

Poststack Seismic Data Denoising Based on 3-D Convolutional Neural Network

Dawei Liu, Wei Wang, Xiaokai Wang[✉], *Member, IEEE*, Cheng Wang, Jiangyun Pei, and Wenchao Chen[✉]

Abstract—Deep learning has been successfully applied to image denoising. In this study, we take one step forward by using deep learning to suppress random noise in poststack seismic data from the aspects of network architecture and training samples. On the one hand, poststack seismic data denoising mainly aims at 3-D seismic data. We designed an end-to-end 3-D denoising convolutional neural network (3-D-DnCNN) that takes raw 3-D cubes as input in order to better extract the features of the 3-D spatial structure of poststack seismic data. On the other hand, denoising images with deep learning require noisy-clean sample pairs for training. In the field of seismic data processing, researchers usually try their best to suppress noise by using complex processes that combine different methods, but clean labels of seismic data are not available. In addition, building training samples in field seismic data has become an interesting but challenging problem. Therefore, we propose a training sample selection method that contains a complex workflow to produce comparatively ideal training samples. Experiments in this study demonstrate that deep learning can directly learn the ability to denoise field seismic data from selected samples. Although the building of the training samples may occur through a complex process, the experimental results of synthetic seismic data and field seismic data show that the 3-D-DnCNN has learned the ability to suppress the Gaussian noise and super-Gaussian noise from different training samples. Moreover, the 3-D-DnCNN network has better denoising performance toward arc-like imaging noise. In addition, we adopt residual learning and batch normalization in order to accelerate the training speed. After network training is satisfactorily completed, its processing efficiency can be significantly higher than that of conventional denoising methods.

Index Terms—3-D, convolutional neural networks (CNNs), seismic data denoising, training sample selection.

Manuscript received February 2, 2019; revised July 28, 2019; accepted September 4, 2019. Date of publication November 6, 2019; date of current version February 26, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 41774135, Grant 41974131, and Grant 41504093, in part by the National Key R&D Plan under Grant 2017YFB0202902, in part by the Fundamental Research Funds for the Central University under Grant xjj2016065, in part by the NSFC-SINOPEC Joint Key Project under Grant U1663207, and in part by the National Science and Technology Major Project under Grant 2017ZX05049-002. (*Corresponding author: Wenchao Chen.*)

D. Liu, X. Wang, and W. Chen are with the School of Information and Communication Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: liudawei2015@stu.xjtu.edu.cn; xkwang@xjtu.edu.cn; wenchao@xjtu.edu.cn).

W. Wang is with InvestBrain, DataYes, Shanghai 200085, China (e-mail: wwangcharles@qq.com).

C. Wang and J. Pei are with the Exploration and Development Research Institute, Daqing Oilfield Company, Ltd., Daqing 163712, China (e-mail: wangcheng88@petrochina.com.cn; jiangyunpei@petrochina.com.cn).

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TGRS.2019.2947149

I. INTRODUCTION

SUPPRESSING random noise in seismic data is a conventional but still an active topic in seismic signal processing because random noise is one of the main factors that lead to reductions in the signal-to-noise ratio (SNR) of seismic data and may further cause unreliable seismic inversion results. Random noise suppression is divided into prestack denoising and poststack denoising. The purpose of prestack denoising is to suppress the noise interference on the original field records as much as possible. The purpose of poststack denoising is to suppress the noise that is not completely suppressed by prestack denoising and the noise caused by processing itself in order to further improve the quality of the section. This study mainly aims at denoising 3-D poststack seismic data.

Many researchers have applied some advanced signal processing techniques to suppress the random noise in seismic data during the past several years. Principle component analysis (PCA) [1], polynomial fitting [2], singular value decomposition (SVD) [3]–[5], K-L transform [6], [7], independent component analysis (ICA) [8], [9], the predicting filter [10]–[12], and the Cadzow filter [13], [14] are the common methods of random noise suppression based on correlation. The above-mentioned methods have achieved some good practical applications. In addition, many time–frequency tools have been applied to suppress the random noise in seismic data. The Fourier transform is the basis of many methods of seismic data denoising [15], including a discrete cosine transform (DCT) [16] and domain deconvolution filtering [17]. In order to make the spectrum effectively represent the local features of time signals, the method of time-windowing is proposed. This includes the short-time Fourier transform (STFT) [18]. The wavelet transform [19], [20] can decompose signals on multiple scales by choosing a suitable wavelet basis function. Therefore, the wavelet transform has good localization characteristics in both the time domain and the frequency domain.

By damping or thresholding the coefficients associated with noise in the wavelet domain, we can reject random noise in seismic data [21]. To better represent anisotropic seismic data, the multidirectional sparse transform was developed. The curvelet transform [22] is a typical sparse multiscale transform and has been widely used in seismic data denoising [23], [24]. In summary, all the above-mentioned signal processing tools denoise random noise by using different basis functions to represent the seismic data and then using a thresholding method in the transform domain. In order to represent seismic

data more sparsely, which makes the differences in the transform domain between a useful signal and random noise more obvious so that a useful signal and noise can be separated more easily, overcomplete dictionaries [25] for sparse representation were proposed. Overcomplete dictionaries can optimally find the best combination of some basis functions to represent data sparsely and have been applied to image denoising [26] and seismic data denoising [27]. In general, the denoising performance of overcomplete dictionaries is superior to that of a conventional fixed dictionary.

In addition, the adaptive filter [28], [29], edge-preserving smoothing [30], [31], empirical mode decomposition [32], nonlinear filter [33]–[35], the Bayesian inversion [36], and nonlocal means [37] were applied to denoise seismic data. The above-mentioned conventional methods can successfully suppress the random noise of seismic data, but each has its own shortcomings. A shared common shortcoming is that the parameters of the above-mentioned methods need to be artificially selected based on different seismic data. In general, the exploration area of seismic data is very large, and the seismic data of different regions possess some different features because of different underground structures. It is difficult to select optimal parameters for different data. In the face of this shortcoming, the supervised deep learning has attracted increased attention because it is a data-driven algorithm and has good robustness.

In recent years, deep learning has made significant achievements in the field of image recognition [38], [39], natural language processing [40], and autonomous driving [41]. Based on the great success of deep learning in the above-mentioned fields, many researchers have introduced deep learning in the field of seismic signal processing. Valentineu and Trampert [42] used an autoencoder to learn seismic waveform features and encoded high-dimensional seismic signal features into low-dimensional features, which is a key step for subsequent nonlinear tomography inversion. Valentine *et al.* [43] used an autoencoder to study geomorphologic features. This has the ability to recognize complex patterns, and the recognition accuracy can exceed 80%. Liu *et al.* [44] used deep learning to extract features automatically and solved the problem of sensor drift effectively. Dahlke *et al.* [45] and Araya-Polo *et al.* [46] used deep learning with different loss functions to predict faults from original seismic data and obtained high accuracy in synthetic data tests.

Moreover, Vikraman *et al.* [47] used deep learning to classify seismic waveforms in real time. After fully training, the identification rates of foreshock, mainshock, and aftershock could reach higher than 99%. Korjani *et al.* [48] used deep learning to predict the petrophysical parameters of a new well from a large amount of logging data. The predicted results were almost identical to the actual records of the new well. Wu and Cao [49] used a continuous restricted Boltzmann machine to extract lithologic features and used an SVM to carry out lithologic identification. The recognition accuracy reached 81.9%, which was higher than that of the principal component analysis. DeVries *et al.* [50] used deep learning to obtain the

effective expression of the viscoelastic solution of a large-scale rheological structure at any time and at any position, and the result had a higher temporal and spatial resolution. Cao [51] used deep learning to predict natural gas reservoirs and found that deep learning could extract differences in the intrinsic features of gas and nongas reservoirs. Wu *et al.* [52] used deep learning to estimate fault orientations and found that deep learning can accurately estimate fault orientations. Ma *et al.* [53] used deep learning to detect faults in 3-D seismic images and found that fault probability calculated by deep learning outperformed coherence technology. Gramstad and Nickel [54] used deep learning to detect salt boundary and found that deep learning could reduce the interpretation turnaround time of both top and base of salt. Li *et al.* [55] used deep learning to suppress the scattered ground-roll noise automatically.

