doi:10.26443/seismica.v2i1.240

SEISMICA

Deep learning for denoising High-Rate Global Navigation Satellite System data

A. M. Thomas *[1], **D. Melgar** [1], **S. N. Dybing** [1], **J. R. Searcy** [2]

[1]Department of Earth Sciences, University of Oregon, Eugene, Oregon, USA, [2]Data Science Initiative, University of Oregon, Eugene, Oregon, USA

Author contributions: *Conceptualization*: A. M. Thomas, D. Melgar. *Methodology*: A. M. Thomas, J. R. Searcy. *Formal Analysis*: A. M. Thomas, D. Melgar. *Software*: A. M. Thomas, D. Melgar, J. R. Searcy. *Resources*: D. Melgar, S. N. Dybing. *Data Curation*: D. Melgar, S. N. Dybing. *Writing - original draft*: A. M. Thomas, D. Melgar. *Writing - review and editing*: A. M. Thomas, D. Melgar, S. N. Dybing, J. R. Searcy. *Project Administration*: A. M. Thomas, D. Melgar. *Funding acquisition*: D. Melgar, A. M. Thomas.

Abstract High-rate global navigation satellite system (HR-GNSS) data records ground displacements and can be used to identify earthquakes and slow slip events. One limitation of such data is the high amplitude, cm-level noise which makes it difficult to identify processes that produce surface displacements smaller than these values. Deep learning has proven adept at performing many useful tasks in seismology and geophysics. Here we explore using deep learning to denoise HR-GNSS data. We develop three different convolutional neural networks with similar architectures but different targets. Training data are synthetic HR-GNSS records and actual noise recordings that are superimposed to generate noisy signals. We train each of the three models to output masks that can be used to reconstruct the true signal. We use a set of performance metrics that quantify the models' ability to denoise the testing data and find that denoising significantly improves the signal-to-noise ratio and the ability to identify first arrivals. Finally, we test the models on HR-GNSS records from the Ridgecrest earthquakes recorded at stations that have nearly colocated strong-motion sites that can be used as ground-truth for the denoising results. We find that the models greatly improve the signal-to-noise ratios in these records and make the P-wave onset clearly identifiable.

Production Editor:
Gareth Funning
Handling Editor:
Mathilde Radiguet
Copy & Layout Editor:
Abhineet Gupta

Received:
September 13, 2022
Accepted:
March 07, 2023
Published:
May 04, 2023

1 Introduction

High-rate Global Navigation Satellite System (HR-GNSS) data record ground displacements at increments of $\geq 1\,\mathrm{Hz}$ and are a type of geophysical measurement commonly used to identify and characterize earthquakes and slip transients (Bock and Melgar, 2016; Larson, 2019). GNSS data can augment traditional seismic data and are superior for some applications. For example, long-period recordings of earthquakes contain diagnostic information about earthquake magnitude (e.g., Melgar et al., 2015). This property of GNSS is used extensively for down-stream applications such as earthquake early warning (e.g., Murray et al., 2018) and tsunami early warning (e.g., Williamson et al., 2020). Inertial recordings of earthquakes suffer from a well-known magnitude saturation problem; recordings of the first arriving seismic waves cannot distinguish between large and very large magnitude earthquakes (e.g., Trugman et al., 2019). In contrast, GNSS faithfully record ground motions at long-periods, including zero-frequency static offsets, and do not suffer from this limitation (e.g., Crowell et al., 2013). Additionally, for modeling finite faults for the largest events it has become commonplace to include HR-GNSS in post-processing (e.g., Goldberg et al., 2020). The data provide fundamental constraints on total moment and on the spatial distribution of slip. For large events it is expected that a credible rupture model will be capable of reproducing these kinds of observations (e.g., Satake and Heidarzadeh, 2017; Lay, 2018). As a final example, some types of crustal deformations such as slow earthquakes and fault creep have long duration source processes and do not effectively generate seismic waves (e.g., Thomas et al., 2016). These slow deformation processes are often identified by a gradual displacement of the Earth's surface, hence characterization of these events relies on GNSS data.

One of the main limitations of HR-GNSS data is the high noise level of $\sim$1–2 cm both in post-processing and in real-time (Geng et al., 2018; Melgar et al., 2020). This limits the utility of HR-GNSS recordings to situations where ground displacements are in excess of these amplitudes such as moderate to large earthquakes at local to regional distances. Several factors contribute to the high noise level of HR-GNSS data. This includes measurement noise, the number and location of satellites, model estimates of orbital errors, satellite and receiver clocks, atmospheric delays, antenna effects, and multipath errors (Melbourne et al., 2021). Several methods have been proposed to eliminate particular forms of noise. For example, sidereal filtering leverages repeating satellite-receiver geometries to correct for noise resulting from multipath errors (i.e., when a transmitted signal arrives at a receiver via an indirect path). Simply stated this technique involves taking displacements recorded during the previous orbital repeat pe-

---

*Corresponding author: amt.seismo@gmail.com riod (the time since the satellite constellation was last in the same configuration), applying a low pass filter (e.g. $11\,\mathrm{s}$ corner frequency), and subtracting the filtered displacement record from the displacement recorded at the present time (e.g., Choi et al., 2004). Spatial filtering targets common-mode noise that is highly correlated across GNSS stations in close spatial proximity (Wdowinski et al., 1997). This technique simply averages detrended records on many nearby stations and the resulting average is subtracted from each station. Principal component analysis (e.g., Dong et al., 2006; He et al., 2015) has been employed to remove long period noise (0.2–$0.1\,\mathrm{cycles/year}$) and various match-filtering approaches have also been employed to reduce noise levels (Frank, 2016; Rousset et al., 2019). With respect to HR-GNSS, there are a dearth of denoising techniques that are both applicable to high rate data and are efficient. For example, sidereal filtering is one of the most commonly employed techniques but because of the low-pass filtering, it does not apply to frequencies higher than the chosen corner frequency. Additionally, Geng et al. (2017) noted that sidereal filtering can also increase noise levels for periods between 20 and $33\,\mathrm{s}$ (those authors used a $10\,\mathrm{s}$ corner frequency). While data-driven denoising strategies are capable of mitigating high-frequency noise (e.g., Li et al., 2018), as proposed, they involve multiple decompositions using techniques such as empirical mode decomposition, which is known to be computationally time consuming.

Machine learning is adept at many commonly performed tasks in seismology and crustal deformation. For example, deep learning (DL) can identify and make phase picks on small magnitude earthquakes (e.g., Zhu and Beroza, 2019; Thomas et al., 2021). DL can recognize crustal deformation patterns measured with HR-GNSS data and accurately estimate the earthquake moment in real time (Lin et al., 2021). Additionally, and of particular interest for the present study, DL methods have been applied to denoise various types of geophysical data. For example, Zhu et al. (2019) developed an algorithm known as Deep Denoiser which they showed drastically improved the signal-to-noise ratio (SNR) in earthquake seismograms. We discuss this method in more detail in Section 2. Additionally, Ende et al. (2021) developed a methodology to denoise distributed acoustic sensing data. Since GNSS measures position and the decomposition into components is arbitrary, noise is often correlated across components on the same station similar to seismic noise. This suggests that deep learning may also be adept at denoising GNSS records.

Motivated by these recent results, here we explore using DL to denoise HR-GNSS records. In Section 2 we discuss the methodology including the generation of training and testing data, network architectures, and the training procedure. In Section 3 we discuss the results of the methodologies including performance on both the training data and real records from the 2019 Ridgecrest, California earthquake sequence. In Section 4 we discuss the successes and limitations of the methods, potential applications, and future directions. Overall, DL appears to be a promising path forward to increasing the utility of HR-GNSS data in rapidly identifying and assessing source properties of intermediate to large magnitude earthquakes and lowering the noise threshold of HR-GNSS data for detecting small magnitude transient slip events.

2 Methods

2.1 Training data for denoising

As a testbed for our proposed approach we focus on the recent Ridgecrest, California earthquake sequence that commenced in July, 2019. This energetic set of earthquakes included an M6.4 foreshock on a NE striking fault plane, followed 34 hours later by an M7.2 mainshock on an adjacent, conjugate fault striking NW (e.g., Ross et al., 2019; Goldberg et al., 2020). The GNSS network surrounding Ridgecrest, shown in Fig. 1A and B, includes multiple near-field stations close enough to record static offsets in both the M6.2 and M7.2 events. Both events were followed by energetic aftershock sequences that included four M5+ events (Fig. 1A and B; Shelly, 2020).

To create an algorithm capable of separating signal from noise we need many thousands of examples of both signal (i.e., HR-GNSS displacement time series from real earthquakes) and noise waveforms. For the noise recordings, we use three-component timeseries of real noise recorded on the 278 GNSS stations closest to the Ridgecrest events, shown in Fig. 1A. These data are from the Network of the Americas (NOTA) sites operated by UNAVCO and the positions are true real-time solutions produced and archived by UNAVCO (Hodgkinson et al., 2020). The timing of the noise waveforms was selected at random times from days between June 1, 2019 to July 31, 2020 that did not have an earthquake above M4.3. This magnitude is a conservative lower bound for the size of an earthquake that one may expect to generate an observable signal in the GNSS data.

Obtaining a similar number of real "signal" waveforms over a range of magnitudes is challenging for two reasons. First, unlike seismic data, which can have SNRs of many orders of magnitude, the high noise floor of GNSS data makes it such that even the largest amplitude displacements have relatively modest SNR and still contain appreciable noise. Second, only large magnitude earthquakes recorded in the near-field generate signals of sufficient magnitude to exceed the noise floor. This is problematic because we would like to include signals that are at or below the noise level (i.e., $\mathrm{SNR} \leq 1$) to train an algorithm capable of detecting small amplitude displacements. These limitations prohibit using real signals for training, and so we turn to generating realistic, synthetic displacements.

The *FakeQuakes* code (Melgar et al., 2016) generates stochastic ruptures using the assumption that slip on a fault is distributed like a random field (Mai and Beroza, 2002). If the statistics of this field are known then random draws can be made to generate arbitrarily large numbers of ruptures. This approach is common in strong motion seismology (e.g., Graves and Pitarka, 2010; Goulet et al., 2015). As is typical, we assume a VonKarman correlation function (VKCF). The along-

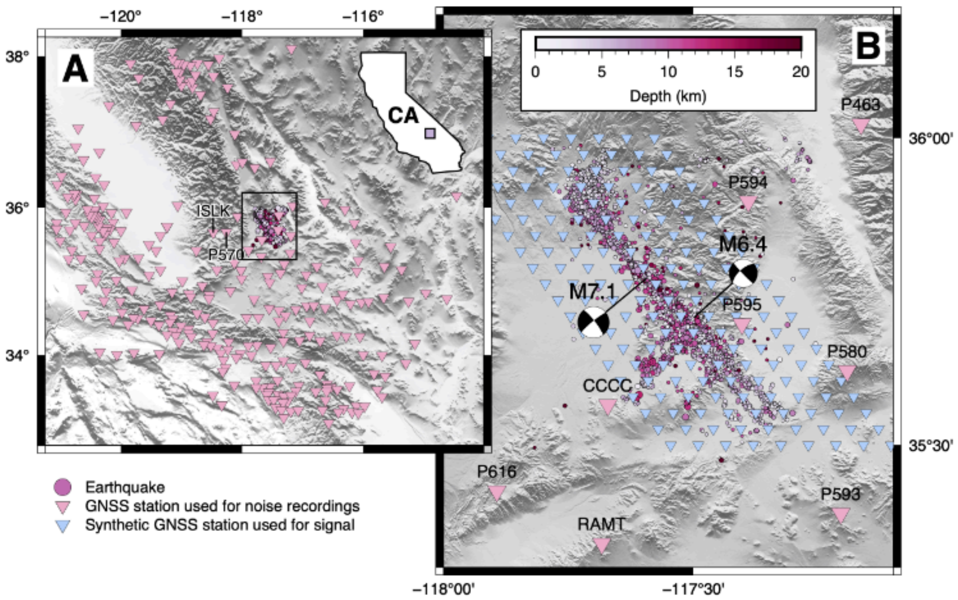

Figure 1 Maps of the Ridgecrest area. Panel A shows a regional map with GNSS stations used for noise recordings (pink) and synthetic signal recordings (blue) shown as inverted triangles. Ridgecrest earthquakes are outlined by the black square. The locations of stations P570 (Fig. 14) and ISLK (Fig. 15) are annotated. Inset shows an outline of the state of California with the region location indicated by the purple square. Panel B shows a closeup of the Ridgecrest earthquake sequence. Events are color coded according to their depth and scaled by magnitude. Focal mechanisms for the M7.1 and M6.4 are also shown.

