and accompaniment (drum, bass, etc.). In the past few years, source separation of audio signals, including

1. Sandia National Laboratories, Albuquerque, New Mexico, U.S.A.; 2. ENSCO, Inc., Springfield, Virginia, U.S.A.; 3. Department of Geology and Geophysics, The University of Utah, Salt Lake City, Utah, U.S.A.

*Corresponding author: rtibi@sandia.gov

Cite this article as Tibi, R., P. Hammond, R. Brogan, C. J. Young, and K. Koper (2021). Deep Learning Denoising Applied to Regional Distance Seismic Data in Utah, *Bull. Seismol. Soc. Am.* **111**, 775–790, doi: [10.1785/0120200292](https://doi.org/10.1785/0120200292)

© Seismological Society of America

musical mixtures, has attracted attention due to the advent of novel processing techniques, most of them stemming from machine learning (ML) (e.g., Huang *et al.*, 2012; Chandna *et al.*, 2017, and references therein). According to several studies, deep CNNs have proven to be efficient in source separation of musical mixtures (Simpson *et al.*, 2015; Chandna *et al.*, 2017; Grais and Plumley, 2017; Jansson *et al.*, 2017; Liu and Yang, 2018). Similar to a musical mixture, a recorded seismic waveform can be thought of as a superposition of the signal of interest and noise. Hence, the seismic noise suppression problem can be formulated as the same sort of source separation problem that is common in the field of music information retrieval.

Inspired by the reported successes of deep learning models in the field of music information retrieval (e.g., Jansson *et al.*, 2017) and a recent work in seismology (Zhu *et al.*, 2019), we implemented a seismic signal denoising method that uses a trained deep CNN model to decompose an input waveform into a signal of interest and noise. To build, test, and assess the efficiency of the CNN denoiser, we use carefully compiled signal and noise datasets of waveforms recorded at local and regional distances by The University of Utah regional seismic network (Pankow *et al.*, 2020). The performance of the denoiser is compared with and contrasted to that of standard band-pass filtering.

METHOD

As mentioned in the [Introduction](#), the approach used in this study is borrowed from music information retrieval (Chandna *et al.*, 2017). A recorded seismic signal, $r(t)$, is assumed to be a superposition of the “noise-free” signal $s(t)$ and “pure” noise $n(t)$. This is expressed mathematically in the time–frequency domain with complex numbers as

$$R(t,f) = S(t,f) + N(t,f), \quad (1)$$

in which $R(t,f)$ is the short-time Fourier transform (STFT) of the recorded signal, $S(t,f)$ is the STFT of the signal, and $N(t,f)$ is the STFT of the noise. To calculate the STFT, we used a segment length of 1 s, and 0.5 s overlap between the segments. We also applied a Hanning window for tapering and zero padding on the input data sampled at 100 Hz. As illustrated in Figure 1, in the encoder part of the denoiser network, the input signal to be processed, $R(t,f)$, is first transformed into a compact, but deep representation. The decoder then works on the encoding and provides a signal mask, $M_S(t,f)$, and a noise mask, $M_N(t,f)$. The estimated signal, $\hat{S}(t,f)$, and estimated noise, $\hat{N}(t,f)$, are obtained by multiplying the respective mask operators with the original input signal, $R(t,f)$, as follows. The symbol \odot in these equations represents the element-wise multiplication operation, also known as Hadamard product:

$$\hat{S}(t,f) = M_S(t,f) \odot R(t,f), \quad (2)$$

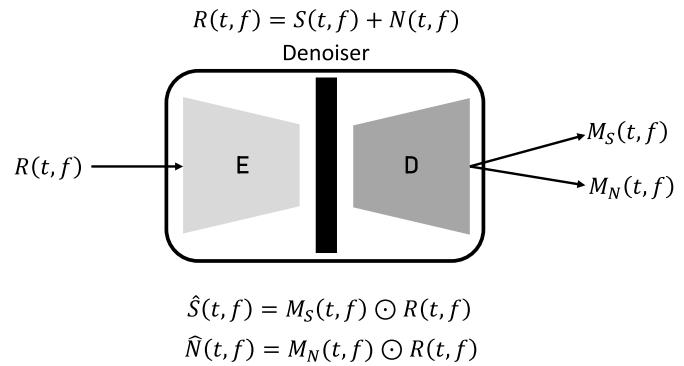


Figure 1. Schematic illustration of the denoiser network (adapted from Mandelli *et al.*, 2019). The encoder (E) and decoder (D) are represented by light gray and dark gray boxes, respectively. Input to the network is the short-time Fourier transform (STFT) of the noisy waveform, $R(t,f)$; and outputs are the signal mask, $M_S(t,f)$, and noise mask, $M_N(t,f)$. See the [Method](#) section for further explanations.

$$\hat{N}(t,f) = M_N(t,f) \odot R(t,f). \quad (3)$$

The estimated signal and noise in the time–frequency domain are inverse transformed into the time domain (real numbers) using the inverse STFT (ISTFT) as follows:

$$\hat{s}(t) = \text{ISTFT}[\hat{S}(t,f)], \quad (4)$$

$$\hat{n}(t) = \text{ISTFT}[\hat{N}(t,f)]. \quad (5)$$

The mask operators, which are matrixes consisting of real numbers with values ranging from 0 to 1, are the targets during the network training stage (see the [Network Training](#) section) and are estimated for the training data (in which signal and noise components are known exactly) as follows:

$$M_S(t,f) = \frac{|S(t,f)|}{|S(t,f)| + |N(t,f)|}, \quad (6)$$

$$M_N(t,f) = \frac{|N(t,f)|}{|S(t,f)| + |N(t,f)|}. \quad (7)$$

NETWORK ARCHITECTURE

The network architecture shown in Figure 2 is adapted from Zhu *et al.* (2019). However, compared with these authors’ version, we made some modifications that include—among other items—a change of the input size and a 30% dropout for each deconvolution layer in the decoder part of the network (discussed in the following). These modifications were necessary to adapt the network to the problem at hand, that is, to get

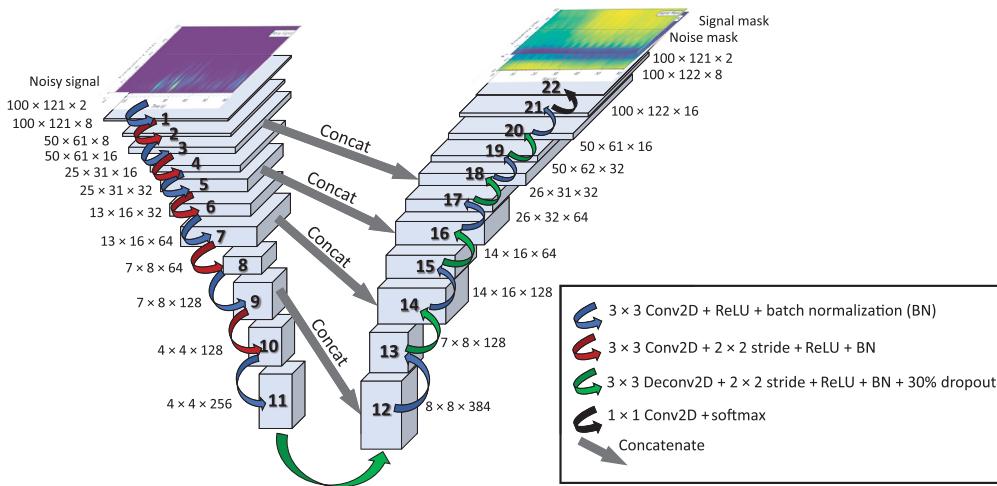


Figure 2. Architecture of the denoiser network. The topmost panel on the left displays an example of input to the network. Light-blue boxes with varying sizes represent the convolutional neural network (CNN) layers, with the box size and number adjacent to each box indicating the dimension and number of channels for the layer. Arrows represent different processes (see legend). Processes represented by the red- and green-curved arrows include a 2×2 stride for down or upsampling, respectively. The two topmost panels on the right show examples of the signal and noise masks outputted by the network. ReLU, rectified linear unit.

the network trained properly with our data. The CNN consists of 20 hidden layers divided equally between the encoder and the decoder.

Inputs to the network are real and imaginary parts of $R(t, f)$, which were previously min–max normalized independent of one another. The encoder (downsampling part) provides a compact, but deep, representation of the inputs and transforms them from a size of $100 \times 121 \times 2$ for the input layer, in which the last number indicates the number of channels, to $4 \times 4 \times 256$ for layer 11 (the last layer of the encoder). This is achieved through a sequential application of the following two processes in the series of layers that make up the encoder:

1. The output of a 2D convolution with a 3×3 weight kernel is passed through a rectified linear unit (ReLU) activation function. This is followed by batch normalization (BN) to improve optimization of the network (Ioffe and Szegedy, 2015; Goodfellow *et al.*, 2016) (blue-curved arrows in Fig. 2).
2. The output of a 2D convolution with a 3×3 kernel and 2×2 stride is passed through a ReLU, followed by BN (red-curved arrows in Fig. 2). From layer 11 to layer 21 in the decoder (upsampling part), process (3), described in the following, and process (1) are applied sequentially in that order.
3. The output of a 2D deconvolution with a 3×3 kernel and 2×2 stride is passed through a ReLU activation function. This is followed by BN and a 30% dropout (green-curved arrows in Fig. 2). The deconvolution process results in upscaling (increase in size) for the layer output with respect

to the input. As discussed in the Network Training section, including dropout was necessary to prevent overfitting.

The output of layer 21 (last hidden layer) is cropped along the second dimension before it is convolved with a 1×1 kernel and passed through a softmax activation function to generate the network outputs (black-curved arrow). These outputs are the signal and noise masks, which have the same dimension as the inputs to the network. To allow a precise reconstruction, output at several levels of the encoder is concatenated with the input at the corresponding decoder levels (Jansson *et al.*, 2017; Mandelli *et al.*, 2019). These

concatenation operations are represented in Figure 2 by gray arrows and are referred to as “skip connections” in ML jargon. The slightly larger sizes of the concatenation inputs from the decoder require them to be cropped along one or both dimensions before each concatenation with the encoder input. Liu and Yang (2018) showed that deep CNNs containing skip connections outperform those without such connections in the domain of music source separation. Zhu *et al.* (2019) also employed skip connections in their signal denoising architecture, which motivated our application here.