As a typical deep learning method, convolutional neural networks (CNNs) have exhibited great denoising performance in the field of image denoising. Many network models have been successfully applied to image denoising, including the multilayer perceptron [56], stacked sparse denoising autoencoders [57], and denoising CNNs (DnCNNs) [58]. Compared with conventional image denoising methods, the above-mentioned network models can obtain a similar or even higher SNR and have a faster calculation speed owing to the heavy use of GPUs. To a certain degree, the random noise of seismic data is similar to the random noise of an image. Because CNNs have achieved good performance in image denoising, our motivation in this study is to explore seismic data denoising via CNNs. It should be pointed out that the random noise in this study also includes arc-like imaging noise, which is not consistent with a stratigraphic structure.

Seismic data are inherently 3-D or higher tensors. However, most of CNN denoising methods are based on the 2-D convolution kernels. While 2-D CNN has been shown useful in solving these problems [59], [60], it intrinsically loses the 3-D context of the original seismic data, which limits the performance of the overall system.

In this article, we propose a 3-D DnCNN (3-D-DnCNN) to suppress random noise in seismic data. The 3-D-DnCNN is expanded to 3-D from a DnCNN by using 3-D convolution kernels. Useful signals in poststack seismic data have coherence in 3-D space, and such a 3-D network can extract its 3-D space structural features more effectively than the 2-D networks. With a residual learning strategy, a 3-D-DnCNN can suppress random noise by implicitly removing the extracted useful 3-D signals in the hidden layers. However, the input data of a 3-D-DnCNN are some 3-D cubes of raw poststack seismic data, and the multidimensional input data bring more challenges compared to 2-D image denoising.

On the other hand, training samples play a significant role in the unparalleled success achieved by deep learning models [61]. However, constructing appropriate training samples in the seismic data denoising field is much more difficult than in the image denoising field. In general, the goal of image

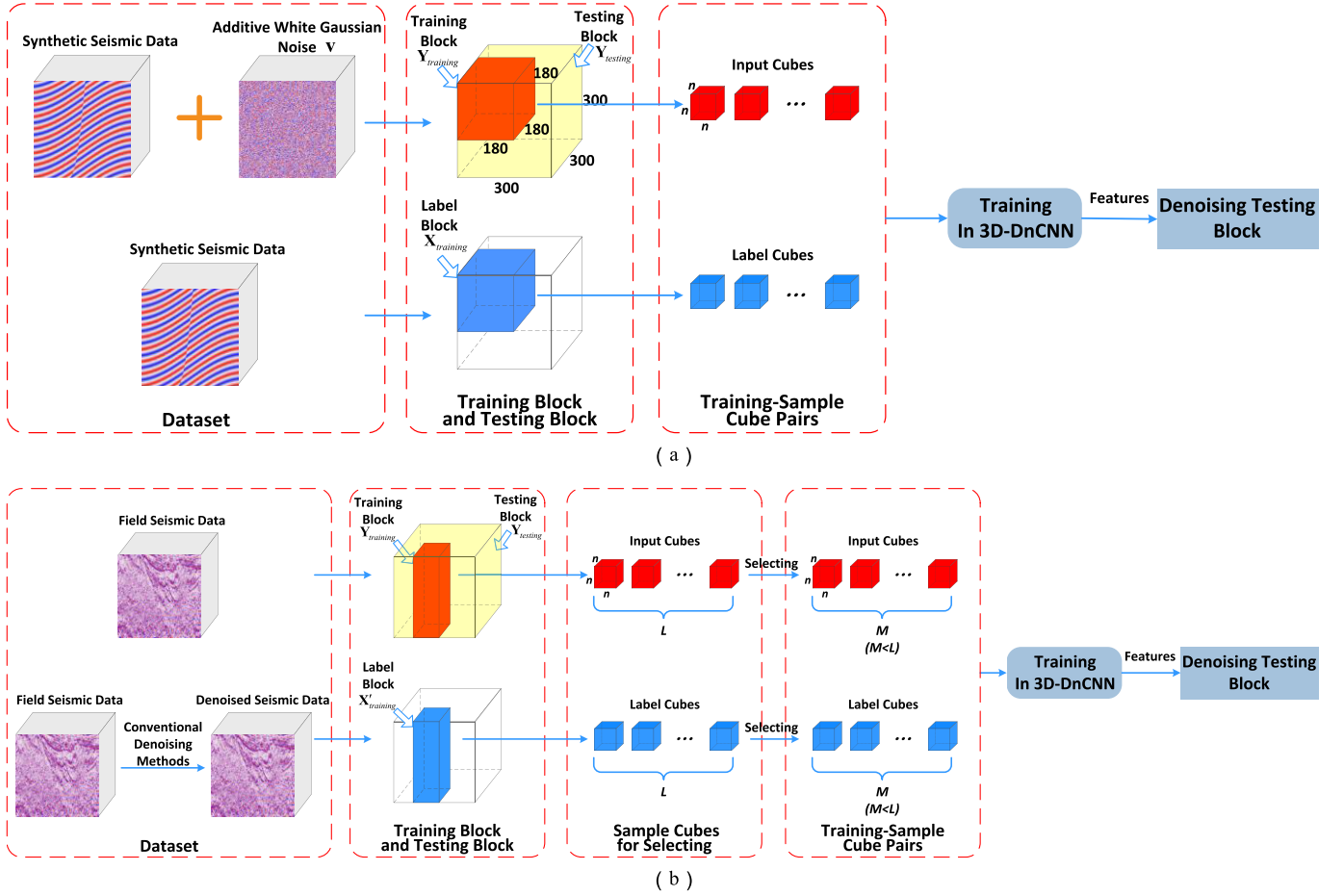


Fig. 1. Framework based on 3-D-DnCNN for seismic data denoising. (a) Denoising framework of synthetic seismic data. (b) Denoising framework of field seismic data.

denoising is to remove additive white Gaussian noise (AWGN) with a special standard deviation. There are clean images and noisy images at our fingertips. However, completely clean seismic data do not exist. Inspired by [62], a natural way to build training sampling is to regard the denoised seismic data by the conventional method as a clean label. However, the denoised seismic data with different conventional denoising methods is not exactly the same everywhere, which makes it difficult to find the ground truth (the clean seismic data) fed to the deep learning models. Therefore, this article aims to propose a 3-D denoising method with a 3-D-DnCNN model and proposes an effective strategy of selecting training samples for training in 3-D poststack seismic data.

To summarize, this article makes the following major contributions.

- 1) We propose an end-to-end trainable deep CNN for seismic data denoising. In contrast to some existing image deep neural network-based denoising methods, which are often 2-D networks, this network consists of 3-D convolution kernels.
- 2) An effective strategy for selecting training samples from field seismic data is proposed, through which relatively ideal training samples fed to the 3-D-DnCNN can be obtained.

- 3) This study validates the effectiveness of batch normalization (BN) and residual learning to denoise 3-D poststack seismic data.
- 4) Its uniform architecture design makes the 3-D-DnCNN a framework that generalizes well for both synthetic seismic data and field seismic data. More importantly, the 3-D-DnCNN can achieve similar or even better denoising results compared to conventional denoising methods when using almost the same parameters.

The remainder of this article is organized as follows. Section II provides the detailed architecture of the 3-D-DnCNN, including building training samples for synthetic seismic data and field seismic data. In Section III, synthetic seismic data and field seismic data denoising experiments are conducted to evaluate the 3-D-DnCNN, and some discussions are offered. Finally, some conclusions are given in Section IV.

II. PROPOSED FRAMEWORK

Fig. 1 shows the entire deep learning framework of 3-D seismic data denoising based on a 3-D-DnCNN. In this framework, the synthetic seismic data and field seismic data are separated into two blocks: a training block and a testing block. The original 3-D noisy seismic data volume Y is divided into $Y_{training}$ and $Y_{testing}$. $Y_{training}$ and the corresponding label