strike and along-dip correlation lengths of the VKCF control the predominant sizes and aspect ratios of asperities in the slip patterns. Meanwhile the Hurst exponent, H, of the VKCF determines the amount of short-wavelength structure in between them. Mai and Beroza (2002) proposed scaling laws for the correlation lengths which depend on fault dimension, and fixed $H = 0.7$. Here we use updated scaling laws from an analysis of the U.S. Geological Survey's database of finite faults (Melgar and Hayes, 2019) and we use $H = 0.4$. A full description of the stochastic approach can be found in Melgar et al. (2016). Once the slip patterns are created, kinematic rupture properties such as rise time, rupture velocities, etc., are set following Graves and Pitarka (2015). We assume a 1D layered Earth model for Eastern California used in the Southern California Earthquake Center's Broadband Platform (Goulet et al., 2015) and produce the waveforms using a frequency-wavenumber approach to generating elastodynamic Green's functions (GFs; Zhu and Rivera, 2002). For this work we introduce an additional modification. After comparing our kinematic rupture waveforms with HR-GNSS data from the Ridgecrest earthquakes we noticed that many waveforms from the Ridgecrest earthquakes had significant "ringing" or long duration coda that was not present in the synthetic waveforms. We hypothesized that these waveforms were affected by either site or basin effects. Since we are not using fully 3D GFs in order to produce waveforms with long durations we created an extra family of "soft layer" waveforms. We varied the thickness of the soft layer between 100 and $900\,\mathrm{m}$ and allowed shear wave speeds as slow as $100\,\mathrm{m\,s^{-1}}$ and as fast as $900\,\mathrm{m\,s^{-1}}$. These two families of waveforms, the ones without and the ones with the soft layers, are then used for training.

The *FakeQuakes* output is then three-component ground displacement timeseries from earthquakes with magnitudes ranging from 4 to 7.2 that rupture the 3D fault geometry of Goldberg et al. (2020). The displacement timeseries from each earthquake are calculated at the 180 synthetic GNSS station locations shown in blue in Fig. 1B. The station distribution shown in Fig. 1B and magnitude range were initially chosen so that the resulting displacement timeseries are not dominated by very small amplitude displacements and instead represent more uniform sampling of peak ground displacement. At each station, displacement waveforms are calculated at $1\,\mathrm{sample/s}$ for a $256\,\mathrm{s}$ window centered on the P-wave arrival expected from ray-tracing through the velocity model. North, east, and vertical displacements for 20 example "fakequakes" are shown in Fig. 2A, B, and C respectively. Despite this selection of magnitudes and station locations, small magnitude peak ground displacements (PGD) still dominate the dataset so we implemented a culling procedure such that we have a nearly uniform distribution of PGD up to $1\,\mathrm{m}$ and then exponentially decreasing numbers of events with PGD $> 1\,\mathrm{m}$. Overall our dataset includes $822\,256$ three-component signal records and $729\,303$ three-component noise records.

2.2 Network architecture

Convolutional neural networks (CNNs) are a deep learning architecture capable of efficiently extracting diagnostic information from multidimensional images. The essential component of any CNN is one or more convolutional layers. In these layers, the user defines both the number of filters and the filter size. The network then convolves these filters with the layer input to create a feature map which is further manipulated in subsequent network layers. Filters can be defined a-priori

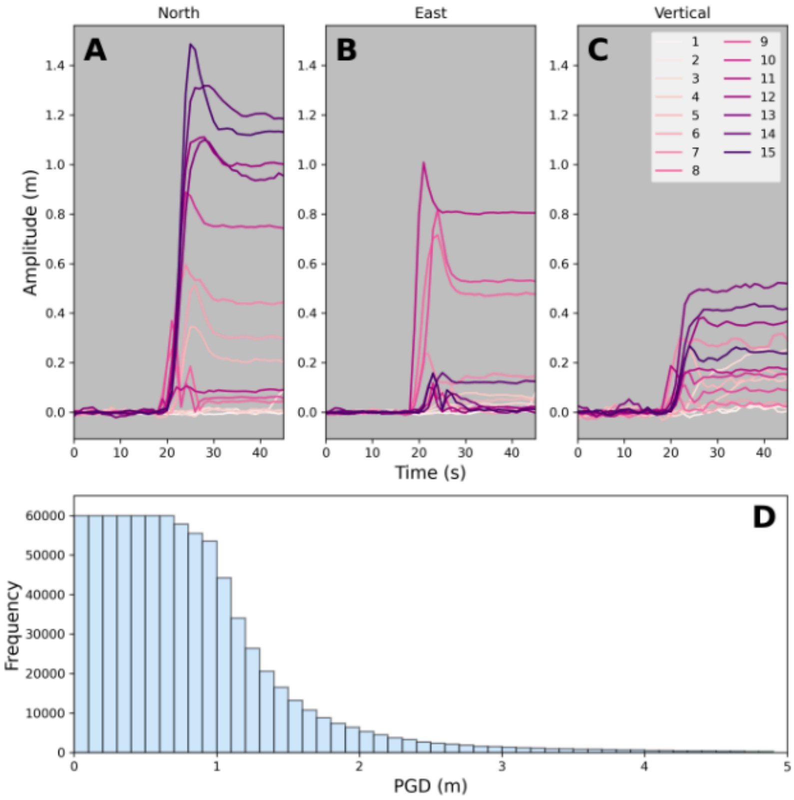

Figure 2 GNSS displacement timeseries and PGD distribution. Panels A, B, and C show north, east, and vertical displacements for 15 different simulated earthquakes, color coded by the example number. Panel D shows PGD distributions for all "fakequake" displacement records in the dataset.

but most often they are learned during training. A U-Net is a special case of a CNN with symmetric contracting and expanding branches (Ronneberger et al., 2015). The contracting branch includes repeated convolutions with a 3x3 filter, rectified linear unit activation functions (Agarap, 2018), and batch normalization. In the contracting branch, each successive convolutional layer has a stride of two and double the number of filters in the previous layer. This results in feature maps that decrease in size but increase in number as network depth increases. In the expanding branch, the filter number is halved while the convolutions upsample their respective feature maps resulting in larger maps after each successive convolutional layer. Skip connections also feed outputs from the contracting branch to inputs of the expanding branch at the same depth in the network, skipping the deeper layers.

Zhu et al. (2019) employed a U-Net called Deep Denoiser to denoise single channel seismic data. In their denoising approach, the noisy signal, $Y(t)$ is the sum of a known signal, $S(t)$, and known noise, $N(t)$, or

$$Y(t) = S(t) + N(t) \tag{1}$$

In the frequency domain, the short-time Fourier Transform (STFT) of the noise, $N(t,f)$, and the STFT of the signal, $S(t,f)$, are summed to represent the STFT of the noisy signal

$$Y(t,f) = S(t,f) + N(t,f) \tag{2}$$

Then the signal mask is defined as

$$M_S(t,f) = \frac{1}{1 + \frac{|N(t,f)|}{|S(t,f)|}} \tag{3}$$

And the noise mask is defined as

$$M_N(t,f) = \frac{\frac{|N(t,f)|}{|S(t,f)|}}{1 + \frac{|N(t,f)|}{|S(t,f)|}} \tag{4}$$

Each mask has the same dimensions as the input STFTs and takes on values between 0 and 1. The network is trained to output an estimate of the signal and noise masks, $\hat{M}_S(t,f)$ and $\hat{M}_N(t,f)$. From these, the estimated signal, $\hat{S}(t,f)$ can be obtained by multiplying $Y(t,f)$ by the respective mask and inverse transforming

$$\hat{S}(t) = STFT^{-1}\left\{Y(t,f)\hat{M}_S(t,f)\right\} \tag{5}$$

Similarly, the estimated noise, $\hat{N}(t,f)$ can be obtained by multiplying $Y(t,f)$ by the noise mask and inverse transforming

$$\hat{N}(t) = STFT^{-1}\left\{Y(t,f)\hat{M}_N(t,f)\right\} \tag{6}$$

Zhu et al. (2019) showed that Deep Denoiser could reliably denoise seismic signals, resulting in significant increases in SNR in the denoised waveforms thereby improving earthquake detection capabilities.

Motivated by the results of Zhu et al. (2019), we develop a similar U-Net, which can be accessed in the repository linked in the Data Availability section, with some noteworthy modifications. First, Deep Denoiser operated on single channel seismic data. For accurate denoising of HR-GNSS data, we anticipate that using all three channels as network input will significantly improve the results given the higher noise levels on the vertical channels and the fact that noise is very highly correlated across channels. As such, each network we present here has inputs derived from three component HR-GNSS data. Second, a notable attribute of Deep Denoiser is that it can distort signal amplitudes during the denoising process, reducing their true amplitudes (Tibi et al., 2021). For some applications, such as earthquake detection and arrival time picking, this amplitude distortion is not a limitation. However, if denoising of HR-GNSS were to be used for early-warning or source studies, amplitude distortion could underpredict early earthquake magnitude estimates. As such we explore three different model variations, described in the next section.

2.3 Inputs, outputs, and model variations

The inputs to each of our networks are generated in the following way. The HR-GNSS data we employ here is sampled at 1 Hz. The noise and signal records described above are 256 s long originally. Leaving the training data in the original form, with the P-wave pick in the middle, would bias the resulting CNN to assign noise prior to the middle of each window and signal thereafter. As such, we use a data generator during training which applies the following modifications to the data prior to input to the network. First, we randomly select a set of N signal timeseries and N noise timeseries, where N is the batch size. These timeseries are added to simulate noisy signals. Second, we randomly select a start-time in the first half of each trace and include only 128 s of data beginning at that time. This has the effect of randomly shifting the earthquake onset in time such that it can occur at any point during the window. The signal can also be shifted such that the window is mostly or entirely noise. The combinations of signal and noise and the time-shifting are data augmentation strategies that significantly increase the size of the training dataset. Third, we compute the STFT of the signal timeseries, the noise timeseries, and the noisy signal timeseries which results in complex, 2D time-frequency representations of the timeseries for each component. To compute the STFT we use a segment length of 31 and 30 points of overlap between segments. The real and imaginary parts of the noisy signal are normalized by dividing the STFTs for each component by the maximum of their absolute value. This accounts for the differences in amplitude of the input data which can vary over several orders of magnitude. The final input to the CNN is six channels (the real and complex parts of the STFT for the north, east, and vertical components), each with dimension 16×128.

We train three different model versions to denoise the HR-GNSS data. In the first model, which we will call Model 1, we adopt an approach very similar to Deep Denoiser (Zhu et al., 2019) The only modifications are that the network ingests six channels of data and we predict only one signal mask for each channel. The noise mask can be determined by taking the complement of the signal mask. For Model 1, the activations on the final layer are sigmoids to accommodate the three channel mask outputs (in the one-component version of Deep Denoiser, the final layer activations were softmax functions). To determine the estimated signal from the input signal, we simply use equation 5.