Our network, which is implemented using Keras (Chollet, 2015) on top of TensorFlow (Abadi *et al.*, 2016), contains ~ 2.4 million trainable parameters and ~ 3000 nontrainable parameters. For a limited number of hyperparameters chosen manually, the choice was based on the values of the validation loss obtained from trial-and-error processes (see the Network Training section). We experimented in reducing the number of free parameters by reducing the number of layers. This resulted in network performance diminishing substantially for both our training and validation data, due to reduced network capacity.

DATA

Even for a seismically active region like Utah, there are a limited number of large magnitude events available to compile a sufficiently large dataset of high-SNR regional waveforms (which will be treated as “noise free”) to train our deep denoising network. Using the University of Utah Seismograph Stations (UUSS) event catalog (University of Utah, 1962)

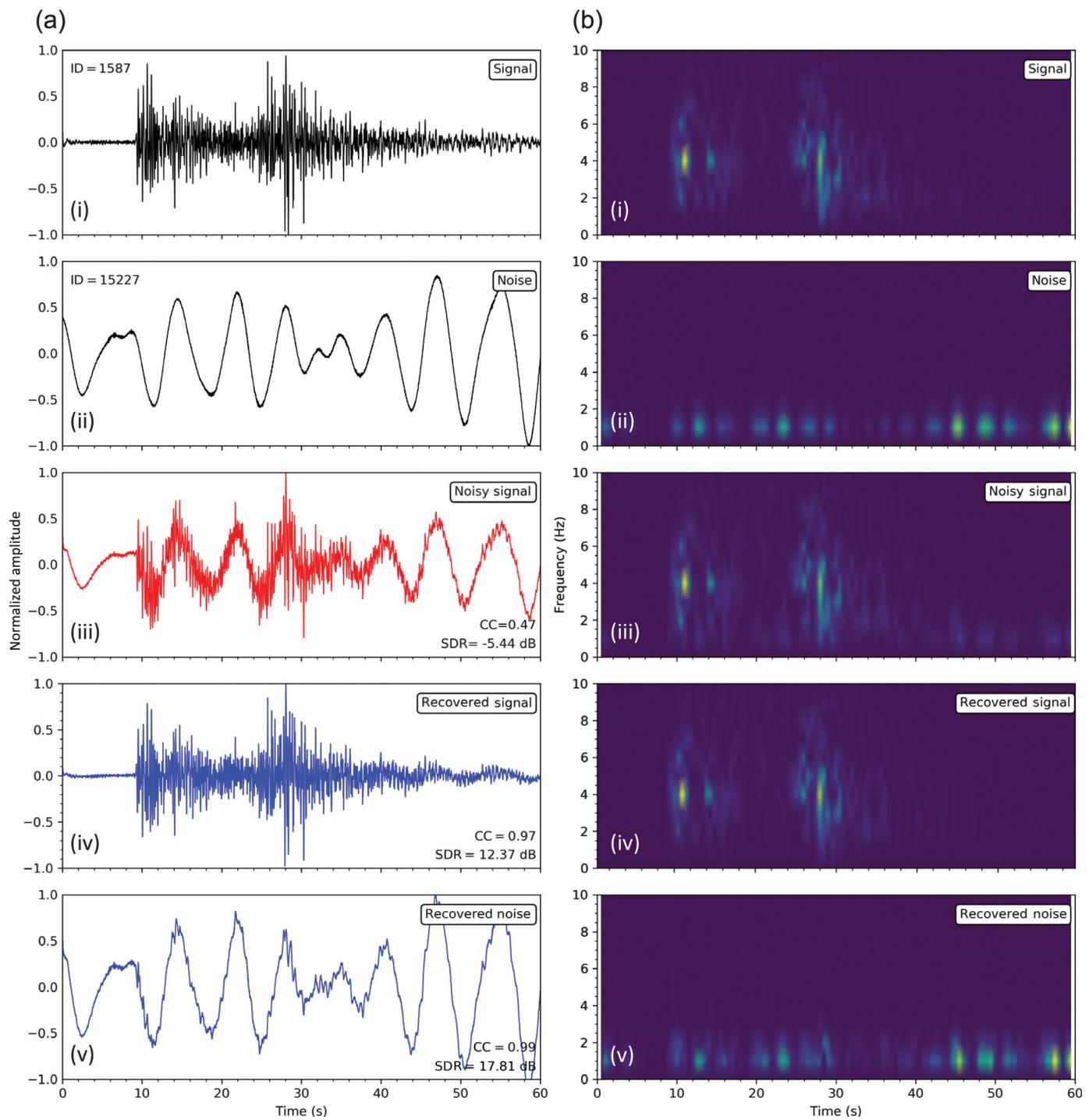


Figure 3. (a) Waveforms and (b) corresponding spectrograms for (i) “noise-free” signal, (ii) “pure” noise, (iii) constructed noisy signal, (iv) recovered signal, and (v) recovered noise. The constructed waveform in panel (iii) consists of the summation of the seismograms shown in panels (i) and (ii). The correlation coefficient (CC) and signal-to-distortion ratio (SDR)

values indicated for the noisy signal in panel (a, iii) and the recovered signal waveform in panel (a, iv) are with respect to the “noise-free” signal waveform displayed in panel (a, i). The CC and SDR values indicated for the recovered noise in panel (a, v) are with respect to the “pure” noise waveform shown in panel (a, ii).

for a 8 yr period (September 2009–December 2017) with more than 30,000 events, we compiled a dataset of “noise-free” signal waveforms that consists of 3188 high-SNR vertical-component

seismograms recorded by the UUSS broadband station BRPU located in central Utah (Fig. S1, available in the supplemental material to this article). We chose signals from only one station

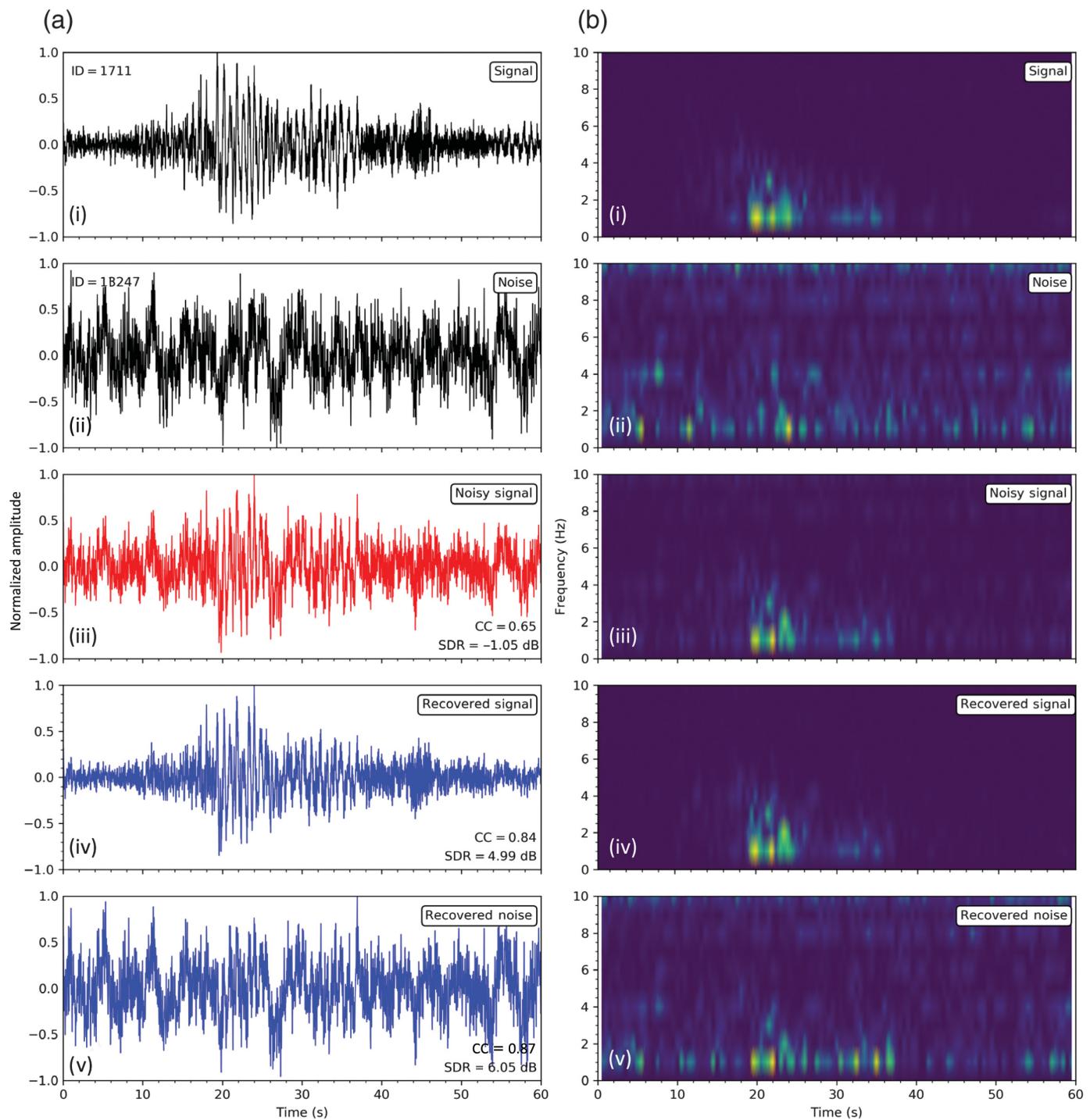


Figure 4. Same as in Figure 3, but for a different waveform.

because we intended to keep the study simple, and explore model transportability within the UUSS network. Station BRPU was selected because it is a high-performing station located in the middle of the state. Waveforms were selected based on an estimated SNR of two or higher, whereas a small subset (total of 427) were selected after visual inspection of

about 2000 seismograms from events with $M_c \geq 2$. Waveform selection based on SNR was adopted after we realized that visual inspection, which resulted in the small subset of 427 waveforms, was too tedious for the large dataset. Before training our model, the compiled high-SNR waveforms, which are from earthquakes with magnitude $M_c \leq 5.3$ and epicentral distances

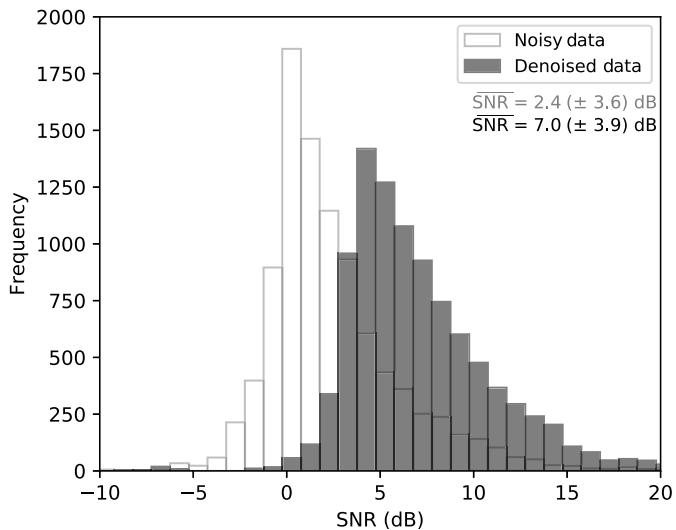


Figure 5. Distribution of signal-to-noise ratios (SNRs) for the constructed, noisy (white) and denoised (black) data. Values indicate the average SNR for each dataset, with the values in parentheses representing the associated one standard deviation.