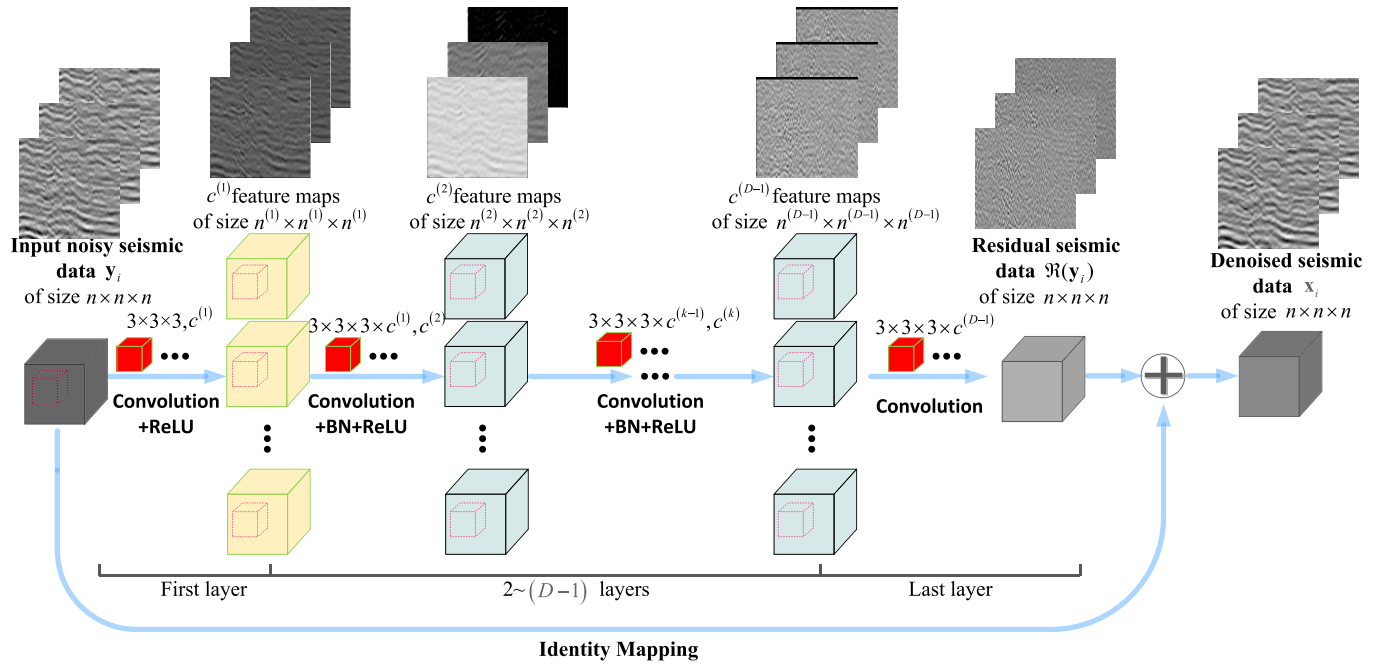


Fig. 2. Architecture of the proposed 3-D-DnCNN network for seismic data denoising.

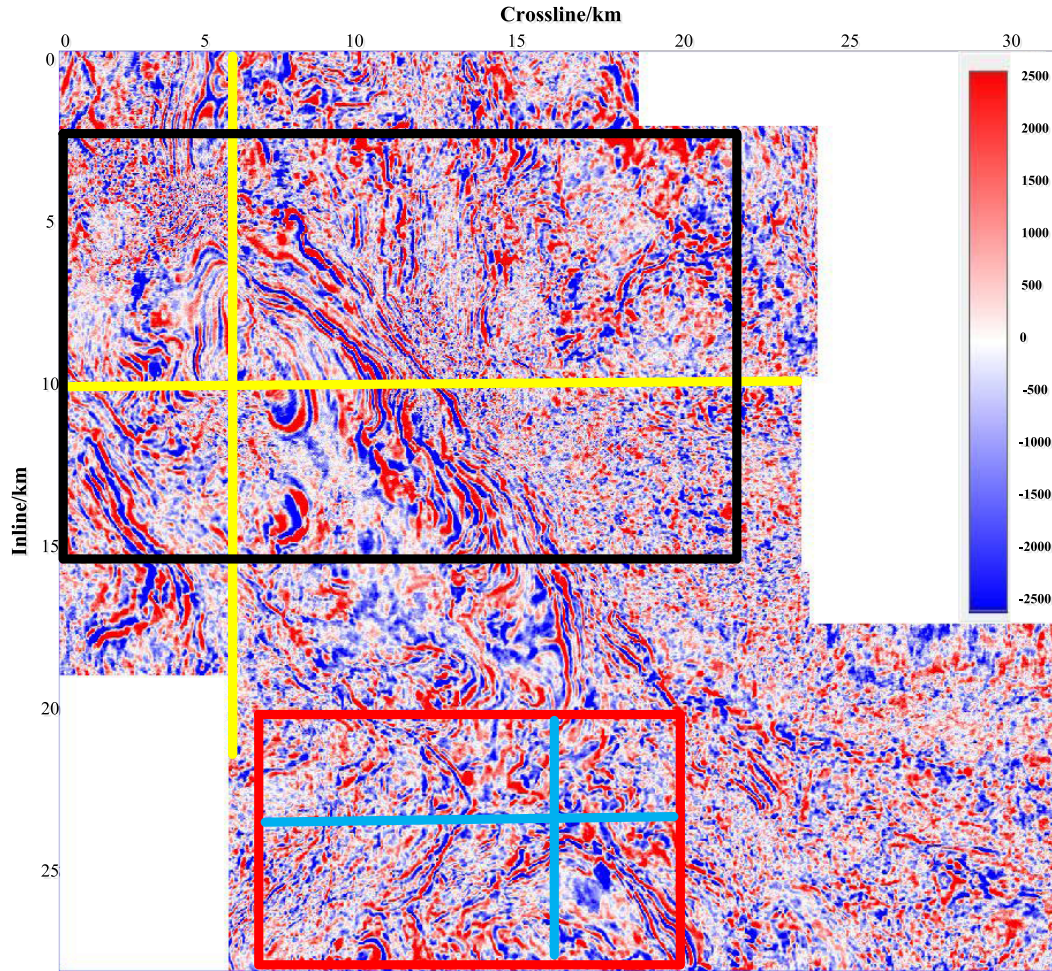


Fig. 3. Time slice of field seismic data.

$\mathbf{X}_{\text{training}}$ lay the foundation for organizing training samples fed to the 3-D-DnCNN. As the synthetic seismic data and

field seismic data have different characteristics, we choose different training-sample selection methods to select training

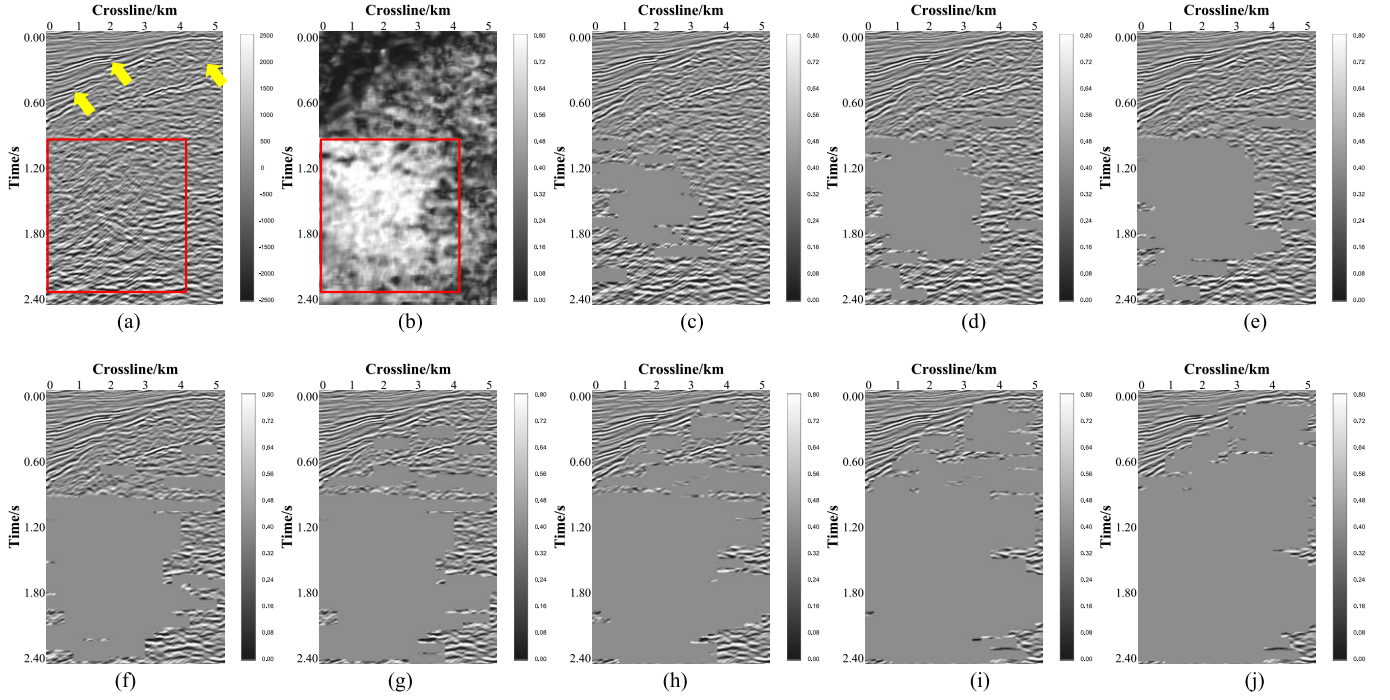


Fig. 4. Sample selection with different thresholds. (a) Original seismic data profile. (b) Calculating faulty confidence with original seismic data profile. (c)–(j) Results with thresholds of 0.80, 0.75, 0.70, 0.65, 0.60, 0.55, 0.50, and 0.45, respectively.

