applied sciences

MDPI

Article

Time–Frequency Domain Seismic Signal Denoising Based on Generative Adversarial Networks

Ming Wei [1], Xinlei Sun [2,*] and Jianye Zong [2,3]

[1] College of Computer Science and Cyber Security, Chengdu University of Technology, Chengdu 610059, China; zpv2jdfc@gmail.com
[2] International Research Center for Planetary Science, College of Earth and Planetary Sciences, Chengdu University of Technology, Chengdu 610059, China; zongjianye@stu.cdut.edu.cn
[3] College of Geophysics, Chengdu University of Technology, Chengdu 610059, China
* Correspondence: xsun@cdut.edu.cn

Abstract: Existing deep learning-based seismic signal denoising methods primarily operate in the time domain. Those methods are ineffective when noise overlaps with the seismic signal in the time domain. Time–frequency domain-based deep learning methods are relatively rare and usually employ single loss function, resulting in suboptimal performance on low SNR signals and potential damage to P wave. This paper proposes a method based on generative adversarial networks (GANs). Compared to convolutional neural networks, the discriminator in GANs helps retain more true signal details by judging denoising performance. Additionally, an attention mechanism is introduced to fully extract signal features, and a perceptual loss is employed to evaluate the difference between the denoised result and the target's high-level features. Experimental results show that this method can effectively improve SNR and ensure that the denoised result is close to the true signal. Furthermore, by comparing DeepDenoiser and ARDU, it is proven that the proposed method achieves better denoising performance, especially for low SNR signals, while causing less damage to the seismic signals.

Keywords: seismic signal; noise reduction; generative adversarial network; perceptual loss

Citation: Wei, M.; Sun, X.; Zong, J. Time–Frequency Domain Seismic Signal Denoising Based on Generative Adversarial Networks. *Appl. Sci.* **2024**, *14*, 4496. https://doi.org/10.3390/app14114496

Academic Editor: Roberto Scarpa

Received: 24 April 2024
Revised: 16 May 2024
Accepted: 23 May 2024
Published: 24 May 2024

1. Introduction

Seismic signals are important for studying the Earth's internal structure, geological exploration, and resource development. By analyzing the propagation paths, velocities, and waveforms of seismic signals, researchers can infer the characteristics of the Earth's interior, including the properties and boundaries of the mantle, crust, and core [1,2]. In the field of geological exploration, seismic signals provide a reliable means of geophysical detection for the exploration and development of resources such as oil and natural gas [3,4]. By analyzing the propagation velocities, reflections, refractions, and other characteristics of seismic signals, explorers can deduce the types, thicknesses, and structures of subsurface rock layers, thereby determining the potential locations and scales of oil and gas reservoirs [5,6]. However, seismic data are often contaminated by various types of noise, such as surface noise (e.g., wind and traffic), instrument noise (e.g., noise from the sensors themselves), and other man-made interference (e.g., electromagnetic interference and human activities). A significant amount of signal data is rendered unusable for further study due to noise interference, and signals containing noise can also introduce biases into research results [7]. For a long time, there has been a persistent need to extract true seismic signals from noisy data.

Traditional seismic signal denoising methods, such as filtering, wavelet decomposition [8], S-transform [9], and empirical mode decomposition [10], aim to extract useful information and achieve denoising by transforming or decomposing the signal. These methods have the advantages of being fast and simple, but they also have many shortcomings. For example, Fourier-based filtering cannot remove noise with the same frequency as

the signal; wavelet transforms and S-transforms require manually selecting appropriate parameters, introducing subjectivity and uncertainty; empirical mode decomposition may suffer from mode mixing, boundary effects, and difficulty in determining the number of signal modes to decompose. Therefore, in the field of seismic signal denoising, it is crucial to seek a denoising method that does not rely on the experience of technicians, achieves superior denoising performance, and has a relatively fast processing speed.

In recent years, with the continuous development of artificial intelligence technology, deep learning has become a research hotspot due to its unique advantages. Time domain deep learning methods for seismic signal denoising have experienced rapid development. Li et al. [11] improved upon DnCNN by adding downsampling and upscaling operations, reducing the training time and memory requirements. Zhong et al. [12] proposed a multi-scale feature extraction DnCNN model with a hierarchical structure capable of extracting features at different scales and utilizing the fused features to more effectively extract seismic data information. Zhao et al. [13] proposed a multiscale CNN based on U-Net. They adjusted the shape of the convolution kernel to fit the shape of the seismic data and used dilated convolutions to extract multiscale features. Lan et al. [14] added an attention module after the convolutional layers in the residual blocks and added channel attention to the skip connections to remove interference.

Time–frequency domain-based seismic signal denoising methods are scarce, and there remains room for improvement. DeepDenoiser [15] obtains the time–frequency spectra of seismic signals via short-time Fourier transform and employs a U-Net-based model for denoising in the time–frequency domain. DnRDB [16] improves upon DeepDenoiser by introducing residual dense block (RDB), further enhancing the SNR. ARDU [17] combines dilated convolutions with RDB, and the SNR improved by 2.9 dB compared to DnRDB.

Addressing the issue of existing methods underperforming in denoising low SNR signals, this paper proposes an innovative seismic signal denoising method based on generative adversarial networks. This method effectively suppresses noise while minimizing damage to the effective signal. Specifically, a lightweight convolutional attention module is introduced in the generator to better extract signal features. When evaluating the denoising results, instead of directly comparing the denoised output with the target at the pixel level, our method incorporates adversarial loss, perceptual loss, and gradient loss from the discriminator to comprehensively assess the denoising results, making the denoised output more closely match the feature distribution of the real signal. Compared to other deep learning-based methods, DeepDenoiser and ARDU, the proposed method obtains better denoised results, especially for low SNR signals, and causes less damage to the true signal.

2. Method

We propose a denoising method for single-channel seismic signals based on generative adversarial networks (GANs). The method takes the short-time Fourier transform (STFT) time–frequency spectrum of a seismic signal as input and employs a generator network to denoise the signal, while a discriminator network is used to evaluate the denoising performance. The denoised spectrum $\hat{s}(t, f)$ can be obtained by

$$\hat{s}(t, f) = Y(t, f) \cdot M(t, f) \tag{1}$$

In Equation (1), $Y(t, f)$ represents the noisy spectrum, $M(t, f)$ is a function that maps $Y(t, f)$ to the desired noise-free spectrum $\hat{s}(t, f)$. Donoho and Johnstone [18] showed that this mapping can be carried out via a simple thresholding in a sparse representation where the thresholding value can be estimated from noise level assuming a Gaussian distribution. Here, we cast the problem as a supervised learning problem. As shown in Figure 1, the generator part of the proposed method will learn a sparse representation of the input data to produce a mask with the same size as the input spectrum, ranging from (0,1), and the clean signal can be obtained by multiplying the mask with the noisy signal.

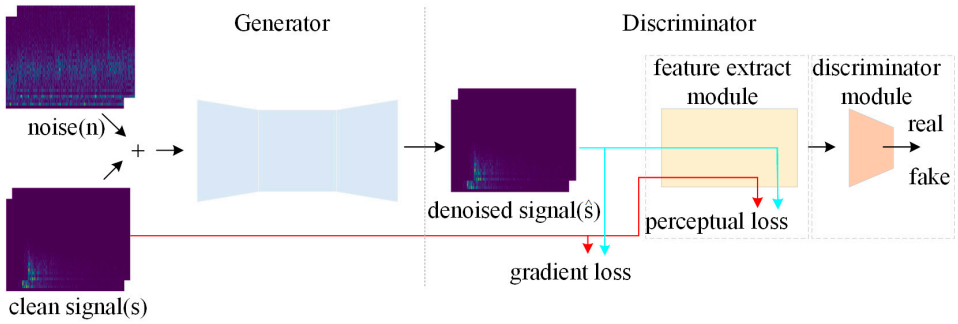

Figure 1. The overall structure of the proposed method.

The generator is built upon the U-Net architecture and incorporates a residual shrinkage structure with a soft thresholding mechanism, allowing the network to focus on signal-relevant features. The discriminator consists of a feature extraction module and a discrimination module. The feature extraction module extracts multi-scale features from the time–frequency spectrum and employs a lightweight convolutional attention module for feature fusion, generating a perceptual feature map. The discrimination module adopts a PatchGAN architecture, evaluating each local region of the perceptual feature map and ultimately outputting a score map. The overall network structure is as follows.