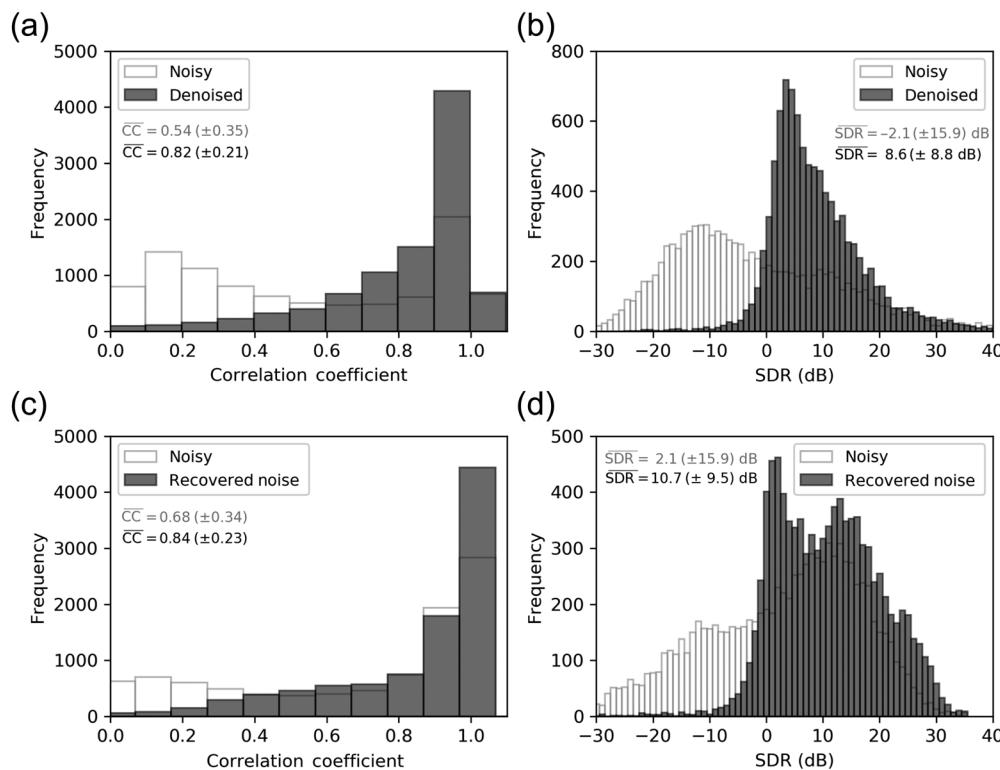


Figure 6. Distribution of (a) CC and (b) SDR values (with respect to the ground truth [GT] signal seismogram) for the noisy (white) and denoised (black) signal waveforms. Average and one standard deviation values for these parameters are indicated. (c) Distribution of CC and (d) SDR values (with respect to the GT noise seismogram) for the noisy and recovered noise waveforms.

between 10–670 km, were all detrended, demeaned, and filtered with a band-pass filter of 1–20 Hz.

The noise dataset contains 15,426 waveforms manually selected by an expert analyst from various noise sources and various stations of the UUSS network. We used many stations in this case because we needed a richer set of noise than would be recorded by a single station. Each waveform in both the signal and noise datasets was 60 s long and sampled at 100 Hz. The signal waveforms were each aligned such that the predicted first P arrival occurs at 10 s after the start time. The two datasets were each randomly divided into training, validation, and test sets using the “70-15-15” partitioning convention. For each set, noisy waveforms were constructed by summing each signal waveform with a randomly selected noise waveform. This was repeated 20 times to generate 44,620 waveforms for training, 9580 for validation, and 9560 for testing. The validation set is used in tuning the network hyperparameters, whereas the test set is used to assess network performance.

NETWORK TRAINING

To train the CNN model, we used the stochastic gradient descent (SGD) optimizer with a learning rate of 5×10^{-4} , a batch size of 128, and a maximum number of epochs of 400. Before settling on the SGD, we tried the Adam optimizer

(Kingma and Ba, 2014), one of the most popular optimizers in ML to date. However, an evaluation of the validation loss indicated that the Adam optimizer was far outperformed by the SGD. We also investigated learning rates of 0.1, 0.01, and 0.001, but these values were too high to promote stable learning. After evaluating mean-squared error and cross-entropy loss functions, both with and without regularization, we chose to minimize the regularized cross-entropy loss. This choice came about because training using cross entropy was substantially faster, and the performance of the resulting model was better; the added penalty from the L_2 regularization ensures that the model also generalizes better. To further prevent overfitting, we applied a 30% dropout in each deconvolution layer. That is, 30% of the layer units

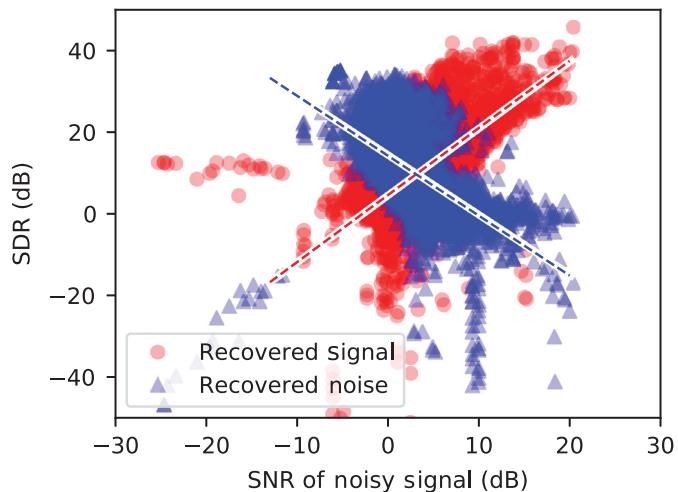


Figure 7. SDR values (with respect to the respective GTs) for the recovered signal (red circles) and recovered noise waveform (blue triangles) as a function of SNR values of the input (noisy) waveform. Each input waveform results in two symbols at the same position on the horizontal axis, one for the recovered signal and one for the recovered noise.

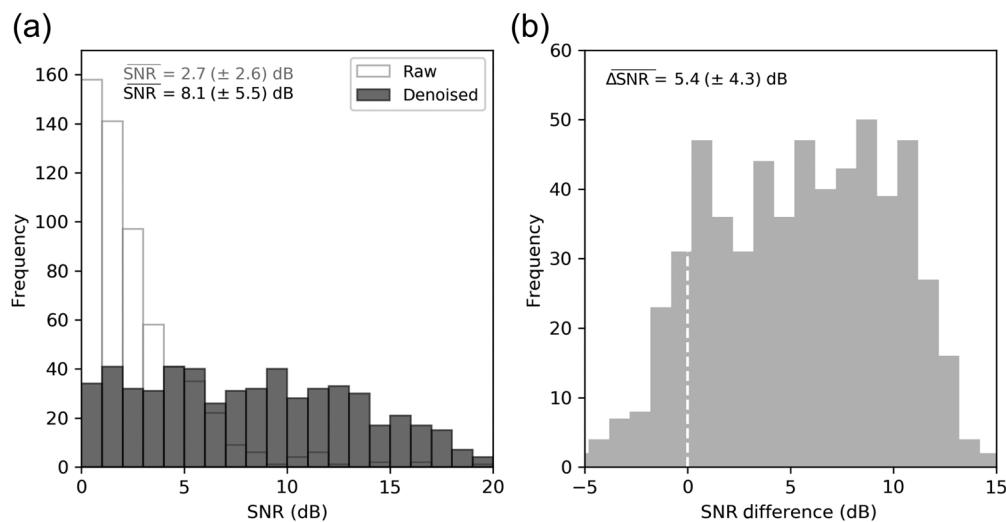


Figure 8. (a) Distribution of SNRs for raw and denoised data. Values indicate the average SNR for each dataset, with the value in parentheses representing the associated one standard deviation. (b) Distribution of the SNR differences between denoised and raw seismograms. The average difference and the associated one standard deviation are indicated. A positive difference indicates improvement in SNR for the denoised waveform over the raw seismogram.

were dropped randomly from the network during training (Srivastava *et al.*, 2014). Using an early stopping patience of 20, training was stopped after 245 epochs, implying that the best model was obtained after 225 epochs (Fig. S2).

MODEL EVALUATION

In the following sections, we assess the performance of the denoiser using both the test dataset of constructed waveforms and real, nonconstructed data. In the assessment, three evaluation metrics (described in the [Evaluation Metrics](#) section) are involved. We also contrast the denoiser performance with band-pass filtering, the most commonly used method in seismology for noise suppression to date.

Evaluation metrics

The effectiveness of the denoiser in suppressing noise is evaluated by comparing the SNRs of the waveforms before and after denoising. The SNR (in decibels) for a waveform is estimated using the following equation:

$$\text{SNR} = 10 \log_{10} \frac{A_S}{A_N}, \quad (8)$$

in which A_S represents the root mean square (rms) of the amplitudes for a 9 s wide signal window starting from the predicted P -wave arrival time, and A_N is the rms for a 9 s wide noise window ending 1 s before the P -wave arrival time.