Model 2 is motivated by Model 1's limitations. Namely that by using a real-valued mask, we cannot account for phase differences between signal and noise which can result in amplitude distortion. As such, in the final layer we use linear activation functions and the model predicts the real and imaginary parts of the complex valued $S(t, f)$ normalized by the maximum absolute value of $Y(t, f)$. We can obtain the signal estimate by multiplying the output by the maximum absolute value of $Y(t, f)$, which can be calculated from the input data, and inverse transforming.

Finally, in Model 3, we explore predicting the ratio $S(t, f)/Y(t, f)$. Model 3 was also motivated by the desire to account for phase differences and minimize amplitude distortion but employs a similar masking technique as Deep Denoiser. Because this ratio can be infinite, we convert the complex representation of the STFT to amplitude and phase and train the network to output the scaled amplitude and phase ratios. Specifically the first output is defined as

$$A(S(t, f), N(t, f)) = \ln\left(|S(t/f)/Y(t/f)| + \epsilon\right) \tag{7}$$

where ϵ is a small magnitude constant added to ensure that the target is never infinite; we set $\epsilon = 10^{-9}$. Direct prediction of the phase is challenging in machine learning because of the modular property of angular measurements. To address this modularity, we use targets for phase prediction defined as

$$\phi_1(S(t, f), N(t, f)) = \cos(\theta) \tag{8}$$

and

$$\phi_2(S(t, f), N(t, f)) = \sin(\theta) \tag{9}$$

where

$$\theta = \tan^{-1}\left(\frac{\mathrm{Im}(S(t, f)/Y(t, f))}{\mathrm{Re}(S(t, f)/Y(t, f))}\right) \tag{10}$$

The functions A, ϕ_1, and ϕ_2 are the targets for each component resulting in nine total predictions for the third model. Like Model 2, Model 3 employs linear activation functions and the estimated signal can be determined using equations 7-10.

Finally, to explore how variations in network size affect performance we vary slightly the network architecture by increasing the number of filters in the convolutional layers by a factor of two.

2.4 Training

We train the three different models for 50 epochs each using Tensorflow (Abadi et al., 2016). In all cases we set aside 10 % of the signal and noise waveforms for testing and train using the remaining 90 %. For all networks, we use the Adam optimization algorithm (Kingma and Ba, 2014) with a learning rate of 0.0001, we also use a drop rate of 0.2. All models used a mean-squared error loss function on all targets.

2.5 Performance metrics

We use three different metrics to quantify model performance. First, we employ the normalized cross-correlation coefficient to measure the similarity between the signal and the model predicted signal, denoted as CC. CC varies between $+1$ when two signals are exactly correlated and -1 when they are exactly anticorrelated. We note that the normalized cross-correlation does not account for amplitude differences between signals. This choice was deliberate such that the inherent amplitude distortion does not bias the result. Second, we compute the SNR using

$$\mathrm{SNR} = \frac{\max(|\mathrm{signal}|)}{2 * \sigma_{\mathrm{noise}}} \tag{11}$$

Here $\max(|\mathrm{signal}|)$ is the maximum absolute value of the signal waveform and σ_{noise} is the standard deviation of the noise evaluated on the time window prior to the P-wave arrival. This metric has a value of $\sim$1 when the signal and noise amplitudes are equal, and is greater (or less) than one when the signal amplitude exceeds (or is less than) the noise amplitude. We note that some other common measures of SNR utilized on seismic data often employ the standard deviation in both the numerator and denominator but this metric does not properly account for the static offsets that represent signal in GNSS data. We also note that the SNR metric can become infinite if the σ_{noise} approaches zero. When we use SNR to quantify general attributes of the waveforms prior to denoising, the numerator of equation 11 is evaluated on the signal waveforms alone while the denominator is evaluated on the noise. We also quantify the change in SNR, $\Delta\mathrm{SNR}$, between noisy and denoised waveforms. We evaluate $\Delta\mathrm{SNR}$ by computing SNR on the noisy signal (i.e., signal + noise) and on the denoised signal and subtracting the two. For all SNR calculations, we require that signal and noise be at least 10 s in duration. Third, we use the Euclidean or L^2 distance as a measure of the difference between the signal and the noisy signal or the denoised signal. The L^2 distance between two vectors $\mathbf{x}$ and $\mathbf{y}$, with components x_i and y_i respectively, is defined as

$$\mathrm{L}^2(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_i (x_i - y_i)^2} \tag{12}$$

As an additional test of applying our simulated training data to real HR-GNSS records, we compare the original HR-GNSS records, denoised HR-GNSS records, and displacements derived from strong motion accelerograms recorded during the Ridgecrest, California earthquake sequence. These GNSS and strong motion stations are the only closely located pairs in proximity to the Ridgecrest earthquakes. For small to moderate ground motions, the strong-motion accelerograms can be used to estimate the three-component ground displacements. For extremely strong ground motions, baseline offsets make the displacement records inaccurate, sometimes by widely large amounts, at periods longer than $\sim$10 s (e.g., Melgar et al., 2013).

We focus our analysis on two site pairs: P570 and WOR, which are separated by 3.5 km and are 59 km from the M7.1 mainshock, and ISA and ISLK, which are 62 m apart and 80.5 km from the mainshock. We take strong motion records from WOR and ISA, integrate twice and correct for gain. We then bandpass filter the strong motion data, raw HR-GNSS, and denoised HR-GNSS between 3 and 15 s for comparison. The results of this analysis are described further in Section 3.

3 Results

3.1 Denoising examples

For each model we select from the testing dataset as denoising examples two records with peak signal amplitudes below 4 cm, since the denoising process will be most useful for these low amplitude signals, and one higher amplitude signal with peak amplitude above 4 cm. As such, the low and higher amplitude signals have different characteristics. For example, the low amplitude signals are typically dominated by transient seismic wave arrivals and have low-amplitude or zero static offset, whereas the higher amplitude signals contain transient features in addition to large amplitude permanent offsets. We also selected examples to showcase model performance on different types of noise, targeting samples with varying amplitude and frequency content.

Example denoising results are shown in Figs 3-11. Figs 3-4 show a low-amplitude example from Model 1. Each row in the figures correspond to a different component: north, east, and vertical. The first column shows the signal waveform in blue and the summed signal and noise in gray. The annotation indicates the SNR of the noisy signal. The second column shows the target signal mask, $M_S(t, f)$, while the third column shows the model predicted mask, $\hat{M}_S(t, f)$. The fourth column is identical to the first but includes the denoised waveform in magenta with the CC and $\Delta\mathrm{SNR}$.

Figs 3-5 show the performance of Model 1 for multiple examples. In Fig. 3, the SNR has comparable values of 2.4 and 1.9 on the north and east components, and lower value of 0.4 on the vertical. After applying Model 1 to denoise, the resulting waveforms have a high CC with the known signal on the horizontals and denoising also recovering much of the signal character for the vertical component, which has $\mathrm{CC} = 0.674$ between the true and denoised vertical components. All components have significant improvements in SNR. Fig. 4 shows another low-amplitude example similar to Fig. 3 but with more complex signal character. The model still recovers much of it, i.e., arrival time, amplitudes, and frequency content, on all three components. Such low amplitudes, like that on the vertical, are sometimes not recoverable (i.e., the model predicts noise) however the model can utilize higher amplitude signals on the

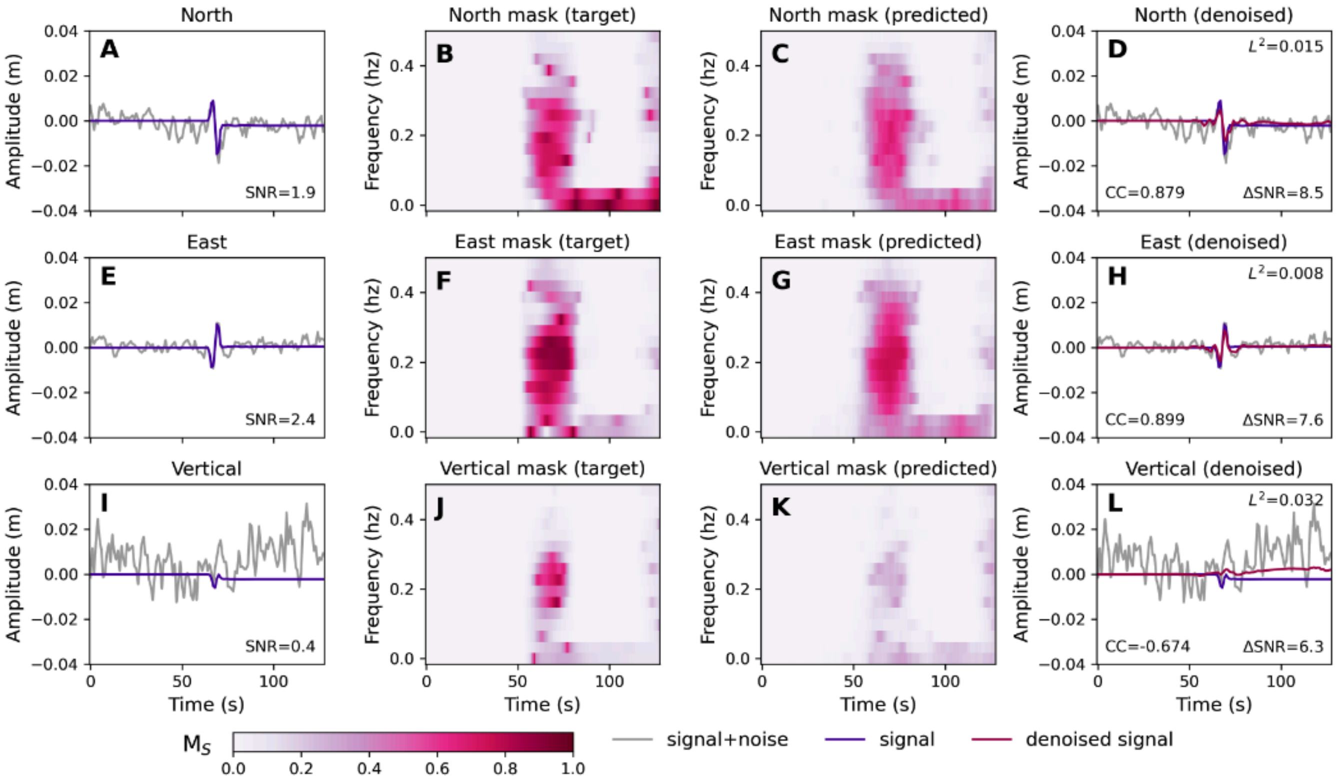

Figure 3 Example denoising using Model 1. Panels A-D, E-H, and I-L correspond to the north, east, and vertical components respectively. The first column shows the signal timeseries (blue) and the noisy signal timeseries (i.e., signal+noise) in gray. The SNR and L^2 distance from equations 11 and 12 respectively are annotated. The second column is the target mask, M_S, for each component. The third column shows the predicted mask for each component. The fourth column shows the signal (blue), noisy signal (gray), and denoised signal (magenta) along with the CC between the signal and denoised signal and the $\Delta\mathrm{SNR}$ between the noisy and denoised signals.