As shown in Figure 1, the proposed method can be divided into two parts: the generator and the discriminator. The generator adopts U-Net architecture, taking the real and imaginary parts of the STFT as input and outputs two masks ranging from 0 to 1. The denoised spectrum is obtained by multiplying the masks with the input. The discriminator consists of two modules: the feature extraction module and the discriminator module. The feature extraction module extracts high-level features from the signal spectrum and calculates the perceptual loss. The discriminator module differentiates the denoised result and the true noise-free signal.

2.1. Generator Network Structure

The generator is based on the U-Net architecture and can be seen as composed of three parts: an encoder, a decoder, and skip connections. As shown in Figure 2, the encoder takes the STFT time–frequency spectrum as input and produces downsampled feature maps. The decoder part upsamples and convolves the feature maps. Finally, via convolutions and outputs of two masked signals ranging from $(0, 1)$, these two masked signals are multiplied with the real and imaginary parts of the input time–frequency spectrum, respectively, yielding the time–frequency spectrum of the effective signal.

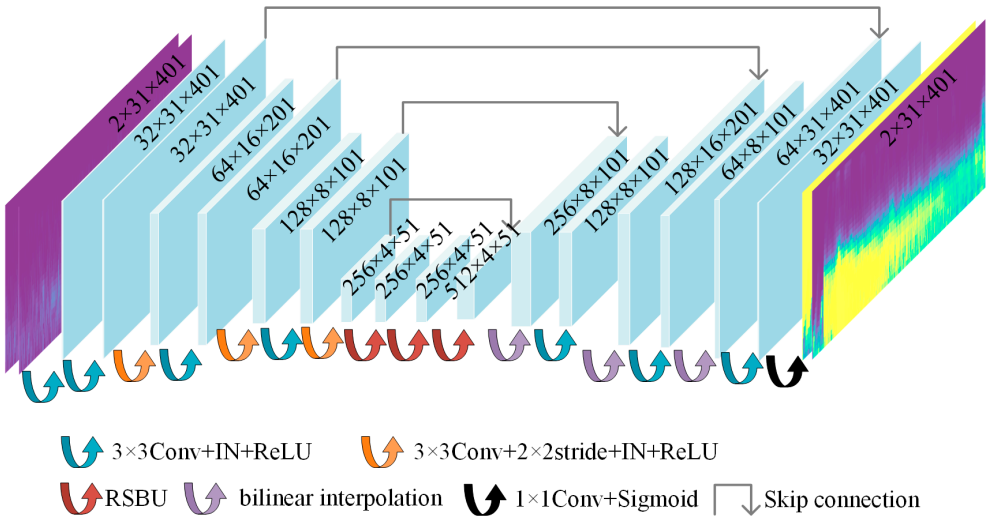

Figure 2. Structure of generator network.

As shown in Figure 2, the generator follows the U-Net architecture, taking the real and imaginary parts of the STFT spectrum as input. The input data are processed through 3×3 convolutional kernels with instance normalization (IN) and ReLU activation function. The encoder part eventually represents the input as a $256 \times 4 \times 51$ feature vector. Then, we use RSBU to further extract the feature (RSBU is the Residual Shrinkage Building Unit depicted in Figure 3). In the decoder part, bilinear interpolation and 3×3 convolutional kernels are used to generate the corresponding high-dimensional non-linear mapping. In the last layer, we use 1×1 convolution and Sigmoid to produce the mask ranging from 0 to 1.

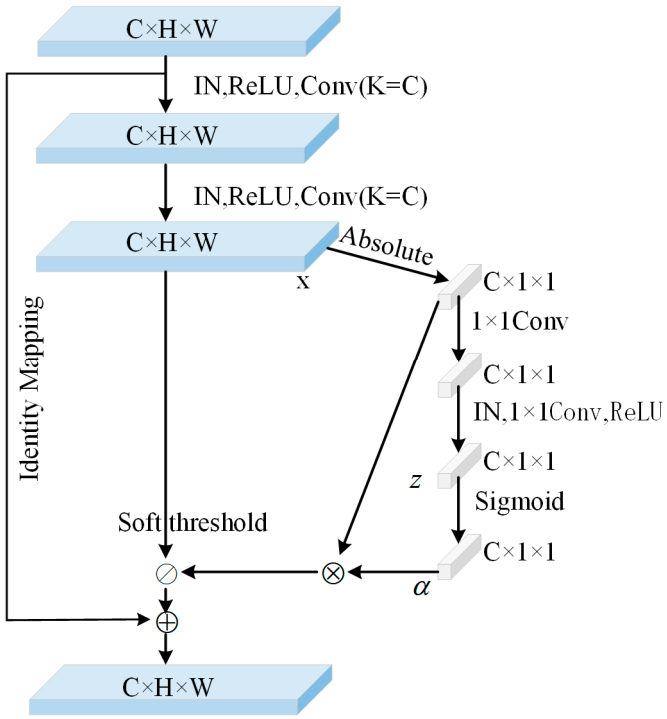

Figure 3. Structure of Residual Shrinkage Building Unit (RSBU).

Compared to the standard U-Net, the generator incorporates improvements in the normalization, downsampling, and upsampling processes, making it more suitable for the task of seismic signal denoising. U-Net employs batch normalization (BN) for normalization. BN introduces noise to individual samples by computing the mean and variance across multiple samples, thereby diminishing the independence between different data points for tasks that require high output detail [19,20]. In contrast, instance normalization (IN) removes the summation across the batch dimension, and the proposed method utilizes IN to independently compute the mean and variance for each channel. IN has been widely applied to tasks that require high output detail, such as generative adversarial networks (GANs) and image super-resolution [21–23], and it can better preserve the time–frequency spectral details of seismic signals [24]. U-Net performs downsampling through max-pooling layers, but Springenberg et al. [25] demonstrated through experiments that using convolutions with larger strides instead of pooling for dimensionality reduction can improve the accuracy of image recognition tasks. In this work, we employ convolutions with a stride of 2 for downsampling. U-Net achieves upsampling through transposed convolutions, which involve padding the input feature map elements and the surrounding areas and then convolving the padded feature map. However, due to the padding between feature map elements, the outputs of transposed convolutions exhibit uneven overlaps, leading to the checkerboard artifact that causes a grid-like pattern of alternating light and dark pixels in the final output image [26]. To avoid this issue, we employ bilinear interpola-

tion for upsampling, which does not produce checkerboard artifacts and is computationally simpler and more efficient.

In deep neural networks, gradients may gradually diminish (vanishing gradient) or amplify (exploding gradient) during the backpropagation process, resulting in ineffective optimization of the parameters in the shallow layers. Residual Networks (ResNets) [27] alleviate the vanishing and exploding gradient problems by introducing a skip connection between the input and output. This allows the loss to be backpropagated not only through the convolutional layers but also directly through the identity mapping function. To enhance the feature learning capability of the network on noisy data, we propose a residual shrinkage structure that incorporates the soft-thresholding function into the residual network. The soft-thresholding function is widely used in signal-denoising algorithms and shrinks the input data toward zero, using the following formula:

$$y = \begin{cases} x - \tau, x > \tau \\ 0 \quad\quad , -\tau \le x \le \tau \\ x + \tau, x < -\tau \end{cases} \tag{2}$$

According to Equation (2), the derivative of the soft-thresholding function is either zero or one, which is beneficial for preventing vanishing or exploding gradients. As illustrated in Figure 3, the Residual Shrinkage Building Unit (RSBU) [28] adaptively learns the coefficient α via a subnetwork and constrains the value of α within the range $(0, 1)$ using the σ activation function. The absolute value of the input feature map is then multiplied by α to obtain an adaptive threshold.

As shown in Figure 3, RSBU performs 1×1 convolution on the absolute value of the input feature x to extract effective features and output a vector α ranging from (0,1) through the Sigmoid function. Then, x is multiplied by α to obtain the threshold τ. x is compared with τ, and the output feature y is determined according to Equation (2). We adopt an identity mapping to directly add the output of the previous layer to y to obtain the final output feature, ensuring that the network can fully utilize the features of the previous layer.