We use cross correlation to measure the similarity—expressed in terms of a normalized correlation coefficient (CC)—between the ground truth (GT) “noise-free” signal or GT “pure” noise waveform and the corresponding signal or noise waveform recovered using the CNN. The CC value indicates the degree of similarity between two waveforms. However, this parameter does not provide information about the differences in amplitudes between the waveforms being compared. To measure the level of distortion in amplitudes for the recovered seismogram (signal or noise) with respect to corresponding GT waveform, we used the signal-to-distortion ratio (SDR). The SDR in decibels is estimated using the following expression (Nakajima *et al.*, 2018):

$$\text{SDR} = 10 \log_{10} \frac{\|W_{\text{GT}}\|^2}{\|\hat{W} - W_{\text{GT}}\|^2}, \quad (9)$$

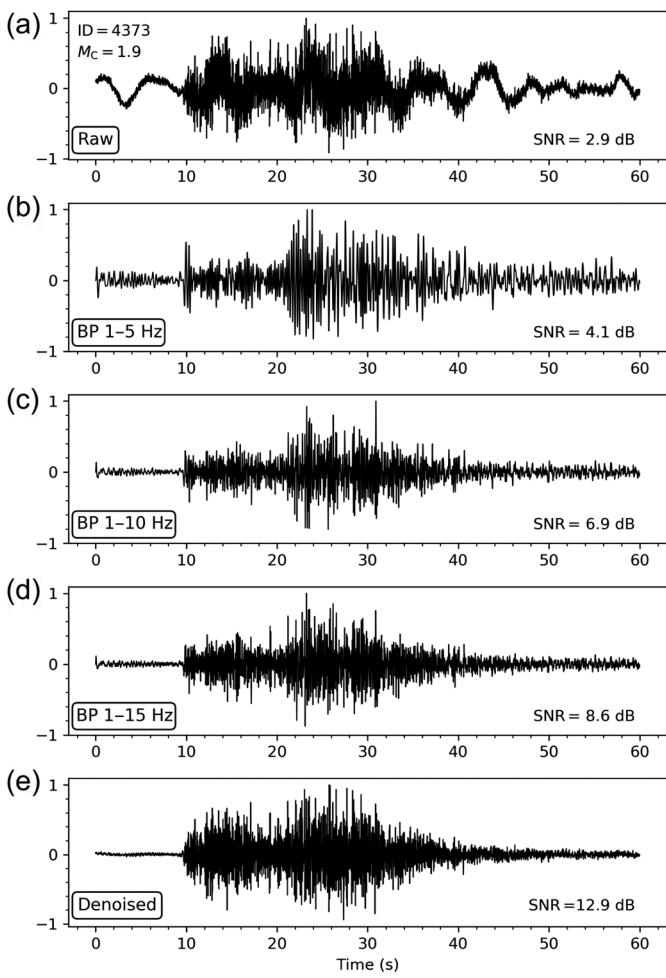


Figure 9. (a) Raw waveform as recorded at station BRPU. (b–d) Waveform shown in panel (a) filtered using a band-pass filter (BP) with a passband of 1–5, 1–10, and 1–15 Hz, respectively. (e) Waveform shown in panel (a) processed with the CNN denoiser. The SNR value for each seismogram is indicated in the bottom-right corner.

in which W_{GT} is the array of amplitudes for the GT seismogram, and \hat{W} is the array of amplitudes for the corresponding waveform recovered by the CNN. When the GT and the recovered waveforms have similar amplitudes (i.e., only little to no distortion has occurred during processing), the denominator in equation (9) goes toward 0, resulting in high SDR values.

Evaluation based on constructed waveforms of the test dataset

We evaluate the performance of the denoiser by processing the 9560 constructed noisy waveforms of the test data. These data were not involved in network training or hyperparameter tuning; as such, they are characterized as “unseen” by the network. Figure 3a(iii) shows an example of constructed seismogram from the test set in red. The associated GT signal and noise

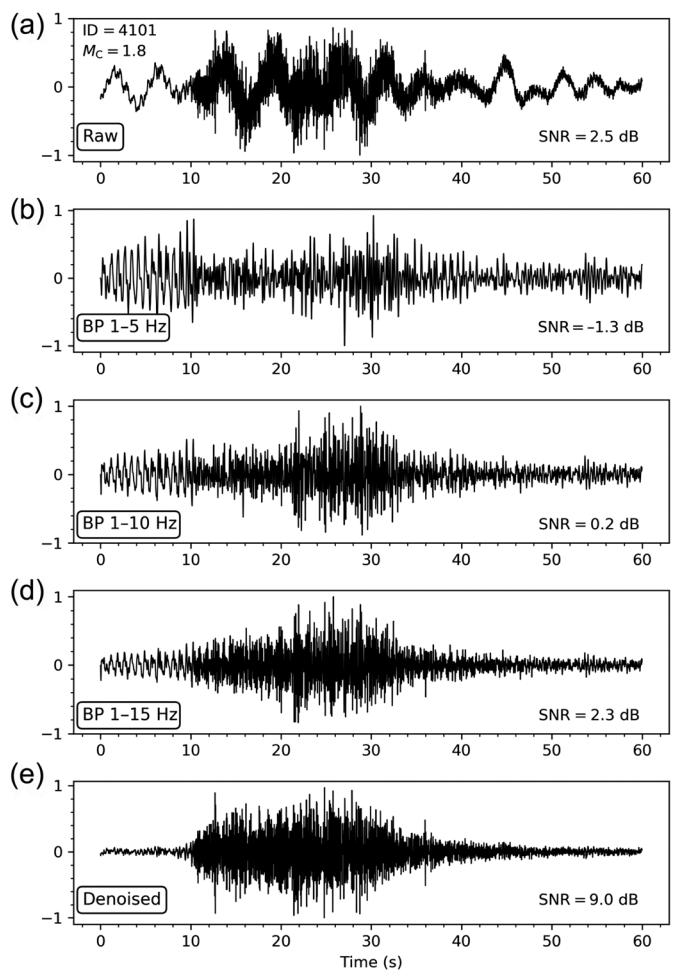


Figure 10. Same as in Figure 9, but for the raw seismogram shown in panel (a).

seismograms are displayed in Figure 3a(i,ii), respectively. This example illustrates a case in which the noise waveform, because of its overall higher amplitudes, is the dominant component of the constructed seismogram. The waveforms shown in blue in Figure 3a(iv,v) represent the signal and noise seismograms recovered by the network, respectively. As inferred from the high CC values (0.97–0.99), the recovered waveforms are very similar to the corresponding GT seismograms. In addition, the high SDR values (12.37–17.81) indicate that the recovered amplitudes show little distortion with respect to the respective GTs. The impact of high-frequency signals that have leaked and contaminated the recovered noise waveform appears to be negligible (Fig. 3a(v)). The spectrograms displayed in Figure 3b confirm these assessments.

Figure 4 shows a similar set of plots for another example of constructed waveform (shown in red in plot Fig. 4a(iii)). Again, the high CC (0.84–0.87) and SDR (4.99–6.05) values of the recovered waveforms with respect to the respective GTs, as well

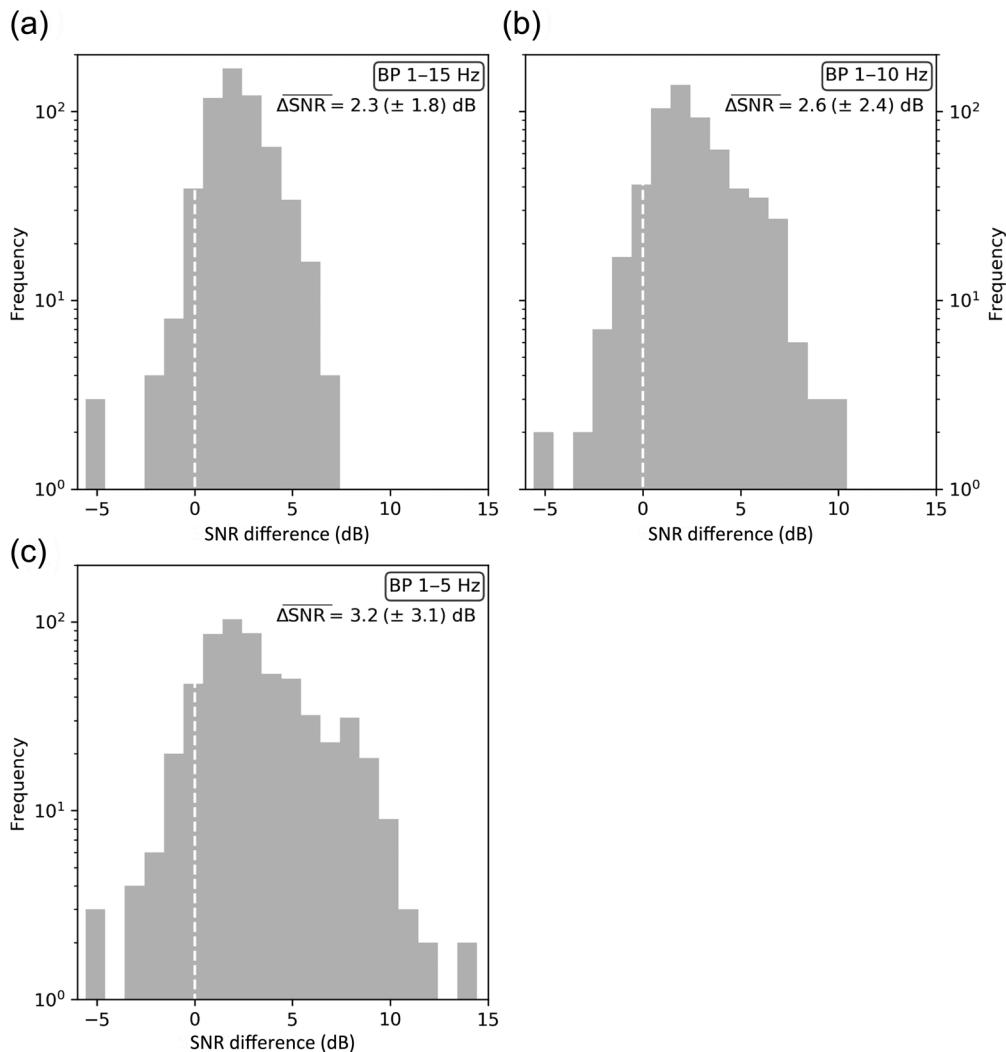


Figure 11. Distribution of SNR differences between the implemented CNN denoiser and band-pass filtering for 584 waveforms recorded at station BRPU. The passband used in filtering is indicated in each plot, that is, (a) 1–15 Hz, (b) 1–10 Hz, and (c) 1–5 Hz. A positive difference indicates improvement in SNR by the deep denoiser over band-pass filtering. The average difference in SNR and the associated one standard deviation for each frequency passband is indicated.

as the consistencies of their respective spectrograms, suggest that the recovered waveforms show high degrees of fidelity to the GTs, in terms of both waveform similarities and amplitudes.