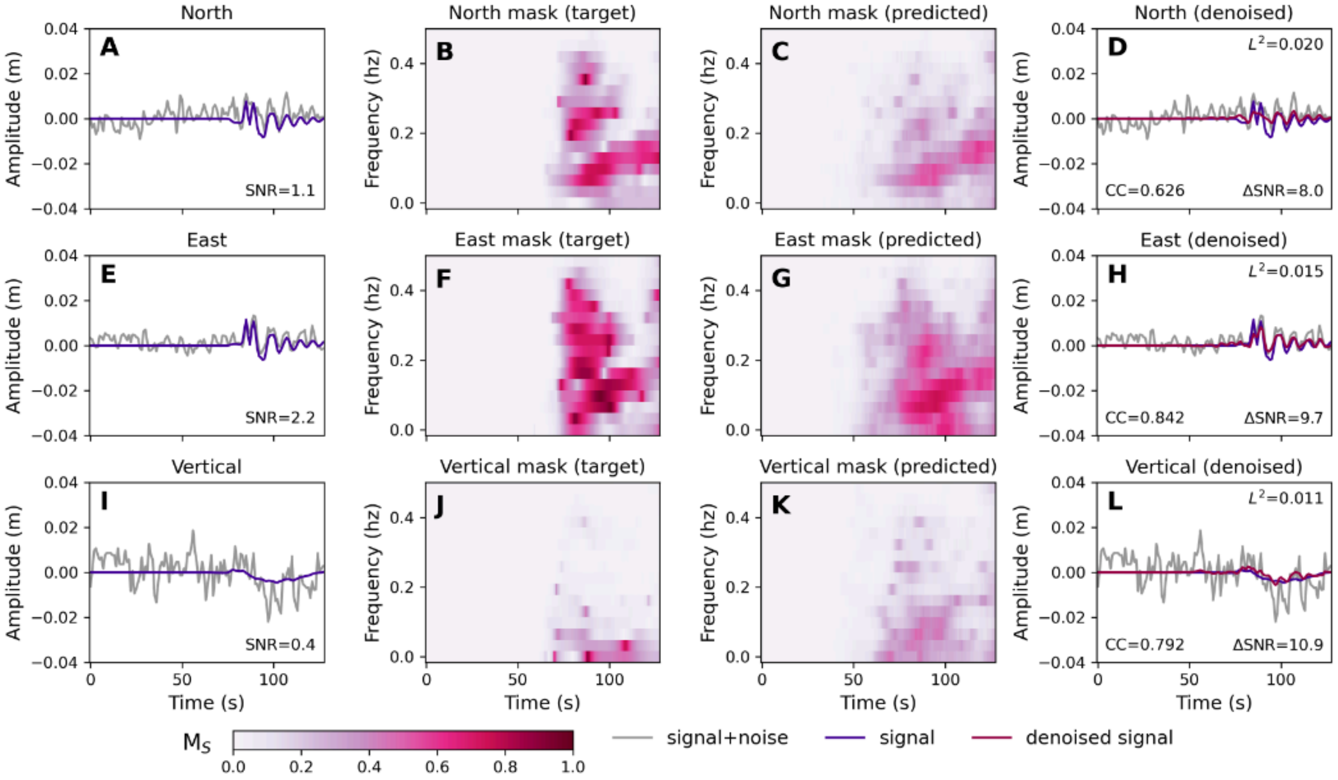

Figure 4 Example denoising using Model 1. Panels are the same as those in Fig. 3.

higher amplitude components to recover some signal character on the components with lower amplitude signals (this is also apparent in Fig. 5). Figs 3 and 4 contain two noteworthy features. First, the signal amplitudes are sometimes underpredicted by Model 1 (see Fig. 3D, H and L and Fig. 4D and H). Second, sometimes the static offset is not well predicted by the model but instead fluctuates with the amplitude of the noisy signal (Panel L in Fig. 3). We discuss these properties more in Section 4. Finally, Fig. 5 shows a higher amplitude demonstration of Model 1 (note the difference in scale between Figs 3-4 and Fig. 5). Denoising essentially does not modify the large amplitude signals on the horizontal components. The main benefit of denoising is to both reduce the amplitude of the noise prior to the first arrivals and to increase the SNR on components with lower signal.

Since the masks predicted by Models 2 and 3 are complex valued, in Figs 6-11 we only show the time domain representations of the noisy signal, signal, and estimated signal (panels D, H, and L of Figs 3-5). Figs 6 and 7 show low amplitude denoising examples using Model 2. In Fig. 6, the three components have SNRs of 0.46, 1.22, and 0.95. Despite low SNR, denoising successfully recovers signal on all components with high CC between all original and estimated signals. Fig. 7 is similar to Fig. 6 showing another example of retrieval of the signal when SNR is low. In particular, the vertical component is well recovered in Fig. 7 despite the high-amplitude and low frequency character of the noise (Fig. 7C). Also note that on the horizontal components, there is constructive interference between the signal and noise (Fig. 7B). In some cases, such as the later arrivals on the north component and the east component, Model 2 is able to successfully partition signal and noise and predicts the true signal amplitude (Fig. 6A and B). In other cases, such as the vertical component in this same example (Fig. 6C), Model 2 can underpredict the signal amplitudes and long term signal character, such as static offset can be influenced by noise. Fig. 8 is a high amplitude example with complex signal character. Model 2 performs well, particularly in predicting the low amplitude vertical component.

Figs 9-11 show examples of denoising using Model 3. Figs 9 and 10 show significant improvements in horizontal SNR even given the relatively low amplitude of the signals. Additionally both examples have high correlations between the true and predicted signals on the horizontal components. While Model 3 generally succeeds on the east component in Fig. 11, the model fails to predict the static offset in the denoised north and vertical signals causing them to deviate from the true static offset. This is commonly observed in Model 3. Figs S1-S10 show the performance of all three models on some common examples.

3.2 Model performance on the testing data

To evaluate the overall performance of each model on the testing dataset ($N = 82\,226$) we compute the CC between the true and denoised signals as a function of the SNR. We then bin the data by SNR and compute the 10th, 50th, and 90th percentiles of the CCs for the signals contained in each bin. We also compute the median CC between the noisy and true signals as a function of SNR. The results are shown in Fig. 12 which includes panels for individual components and all signals combined. While many horizontal components have $\mathrm{SNR} > 6$, we chose this particular cutoff to highlight model behavior at lower amplitudes. As expected, model performance improves significantly with SNR. At very low SNR, there is a large spread in CC for all models. For SNRs of 0.5 to 2, the models have significantly larger CCs than the noisy data with the true signal. As SNR increases further, this difference diminishes as the noise becomes a smaller fraction of the overall signal and CCs become high approaching 1 at very large SNR. For all components, Models 1 and 3 have median CCs of 0.51 and 0.41 for an SNR of 0.5. These values increase to 0.78 and 0.74 at an SNR of 1. Model 2 has better performance with a $\mathrm{CC} = 0.70$ at $\mathrm{SNR} = 0.5$ and $\mathrm{CC} = 0.89$ at $\mathrm{SNR} = 1$. For the horizontal components alone, we note that the percentiles are determined from a smaller numbers of events. This is because the horizontal components typically have larger amplitudes for the strike-slip geometry that we considered in this study and lower noise levels than the vertical components, hence there are fewer horizontal records that satisfy the $\mathrm{SNR} < 6$ criterion. Additionally, we note that at very low SNR (i.e., those < 1), the vertical components typically have higher CC than the horizontal components. This is due to two factors. First, the models can use higher amplitude information recorded on the horizontal components to make predictions about content on the vertical component despite the higher amplitude noise. These predictions often result in higher CC than that between the true and noisy waveforms. Second, the high amplitude noise in the noisy waveforms can result in lower CC with the true signal than does a prediction with a near zero value prior to the first arrival and a low amplitude signal, even if the predicted signal departs significantly from the true signal.

Similar to Fig. 12, Fig. 13 shows the L^2 distances between the true and denoised signals as a function of the SNR. The data are binned by SNR and we compute the 10th, 50th, and 90th percentiles of the L^2 distances for the signals contained in each bin. The black line shows the distribution of L^2 distances between the noisy signal and true signal, while the colored lines show the distances between the true signal and those estimated using Models 1, 2, and 3. For all models and components, denoising results in smaller L^2 distances between the true signal and denoised signal than the noisy data independent of SNR. On the horizontal components, the distances between denoised and true signals are small at low SNR because the denoising algorithm generally predicts noise (i.e., zero amplitude true signal) in these scenarios. As the SNR increases and the models make more non-zero signal predictions the L^2 distances increase until, at intermediate to high SNR they plateau. In this region, model distances are still less than the original distances likely due to the zero amplitude prediction of the models prior to the first arrivals. The overall larger distances on the vertical result from the higher

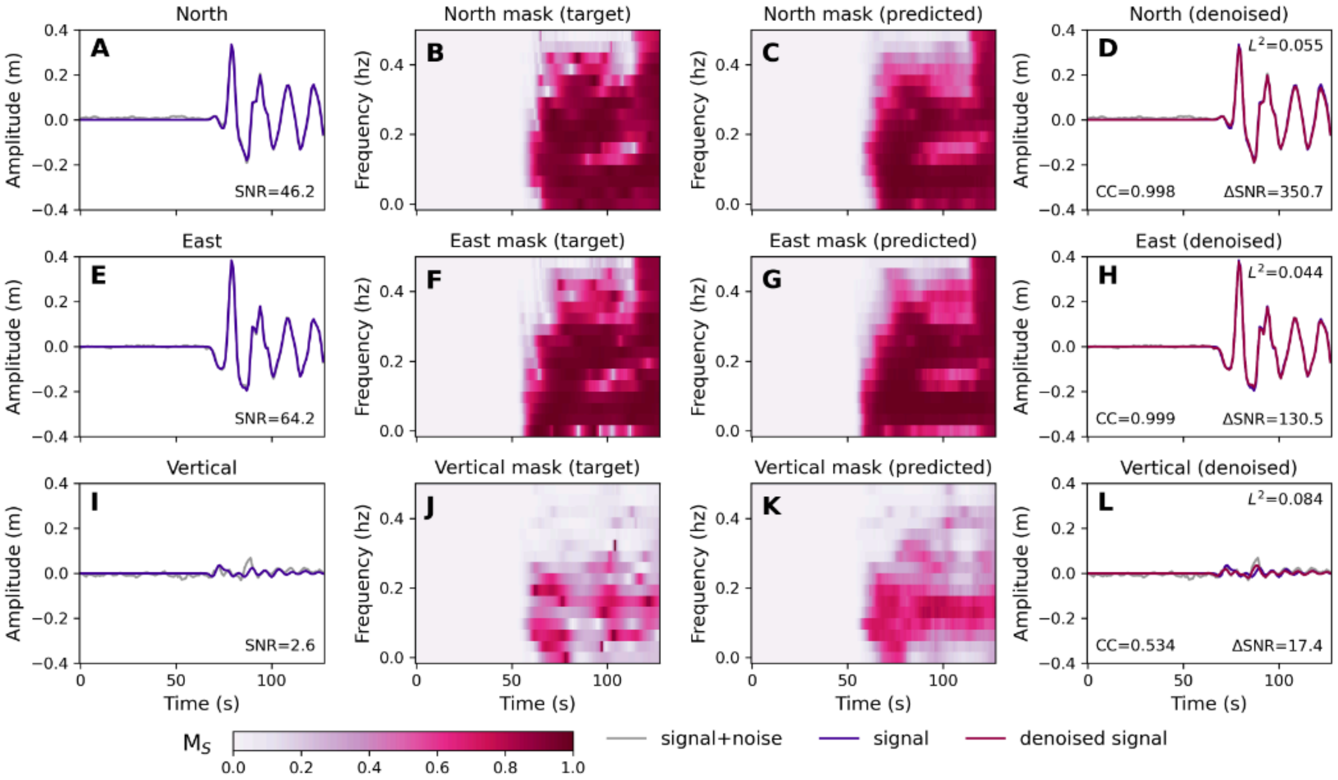

Figure 5 Higher amplitude example denoising using Model 1. Panels are the same as those in Fig. 3.