2.2. Discriminator Network Structure

For generative adversarial networks (GANs), the quality of the generator's output relies heavily on the performance of the discriminator, making the optimization of the discriminator crucial. In this work, we build upon GANs and divide the discriminator into a feature extraction module and a discrimination module. Taking two-channel time–frequency spectra as input, the feature extraction module adaptively extracts time–frequency spectral features and computes the perceptual loss, which measures the high-level feature differences between the generator's output and the real signal. The discrimination module evaluates the denoising performance of the generator.

Perceptual loss has been applied in seismic image denoising tasks, where SeisGAN [29] and MSRD-GAN [30] both adopted a supervised approach, extracting perceptual features through a pre-trained VGG network [31] and combining pixel-level loss, perceptual loss, and adversarial loss to compute the generator loss according to Equation (3).

$$\ell_G = \ell_{MSE} + \alpha \ell_{VGG} + \beta \ell_{ADV} \tag{3}$$

Compared to using only pixel loss, the inclusion of perceptual loss results in denoised data with more detailed information. The perceptual loss compares the convolutional features of the generated image and the real image, enabling the capture of more image details. In tasks such as image super-resolution and style transfer, pre-trained VGG networks are commonly used to extract perceptual features. However, using the VGG network to extract perceptual features for seismic signals and their time–frequency representations has certain limitations:

(1) The VGG network was trained on the ILSVRC-2012 [32] dataset, which contains 1000 different categories of natural images, including animals, vehicles, natural landscapes,

and buildings. The features of these images differ significantly from those of seismic time–frequency spectra, limiting the effectiveness of using the VGG network to extract features from seismic time–frequency spectra.

(2) The VGG network takes RGB three-channel image data as input, while seismic signals represented in the time–frequency domain after a short-time Fourier transform (STFT) have two channels: real and imaginary parts. Therefore, before using the VGG network to extract features from seismic time–frequency spectra, the data need to be padded with zeros or other means to form a three-channel input. However, these padded channels do not contain any useful information, leading to a waste of computational resources. In this paper, the discriminator is divided into two components: a feature extraction module and a discrimination module. Taking two-channel time–frequency spectra as input, the feature extraction module adaptively extracts features from the time–frequency spectra, while the discrimination module evaluates the denoising performance of the generator

The feature extraction module employs a three-step strategy for feature extraction: First, it extracts shallow features through a densely connected structure; second, it applies pooling operations of different kernel sizes to the shallow features and then upsamples the resulting feature maps using the PixShuffle [33] layer, concatenating them to form a multi-scale feature representation; finally, it introduces a channel attention mechanism to assign different weights to different channels of the multi-scale feature maps, thereby enhancing the overall expressive capability of the feature extraction module.

In Figure 4, the feature extraction module extracts a feature map of size $32 \times 16 \times 200$ through dense connections, downsamples the feature map through max pooling, and upsamples it through the PixShuffle layer. After concatenating the obtained features, we extract high-level features through LighWeight Convolutional Block Attention Module (LCBAM). Similar to CBAM [34], LCBAM replaces the fully connected layer with 1×1 convolution in channel attention and replaces the 7×7 convolutional kernel with dilated convolution with a dilation rate of 2 in spatial attention.

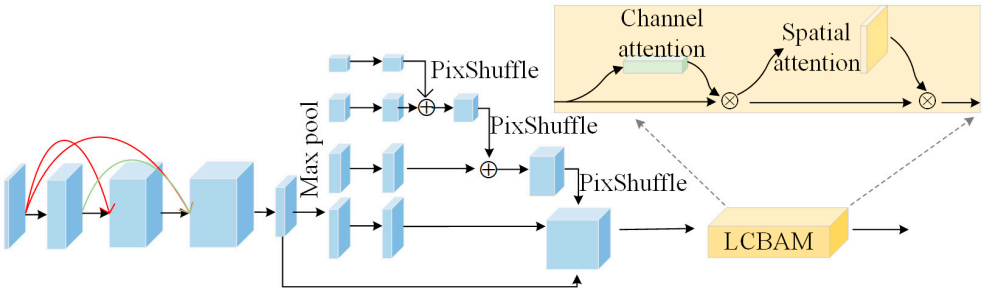

Figure 4. Structure of feature extraction module.

In a traditional generative adversarial network, the discriminator maps the input features to a single scalar representing the probability of being real or fake. This scalar is essentially the discriminator's normalized evaluation of the entire input data, which is an average assessment of all local features. Taking seismic signal denoising as an example, when the generator suppresses most of the noise but still leaves some residual noise in certain frequency bands, the overall quality of the denoised signal is good. Using a single scalar to evaluate the entire time–frequency spectrum will result in an assessment close to the real label, weakening the impact of the local residual noise. Therefore, evaluating the overall quality of the generated data with a single scalar may lead to an inaccurate assessment of some local features by the discriminator, affecting the generator's ability to remove weak noise. In this paper, the discrimination module is based on the PatchGAN structure, where the input data are divided into multiple patches, and each patch is independently evaluated. The evaluation result is an $8 \times 1 \times 13$ matrix, with each element in the matrix representing a local region of the input data. The evaluation results have a strong locality, enabling better capture of the data details and improving the denoising

performance. Additionally, PatchGAN does not require fully connected layers, reducing computational complexity. The structure of the discrimination module is shown in Figure 5.

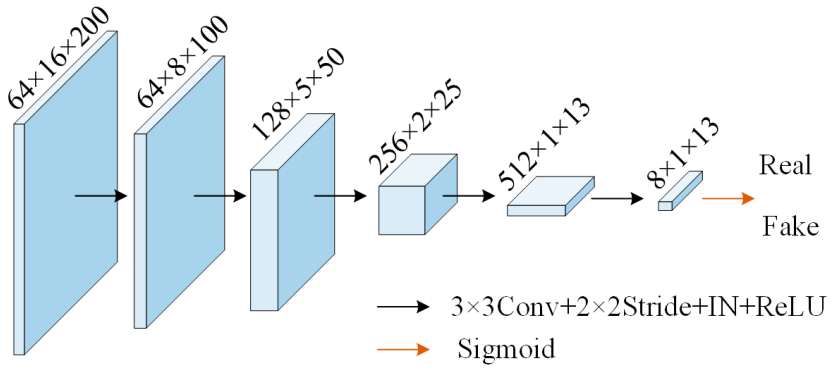

Figure 5. Structure of discrimination module.

2.3. Loss Function

The loss function proposed in this chapter consists of three components: adversarial loss, perceptual loss, and gradient loss [35]. Adversarial loss is the inherent loss function in generative adversarial networks. By incorporating the perceptual loss function, the detailed features of the denoised time–frequency spectrum can be enhanced. The gradient loss function serves to prevent the generator from distorting the edges of the time–frequency signal [36,37]. The overall loss function is formulated as follows:

$$\ell = \ell_{\text{adv}} + \alpha \cdot \ell_{per} + \beta \cdot \ell_{grad} \tag{4}$$

In Equation (4), α and β represents the coefficients of perceptual loss and gradient loss, respectively. In this paper, we simply set both of them to 1.

The adversarial loss enables the generator to produce more realistic samples, allowing the discriminator to more accurately distinguish between real and generated samples. For the discriminator, when the input is the time–frequency spectrum of a real signal, the discriminator output should be close to 1, whereas when the input is a time–frequency spectrum generated by the generator, the discriminator output should be close to 0. The standard GAN utilizes the negative log-likelihood loss as the adversarial loss, but this can lead to vanishing gradients. In this work, we follow the principle of the Least Squares GAN (LSGAN) and adopt the least squares loss function for the adversarial loss:

$$\ell_{adv} = \mathbb{E}_{s\sim P_{data}(s)}\left[(D(s)-1)^2\right] + \mathbb{E}_{y\sim P_{data}(y)}\left[(D(G(y)))^2\right] \tag{5}$$

For perceptual loss ℓ_{per}, the feature extraction module of the discriminator adaptively extracts perceptual features from the time–frequency spectrum:

$$\ell_{per} = \mathbb{E}_{s\sim P_{data}(s)}\left[\|\phi_D(G(s_i)) - \phi_D(y_i)\|_2^2\right] \tag{6}$$

In Equation (6), ϕ_D represents the feature extraction module.