The SNR distribution for the 9560 constructed noisy waveforms and corresponding denoised seismograms is displayed in Figure 5. For this dataset, the denoiser achieves an average improvement in SNR of about 5 dB. For the same dataset, Figure 6 shows the distributions of CC and SDR values for the noisy waveforms (histograms in red) and recovered seismograms (histograms in navy) with respect to the GTs. The high-average CC values (0.82–0.84) for both the recovered signal and recovered noise waveforms suggest that most of the

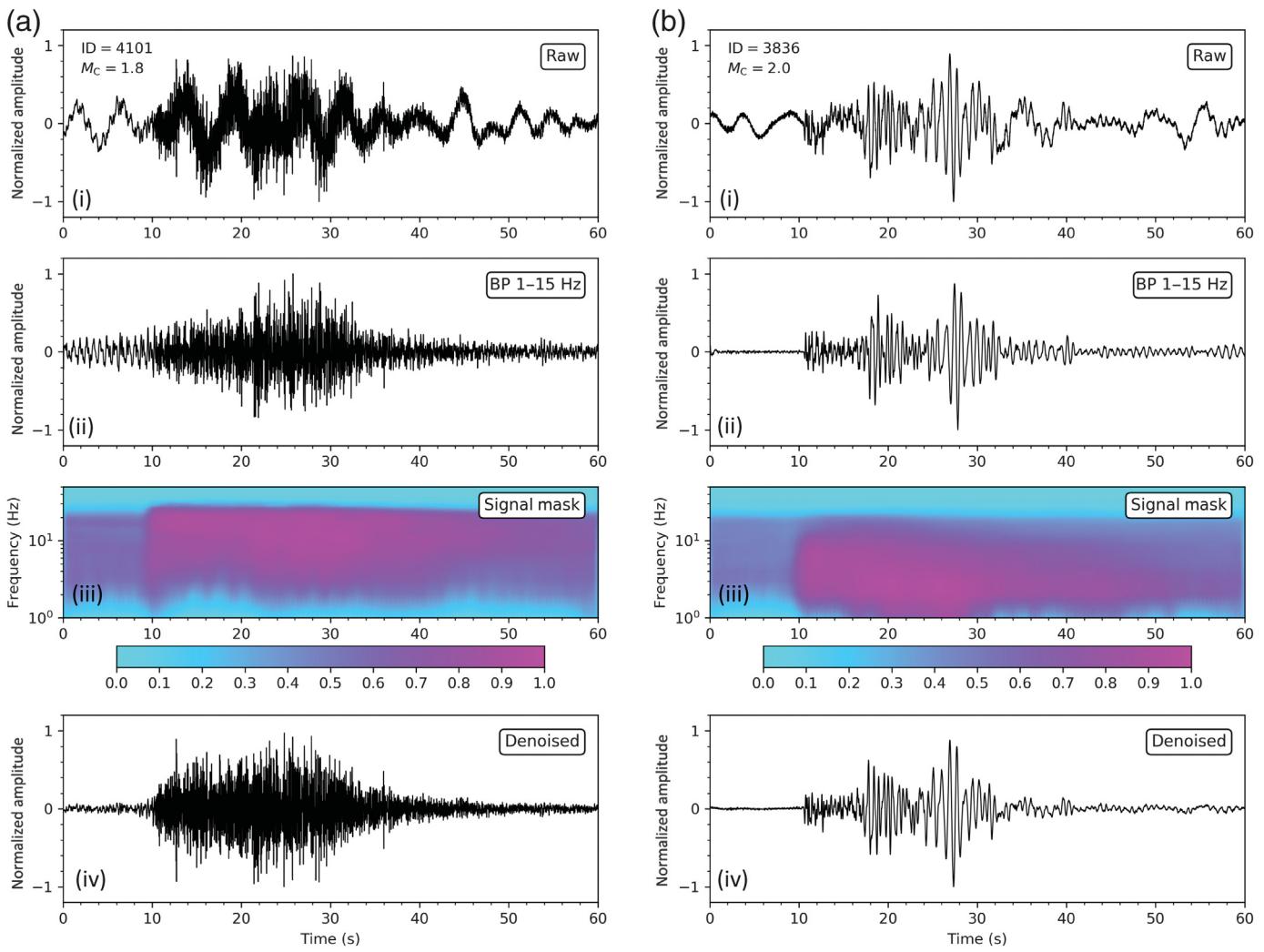
waveforms recovered using the network are very similar to their respective GTs, and as implied by the high-average SDRs (8.6–10.7), the majority of the recovered waveforms suffer little amplitude distortion.

Figure 7 shows the estimated SDR values with respect to the respective GTs for both the recovered signal (red circles) and noise (blue triangles) waveforms, each plotted as a function of SNR of the (input) constructed noisy waveforms. Higher SNRs for the noisy seismogram suggest that the signal waveform is the more dominant component (in terms of amplitudes), and lower SNRs indicate increasing importance of the noise waveform. As illustrated by the two crossing linear regression lines in Figure 7, and also confirmed—although to a lesser extent—by a similar plot of CC versus SNR values (not shown), the CNN model suffers somewhat from the “X-trend” problem. That is, between signal and noise, the dominant component in the input (noisy) waveform is recovered with a higher degree of accuracy (higher SDR and CC values) compared with the weaker component.

Evaluation based on real data

Using our denoiser, we processed a set of real data that consists of 584 seismograms from events in Utah with M_c from −0.1 and lower to 4.5 recorded at BRPU. Figure 8a shows the distribution of SNRs for these waveforms before and after denoising, and Figure 8b shows the histogram of the difference in SNRs between the raw and denoised data. A positive difference indicates improvement in SNR for the denoised seismogram over the raw data. The denoiser achieves an average improvement of ~5 dB over raw data, with some seismograms showing improvement as high as 15 dB.

Using the same dataset, we compared the performance of the denoiser with that of band-pass filtering, involving a



Butterworth filter with three of the passbands typically used for local and near-regional data (1–5, 1–10, and 1–15 Hz). Figure 9b–d shows the processed waveforms for the raw seismogram shown in Figure 9a. The SNR for the raw waveform is 2.9 dB. Band-pass filtering achieves SNRs of 4.1–8.6 dB. Using the denoiser, the SNR increases to 12.9 dB (Fig. 9e). Compared with both the raw and filtered waveforms, the denoised seismogram is characterized by reduced pre-*P* and enhanced *P* amplitudes, resulting in the improved SNR (Fig. 9). Figure 10a displays another example of a raw waveform. Long-period pre-*P* harmonic noise is between 0 and 10 s time. Band-pass filtering appears to amplify that harmonic, resulting in decreased SNR, from 2.5 dB for the raw seismogram to values between –1.3 and 2.3 dB for the filtered waveforms (Fig. 10b–d). The denoiser is able to successfully suppress the pre-*P* harmonic, while enhancing the signal at the same time, resulting in an SNR value of 9 dB for the denoised waveform (Fig. 10e).

Figure 11 shows the distribution of the SNR differences between the denoised seismograms and the corresponding filtered ones for each of the three band-pass filters used. A positive difference suggests that, for the same raw seismogram,

Figure 12. (a) (i) Raw waveform (ID 4101). (ii) Seismogram in panel (i) filtered with a BP of 1–15 Hz. (iii) Signal mask operator obtained from the CNN that was used to denoise the waveform in panel (i). (iv) Seismogram in panel (i) processed with the denoiser using the signal mask displayed in panel (iii). Panel (b) is same as in panel (a), but for the waveform with ID 3836. The waveform from a shallow event located at about 39 km distance shows *Pg*, *Sg*, and also *Rg* phases, characterized by lower frequencies.

denoising results in a higher SNR compared with filtering. For this dataset, the denoiser achieves an average improvement in SNR of about 2–3 dB over band-pass filtering. For individual waveforms, the improvement can be as high as ~15 dB (Fig. 11c). In general, the gap in performance between denoising and filtering increases with decreasing filter bandwidth, due to decreasing SNRs for the filtered waveforms (Fig. 11). We noticed that in some of the seismograms, there seem to be lot of noise with a dominant frequency around 1 Hz. For that reason, we increased the low-cut frequency of the filters to 2 Hz and experimented with the passbands of 2–5, 2–10, and 2–15 Hz. Figure S3 shows that most of the conclusions

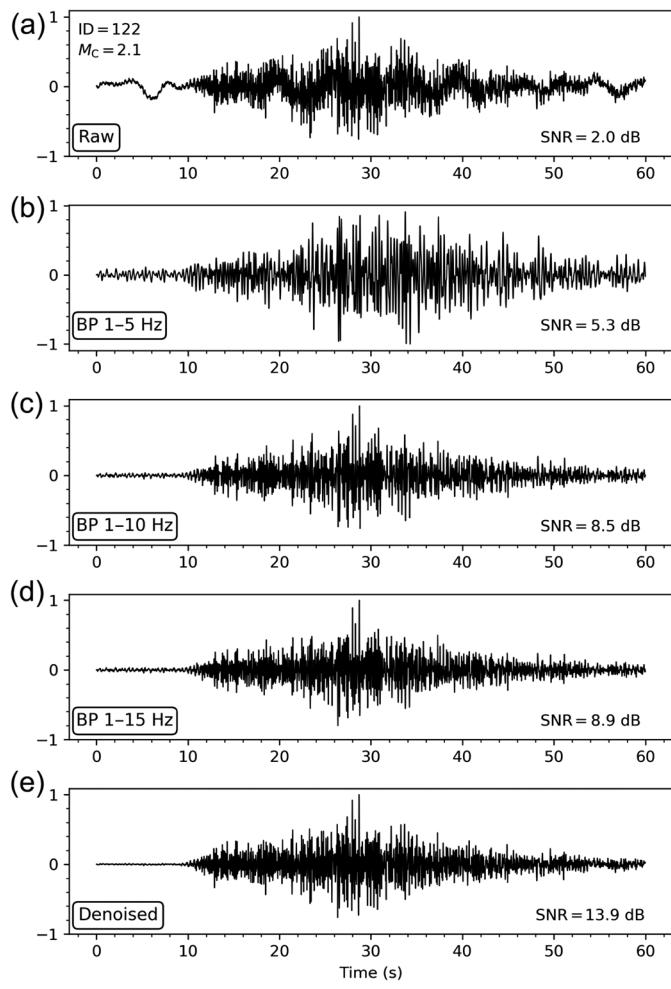


Figure 13. (a) Raw waveform as recorded at station ZNPU. (b–d) Waveform shown in panel (a) filtered using a BP with a passband of 1–5, 1–10, and 1–15 Hz, respectively. (e) Waveform shown in panel (a) processed with the CNN denoiser. The SNR value for each seismogram is indicated in the bottom-right corner.

reached earlier for the filters with a low cut of 1 Hz still hold. A visual inspection of a multitude of processed seismograms like those displayed in Figures 9 and 10 suggests that the denoiser can suppress many types of noise that band-pass filtering cannot.