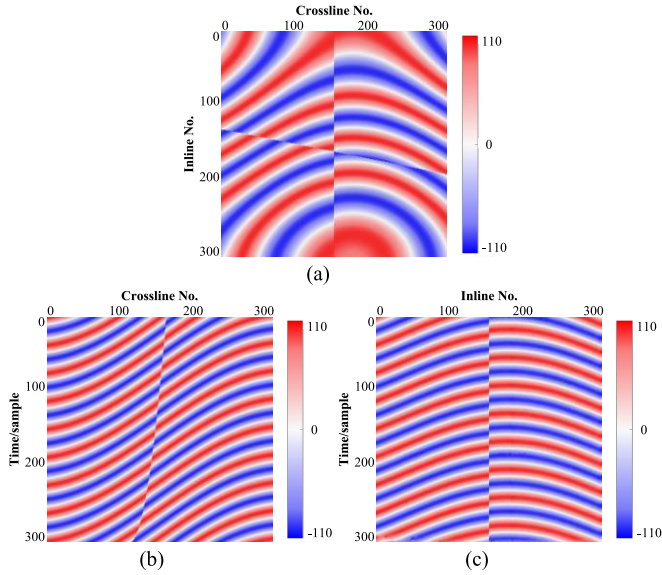


Fig. 5. Synthetic seismic data. (a) Time slice of synthetic seismic data at the 100th sampling point. (b) Inline profile of synthetic seismic data at the 100th sampling point. (c) Crossline profile of synthetic seismic data at the 100th sampling point.

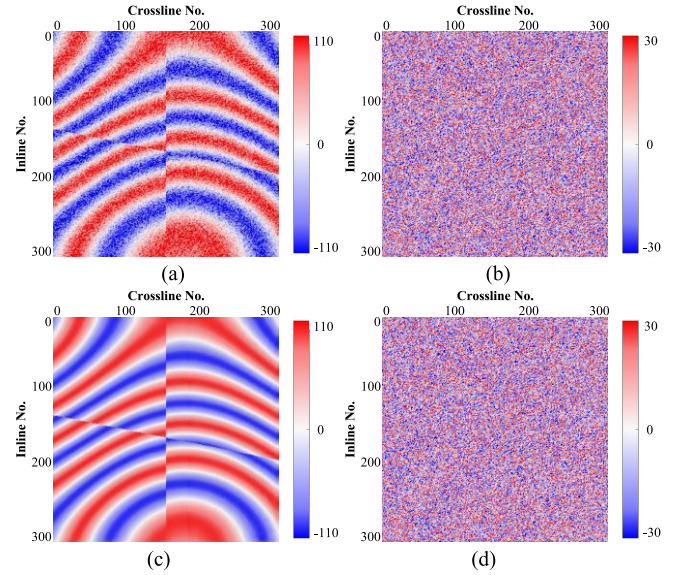


Fig. 6. Noise suppression on time slice. (a) Noisy data. (b) Noise added to time slice. (c) Denoised by the proposed method. (d) Noise removed by the proposed method.

samples from $\mathbf{Y}_{\text{training}}$ and $\mathbf{X}_{\text{training}}$. Then, the selected training samples are fed to the 3-D-DnCNN. After the hyperparameters for training are configured, the 3-D-DnCNN is trained for dozens of epochs. Finally, the testing block $\mathbf{Y}_{\text{testing}}$ is employed to test the denoising performance of the trained 3-D-DnCNN.

A. Training Samples

Training samples are very important because they determine the features extracted by the network model and affect

the denoising performance of the network model. Different training-sample selection methods are adopted for the synthetic seismic data and field seismic data.

1) *Training Samples for Synthetic Seismic Data:* Similar to the ground truth in image denoising, the synthetic seismic data $\mathbf{X}_{\text{training}}$ are regarded as a completely clean label. Then, $\mathbf{Y}_{\text{training}}$ can be obtained by $\mathbf{Y}_{\text{training}} = \mathbf{X}_{\text{training}} + \mathbf{v}$, where \mathbf{v} is the AWGN. Note that noise \mathbf{v} in the synthetic seismic data is a stationary signal. Therefore, training-sample cube pairs

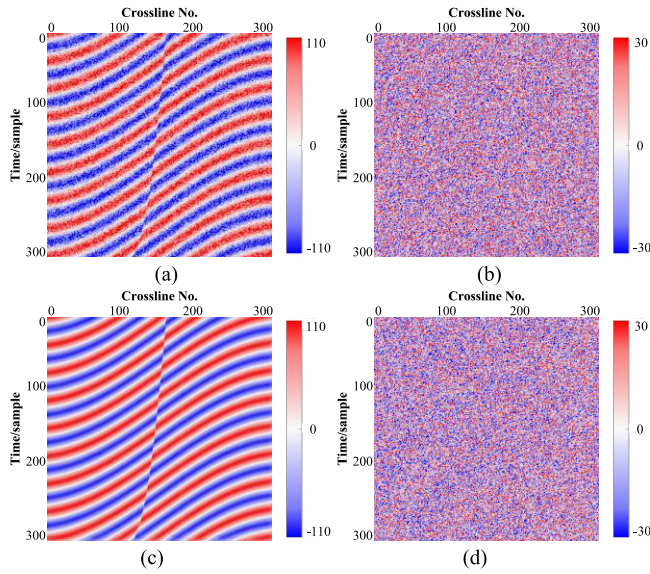


Fig. 7. Noise suppression on inline profile. (a) Noisy data. (b) Noise added to the profile. (c) Denoised by the proposed method. (d) Noise removed by the proposed method.

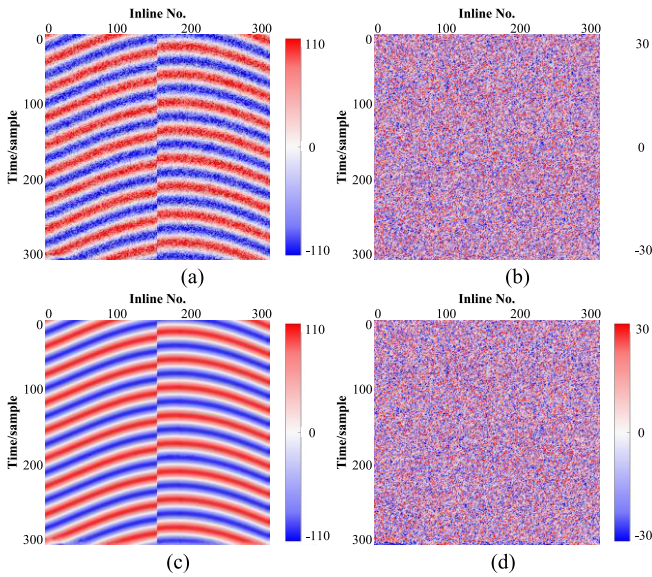


Fig. 8. Noise suppression on crossline profile. (a) Noisy data. (b) Noise added to the profile. (c) Denoised by the proposed method. (d) Noise removed by the proposed method.

in synthetic seismic data can be synchronously and uniformly generated from $\mathbf{Y}_{\text{training}}$ and $\mathbf{X}_{\text{training}}$.

2) *Training Samples for Field Seismic Data:* Unlike synthetic seismic data, field seismic data inevitably receive noise pollution during the exploration and data processing stages and do not have completely clean data as a label. In the absence of label data $\mathbf{X}_{\text{training}}$, which is absolutely clean, finding the cleanest $\mathbf{X}'_{\text{training}}$ is a key step in denoising with network models. With better training samples, the network model can extract useful signals more clearly. Generally, the underground structures in a large survey region have similarities, which means that the structures of the seismic imaging useful signals are similar. Based on this, the useful signal features extracted

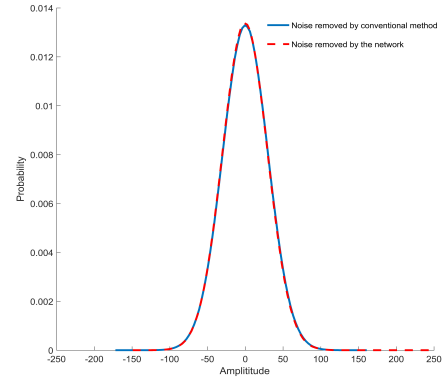


Fig. 9. Probability density distribution of original noise and noise removed by the network.

from a small high-SNR block can be used to infer the useful signals of the rest block of the seismic data. In this way, the powerful denoising ability of the high-SNR block can be extended to all of the seismic data. Therefore, providing better training samples can increase the ability to extract useful signals, thereby can significantly boost the denoising performance of the network.