While the perceptual loss can effectively enhance the output details of the generator, it can also lead to distortions and artifacts in the signal, causing deformations in the edge regions of the time–frequency spectrum. By computing gradients and comparing the rate of change of elements along the x and y axes, the outlines of signal edges can be reinforced, improving the stability of output details. In this paper, we employ the Sobel operator, using the convolution kernels G_x and G_y defined in Equation (7) to calculate the gradients along the x and y directions:

$$\begin{aligned} G_x &= \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \\ G_y &= \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \\ grad(s) &= |Conv(s, G_x)| + \left|Conv(s, G_y)\right| \end{aligned} \tag{7}$$

And the gradient loss ℓ_{grad} is

$$\ell_{grad} = \|grad(s) - grad(\hat{s})\|^2 \tag{8}$$

3. Results

We used the STanford EArthquake Dataset (STEAD) [38] as the dataset. STEAD is a high-quality, large-scale, and global dataset of local earthquake and non-earthquake signals recorded by seismic instruments. In this paper, 47,335 signals with an SNR greater than 50 dB were randomly selected from the STEAD as the clean signal set, and 100,000 noise signals were randomly selected as the noise set. These were then divided into training, validation, and test sets in a 3:1:1 ratio. During the training process, the signal was randomly selected from the training clean signal set and the training noise set, respectively, and added together to obtain a noisy signal. The validation process is similar to the training process. For the synthetic test set, we generate a set of random numbers, where each random number represents the index of the corresponding noise signal from the test noise set for each clean signal in the test clean signal set. During the testing process, these two signals are added together to obtain a noisy signal. For the field test set, 20,000 signals with SNR lower than 20 dB were randomly selected from the STEAD as the test set. The SNR formula used in this paper is different from the one used in STEAD, which is shown in Equation (9). We use the same formula as the one used in DeepDenoiser.

$$\text{SNR} = 10 \log_{10} \frac{\|S\|^2}{\|N\|^2} \tag{9}$$

We compare our proposed method with DeepDenoiser and ARDU on synthetic and field seismic signals. The performance is assessed using metrics of signal-to-noise ratio (SNR), correlation coefficient, and mean absolute error (MAE). The corresponding formulas defined are as follows:

$$\begin{aligned} \text{SNR} &= 10 \log_{10} \left(\frac{\sigma_s}{\sigma_n} \right) \\ r(X, \hat{X}) &= \frac{Cov(X, \hat{X})}{\sqrt{Var[X] \cdot Var[\hat{X}]}} \\ \text{MAE} &= \frac{1}{n} \sum_{i=1}^{n} \left| \hat{X}(i) - X(i) \right| \end{aligned} \tag{10}$$

In Equation (10), σ_s and σ_n represent the standard deviations of the signal before and after the arrival of the P-wave. *Cov* represents covariance, and *Var* represents variance.

3.1. Synthetic Data

The synthetic signal test set is derived from the STEAD dataset, including 9476 noise-free signals and 20,000 pure noise signals. To evaluate the denoising performance, we add noise signals to the noise-free signals to synthesize noisy signals as the input to the model. We then perform quantitative and qualitative analyses by comparing the noise-free signals with the denoised outputs from the model. To obtain low-SNR signals, we amplify the pure noise signals by a factor of 5 and add them to the noise-free signals, thereby synthesizing noisy signals with low SNR.

Figure 6 illustrates a denoising example, where, due to the strong noise signal, DeepDenoiser misinterpreted the entire signal as pure noise, resulting in a zero-output signal. During the denoising process, ARDU caused significant damage to the signal. In contrast, the denoised signal obtained through the proposed method exhibits the closest resemblance to the original noise-free signal.

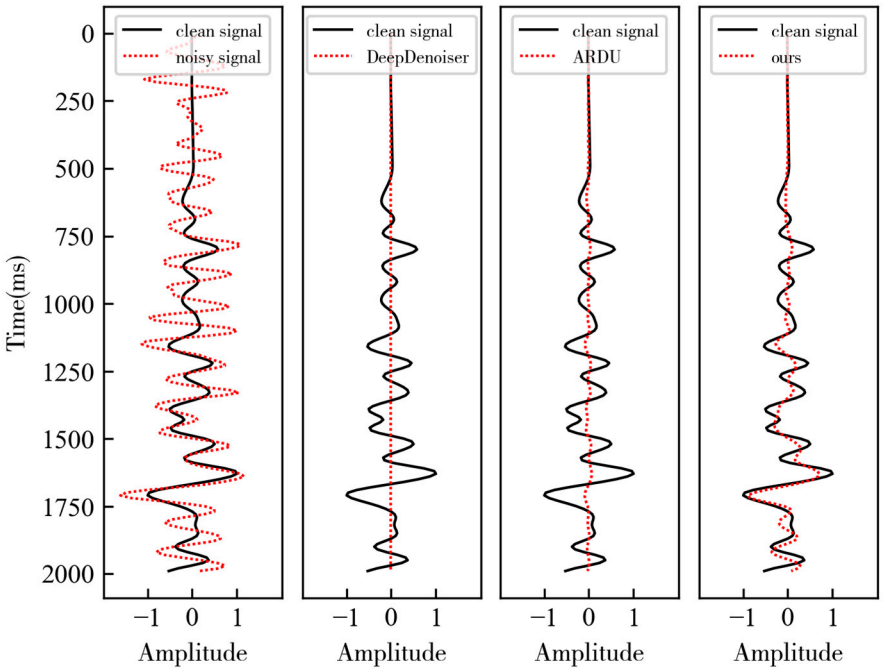

Figure 6. Noise reduction comparison in the time domain. The black solid line represents the original noise-free signal waveform, while the red dashed lines represent the denoising results of different models. DeepDenoiser and ARDU cause more damage to the effective signal, with average absolute errors of 0.159 and 0.148, respectively. The denoising result of the method proposed in this paper is closer to the original noise-free signal, with an average absolute error of 0.113.

To figure out the denoising performance of the three methods in the time–frequency domain, a comparative analysis of the time–frequency representations before and after denoising is conducted. As illustrated in Figure 7, the seismic signal is mainly in the low-frequency band below 10 Hz, and low-frequency noise within the same frequency range is added. After denoising, both DeepDenoiser and ARDU cause significant damage to the P-wave, while the proposed method in this paper preserves the majority of the P-wave energy, yielding the closest resemblance to the original noise-free signal. In Figure 8, we added high-frequency noise to the clean signal (a). Both DeepDenoiser and ARDU have different degrees of residual high-frequency noise. The denoising result of the proposed method is the closest to (a).