We investigated why the denoiser outperforms frequency filtering by carefully examining some of the signal masks provided by the CNN network (Fig. 12a,b and Fig. S4). The values of the elements in the mask operator vary with both time and frequency in the range from 0 to 1. These variations are particularly significant for input waveforms that include phases with different dominant frequencies (e.g., Pg, Sg, and Rg; see Fig. 12b and Fig. S4c). The values of the elements in the mask operator indicate the proportion of energy from each point of the input seismogram (in the time–frequency domain) that is converted into signal, and the remaining energy is considered to be noise. In contrast to the masks shown in Figure 12a(iii),b(iii)

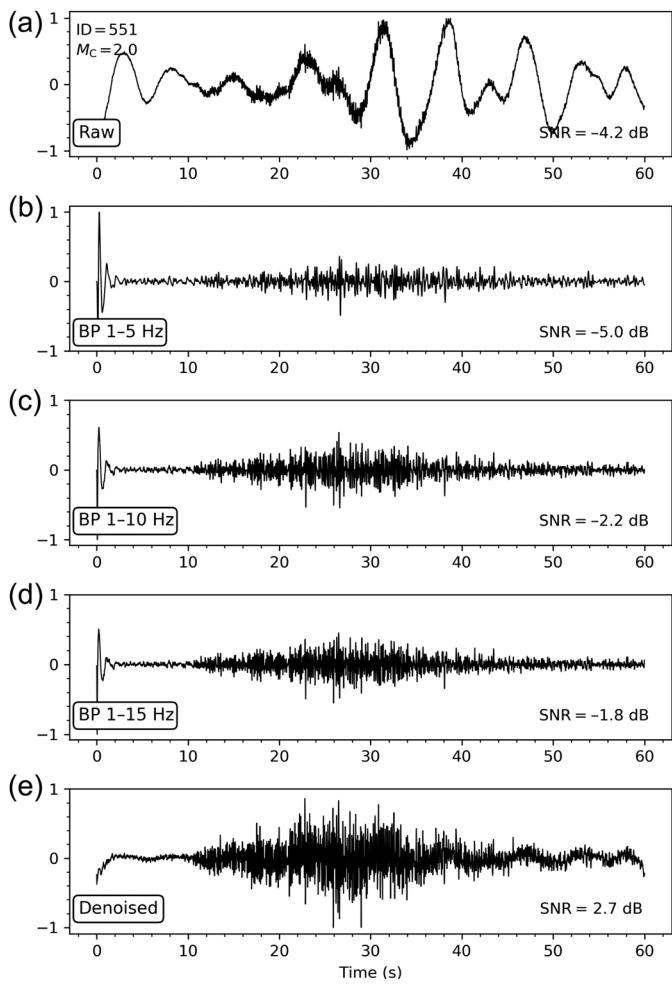


Figure 14. Same as in Figure 13, but for the raw seismogram shown in panel (a).

and Figure S4a–c(iii), the operator for a band-pass filter would appear as a perfect horizontal streak of ones within the frequency passband with relatively sharp drop-off to 0 above and below. The variations of the elements in the mask operators with time and frequency allow the denoiser to adapt to changing characteristics of the input waveform. This adaptability is undoubtedly the reason for the observed edge of the denoiser over band-pass filtering.

Model transportability

An important question is how the trained CNN model performs for stations not involved in the training (i.e., not BRPU). To help answer this question, we tested the denoiser on data recorded by two other stations in the UUSS network, ZNPU and SPU (Fig. S1). Station ZNPU is located in the southwestern region of Utah, about 314 km from BRPU, the training station. Station SPU is located north–northwest of Salt Lake City, about 212 km from BRPU. The large geographical separations between these stations are intended to ensure that regional

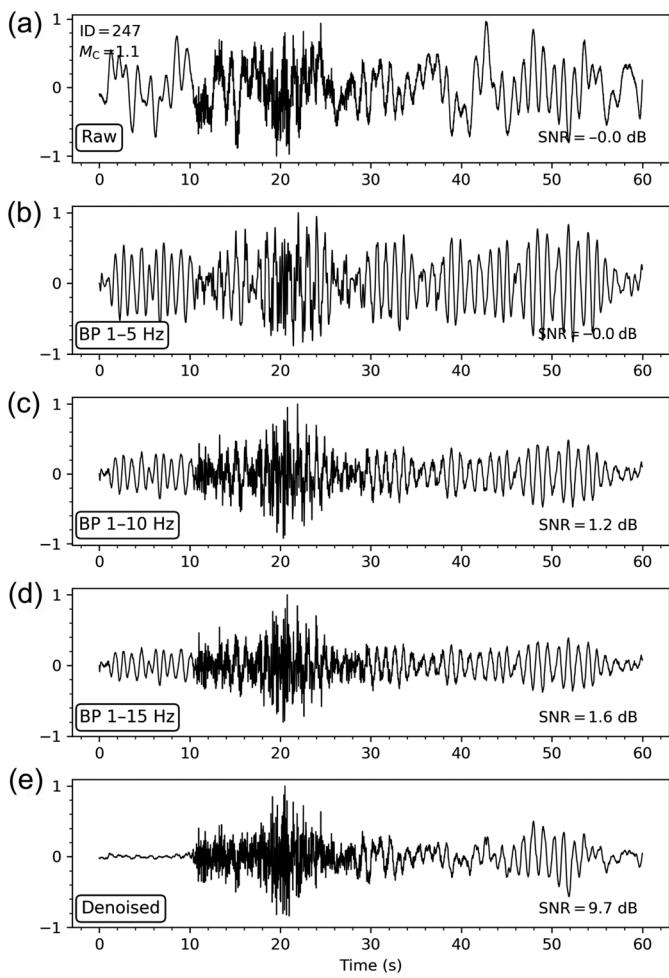


Figure 15. (a) Raw waveform as recorded at station SPU. (b–d) Waveform shown in panel (a) filtered using a BP with a passband of 1–5, 1–10, and 1–15 Hz, respectively. (e) Waveform shown in panel (a) processed with the CNN denoiser. The SNR value for each seismogram is indicated in the bottom-right corner.

scale variabilities in terms of propagation effects and background noise are sufficiently captured by the waveforms to be processed. Station ZNPU did not contribute to the signal dataset used in network training but did contribute to the noise dataset, whereas station SPU contributed to neither of the two datasets. For station ZNPU, we processed 844 waveforms from events with magnitudes of $M_c = -0.87$ to 5.3. For SPU, 583 seismograms from events with magnitude of $M_c = 0.1$ –4.5 were used.

Figure 13a shows an example of a raw seismogram recorded at ZNPU. Band-pass filtering improves the SNR for that waveform from 2 dB to values of 5.3–8.9 dB (Fig. 13b–d). Denoising improves the SNR further to 13.9 dB, resulting from the combined effect of a reduced pre- P noise (between 0 and 10 s) and enhanced signal (Fig. 13e). Figure 14a displays another example seismogram from the same station. In this case, the raw waveform is dominated by long-period noise of varying

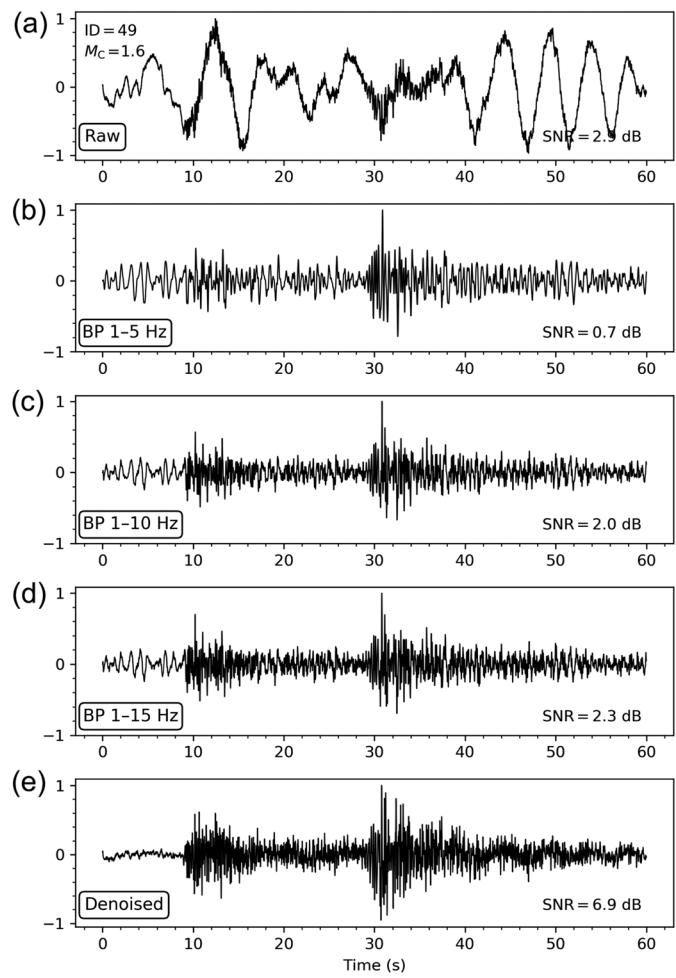


Figure 16. Same as Figure 15, but for the raw seismogram shown in panel (a).

amplitudes, and no arrival is visible (Fig. 14a). Although band-pass filtering is able to remove the long-period noise, allowing the P coda to be visible, the P onset is hardly distinguishable from background noise (Fig. 14b–d). In a processing pipeline involving frequency filtering as the only means for noise suppression, the P onset would be difficult to determine in this case. On the other hand, the P onset is clearly visible in the denoised waveform at about the expected time of 10 s (Fig. 14e), and the SNR has improved to 2.7 dB from subzero values for the raw and filtered seismograms.