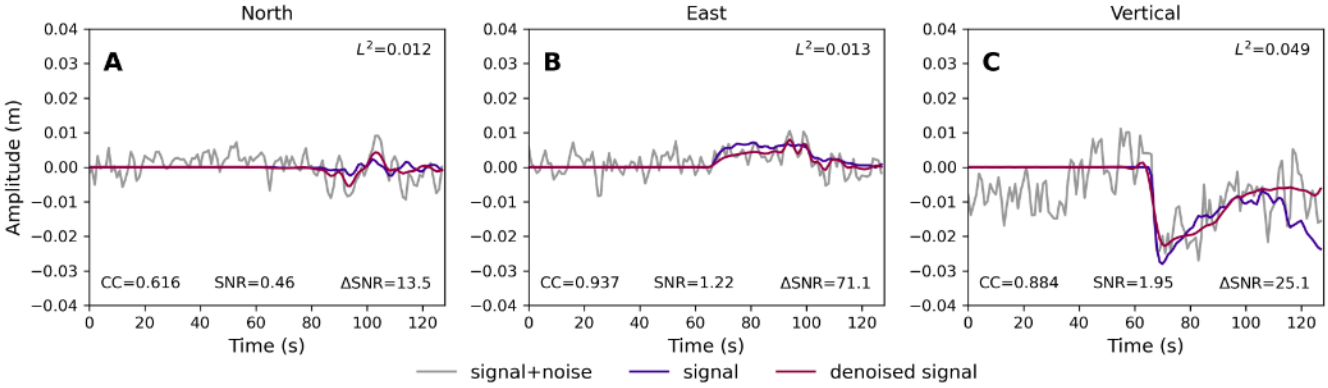

Figure 6 Example denoising using Model 2. Panels A-C correspond to the north, east, and vertical components respectively. Each panel shows the signal timeseries (blue), the noisy signal timeseries (i.e. signal+noise) in gray, and the denoised signal in pink. The SNR (equation 11), CC, L^2 distance (equation 12), and $\Delta\mathrm{SNR}$ between the noisy and denoised signals are annotated.

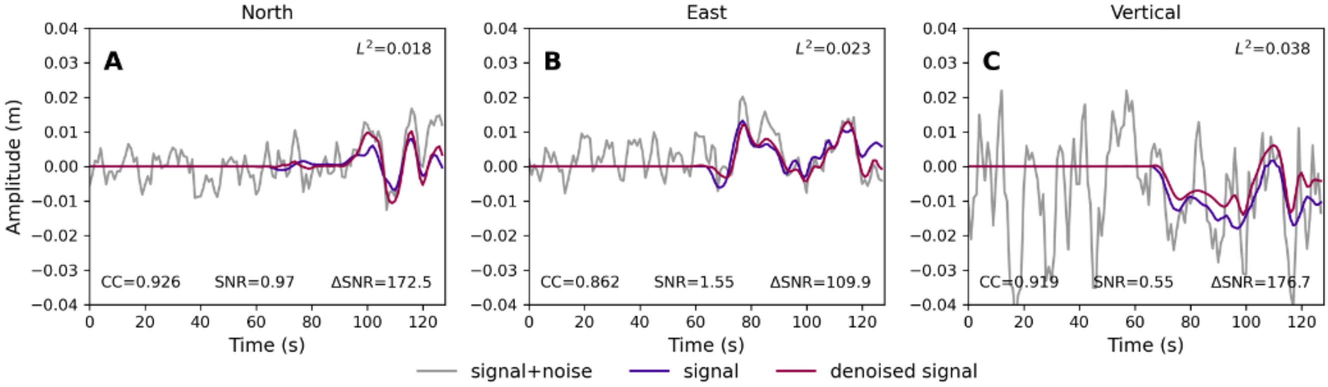

Figure 7 Example denoising using Model 2. Panels are the same as those in Fig. 6.

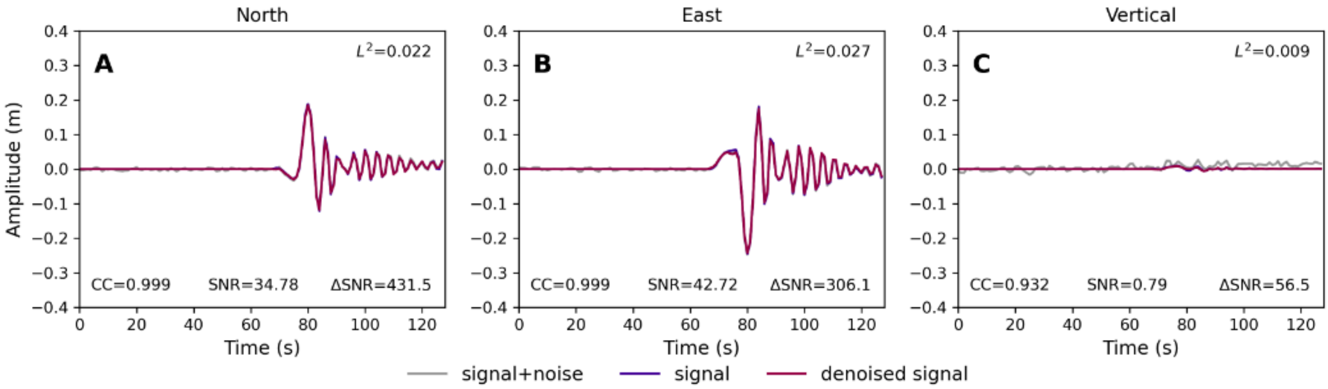

Figure 8 Higher amplitude example denoising using Model 2. Panels are the same as those in Fig. 6.

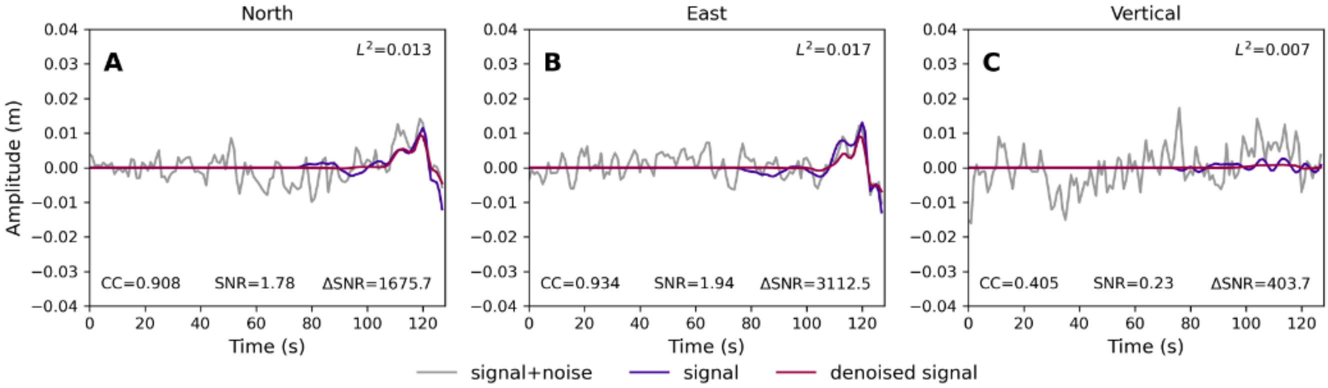

Figure 9 Example denoising using Model 3. Panels A-C correspond to the north, east, and vertical components respectively. Each panel shows the signal timeseries (blue), the noisy signal timeseries (i.e. signal+noise) in gray, and the denoised signal in pink. The SNR (equation 11), CC, L^2 distance (equation 12), and $\Delta\mathrm{SNR}$ between the noisy and denoised signals are annotated.

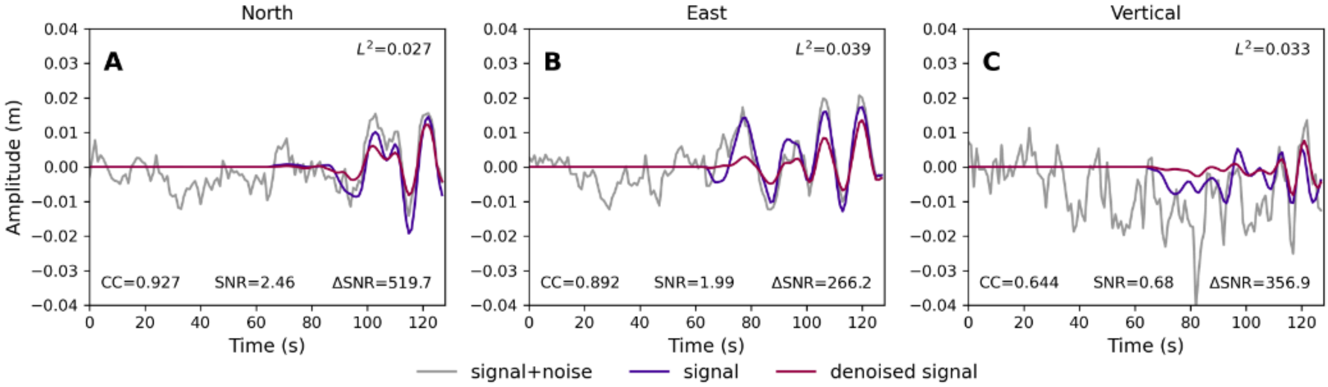

Figure 10 Example denoising using Model 3. Panels are the same as those in Fig. 9.

amplitude noise that is generally present in HR-GNSS data and the lower amplitude signals that result from the choice to simulate earthquakes on strike-slip fault systems.

3.3 Application to the 2019 Ridgecrest earthquakes

As a performance test of our trained models, we apply the denoising algorithms to the Ridgecrest earthquake sequence. While the aftershocks of these events included a number of intermediate magnitude earthquakes, determining how well the denoising models perform is hampered by lack of knowledge of what the true signal is. GNSS sites with nearby strong motion stations obviate this issue; the strong-motion data can be used to estimate true ground motions.

Denoising results at P570 and WOR are shown in Fig. 14. The model successfully identifies the arrival time (vertical black line) and suppresses noise prior to the first arrivals (SNRs are shown in bottom right of each panel). All models also successfully predict signal amplitudes, particularly on the vertical component. The SNR is significantly improved in the denoised versions of all components. The models generally do well

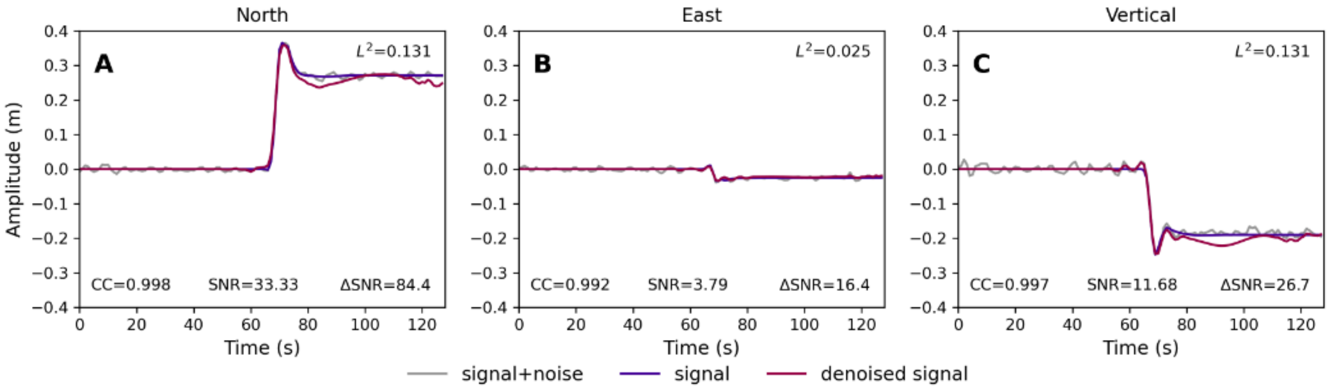

Figure 11 Example denoising using Model 3. Panels are the same as those in Fig. 9.