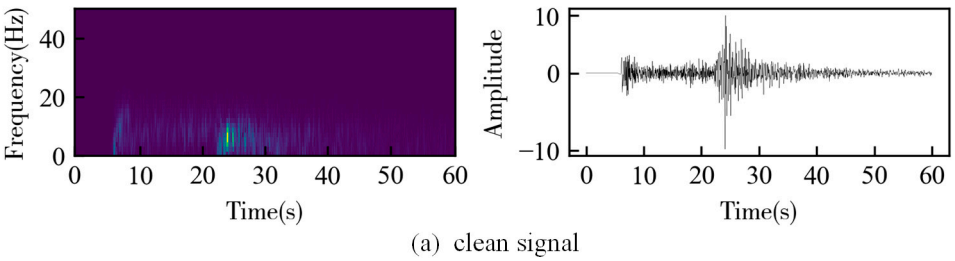

Figure 7. *Cont.*

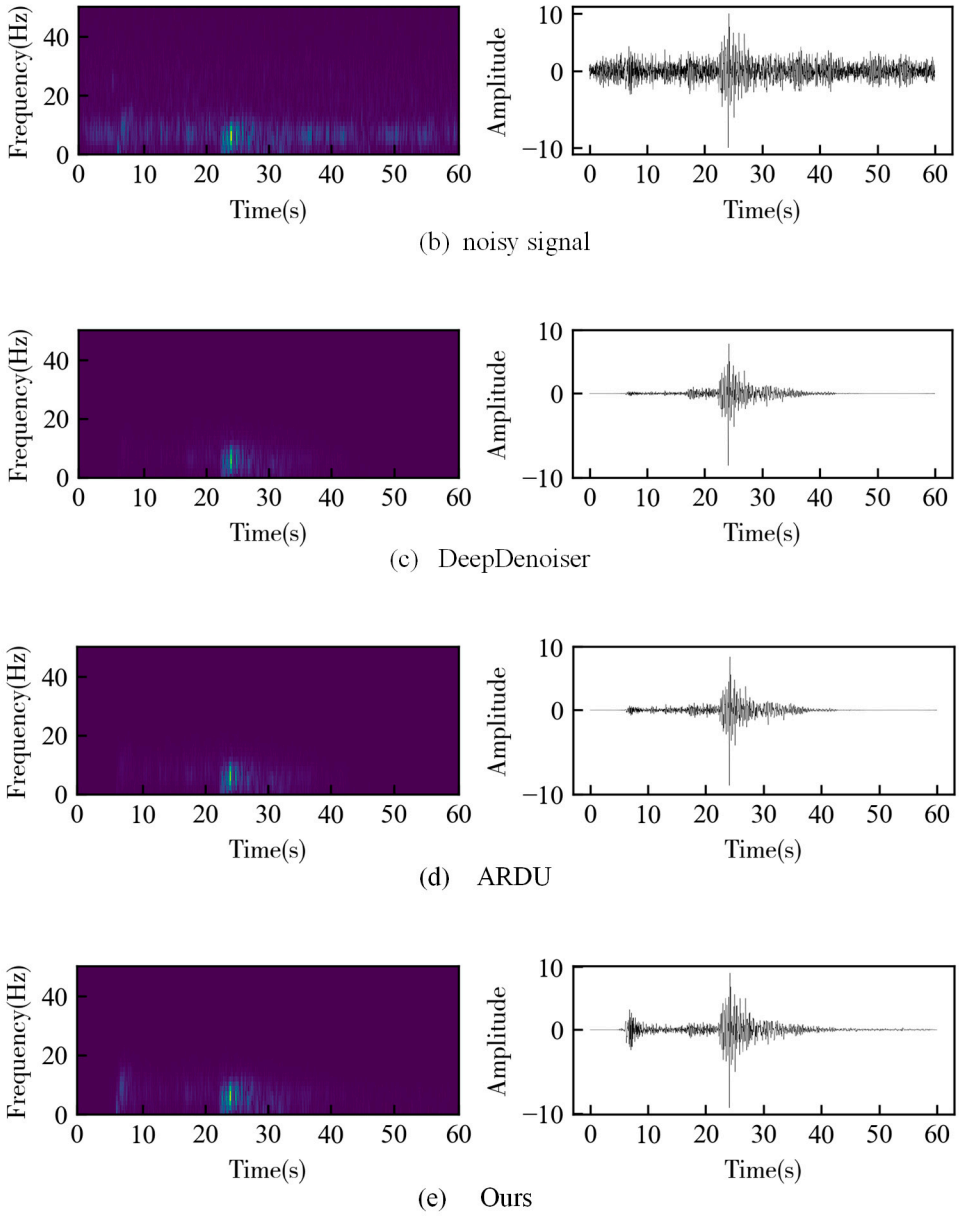

Figure 7. Noise reduction comparison for low-frequency noise. Each subfigure shows the STFT spectrum on the left side and the time-series signal on the right side. In (**c**,**d**), although DeepDenoiser and ARDU removed noise across all frequency ranges, they caused significant damage to the energy of the P-wave, substantially attenuating the P-wave amplitude. In (**e**), the method proposed in this paper not only removes noise but also preserves the energy of the effective signal.

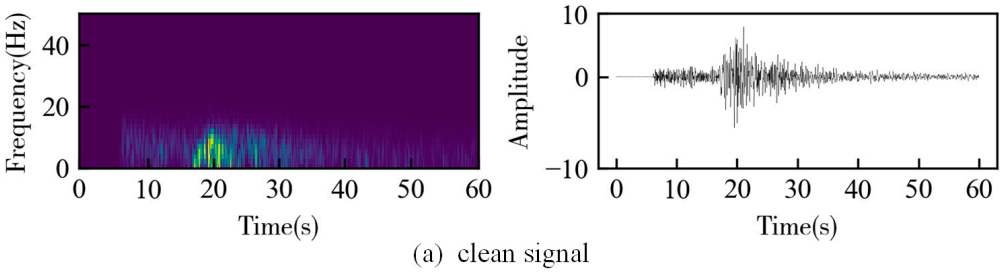

Figure 8. *Cont.*

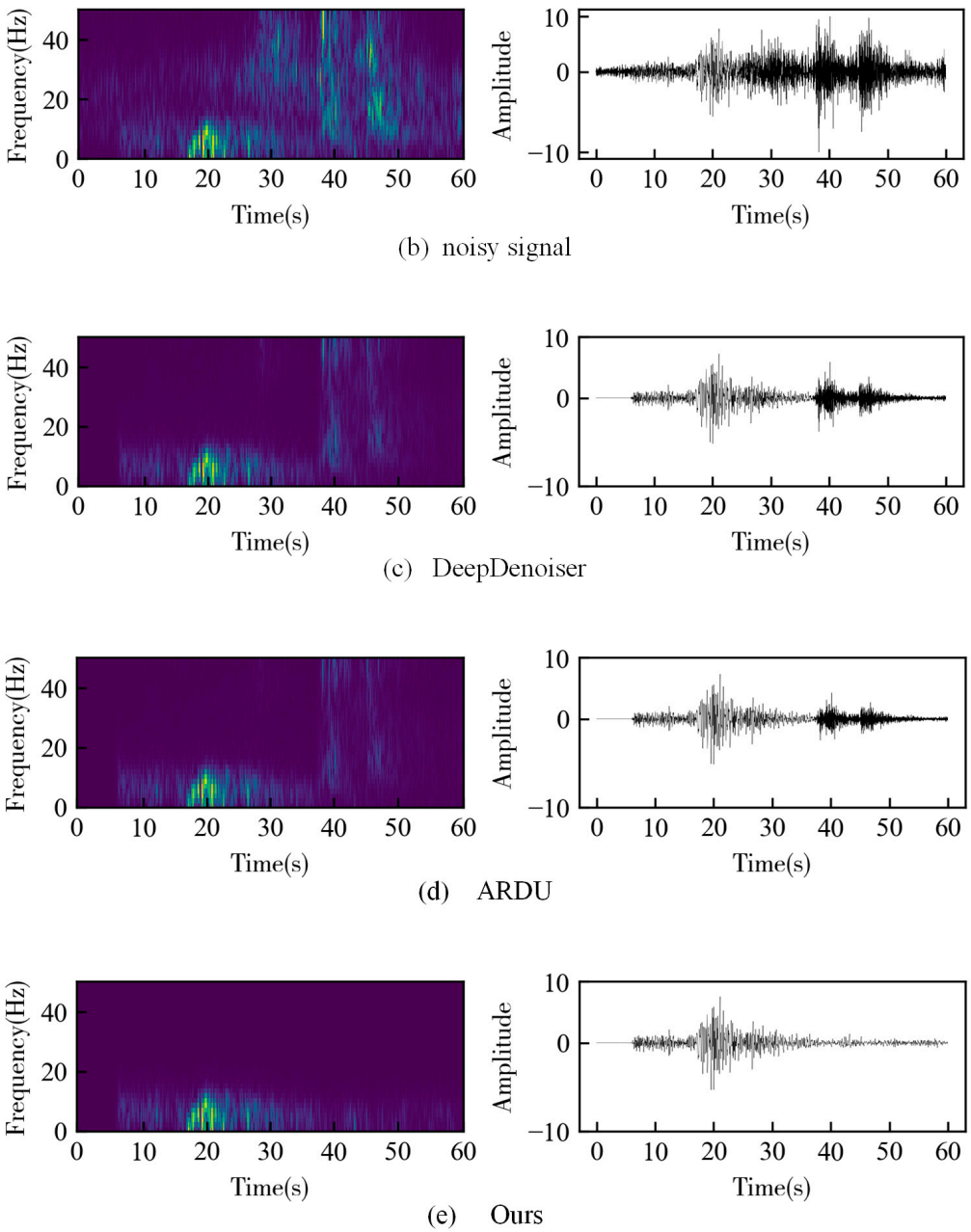

Figure 8. Noise reduction comparison for high-frequency noise. Each subfigure shows the STFT spectrum on the left side and the time-series signal on the right side. In (**c**), the denoising result of DeepDenoiser retains obvious high-frequency noise. In (**d**), from the time-series signal, we can see that ARDU also retains some high-frequency noise. In (**e**), most of the noise has been removed and the denoising result of the proposed method is closest to (**a**).