Figure 15a displays a raw waveform recorded at station SPU. In this case, filtering has reduced the noise only marginally (Fig. 15b–d). In contrast, the efficiency with which the denoiser has suppressed the noise, as demonstrated by very low pre- P noise level, is outstanding (Fig. 15e). This resulted in a dramatic improvement in SNR ratio from 0–1.6 dB for the raw and filtered seismograms to a value of 9.7 dB for the denoised waveform (Fig. 15). In a second raw seismogram from SPU, arrivals are invisible, except marginal hints coming

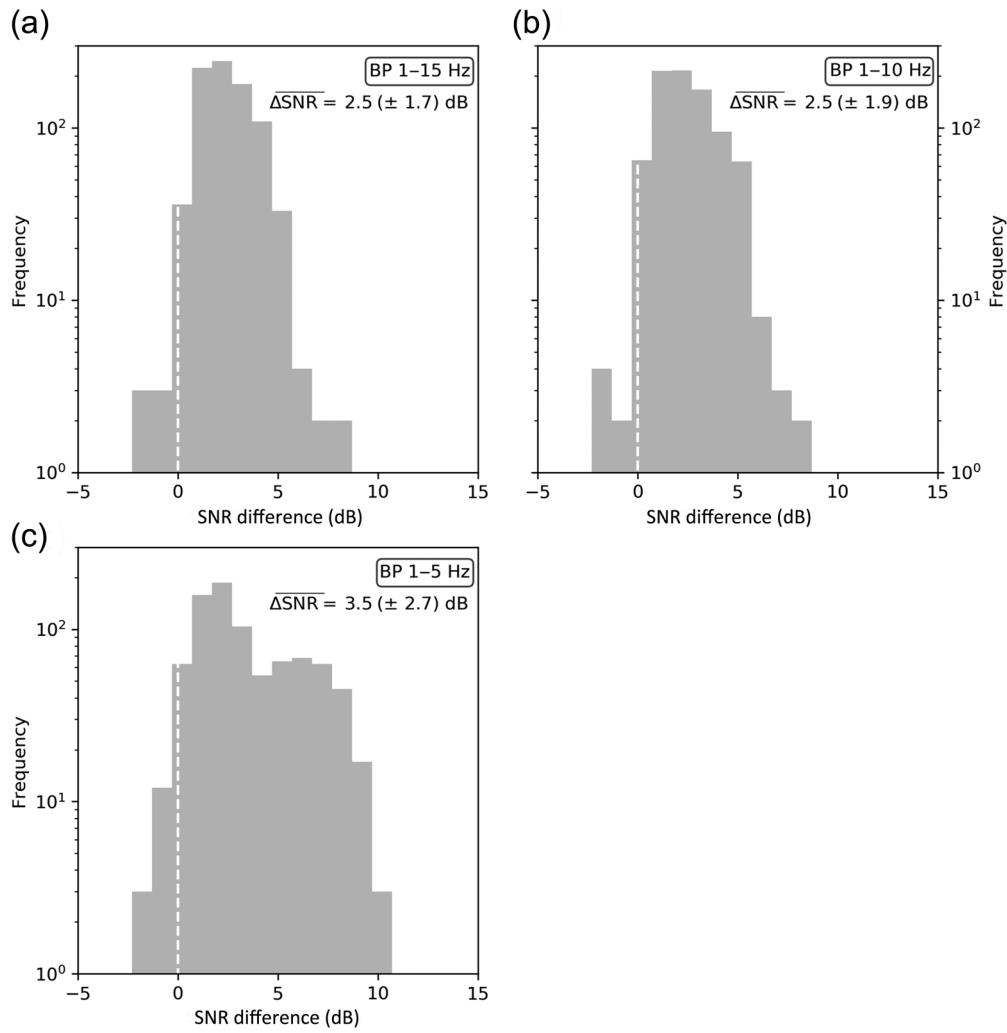


Figure 17. Distribution of SNR differences between the implemented CNN denoiser and band-pass filtering for 844 waveforms recorded at station ZNPU. The passband used in filtering is indicated in each plot, that is (a) 1–15 Hz, (b) 1–10 Hz, and (c) 1–5 Hz. A positive difference indicates improvement in SNR by the deep denoiser over band-pass filtering. The average difference in SNR and the associated one standard deviation for each frequency passband is indicated.

from changes in frequency contents (Fig. 16a). Although *P* and *S* arrivals are visible after filtering (Fig. 16b–d), again, the denoiser does much better at noise suppression. This is demonstrated by the low pre-*P* noise level and confirmed by the high-SNR value of 6.9 dB (Fig. 16e).

The results of the comparison between denoising and filtering for the 844 waveforms processed for station ZNPU are summarized in Figure 17. Shown are the distribution of the difference in SNRs between denoising and band-pass filtering for the frequency bands of 1–5, 1–10, and 1–15 Hz using the same raw waveforms. Similarly, the results for the 583 seismograms processed for the station SPU are shown in Figure 18. For these stations, denoising achieved an average improvement of ~3–5 dB over band-pass filtering. This range is slightly higher than the average improvement of ~2–3 obtained for

the station BRPU that was used to train the network. Undoubtedly, this suggests that the CNN model—although built using mainly data recorded by a single station—is transportable to other stations in Utah, and possibly also in neighboring regions, where wave-propagation characteristics and the ambient noise wavefield are not significantly different from those of the Utah region. In support of this theory, we obtained mixed results when attempting to denoise regional data recorded by station MDJ located in northeast China, which has very different local scale Earth structure than Utah (Fig. S5). This suggests that the propagation characteristics must be similar to those of the CNN training data in order for the denoiser to perform adequately.

The CNN model was trained and tested using 60 s long waveforms that were each aligned beforehand such that the predicted first *P* arrival occurred 10 s after the start time. A question is whether this configuration constitutes a handicap when processing continuous data, in which case arrival timings vary. To address this question, we proc-

essed seismograms with first *P* arrivals occurring at different times (5, 20, and 30 s after the start time). Figure 19 shows a set of waveforms associated with a *P* arrival timed at 30 s. In this and other cases, the denoiser performed well, no matter the timing of the arrival, suggesting that the CNN model in its current state can be successfully used to process continuous data.

CONCLUSIONS

We implemented a seismic denoising method that uses a trained deep CNN model to decompose an input waveform into a signal of interest and noise. Test results involving more than 9000 constructed noisy waveforms suggest that the vast majority of the seismograms recovered by the network show high degrees of fidelity to their respective GTs, in terms of both

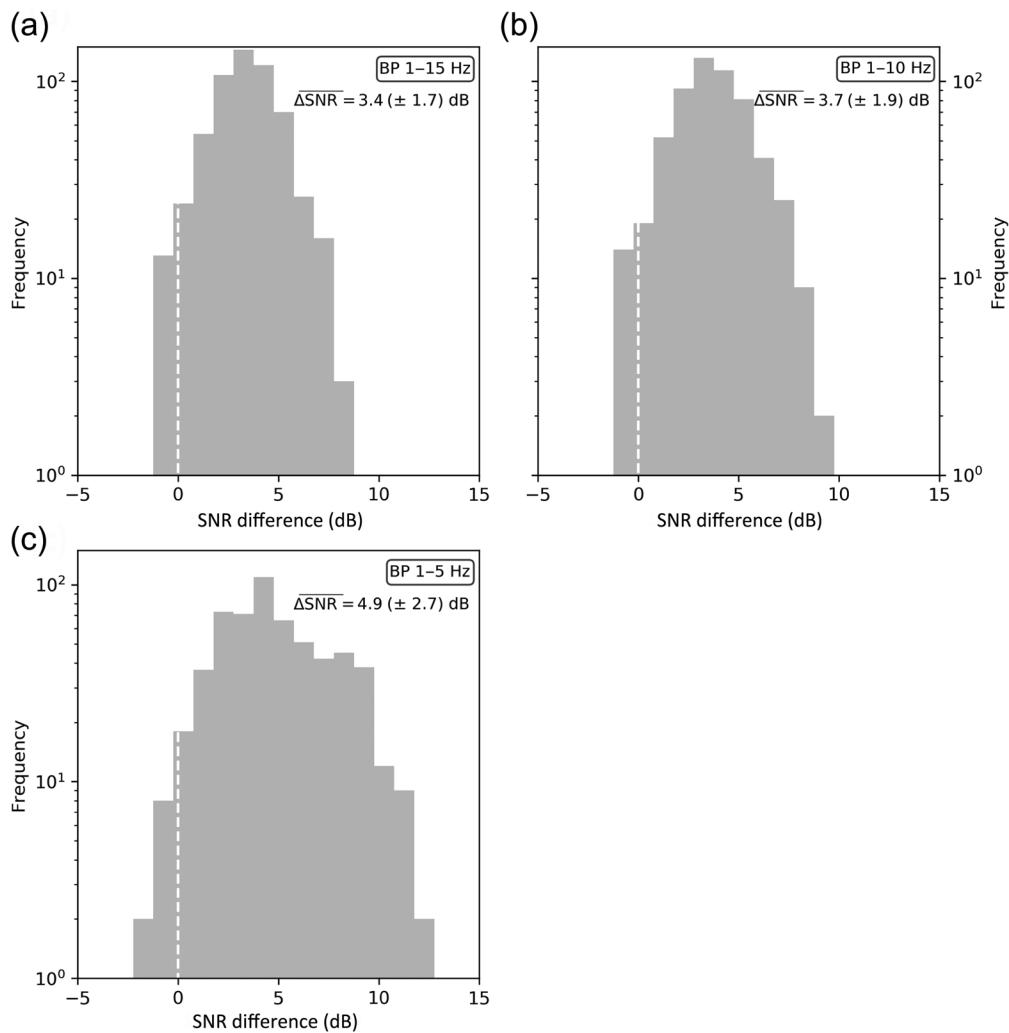


Figure 18. Same as Figure 17, but for the 583 seismograms processed for the station SPU.

waveform similarities and amplitudes. To a limited extent, the network suffers from the “X-trend” problem. That is, between noise and signal, the dominant component in the input waveform is recovered with higher level of accuracy compared with the weaker component; hence, for low SNR input waveforms, recovery of noise is emphasized over recovery of signal. Processing of real seismograms suggests that the CNN denoiser achieves an average improvement in SNR of ~ 5 and $\sim 2\text{--}5$ dB over the raw and band-pass filtered data, respectively, and the improvement can be as high as ~ 15 dB for individual waveforms. The denoiser can suppress many types of noise that band-pass filtering cannot. The edge of the denoiser over band-pass filtering is due to the adaptability of the denoiser to changing characteristics of the input seismogram over time. This adaptability is driven by the variations of the mask operator with time and frequency. The constructed CNN model also works well for UUSS stations not involved in model training, suggesting that it is transportable around Utah, and possibly also to neighboring regions with similar wave-propagation

characteristics and background noise. For each input waveform, the network provides both the signal and noise components. Hence, as pointed out by Zhu *et al.* (2019), it could also be a useful tool for separating noise from transient signals in noise correlation studies. Mousavi *et al.* (2016) and recently Langston and Mousavi (2019) have developed an approach that uses continuous wavelet transform to denoise seismic waveform data. A possible future extension of our work could consist of using wavelet transforms, instead of Fourier transforms, as the framework for generating signal and noise masks.