We propose an effective strategy to select relatively ideal labels from $\mathbf{X}'_{\text{training}}$ and then build noise-clean training-sample cube pairs together from $\mathbf{Y}_{\text{training}}$. This strategy is divided into three steps.

First, we select a relatively high-SNR block in the 3-D seismic data as a training block $\mathbf{Y}_{\text{training}}$, and the rest of the blocks are selected for testing block \mathbf{Y}_{test} . It is generally known that there are many factors affecting the SNR of seismic data during the exploration of field seismic data. For example, some local exploration regions used high-density exploration technology and hence can acquire high-SNR seismic data. In addition, high-SNR seismic data can also be obtained in exploration regions whose surface conditions are conducive to seismic exploration. Taking a loess-covered region as an example, the SNR of the seismic data in a thin loess layer is better than that in a thick loess layer. In combination with these favorable factors, a small block of relatively high SNR from all of the seismic data usually exists. In the experiment involving field seismic data in this study, a small block of high SNR is selected as training block $\mathbf{Y}_{\text{training}}$, as indicated by the red box in Fig. 1(b).

Second, conventional denoising methods that are as advanced as possible are applied to suppress the noise of the training block $\mathbf{Y}_{\text{training}}$ in order to construct the desired training sample pairs. After careful comparison, a three-step denoising workflow proposed by [63], which is a very effective conventional denoising method, is suitable to denoise field seismic data in this study and is used to denoise $\mathbf{Y}_{\text{training}}$. The above-mentioned workflow includes the Cadzow filtering [13] from the inline direction and the crossline direction and edge-preserving filtering [64]. For the sake of simplicity, the above-mentioned workflow is called the conventional method in this article. Thus, $\mathbf{X}'_{\text{training}}$ is obtained. After the first two steps, the SNR of $\mathbf{X}'_{\text{training}}$ is almost satisfactory. However, considering the complexity of the field seismic data, the denoising

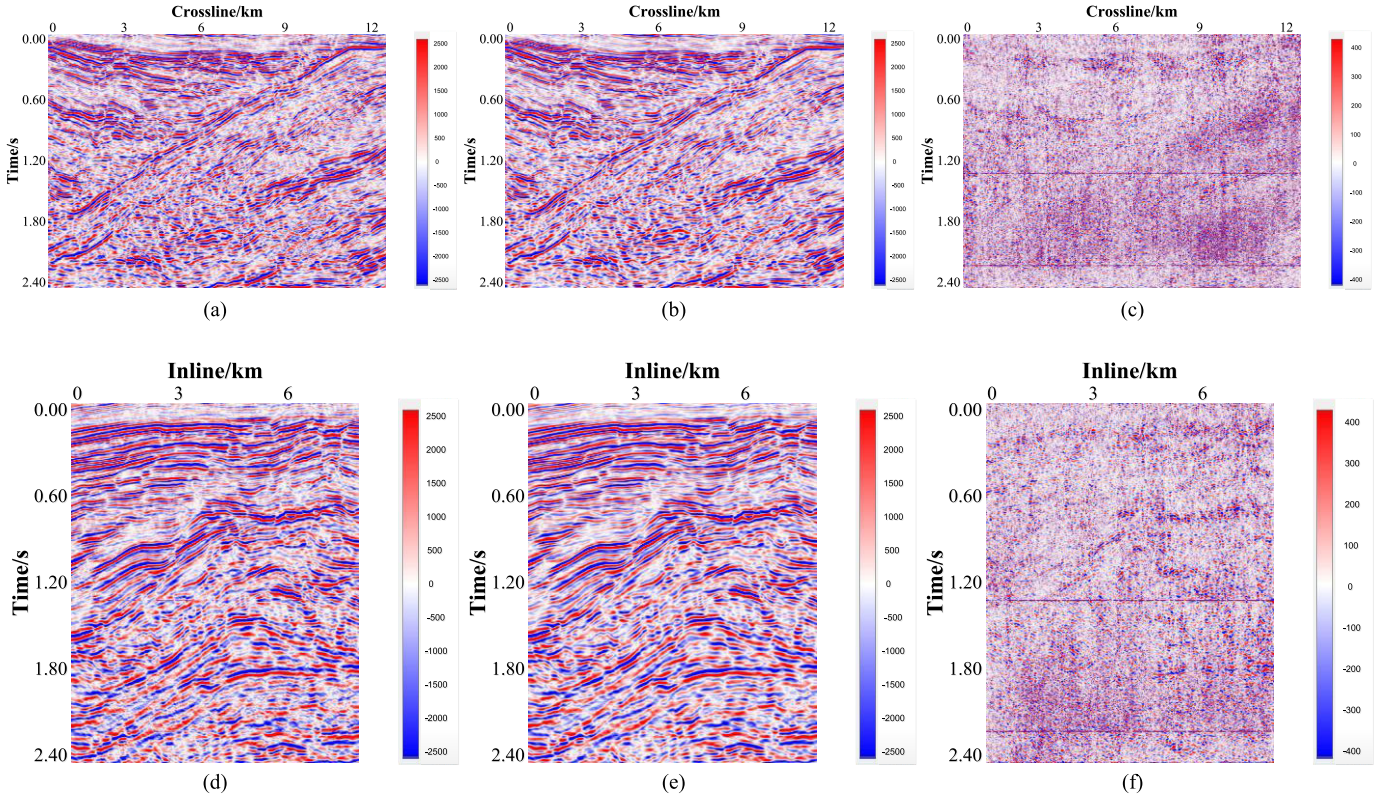


Fig. 10. Training data profiles. (a) Noisy inline profile. (b) Denoised inline profile by the conventional method. (c) Noise removed in the inline profile. (d) Noisy crossline profile. (e) Denoised crossline profile by the conventional method. (f) Noise removed in crossline profile.

performance of some conventional denoising methods will vary with the location and structure of the seismic data. There are still some regions of $\mathbf{X}'_{\text{training}}$ where random noise resides and the continuity of events is affected. Therefore, it is necessary to further select better training samples and remove training samples that have poor denoising performance.

Third, we use faulty confidence obtained by a gradient structure tensor (GST) to select regions with good event continuity from $\mathbf{X}'_{\text{training}}$ to generate the selected training-sample pairs. The GST [65] is a widely used tool for estimating seismic structural and stratigraphic features, such as detecting faults and measuring the continuity of events. For this reason, GST is applied to further select training-sample cube pairs from $\mathbf{Y}_{\text{training}}$ and $\mathbf{X}'_{\text{training}}$. Then, the detailed steps for calculating the faulty confidence and selecting sample pairs are as follows. $\hat{\mathbf{X}}'_{\text{training}}$ is obtained by $\mathbf{X}'_{\text{training}}$ after a Hilbert transform, as follows:

$$\hat{\mathbf{X}}'_{\text{training}} = \mathbf{H}[\mathbf{X}'_{\text{training}}] = \mathbf{X}'_{\text{training}} * \frac{1}{\pi t}. \quad (1)$$

The instantaneous amplitude A and instantaneous phase ψ of $\mathbf{X}'_{\text{training}}$ can be calculated as follows:

$$A = \sqrt{\mathbf{X}'_{\text{training}}^2 + \hat{\mathbf{X}}'^2_{\text{training}}} \quad (2)$$

$$\psi = \arctan \frac{\hat{\mathbf{X}}'_{\text{training}}}{\mathbf{X}'_{\text{training}}}. \quad (3)$$