As shown in Table 1, all three methods significantly improve the SNR after denoising. When the input SNR is lower than 4, the proposed method achieves a higher denoised SNR. However, for input signals with SNR greater than 4, the performance of the proposed method is outperformed by the other two methods. DeepDenoiser and ARDU tend to completely suppress the portion before the P-wave arrival to zero, thereby producing higher SNR. However, the proposed method uses a discriminator to evaluate the difference between the denoised result and the true signal. Since high-quality real signals also exhibit small fluctuations before the P-wave arrival, the denoising results of this method sometimes do not completely suppress

the portion before the P-wave arrival to zero, leading to a relatively lower SNR of the denoised results.

Table 1. Comparison of SNR after noise reduction.

Signal SNR (dB)	DeepDenoiser	ARDU	Ours
−1	2.568	4.933	6.945
0	4.086	7.593	9.384
1	10.802	11.650	14.471
2	12.643	13.001	14.229
3	14.847	15.480	15.502
4	19.111	20.444	16.081
5	21.425	21.065	16.021
6	21.934	22.351	15.857
7	23.350	23.015	16.173

The correlation coefficient serves as a metric to quantify the similarity between the denoised signal and the original noise-free signal. As presented in Table 2, after denoising, the proposed method yields signals that exhibit a higher correlation with the original noise-free signals compared to the other two methods.

Table 2. Comparison of correlation coefficient after noise reduction.

Signal SNR (dB)	DeepDenoiser	ARDU	Ours
−1	0.387	0.391	0.499
0	0.453	0.466	0.558
1	0.703	0.709	0.772
2	0.793	0.797	0.841
3	0.870	0.873	0.914
4	0.910	0.910	0.954
5	0.932	0.935	0.978
6	0.938	0.957	0.976
7	0.942	0.960	0.974

The mean absolute error evaluates the degree of distortion inflicted upon the effective signal during the denoising process. As illustrated in Table 3, the proposed method incurs the least signal distortion compared to the other two methods.

Table 3. Comparison of mean absolute error after noise reduction.

Signal SNR (dB)	DeepDenoiser	ARDU	Ours
−1	0.249	0.242	0.239
0	0.248	0.242	0.237
1	0.197	0.194	0.187
2	0.185	0.161	0.163
3	0.170	0.147	0.132
4	0.151	0.126	0.109
5	0.139	0.115	0.096
6	0.147	0.119	0.096
7	0.147	0.111	0.098

Tables 1–3 present the denoising results for the test set, which contains 9476 signals. We excluded signals with an SNR lower than −1 dB or higher than 7 dB from the test set, and the tables report the statistics from the remaining 3211 signals.

To figure out the computational costs of different models, we compared the floating-point operations (FLOPs) and a number of parameters of DeepDenoiser, ARDU, and the proposed method. As shown in Table 4, DeepDenoiser requires the least FLOPs and parameters, and the proposed method has slightly lower FLOPs and parameters than ARDU.

Considering that ARDU is an improvement over DeepDenoiser, incorporating dilated convolutions and residual dense blocks, simply increasing the parameters of DeepDenoiser to match the proposed method does not necessarily achieve better denoising performance. It can be considered that with the same number of parameters, the proposed method would achieve the best results.

Table 4. Comparison of computational cost.

Method	Floating-Point Operations (G)	Number of Parameters (M)
DeepDenoiser	0.31	2.65
ARDU	6.83	40.87
Ours	6.67	17.27

3.2. Field Data

In a real environment, the data acquisition environment for seismic signals is often more complex, facing various interference sources and errors introduced by different equipment. To evaluate the performance of various denoising methods on real data, we use a subset of low signal-to-noise ratio seismic data from the STEAD dataset and a subset of data from the ChinArray dataset as test sets and compare the denoising effects of different models on these field data.

The average SNR of the original data was 4.119 dB. After denoising, the SNR achieved by DeepDenoiser was 9.865 dB, while ARDU attained an SNR of 11.685 dB. In contrast, the proposed method in this study yielded a significantly higher denoised SNR of 14.531.

We also compared the denoising effects of the three methods on the earthquake recorded by the ChinArray X1 network at the intersection of Nantou County, Chiayi County, and Kaohsiung County in Taiwan (occurring on 31 December 2012, at 00:03:25, with a magnitude of 4.8 and a depth of 10 km). As shown in Figure 9, We arranged the signals according to the epicentral distance, and took the theoretical arrival time of the P-wave as the zero moment. It is expected that high-quality seismic images can clearly observe the arrival of the P wave and S wave. All three methods effectively improved the quality of the seismic image, making the seismic phases more distinct and enhancing their continuity, demonstrating their effectiveness in suppressing noise to varying degrees. In (b), it can be observed that the denoised results obtained by DeepDenoiser have a lower resolution, with noticeable noise residuals between 1600 and 1800 km from the epicenter. In (c), the denoised results with ARDU are clearer, but compared to the proposed method, there are still noise residuals around 1500 km from the epicenter. Therefore, the proposed method can significantly improve the quality of the seismic profile, making the seismic phases more stable and distinct.

We also calculated the noise residuals after denoising to compare the three methods' effectiveness in preserving the actual seismic signal amplitudes. The residual of the denoising result is the separated noise, which is supposed to be disorderly and cannot observe the P wave and S wave at all. As shown in Figure 10, apparent seismic phases can be observed in the noise residual of DeepDenoiser, indicating that it has damaged the effective signal while suppressing noise. In contrast, no seismic phases are observable in the noise residual of ARDU and the proposed method, suggesting that they have caused less damage to the true signal.