DATA AND RESOURCES

Waveform data used in this study were obtained from the Incorporated Research Institutions for Seismology database (<http://www.iris.edu>, last accessed April 2020) using the network code UU. The supplemental material contains a map showing the geographic locations of the stations BRPU, ZNPU, and SPU; a figure showing the training curve for the convolutional neural network (CNN); a figure displaying the distribution of signal-to-noise ratio (SNR) differences between the implemented CNN denoiser and band-pass filtering for a low-cut frequency of 2 Hz; a figure showing three raw seismograms, their respective filtered waveforms, signal mask operators from the CNN, and denoised waveforms; and a figure displaying two raw seismograms recorded at station MDJ in northeast China, the respective filtered and denoised waveforms.

convolutional neural network (CNN); a figure displaying the distribution of signal-to-noise ratio (SNR) differences between the implemented CNN denoiser and band-pass filtering for a low-cut frequency of 2 Hz; a figure showing three raw seismograms, their respective filtered waveforms, signal mask operators from the CNN, and denoised waveforms; and a figure displaying two raw seismograms recorded at station MDJ in northeast China, the respective filtered and denoised waveforms.

ACKNOWLEDGMENTS

This research was funded by the U.S. Department of Energy. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy’s National Nuclear Security Administration under Contract Number DE-NA-0003525. The views expressed in this article do not necessarily represent the views of the U.S. Department of Energy or the U.S. Government. This research was also funded by the National Nuclear Security Administration, Defense Nuclear Nonproliferation Research and Development (NNSA DNN R&D). The authors acknowledge important interdisciplinary

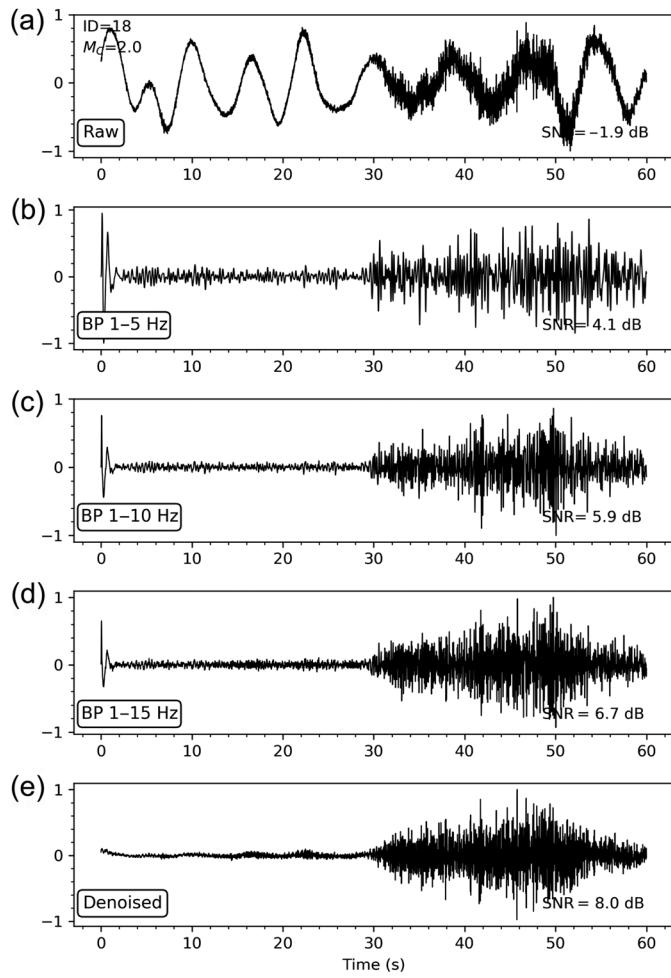


Figure 19. Same as Figure 9, but in this case, the timing of the predicted P arrival is 30 s after the start time.

collaboration with scientists and engineers from Los Alamos National Laboratory, Lawrence Livermore National Laboratory, Mission Support and Test Services, Pacific Northwest National Laboratory, and Sandia National Laboratories. The authors would like to thank Ben Baker, Lisa Linville, and Amy Sundermier for their comments on an earlier version of the article. The authors also thank Associate Editor Cleat Zeiler and two anonymous reviewers for their thorough reviews that help improve the article.

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 systems, available at [arxiv.org/pdf/1603.04467](https://arxiv.org/pdf/1603.04467.pdf) (last accessed September 2020).
- Bharadwaj, P., L. Demanet, and A. Fournier (2017). Deblending random seismic sources via independent component analysis, *SEG Technical Program Expanded Abstracts*, Society of Exploration Geophysicists, 4898–4902.
- Chandna, P., M. Miron, J. Janer, and E. Gómez (2017). Monoaural audio source separation using deep convolutional neural networks, *Proc. of International Conf. on Latent Variable Analysis and Signal Separation*, 258–266.
- Chollet, F. (2015). Keras, available at <https://github.com/fchollet/keras> (last accessed September 2020).
- Douglas, A. (1997). Bandpass filtering to reduce noise on seismograms: Is there a better way? *Bull. Seismol. Soc. Am.* **87**, 770–777.
- Goodfellow, I., Y. Bengio, A. Courville, and Y. Bengio (2016). *Deep Learning*, Vol. 1, MIT Press, Cambridge, Massachusetts.
- Grais, G. M., and M. D. Plumley (2017). Single channel audio source separation using convolutional denoising autoencoders, *IEEE Global Conf. on Signal and Information Processing (GlobalSIP)*, Montreal, Quebec, 1265–1269, doi: [10.1109/GlobalSIP.2017.8309164](https://doi.org/10.1109/GlobalSIP.2017.8309164).
- Han, J., and M. van der Baan (2015). Microseismic and seismic denoising via ensemble empirical mode decomposition and adaptive thresholding, *Geophysics* **80**, KS69–KS80.
- Huang, P.-S., S. D. Chen, P. Smaraglis, and M. Hasegawa-Johnson (2012). Singing-Voice separation from monaural recordings using robust principal component analysis, *IEEE International Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 57–60.
- Ioffe, S., and C. Szegedy (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift, available at [arXiv:1502.03167v3\[cs.LG\]](https://arxiv.org/pdf/1502.03167v3.pdf) (last accessed June 2020).
- Jansson, A., E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde (2017). Singing voice separation with deep U-Net convolutional networks, *Proc. of 18th International Society for Music Information Retrieval Conf.*, Suzhou, China, 745–751.
- Kingma, D. P., and J. Ba (2014). Adam: A method for stochastic optimization, available at [arXiv:1412.6980](https://arxiv.org/pdf/1412.6980.pdf) (last accessed September 2020).
- Langston, C. A., and S. M. Mousavi (2019). Separating signal from noise and from other signal using nonlinear thresholding and scale-time windowing of continuous wavelet transforms, *Bull. Seismol. Soc. Am.* **109**, 1691–1700.
- Liu, J.-Y., and Y.-H. Yang (2018). Denoising auto-encoder with recurrent skip connections and residual regression for music source separation, *IEEE International Conf. on Machine Learning and Applications*, 773–778, doi: [10.1109/ICMLA.2018.00123](https://doi.org/10.1109/ICMLA.2018.00123).
- Mandelli, S., V. Lipari, and P. Bestagini (2019). Interpolation and denoising of seismic data using convolutional neural networks, available at [arXiv:1901.07927v4\[cs.NE\]](https://arxiv.org/pdf/1901.07927v4.pdf) (last accessed March 2020).
- Mousavi, S. M., S. P. Horton, and C. A. Langston (2016). Automatic microseismic denoising and onset detection using the synchrosqueezed continuous wavelet transform, *Geophysics* **81**, no. 4, V341–V355, doi: [10.1190/geo2015-0598.1](https://doi.org/10.1190/geo2015-0598.1).
- Nakajima, H., Y. Takahashi, K. Kondo, and Y. Hisamimato (2018). Monoaural source enhancement maximizing source-to-distortion ratio via automatic differentiation, available at [arXiv:1806.05791v1\[cs.SD\]](https://arxiv.org/pdf/1806.05791v1.pdf) (last accessed September 2020).
- Pankow, K. L., M. Stickney, J. Y. Ben-Horin, M. Litherland, S. Payne, K. D. Koper, S. L. Bilek, and K. Bogolub (2020). Regional seismic network monitoring in the eastern Intermountain West, *Seismol. Res. Lett.* **91**, 631–646, doi: [10.1785/0220190209](https://doi.org/10.1785/0220190209).
- Simpson, A. J. R., G. Roma, and M. D. Plumley (2015). Deep karaoke: Extracting vocals from musical mixtures using a convolutional deep neural network, in *Latent Variable Analysis and Signal*

- Separation*, E. Vincent, A. Yeredor, Z. Koldovský, and P. Tichavský (Editors), LVA/ICA 2015, Lecture Notes in Computer Science, Vol. 9237, Springer, Cham, Switzerland.
- Srivastava, N., G. Hinton, A. Krizhevsky, and R. Salakhutdinov (2014). Dropout: A simple way to prevent neural networks from overfitting, *J. Machine Learn. Res.* **15**, 1929–1958.
- University of Utah (1962). University of Utah Regional Seismic Network, International Federation of Digital Seismograph Networks, Other/Seismic Network, doi: [10.7914/SN/UU](https://doi.org/10.7914/SN/UU).
- Zhu, W., S. M. Mousavi, and G. C. Beroza (2019). Seismic signal denoising and decomposition using deep neural networks, *IEEE Trans. Geosci. Remote Sens.* **57**, no. 11, 9476–9488, doi: [10.1109/TGRS.2019.2926772](https://doi.org/10.1109/TGRS.2019.2926772).

Manuscript received 16 September 2020

Published online 19 January 2021