Neighboring cubes centered at every point in the ψ form a new group of data sets $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N\} \in \mathbb{R}^{w \times w \times b}$, where w indicates the length and weight of the cubes, b

indicates the height of the cubes, and N indicates the total number of voxels in ψ . Then, we calculate the GST for each neighboring cube in \mathbf{Z} according to the following formula:

$$\mathbf{g}_i = \begin{bmatrix} \overline{g_x^2} & \overline{g_x g_y} & \overline{g_x g_z} \\ \overline{g_x g_y} & \overline{g_y^2} & \overline{g_y g_z} \\ \overline{g_x g_z} & \overline{g_y g_z} & \overline{g_z^2} \end{bmatrix} \quad (4)$$

where g_x, g_y, g_z indicate the time direction derivative, crossline direction derivative, and inline direction derivative of \mathbf{z}_i , and \bar{g} indicates the average of all points in \mathbf{z}_i . For each \mathbf{g}_i in the new data set $\mathbf{G} = \{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_N\} \in \mathbb{R}^{3 \times 3}$, three eigenvalues λ_1^i, λ_2^i , and λ_3^i in order of large to small can be obtained through an eigen decomposition. λ_1^i, λ_2^i , and λ_3^i extract the relevant information contained in \mathbf{g}_i . Inspired by [65], the fault confidence C is applied to determine whether the training sample cubes are discarded. C is given by

$$C = \frac{2\lambda_2(\lambda_2 - \lambda_3)}{(\lambda_1 + \lambda_2)(\lambda_2 + \lambda_3)}. \quad (5)$$

This fault confidence measure takes values between 0 and 1. Interval $[0, 1]$ can be divided into two parts by setting the threshold value T . In general, training block $\mathbf{Y}_{\text{training}}$ and corresponding label $\mathbf{X}'_{\text{training}}$ are too large to directly feed to the 3-D-DnCNN. Therefore, $\mathbf{Y}_{\text{training}}$ and $\mathbf{X}'_{\text{training}}$ are correspondingly divided into some cube sets $\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_L\} \in \mathbb{R}^{n \times n \times n}$ and $\{\mathbf{x}'_1, \mathbf{x}'_2, \dots, \mathbf{x}'_L\} \in \mathbb{R}^{n \times n \times n}$ within a sliding 3-D window of $n \times n \times n$ voxels. This study regards C of \mathbf{x}' as a criterion for selecting training-sample pairs from cube pairs $\{(\mathbf{y}_1, \mathbf{x}'_1), (\mathbf{y}_2, \mathbf{x}'_2), \dots, (\mathbf{y}_L, \mathbf{x}'_L)\}$. Cube pairs belonging

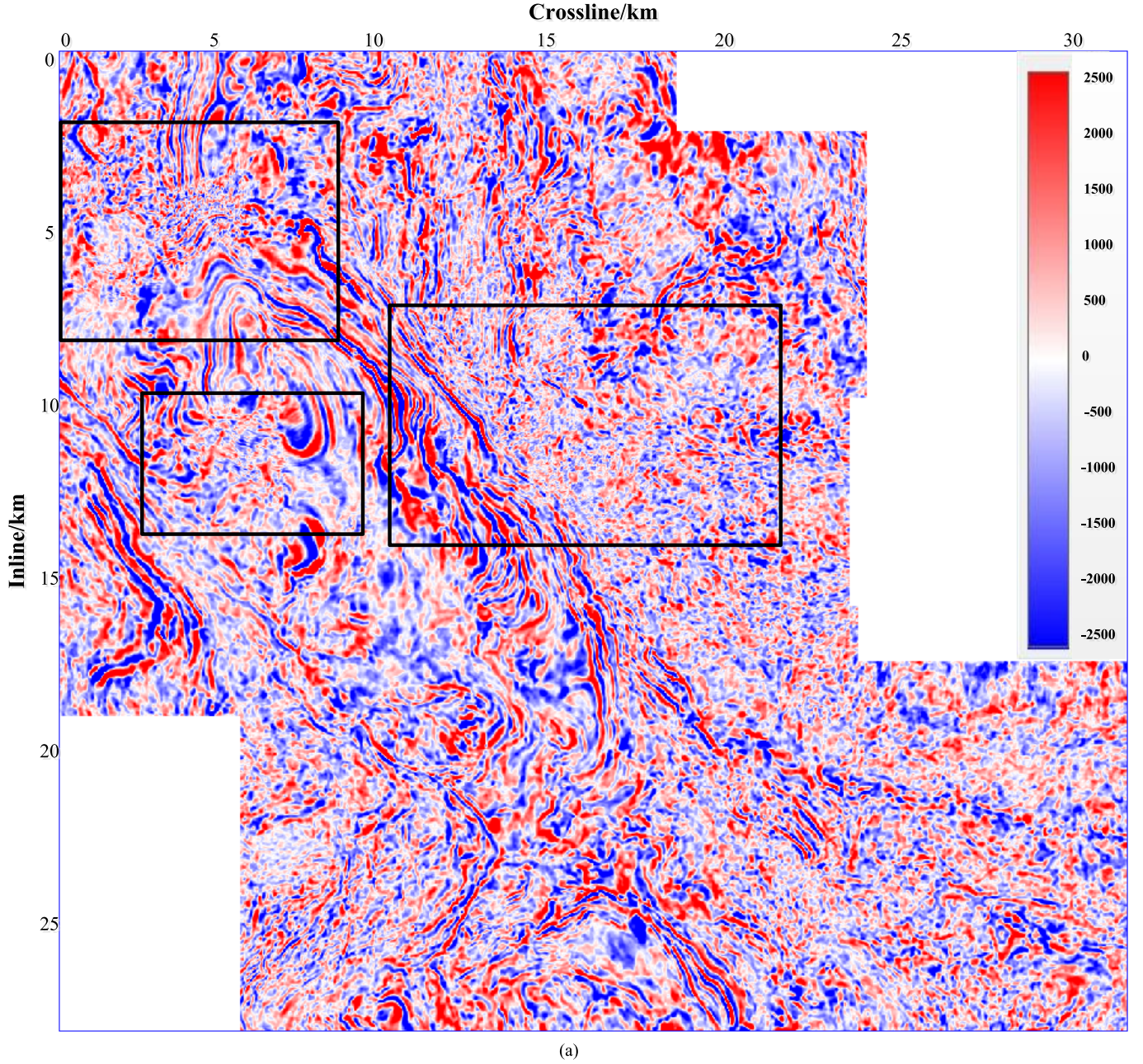


Fig. 11. Time slice denoised by the 3-D-DnCNN network. (a) Time slice of the denoised results based on the conventional method. (b) Time slice of the denoised results based on the 3-D-DnCNN network. (c) Time slice of the denoised results based on the 2-D-DnCNN network (inline cascade crossline). (d) Time slice of the denoised results based on the 2-D-DnCNN network (crossline cascade inline).

to $\{(\mathbf{y}_1, \mathbf{x}'_1), (\mathbf{y}_2, \mathbf{x}'_2), \dots, (\mathbf{y}_L, \mathbf{x}'_L)\}$ with a cube center fault confidence value of $C \in [0, T]$ are reserved, which indicates flat continuous reflectors and fault structures. Meanwhile, cube pairs with a cube center fault confidence value of $C \in [T, 1]$ are dropped, which indicates a random structure and cannot reflect the underground clearly. Following the above-mentioned steps, the relatively ideal training-sample cube pair set $\{(\mathbf{y}_1, \mathbf{x}'_1), (\mathbf{y}_2, \mathbf{x}'_2), \dots, (\mathbf{y}_M, \mathbf{x}'_M)\} (M < L)$ is generated and can be fed to the 3-D-DnCNN for training.