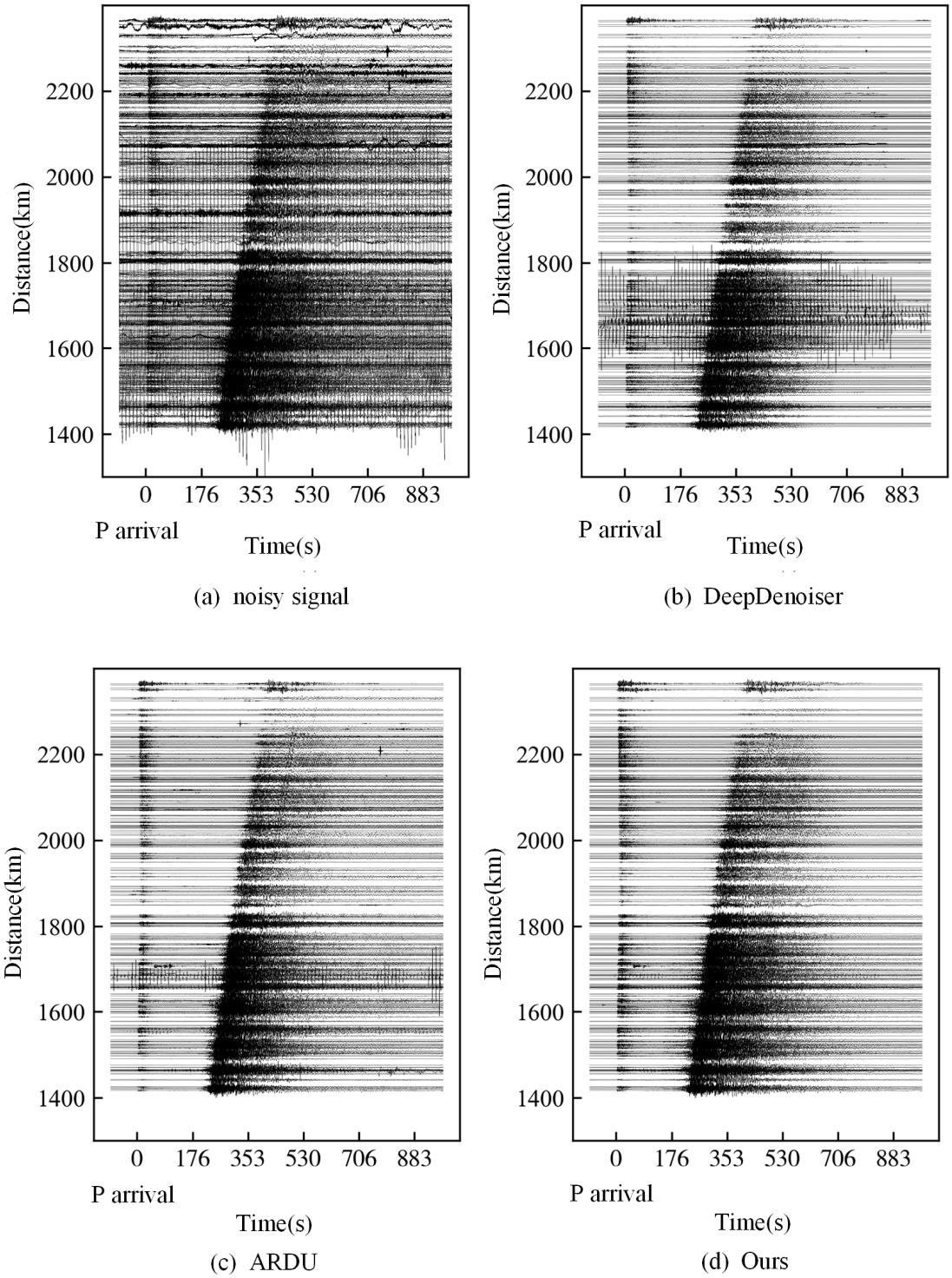

Figure 9. Noise reduction comparison for ChinArray X1 Network. (**a**) shows the original data. (**b**–**d**) are the denoising results of DeepDenoiser, ARDU, and the method proposed in this paper, respectively. All seismic signals are arranged according to the epicentral distance. In the denoising result of DeepDenoiser, records with epicentral distances between 1600 and 1800 km have noticeable noise. In the denoising result of ARDU, records around 1700 km epicentral distance have some residual noise. In the denoising result of the proposed method, there is no noticeable residual noise, and the in-phase axis is relatively clear.

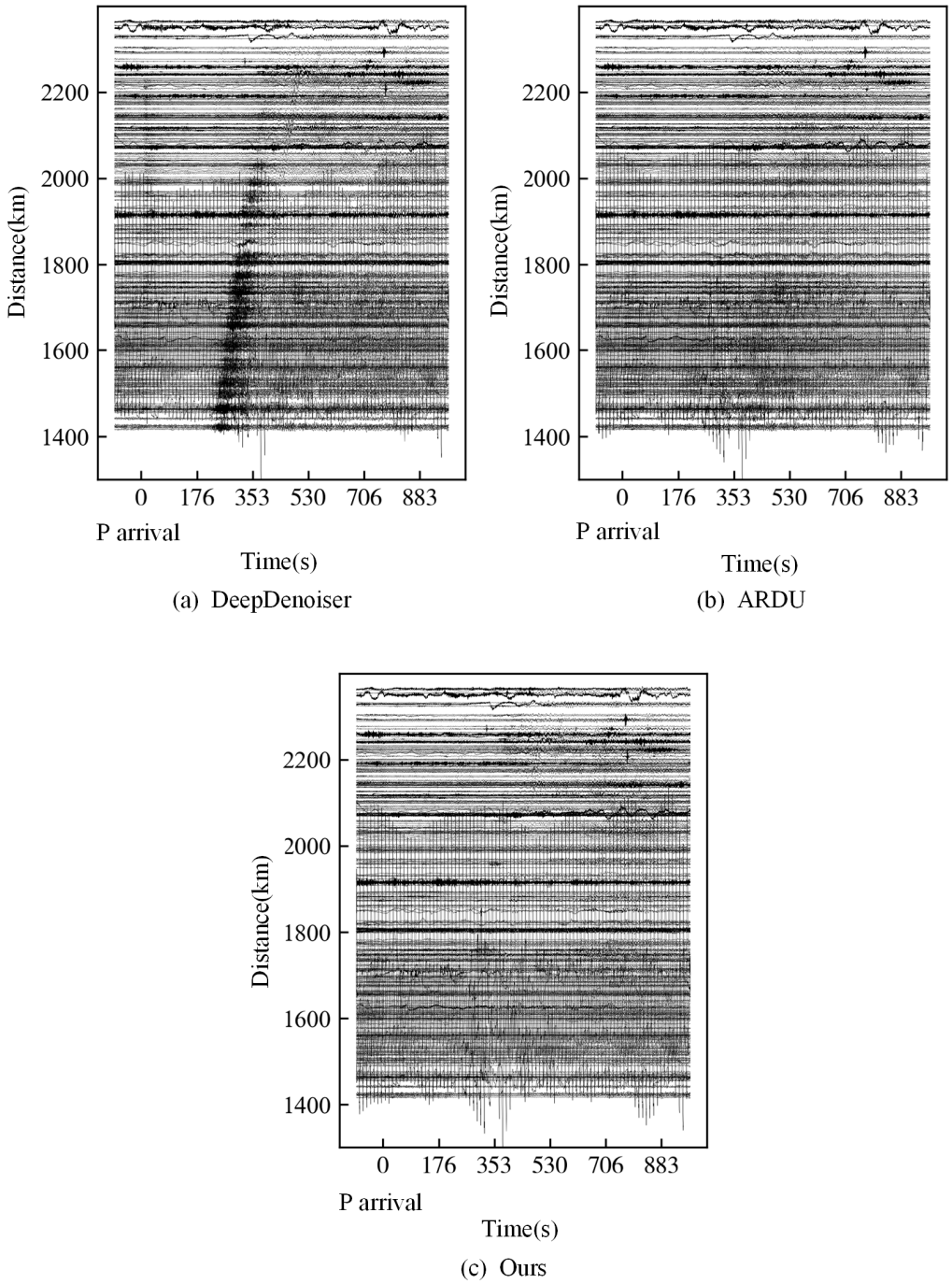

Figure 10. Noise residual comparison for ChinArray X1 Network. (**a**–**c**) represent the noise signals separated by DeepDenoiser, ARDU, and the method proposed in this paper, respectively. In (**a**), the S-wave and weak P-wave phases can be clearly observed, indicating that DeepDenoiser damaged the effective signal during the denoising process. In (**b**,**c**), there are no obvious seismic phases, suggesting that both ARDU and the proposed method cause less damage to the effective signal.

4. Discussion

This paper proposes a novel deep learning method for seismic signal denoising based on the time–frequency domain representation of the signal. The proposed method utilizes a generative adversarial network (GAN) to separate the seismic signal and noise in the time–frequency domain. The generator network takes the real and imaginary parts of the short-time Fourier transform (STFT) of the seismic signal as input and outputs two time–frequency masked signals to separate the seismic signal and noise. The discriminator

network consists of a feature extraction module and a discrimination module. The feature extraction module adaptively extracts signal features and calculates perceptual losses, while the discrimination module determines whether the input signal is a true signal or noise. Compared to existing deep learning methods, the proposed method employs multiple loss functions to comprehensively evaluate the denoising effectiveness, achieving better denoising performance, especially for low SNR signals. To demonstrate the effectiveness of the proposed method, this paper conducts qualitative and quantitative analyses of the denoising results on synthetic and real seismic data. The experimental results demonstrate that compared to DeepDenoiser and ARDU, the proposed method can more effectively suppress noise while preserving the true signal.