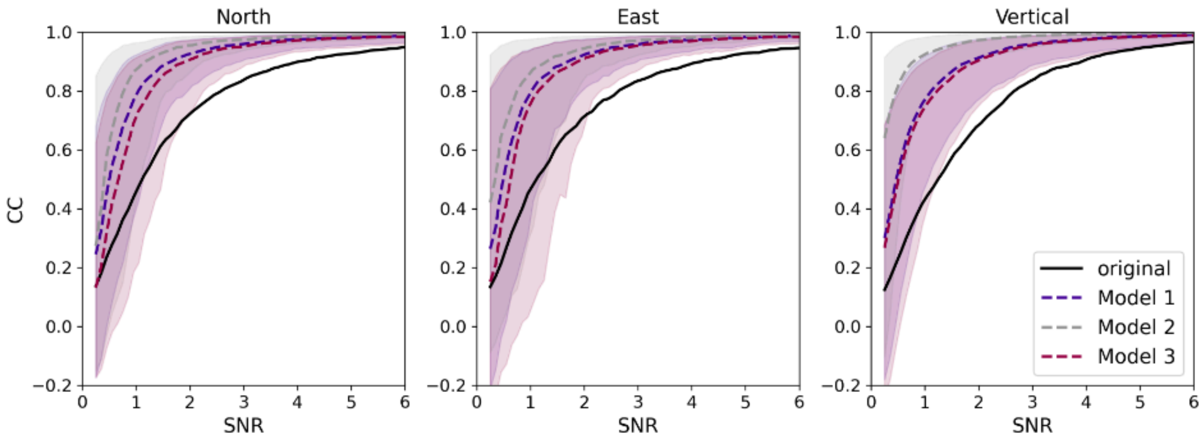

Figure 12 CC as a function of SNR. The median value of the CC between the true signal and noisy signal is shown in black. Median CC values between the true signal and signals denoised by Models 1, 2, and 3 are indicated by the blue, grey, and pink lines respectively. The shaded areas represent the region between the 10th and 90th percentiles that contain $80\,\%$ of the testing data and have the same color coding as the median CC.

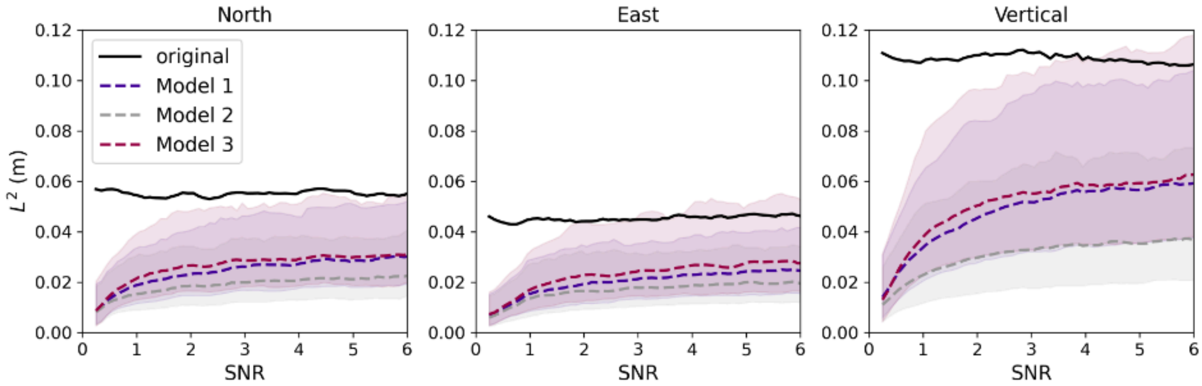

Figure 13 L^2 distances as a function of SNR. The median value of the L^2 distance between the true signal and noisy signal is shown in black. Median L^2 values between the true signal and signals denoised by Models 1, 2, and 3 are indicated by the blue, grey, and pink lines respectively. The blue, grey, and pink shaded areas represent the region between the 10th and 9-th percentiles that contain $80\,\%$ of the testing data for Models 1, 2, and 3 respectively.

early on ($\sim$35–60 s) however the coda is not well predicted i.e., the noisy GNSS better predicts these later arrivals ($\sim$60–80 s). Performance is similar for the ISA and ISLK station pair shown in Fig. 15. The models suppress noise prior to the first arrivals, predict phase and amplitude well early in the wave train, and then trend toward zero amplitude prediction for the later coda arrivals. Results for the M6.4 are shown in Figs S11 and S12. We will discuss this more in the following section.

Finally, as denoising may be applied to HR-GNSS in real time, we explored the false positive rate of Models 1, 2, and 3 on additional 10 000 noise waveforms. The results of this exercise is shown in Fig. S13. The false positive rate at a decision threshold of 1 cm PGD

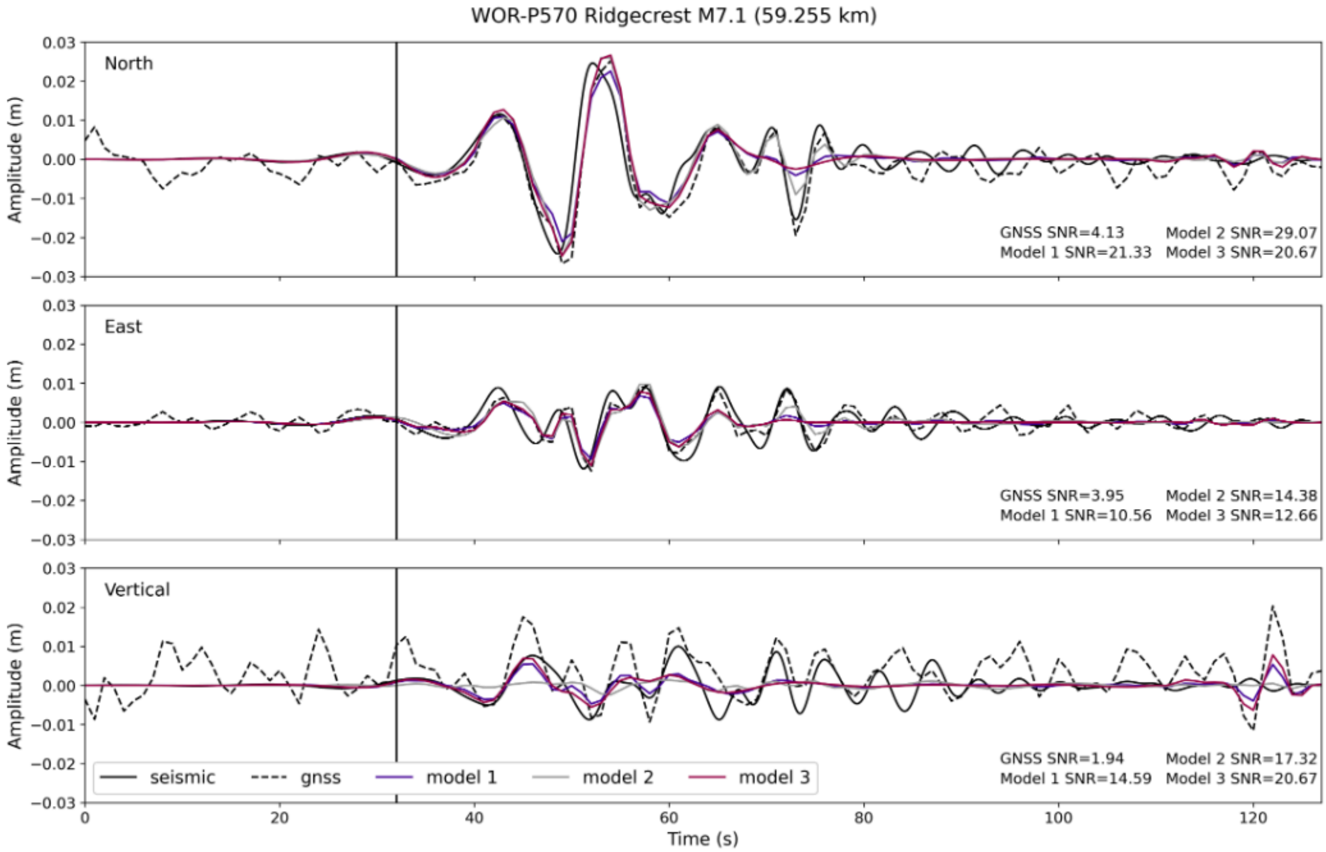

Figure 14 Comparison of recordings of the M7.1 Ridgecrest earthquake at strong motion station WOR and GNSS station P570. GNSS and integrated strong motion data are shown as dashed and solid black lines respectively. GNSS records denoised by Models 1, 2, and 3 are shown in blue, grey, and pink respectively. Vertical black line shows the theoretical P-wave arrival time. SNRs of the original HR-GNSS data, Models 1, 2, and 3 are shown in the bottom right.

is $2.05\,\%$, $2.51\,\%$, and $0.16\,\%$ for Models 1, 2, and 3 respectively. The false positive rate at a decision threshold of $2\,\mathrm{cm}$ PGD is $0.4\,\%$, $0.53\,\%$, and $0.06\,\%$ for Models 1, 2, and 3 respectively. These values are far smaller than the PGD of the noise waveforms themselves. Additionally, they could be further reduced by considering waveform character at multiple stations as all earthquake early warning algorithms do.

4 Discussion

As is evident in Figs 12 and 13, denoising HR-GNSS data offers significant improvements over utilizing noisy records. On all components, the median performance of Model 1, Model 2, and Model 3 exceeds that of the noisy records for all SNRs for both the CC and distance based performance metrics. Generally the models are excellent at suppressing noise prior to the first arrivals making identification of earthquake onsets more apparent. P-waves are almost always well below the GNSS noise, and, as shown in Figs 14 and 15, the P-wave arrival is clear in the denoised GNSS waveforms. Additionally, models can recover signals with amplitudes between $5\,\mathrm{mm}$ and $1\,\mathrm{cm}$, i.e., with $\mathrm{SNR} < 1$. Because of our choice to model strike-slip motion on vertical faults, the horizontal components are typically larger than the vertical. Additionally the vertical component generally has higher noise levels. Despite this, the models often predict well the timing and amplitude on the vertical component by utilizing information on the horizontal components to inform its character. The greatest improvements in CC are between SNRs of slightly below 1 to 3. At high SNR, applying denoising does not result in significant improvements as expected. At low SNR, amplitudes on all three components diminish, the model generally predicts zero amplitude signal on all components.

Models 2 and 3 were motivated by the amplitude distortion inherent in Model 1 which was originally developed for seismic data (Zhu et al., 2019). After assessing the performance of all models we find that Model 2, simple direct amplitude prediction, generally performs better than Models 1 and 3. It does not suffer from the amplitude distortion and is better at predicting both ringing and static offsets than the other models. By both performance metrics utilized here, it performs better at all SNRs than Model 1 and Model 3.