B. Network Architecture

The input of the 3-D-DnCNN is noisy seismic data $\mathbf{y}_i = \mathbf{x}_i + \mathbf{v}_i$, where \mathbf{x}_i is the useful signal and \mathbf{v}_i is the noise that this study aims to suppress. $\{(\mathbf{y}_i, \mathbf{x}_i)\}_{i=1}^M$ represents M noisy-clean

training-sample cube pairs after selection (replacing \mathbf{x}_i with \mathbf{x}'_i for the field seismic data). For the 3-D-DnCNN, we employ the residual learning method to train a residual mapping $\mathfrak{R}(\mathbf{y}_i) \approx \mathbf{v}_i$. Then, we obtain the output $\hat{\mathbf{x}}_i = \mathbf{y}_i - \mathfrak{R}(\mathbf{y}_i)$, which indicates a useful denoised signal. We adopt the averaged mean squared error between the desired denoised seismic data \mathbf{x}_i and estimated ones $\hat{\mathbf{x}}_i$ from the original noisy seismic data as a loss function, as follows:

$$l(\Theta) = \frac{1}{2N} \sum_{i=1}^M \|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_F^2 \quad (6)$$

where Θ indicates the trainable parameters in the 3-D-DnCNN. These are updated through backpropagating the

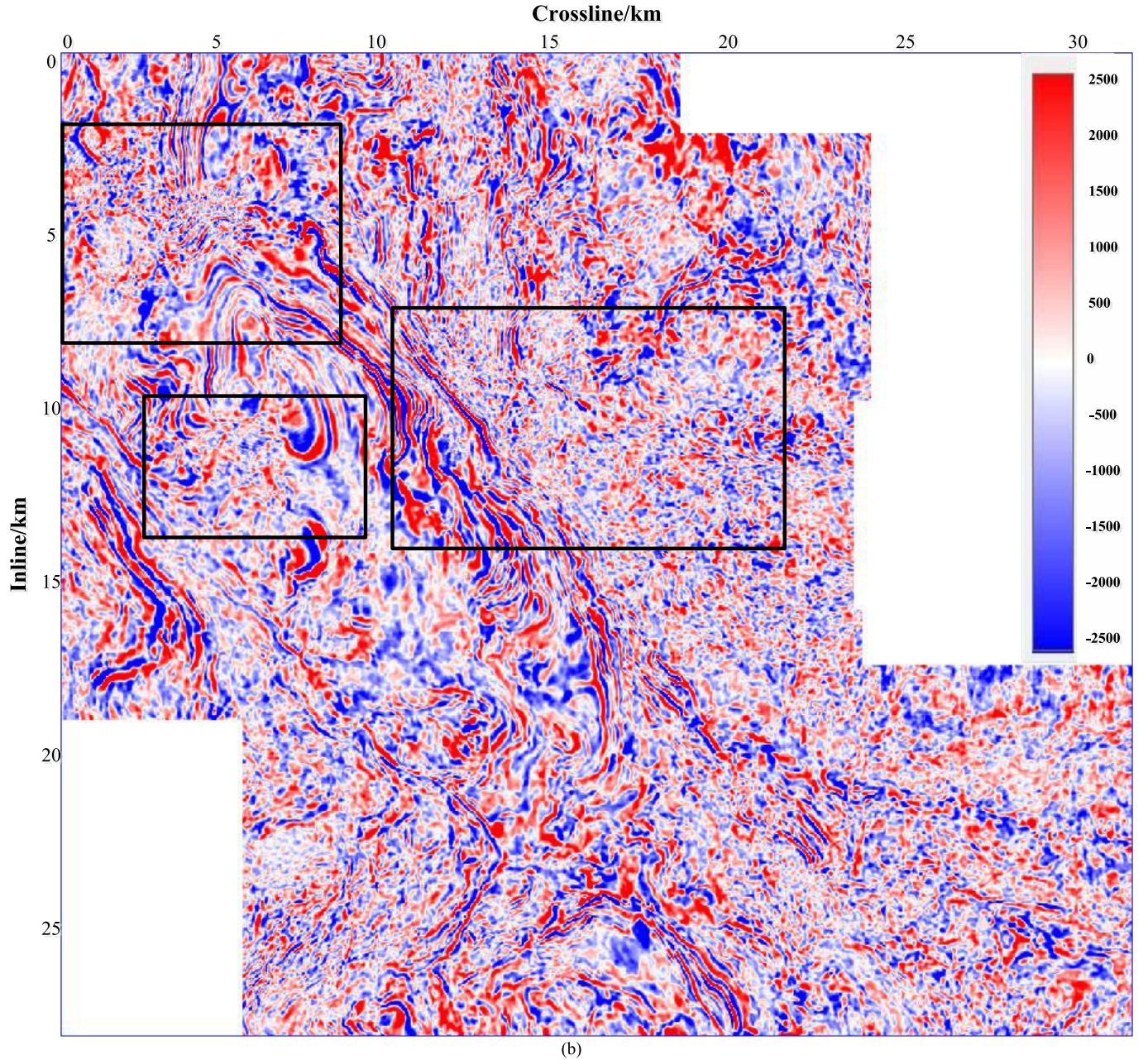


Fig. 11. (Continued.) Time slice denoised by the 3-D-DnCNN network. (a) Time slice of the denoised results based on the conventional method. (b) Time slice of the denoised results based on the 3-D-DnCNN network. (c) Time slice of the denoised results based on the 2-D-DnCNN network (inline cascade crossline). (d) Time slice of the denoised results based on the 2-D-DnCNN network (crossline cascade inline).

gradients of $l(\Theta)$. Equation (6) can be also unfolded into a new form as follows because we employ residual learning:

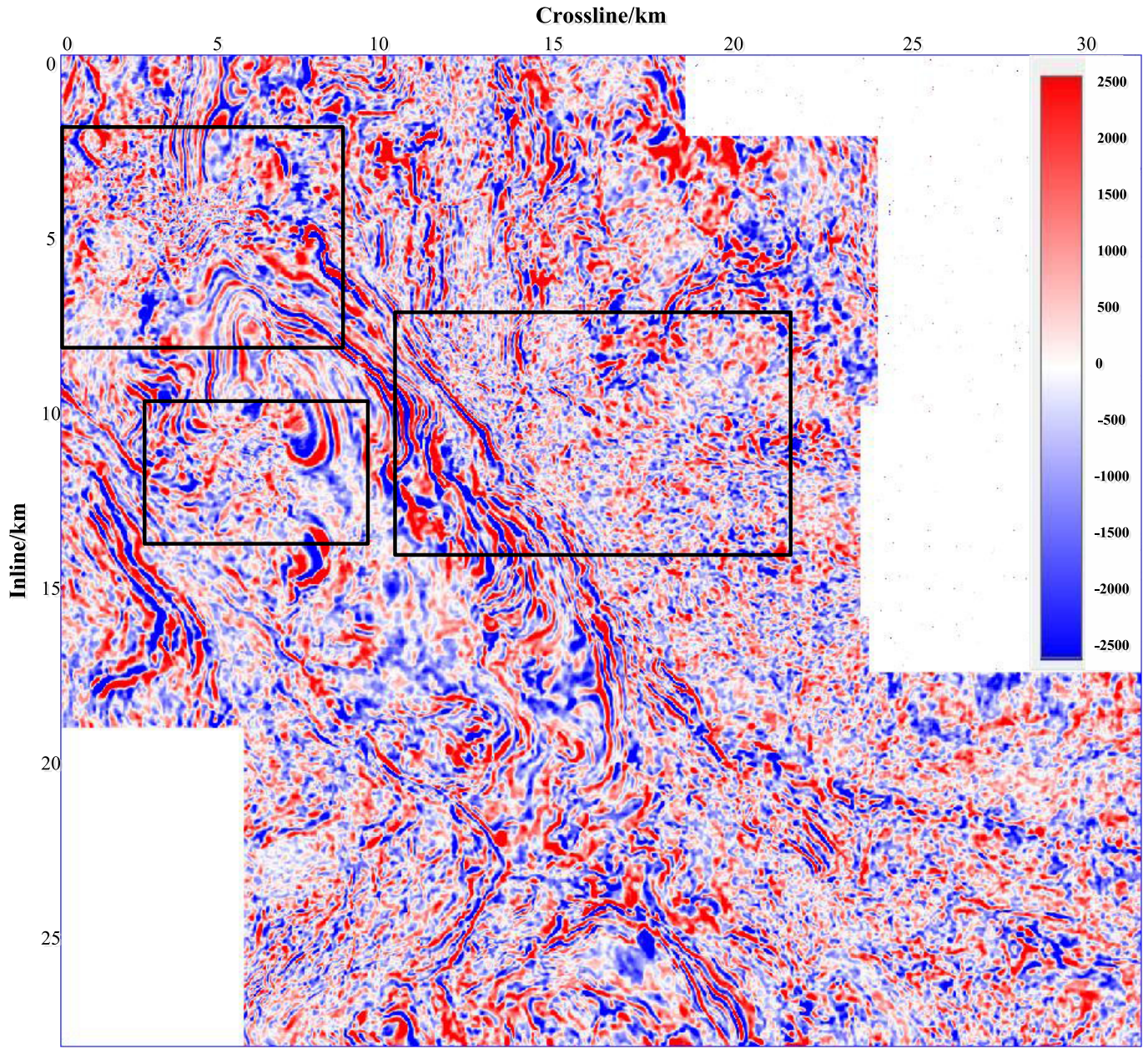
$$l(\Theta) = \frac{1}{2N} \sum_{i=1}^M \|\Re(\mathbf{y}_i; \Theta) - (\mathbf{y}_i - \mathbf{x}_i)\|_F^2 \quad (7)$$

where $l(\Theta)$ measures the averaged mean squared error between the desired residual noise \mathbf{v}_i and the estimated noise $\Re(\mathbf{y}_i)$. Fig. 2 illustrates the architecture of the proposed 3-D-DnCNN. In the following, we elaborate on the architecture of the 3-D-DnCNN and the strategy for reducing the boundary effect.