Author Contributions: Formal analysis, M.W. and J.Z.; methodology, M.W.; writing, M.W.; review and editing, X.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (grant No. XDB42020304) and the National Natural Science Foundation of China (grant No. 42074059).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data source: STEAD ref. [38]. ChinArray X1 Network waveform data were provided by the International Earthquake Science Data Center (Doi:10.11998/IESDC).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Brett, H.; Hawkins, R.; Waszek, L.; Lythgoe, K.; Deuss, A. 3D Transdimensional Seismic Tomography of the Inner Core. *Earth Planet. Sci. Lett.* **2022**, *593*, 117688. [CrossRef]
2. Tassiopoulou, S.; Koukiou, G.; Anastassopoulos, V. Algorithms in Tomography and Related Inverse Problems—A Review. *Algorithms* **2024**, *17*, 71. [CrossRef]
3. Zhang, J.; Shi, M.; Wang, D.; Tong, Z.; Hou, X.; Niu, J.; Li, X.; Li, Z.; Zhang, P.; Huang, Y. Fields and Directions for Shale Gas Exploration in China. *Nat. Gas Ind. B* **2022**, *9*, 20–32. [CrossRef]
4. Wang, W.; Xue, C.; Zhao, J.; Yuan, C.; Tang, J. Machine Learning-Based Field Geological Mapping: A New Exploration of Geological Survey Data Acquisition Strategy. *Ore Geol. Rev.* **2024**, *166*, 105959. [CrossRef]
5. da Silva, S.L.; Costa, F.; Karsou, A.; Capuzzo, F.; Moreira, R.; Lopez, J.; Cetale, M. Research Note: Application of Refraction Full-Waveform Inversion of Ocean Bottom Node Data Using a Squared-Slowness Model Parameterization. *Geophys. Prospect.* **2024**, *72*, 1189–1195. [CrossRef]
6. Fehler, M.C.; Huang, L. Modern Imaging Using Seismic Reflection Data. *Annu. Rev. Earth Planet. Sci.* **2002**, *30*, 259–284. [CrossRef]
7. Wang, F.; Yang, B.; Wang, Y.; Wang, M. Learning From Noisy Data: An Unsupervised Random Denoising Method for Seismic Data Using Model-Based Deep Learning. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–14. [CrossRef]
8. Cao, S.; Chen, X. The Second-Generation Wavelet Transform and Its Application in Denoising of Seismic Data. *Appl. Geophys.* **2005**, *2*, 70–74. [CrossRef]
9. Chen, X.; He, Z. Improved S-Transform and Its Application in Seismic Signal Processing. Available online: https://www.researchgate.net/publication/298264611_Improved_S-transform_and_its_application_in_seismic_signal_processing (accessed on 15 March 2024).
10. Jicheng, L.; Gu, Y.; Chou, Y.; Gu, J. Seismic Data Random Noise Reduction Using a Method Based on Improved Complementary Ensemble EMD and Adaptive Interval Threshold. *Explor. Geophys.* **2021**, *52*, 137–149. [CrossRef]
11. Li, W.; Liu, H.; Wang, J. A Deep Learning Method for Denoising Based on a Fast and Flexible Convolutional Neural Network. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–13. [CrossRef]
12. Zhong, T.; Cheng, M.; Dong, X.; Wu, N. Seismic Random Noise Attenuation by Applying Multiscale Denoising Convolutional Neural Network. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–13. [CrossRef]
13. Zhao, H.; Zhou, Y.; Bai, T.; Chen, Y. A U-Net Based Multi-Scale Deformable Convolution Network for Seismic Random Noise Suppression-All Databases. Available online: https://webofscience.clarivate.cn/wos/alldb/full-record/WOS:001072548400001 (accessed on 3 February 2024).
14. Lan, T.; Han, L.; Zeng, Z.; Zeng, J. An Attention-Based Residual Neural Network for Efficient Noise Suppression in Signal Processing. *Appl. Sci.* **2023**, *13*, 5262. [CrossRef]
15. Zhu, W.; Mousavi, S.M.; Beroza, G.C. Seismic Signal Denoising and Decomposition Using Deep Neural Networks. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 9476–9488. [CrossRef]

16. Gao, Z.; Zhang, S.; Cai, J.; Hong, L.; Zheng, J. Research on Deep Convolutional Neural Network Time-Frequency Domain Seismic Signal Denoising Combined with Residual Dense Blocks. *Front. Earth Sci.* **2021**, *9*, 681869. [CrossRef]
17. Cai, J.; Wang, L.; Zheng, J.; Duan, Z.; Li, L.; Chen, N. Denoising Method for Seismic Co-Band Noise Based on a U-Net Network Combined with a Residual Dense Block. *Appl. Sci.* **2023**, *13*, 1324. [CrossRef]
18. Donoho, D.L.; Johnstone, I.M. Ideal Spatial Adaptation by Wavelet Shrinkage. *Biometrika* **1994**, *81*, 425–455. [CrossRef]
19. Li, Y.; Wang, N.; Shi, J.; Liu, J.; Hou, X. Revisiting Batch Normalization For Practical Domain Adaptation. *arXiv* **2019**, arXiv:1603.04779.
20. Singh, A.; Hingane, S.; Gong, X.; Wang, Z. SAFIN: Arbitrary Style Transfer with Self-Attentive Factorized Instance Normalization. In Proceedings of the 2021 IEEE International Conference on Multimedia and Expo (ICME), Shenzhen, China, 5 July 2021; IEEE: New York, NY, USA, 2021; pp. 1–6.
21. Huang, L.; Qin, J.; Zhou, Y.; Zhu, F.; Liu, L.; Shao, L. Normalization Techniques in Training DNNs: Methodology, Analysis and Application. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 10173–10196. [CrossRef] [PubMed]
22. Shi, Y.; Huang, Z.; Huang, Z.; Hua, X.; Hong, H.; Li, L. HINRDNet: A Half Instance Normalization Residual Dense Network for Passive Millimetre Wave Image Restoration. *Infrared Phys. Technol.* **2023**, *132*, 104722. [CrossRef]
23. Tarasiewicz, T.; Nalepa, J.; Farrugia, R.A.; Valentino, G.; Chen, M.; Briffa, J.A.; Kawulok, M. Multitemporal and Multispectral Data Fusion for Super-Resolution of Sentinel-2 Images. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 1–19. [CrossRef]
24. Chen, L.; Lu, X.; Zhang, J.; Chu, X.; Chen, C. HINet: Half Instance Normalization Network for Image Restoration. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Nashville, TN, USA, 19–25 June 2021; IEEE: New York, NY, USA, 2021; pp. 182–192.
25. Springenberg, J.T.; Dosovitskiy, A.; Brox, T.; Riedmiller, M. Striving for Simplicity: The All Convolutional Net. *arXiv* **2019**, arXiv:1412.6806.
26. Odena, A.; Dumoulin, V.; Olah, C. Deconvolution and Checkerboard Artifacts. *Distill* **2016**, *1*, e3. [CrossRef]
27. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; IEEE: New York, NY, USA, 2016; pp. 770–778.
28. Zhao, M.; Zhong, S.; Fu, X.; Tang, B.; Pecht, M. Deep Residual Shrinkage Networks for Fault Diagnosis. *IEEE Trans. Ind. Inform.* **2020**, *16*, 4681–4690. [CrossRef]
29. Lin, L.; Zhong, Z.; Li, C. SeisGAN: Improving Seismic Image Resolution and Reducing Random Noise Using a Generative Adversarial Network | Mathematical Geosciences. Available online: https://link.springer.com/article/10.1007/s11004-023-10103-8 (accessed on 13 December 2023).
30. Li, Y.; Wang, S.; Jiang, M. Seismic Random Noise Suppression by Using MSRD-GAN. *Geoenergy Sci. Eng.* **2023**, *222*, 211410. [CrossRef]
31. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556.
32. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
33. Shi, W.; Caballero, J.; Huszar, F.; Totz, J.; Aitken, A.P.; Bishop, R.; Rueckert, D.; Wang, Z. Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; IEEE: New York, NY, USA, 2016; pp. 1874–1883.
34. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In *Computer Vision—ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2018; Volume 11211, pp. 3–19, ISBN 978-3-030-01233-5.
35. Canny, J. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *PAMI-8*, 679–698. [CrossRef]
36. Ma, C.; Rao, Y.; Cheng, Y.; Chen, C.; Lu, J.; Zhou, J. Structure-Preserving Super Resolution with Gradient Guidance. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2020, Seattle, WA, USA, 13–19 June 2020; pp. 7769–7778.
37. Furnari, A.; Farinella, G.M.; Bruna, A.R.; Battiato, S. Generalized Sobel Filters for Gradient Estimation of Distorted Images. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 27–30 September 2015; pp. 3250–3254.
38. Mousavi, S.M.; Sheng, Y.; Zhu, W.; Beroza, G.C. STanford EArthquake Dataset (STEAD): A Global Data Set of Seismic Signals for AI. *IEEE Access* **2019**, *7*, 179464–179476. [CrossRef]