The main shortcoming of the models are the mispredictions of static offsets apparent in Figs 3L, 6C, 11A, and 11C. These mispredictions in some cases result from the noise character; the model predicts the average noise level as the static offset so noise with complex character in the presence of a static offset can result in a nonstationary prediction (e.g., Fig. 6C). In other cases these mispredictions appear to result from the frequency decomposition as is apparent from the oscilla-

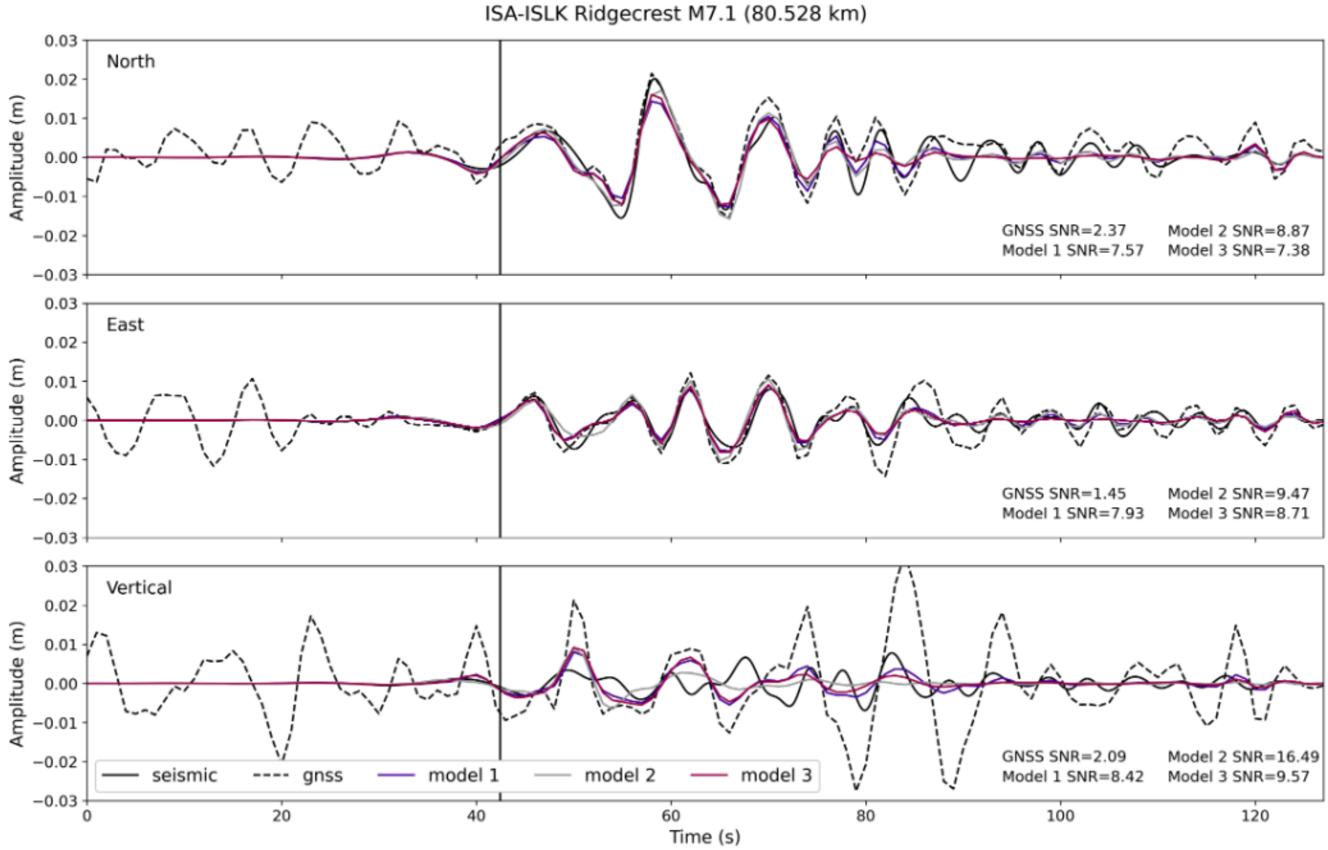

Figure 15 Comparison of recordings of the M7.1 Ridgecrest earthquake at strong motion station ISA and GNSS station ISLK. GNSS and integrated strong motion data are shown as dashed and solid black lines respectively. GNSS records denoised by Models 1, 2, and 3 are shown in blue, grey, and pink respectively. Vertical black line shows the theoretical P-wave arrival time. SNRs of the original HR-GNSS data, Models 1, 2, and 3 are shown in the bottom right.

tions in Fig. 11A and C that are unrelated to noise character and the overshoot in Fig. 11A that is reminiscent of Gibbs phenomenon. In the application of the models to the Ridgecrest data, the main shortcoming of the model was the misprediction of the lower amplitude, later arriving coda. We note that the amplitudes of these arrivals were $< 1\,\mathrm{cm}$ measured on the horizontal components. This may simply be too low SNR for the model to detect these arrivals. Incorporating more long duration low amplitude data may improve denoising in this scenario.

The single station algorithms we have developed here show promise for denoising HR-GNSS data. However, we made a number of decisions in the process of developing the models that might not make them more generally applicable. First, we utilized noise data from NOTA sites closest to the Ridgecrest earthquakes processed by UNAVCO. GNSS noise is network and processing algorithm dependent. Application to a different location or to a different processing algorithm will require re-training. Second, we simulated earthquakes on predominantly vertically dipping strike-slip faults. The resulting crustal deformation patterns are clearly not representative of earthquakes in other tectonic environments and simulation of such events would be needed for an all-purpose denoiser. Third, by utilizing "Fakequakes" to generate our synthetic earthquake database we make the assumption that the parameterizations and design choices therein are representative of real earthquakes. Fourth, by utilizing a 1D velocity structure we neglect 3D wave propagation effects. We have attempted to obviate this by including some simulations with soft layers that mimic basin resonance apparent in the two Ridgecrest examples we showed but this is a shortcut that could be improved upon with fully 3D GFs. Finally, we made a number of design choices in the deep learning implementation of denoising that could be improved upon. For example, the frequency decomposition technique may not be the best approach, other architectures may work better than the traditional CNN we employed here, etc.

While we have demonstrated that the models we developed are capable of denoising HR-GNSS data, their main limitation in our opinion, is that they are single station algorithms that do not utilize information from multiple stations. Certain types of GNSS noise, such as common mode errors, are highly correlated on nearby stations. For example, Fig. 16 shows normalized cross-correlations of noise windows prior to the M6.4 Ridgecrest earthquake. The $120\,\mathrm{s}$ of noise recorded on station P811 prior to the start of the event is cross-correlated with the same time window recorded on nearby stations. CC values between P811 and CCCC ($55.5\,\mathrm{km}$) are 0.81. CC values decrease to smaller values of 0.52–0.53 on stations P091 ($168\,\mathrm{km}$), ISLK ($70.2\,\mathrm{km}$), P595 ($82.3\,\mathrm{km}$), and P570 ($73.9\,\mathrm{km}$). Developing other

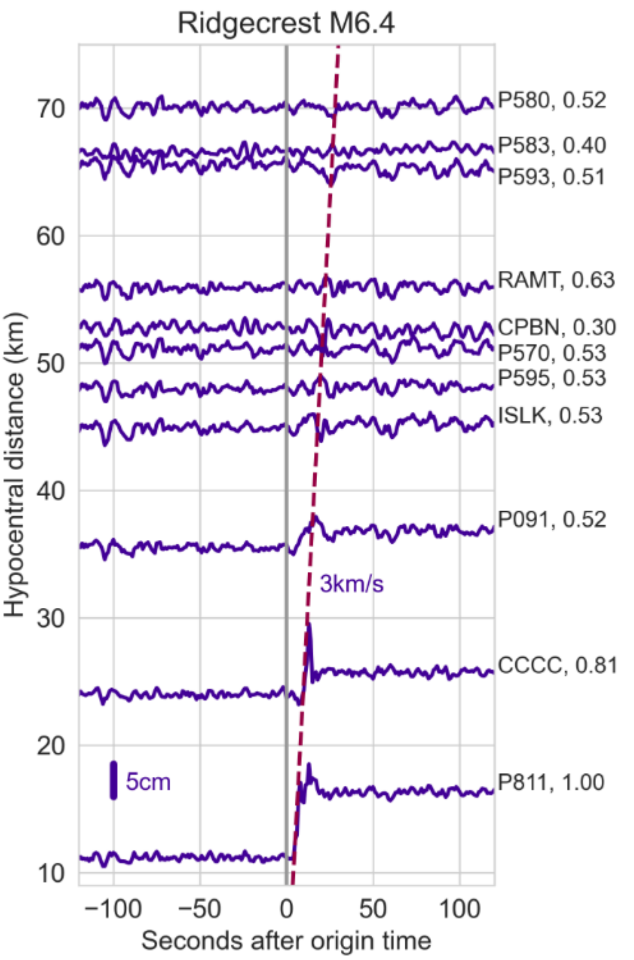

Figure 16 Comparison of recordings of the M6.4 Ridgecrest earthquake (blue timeseries) on stations with hypocentral distances up to $70\,\mathrm{km}$. Noise windows in the $120\,\mathrm{s}$ prior to the start time of the M6.4 Ridgecrest earthquake (start time is vertical grey line, moveout is dashed pink line) are cross-correlated with noise recorded on P811. Station names and CC values are annotated on the right.

GNSS denoising algorithms that are network based, include information from many stations, and can leverage the similarity of noise character on nearby stations will undoubtedly be more successful than the single station approach developed here. Such an algorithm would have a number of applications. For example, the identification of small magnitude slip transients that are typically challenging to detect in GNSS data but have limited to no seismic expression (e.g., Rousset et al., 2019), first arrivals of large magnitude earthquakes that, if rapidly identified, decrease the latency of early warning systems (e.g., Lin et al., 2021), and the incorporation of far-field, lower SNR GNSS data to constrain earthquake slip inversions.

5 Conclusions

The goal of this work was to explore the performance of single station, three component denoising algorithms on high-rate GNSS data. Single station denoising algorithms have been developed for seismic records, but their application to GNSS data has not been explored. To this end, we adapted frameworks proposed for denoising seismic data to perform on GNSS data by modifying such algorithms to work on three component data and proposing two additional slightly modified approaches that attempt to overcome amplitude distortion inherent in seismic algorithms. Earthquake metrics derived from HR-GNSS such as PGD rely heavily on signal amplitudes so it's important for any denoising algorithm to preserve amplitude information. After training three machine learning models on synthetic data designed to simulate the 2019 Ridgecrest earthquake sequence, we find that denoising can significantly improve the SNR of HR-GNSS waveforms. The denoised waveforms effectively suppress noise prior to the first arrivals making them more apparent. Additionally, small amplitude signals with SNR at or lower than noise levels can be identified and characterized. Also, because the three models utilize all three components, we are often able to recover much of the signal character on components with low SNR (such as the vertical) by utilizing signal character on the horizontal components. Overall this approach to denoising is promising and if employed in real-time may reduce the latency and improve the accuracy of early warning algorithms.

Acknowledgements

This work used the Python programming language (Van Rossum and Drake Jr, 1995), the Obspy package (Beyreuther et al., 2010; Krischer et al., 2015), Tensorflow (Abadi et al., 2016), and Keras (Chollet, 2015). Figures were made with the Generic Mapping Tools software (Wessel et al., 2019), the matplotlib software package (Hunter, 2007), and Netron for model visualization (https://github.com/lutzroeder/netron). This work is supported by NASA Grants 80NSSC19K0360, 80NSSC19K0741, and 80NSSC22K0458.

Data and code availability

GNSS noise data are from NOTA stations, funded by the National Science Foundation and operated by UNAVCO, Inc. Data based on services provided by the GAGE Facility, operated by UNAVCO, Inc. with support from the National Science Foundation and the National Aeronautics and Space Administration under NSF Cooperative Agreement EAR-1724794.

The synthetic records calculated from fakequakes and noise data are available on Zenodo at https://doi.org/10.5281/zenodo.7075919. The code used to develop and train the models, and the trained models are available at https://github.com/amtseismo/hrgnss_denoising.

Competing interests

The authors have no competing interests.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., and Zheng, X. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *Preprint*, 2016. doi: 10.48550/arXiv.1603.04467.

Agarap, A. Deep learning using rectified linear units (ReLU). *Preprint*, 2018. doi: 10.48550/arXiv.1803.08375.

Beyreuther, M., Barsch, R., Krischer, L., Megies, T., Behr, Y., and Wassermann, J. ObsPy: A Python toolbox for seismology. *Seismological Research Letters*, 81(3):530–533, 2010. doi: 10.1785/gssrl.81.3.530.

Bock, Y. and Melgar, D. Physical applications of GPS geodesy: A review. *Reports on Progress in Physics*, 79(10):10 1088 0034–4885 79 10 106801, 2016. doi: 10.1088/0034-4885/79/10/106801.

Choi, K., Bilich, A., Larson, K., and Axelrad, P. Modified sidereal filtering: Implications for high-rate GPS positioning. *Geophysical research letters*, 31(22), 2004. doi: 10.1029/2004GL021621.

Chollet, F. Keras. https://github.com/keras-team/keras. Retrieved from GitHub.

Crowell, B., Melgar, D., Bock, Y., Haase, J., and Geng, J. Earthquake magnitude scaling using seismogeodetic data. *Geophysical Research Letters*, 40(23):6089–6094, 2013. doi: 10.1002/2013GL058391.

Dong, D., Fang, P., Bock, Y., Webb, F., Prawirodirdjo, L., Kedar, S., and Jamason, P. Spatiotemporal filtering using principal component analysis and Karhunen-Loeve expansion approaches for regional GPS network analysis. *Journal of geophysical research: solid earth*, 111(B3), 2006. doi: 10.1029/2005JB003806.

Ende, M., Lior, I., Ampuero, J., Sladen, A., Ferrari, A., and Richard, C. A self-supervised deep learning approach for blind denoising and waveform coherence enhancement in distributed acoustic sensing data. *IEEE Transactions on Neural Networks and Learning Systems*, 2021. doi: 10.1109/TNNLS.2021.3132832.

Frank, W. Slow slip hidden in the noise: The intermittence of tectonic release. *Geophysical Research Letters*, 43(19):10–125, 2016. doi: 10.1002/2016GL069537.

Geng, J., Jiang, P., and Liu, J. Integrating GPS with GLONASS for high-rate seismogeodesy. *Geophysical research letters*, 44(7): 3139–3146, 2017. doi: 10.1002/2017GL072808.

Geng, J., Pan, Y., Li, X., Guo, J., Liu, J., Chen, X., and Zhang, Y. Noise characteristics of high-rate multi-GNSS for subdaily crustal deformation monitoring. *Journal of Geophysical Research: Solid Earth*, 123(2):1987–2002, 2018. doi: 10.1002/2018JB015527.

Goldberg, D., Melgar, D., Sahakian, V., Thomas, A., Xu, X., Crowell, B., and Geng, J. Complex rupture of an immature fault zone: A simultaneous kinematic model of the 2019 Ridgecrest, CA earthquakes. *Geophysical Research Letters*, 47(3):2019 086382, 2020. doi: 10.1029/2019GL086382.

Goulet, C., Abrahamson, N., Somerville, P., and Wooddell, K. The SCEC broadband platform validation exercise: Methodology for code validation in the context of seismic-hazard analyses. *Seismological Research Letters*, 86(1):17–26, 2015. doi: 10.1785/0220140104.

Graves, R. and Pitarka, A. Broadband ground-motion simulation using a hybrid approach. *Bulletin of the Seismological Society of America*, 100(5A):2095–2123, 2010. doi: 10.1785/0120100057.

Graves, R. and Pitarka, A. Refinements to the Graves and Pitarka (2010) broadband ground-motion simulation method. *Seismological Research Letters*, 86(1):75–80, 2015. doi: 10.1785/0220140101.

He, X., Hua, X., Yu, K., Xuan, W., Lu, T., Zhang, W., and Chen, X. Accuracy enhancement of GPS time series using principal component analysis and block spatial filtering. *Advances in Space Research*, 55(5):1316–1327, 2015. doi: 10.1016/j.asr.2014.12.016.

Hodgkinson, K., Mencin, D., Feaux, K., Sievers, C., and Mattioli, G. Evaluation of earthquake magnitude estimation and event detection thresholds for real-time GNSS networks: Examples from recent events captured by the network of the Americas. *Seismological Research Letters*, 91(3):1628–1645, 2020. doi: 10.1785/0220190269.

Hunter, J. Matplotlib: A 2D Graphics Environment. *Computing in Science and Engineering*, 9(3):90–95,, 2007. doi: 10.1109/MCSE.2007.55.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. *Preprint*, 2014. doi: 10.48550/arXiv.1412.6980.

Krischer, L., Megies, T., Barsch, R., Beyreuther, M., Lecocq, T., Caudron, C., and Wassermann, J. ObsPy: A bridge for seismology into the scientific Python ecosystem. *Computational Science and Discovery*, 8(1):014003, 2015. doi: 10.1088/1749-4699/8/1/014003.

Larson, K. Unanticipated uses of the global positioning system. *Annual Review of Earth and Planetary Sciences*, 47:19–40, 2019. doi: 10.1146/annurev-earth-053018-060203.

Lay, T. A review of the rupture characteristics of the 2011 Tohoku-oki Mw 9.1 earthquake. *Tectonophysics*, 733:4–36, 2018. doi: 10.1016/j.tecto.2017.09.022.

Li, Y., Xu, C., Yi, L., and Fang, R. A data-driven approach for denoising GNSS position time series. *Journal of Geodesy*, 92:905–922, 2018. doi: 10.1007/s00190-017-1102-2.

Lin, J., Melgar, D., Thomas, A., and Searcy, J. Early warning for great earthquakes from characterization of crustal deformation patterns with deep learning. *Journal of Geophysical Research: Solid Earth*, 126(10), 2021. doi: 10.1029/2021JB022703.

Mai, P. and Beroza, G. A spatial random field model to characterize complexity in earthquake slip. *Journal of Geophysical Research: Solid Earth*, 107(B11):–10, 2002. doi: 10.1029/2001JB000588.

Melbourne, T., Szeliga, W., Marcelo Santillan, V., and Scrivner, C. Global Navigational Satellite System Seismic Monitoring. *Bulletin of the Seismological Society of America*, 111(3):1248–1262, 2021. doi: 10.1785/0120200356.

Melgar, D. and Hayes, G. The Correlation Lengths and Hypocentral Positions of Great EarthquakesThe Correlation Lengths and Hypocentral Positions of Great Earthquakes. *Bulletin of the Seismological Society of America*, 109(6):2582–2593, 2019. doi: 10.1002/2016JB013314.

Melgar, D., Bock, Y., Sanchez, D., and Crowell, B. On robust and reliable automated baseline corrections for strong motion seismology. *Journal of Geophysical Research: Solid Earth*, 118(3): 1177–1187, 2013. doi: 10.1002/jgrb.50135.

Melgar, D., Crowell, B., Geng, J., Allen, R., Bock, Y., Riquelme, S., and Ganas, A. Earthquake magnitude calculation without saturation from the scaling of peak ground displacement. *Geophysical Research Letters*, 42(13):5197–5205, 2015. doi: 10.1002/2015GL064278.

Melgar, D., LeVeque, R., Dreger, D., and Allen, R. Kinematic rupture scenarios and synthetic displacement data: An example application to the Cascadia subduction zone. *Journal of Geophysical Research: Solid Earth*, 121(9):6658–6674, 2016. doi: 10.1002/2016JB013314.

Melgar, D., Crowell, B., Melbourne, T., Szeliga, W., Santillan, M., and Scrivner, C. Noise characteristics of operational real-time high-rate GNSS positions in a large aperture network. *Journal of Geophysical Research: Solid Earth*, 125(7):2019 019197, 2020. doi: 10.1029/2019JB019197.

Murray, J., Crowell, B., Grapenthin, R., Hodgkinson, K., Langbein, J., Melbourne, T., and Schmidt, D. Development of a geodetic component for the US West Coast earthquake early warning system. *Seismological Research Letters*, 89(6):2322–2336, 2018. doi: 10.1785/0220180162.

Ronneberger, O., Fischer, P., and Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, page 234–241, Cham, 2015. Springer. doi: 10.1007/978-3-319-24574-4_28.

Ross, Z., Idini, B., Jia, Z., Stephenson, O., Zhong, M., Wang, X., and Jung, J. Hierarchical interlocked orthogonal faulting in the 2019 Ridgecrest earthquake sequence. *Science*, 366(6463):346–351, 2019. doi: 10.1126/science.aaz0109.

Rousset, B., Bürgmann, R., and Campillo, M. Slow slip events in the roots of the San Andreas fault. *Science advances*, 5(2), 2019. doi: 10.1126/sciadv.aav3274.

Satake, K. and Heidarzadeh, M. A review of source models of the 2015 Illapel, Chile earthquake and insights from tsunami data. *Pure and Applied Geophysics*, 174(1):1–9, 2017. doi: 10.1007/978-3-319-57822-4_1.

Shelly, D. A high-resolution seismic catalog for the initial 2019 Ridgecrest earthquake sequence: Foreshocks, aftershocks, and faulting complexity. *Seismological Research Letters*, 91(4): 1971–1978, 2020. doi: 10.1785/0220190309.

Thomas, A., Beroza, G., and Shelly, D. Constraints on the source parameters of low-frequency earthquakes on the San Andreas Fault. *Geophysical Research Letters*, 43(4):1464–1471, 2016. doi: 10.1002/2015GL067173.

Thomas, A., Inbal, A., Searcy, J., Shelly, D., and Bürgmann, R. Identification of Low-Frequency Earthquakes on the San Andreas Fault With Deep Learning. *Geophysical Research Letters*, 48(13): 2021 093157, 2021. doi: 10.1029/2021GL093157.

Tibi, R., Hammond, P., Brogan, R., Young, C., and Koper, K. Deep learning denoising applied to regional distance seismic data in Utah. *Bulletin of the Seismological Society of America*, 111(2): 775–790, 2021. doi: 10.1785/0120200292.

Trugman, D., Page, M., Minson, S., and Cochran, E. Peak ground displacement saturates exactly when expected: Implications for earthquake early warning. *Journal of Geophysical Research: Solid Earth*, 124(5):4642–4653, 2019. doi: 10.1029/2018JB017093.

Van Rossum, G. and Drake Jr, F. L. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995. https://www.narcis.nl/publication/RecordID/oai%3Acwi.nl%3A5008.

Wdowinski, S., Bock, Y., Zhang, J., Fang, P., and Genrich, J. Southern California permanent GPS geodetic array: Spatial filtering of daily positions for estimating coseismic and postseismic displacements induced by the 1992 Landers earthquake. *Journal of Geophysical Research: Solid Earth*, 102(B8):18057–18070, 1997. doi: 10.1029/97JB01378.

Wessel, P., Luis, J., Uieda, L., Scharroo, R., Wobbe, F., Smith, W., and Tian, D. The Generic Mapping Tools version 6. *Geochemistry, Geophysics, Geosystems*, 20:5556–5564, 2019. doi: 10.1029/2019GC008515.

Williamson, A., Melgar, D., Crowell, B., Arcas, D., Melbourne, T., Wei, Y., and Kwong, K. Toward near-field tsunami forecasting along the Cascadia subduction zone using rapid GNSS source models. *Journal of Geophysical Research: Solid Earth*, 125(8):2020 019636, 2020. doi: 10.1029/2020JB019636.

Zhu, L. and Rivera, L. A note on the dynamic and static displacements from a point source in multilayered media. *Geophysical Journal International*, 148(3):619–627, 2002. doi: 10.1046/j.1365-246X.2002.01610.x.

Zhu, W. and Beroza, G. PhaseNet: a deep-neural-network-based seismic arrival-time picking method. *Geophysical Journal International*, 216(1):261–273, 2019. doi: 10.1093/gji/ggy423.

Zhu, W., Mousavi, S., and Beroza, G. Seismic signal denoising and decomposition using deep neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 57(11):9476–9488, 2019. doi: 10.1109/TGRS.2019.2926772.