1) *3-D Architecture*: Recently, many practical applications have benefited from converting 2-D convolution kernels

to 3-D convolution kernels, such as lung nodule detection [66], cerebral microbleeds detection [67], action recognition [68], and remote sensing image classification [69]. Different results for medical image segmentation with 2-D patches, triplanar patches, and 3-D patches are compared in [70] and showed that the 3-D approach performs best at patch classification. This is beneficial because 3-D convolution kernels can capture the full range of spatial variation.

However, there are significant challenges in generalizing 2-D convolutional networks to 3-D. First, the computation cost is a big blocking because 3-D convolutions are more computationally intensive than 2-D convolutions given the same computing



(c)

Fig. 11. (Continued.) Time slice denoised by the 3-D-DnCNN network. (a) Time slice of the denoised results based on the conventional method. (b) Time slice of the denoised results based on the 3-D-DnCNN network. (c) Time slice of the denoised results based on the 2-D-DnCNN network (inline cascade crossline). (d) Time slice of the denoised results based on the 2-D-DnCNN network (crossline cascade inline).

resources. The computational complexity of 3-D volumes has a fatal impact on recognition tasks. However, different from the recognition task, the denoising task can work well with a small receptive field, and our network does not need a fully connected layer that is computationally intensive. To circumvent this, we train our network on smaller 3-D cubes. The accessibility of affordable parallel computing resources via GPUs has made it feasible to train some 3-D networks. Second, training a 3-D network is more difficult than training a 2-D network. The main reason is that there are much more parameters in the 3-D network, which are easy to be overfitting without enough training data, resulting in the

poor generalization ability of the network. This is exactly the reason why 3-D network sometimes loses to 2-D network (for example, relatively small data scale of video data sets [68]). However, seismic data sets are relatively large scale and are sufficient for optimizing the enormous number of parameters in 3-D network. In addition, we adopt some training tricks, such as BN and residual learning, to get over it. The denoising results for the synthetic seismic data and field seismic data in subsequent experiments demonstrated that our method has good denoising generalization ability. Thirdly, 3-D network requires 3-D annotated training data. We use a 3-D denoising conventional method to obtain the 3-D annotated training data

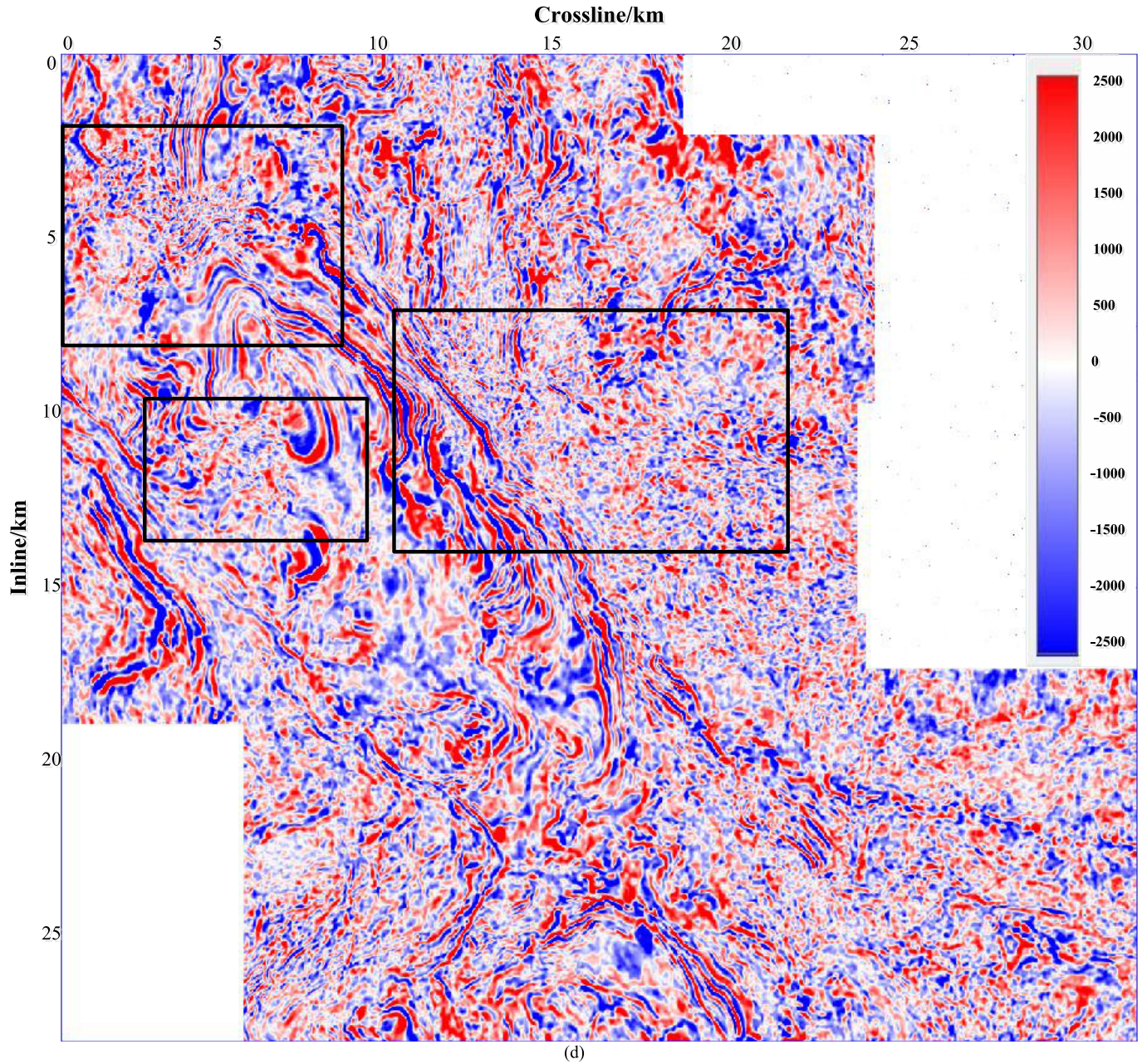


Fig. 11. (Continued.) Time slice denoised by the 3-D-DnCNN network. (a) Time slice of the denoised results based on the conventional method. (b) Time slice of the denoised results based on the 3-D-DnCNN network. (c) Time slice of the denoised results based on the 2-D-DnCNN network (inline cascade crossline). (d) Time slice of the denoised results based on the 2-D-DnCNN network (crossline cascade inline).

and use fault confidence, which is also calculated in 3-D to further select the training data.

Then, we discuss our proposed 3-D network architecture in detail. The 3-D convolutional layers are basic elements of 3-D-DnCNNs with depth D . Inspired by the Visual Geometry Group network (VGG-net) [38], we set the size of the 3-D convolutional kernels to be $3 \times 3 \times 3$ and remove the pooling layers. Incorporating with the rectified linear units [ReLU, $\max(0, x)$] [71] and BN [72], there are three types of layers, as shown in Fig. 2, with three different colors. For the first layer, the input is one cube y_i of size $n \times n \times n$, and the output is $c^{(1)}$ feature cubes of size $n^{(1)} \times n^{(1)} \times n^{(1)}$ generated by $c^{(1)}$ filters of size $3 \times 3 \times 3$. ReLUs are used for nonlinearity.

For layer $k \in [2, D - 1]$, if the $(k + 1)$ th layer has $c^{(k)}$ input feature cubes of size $n^{(k)} \times n^{(k)} \times n^{(k)}$ and a convolutional filter bank that contains $c^{(k+1)}$ convolutional filters of size $3 \times 3 \times 3 \times c^{(k)}$, then this layer generates $c^{(k+1)}$ output feature cubes of size $n^{(k+1)} \times n^{(k+1)} \times n^{(k+1)}$. BN is added between the convolution and ReLU. For the last layer, a convolutional filter of size $3 \times 3 \times 3 \times c^{(D-1)}$ is used to reconstruct the output.

To summarize, our 3-D-DnCNN model has two main features: on the one hand, all convolutional filters are 3-D, which makes the network easier to extract 3-D spatial structural features of the 3-D seismic data volume compared to 2-D convolutional filters. On the other hand, the residual learning

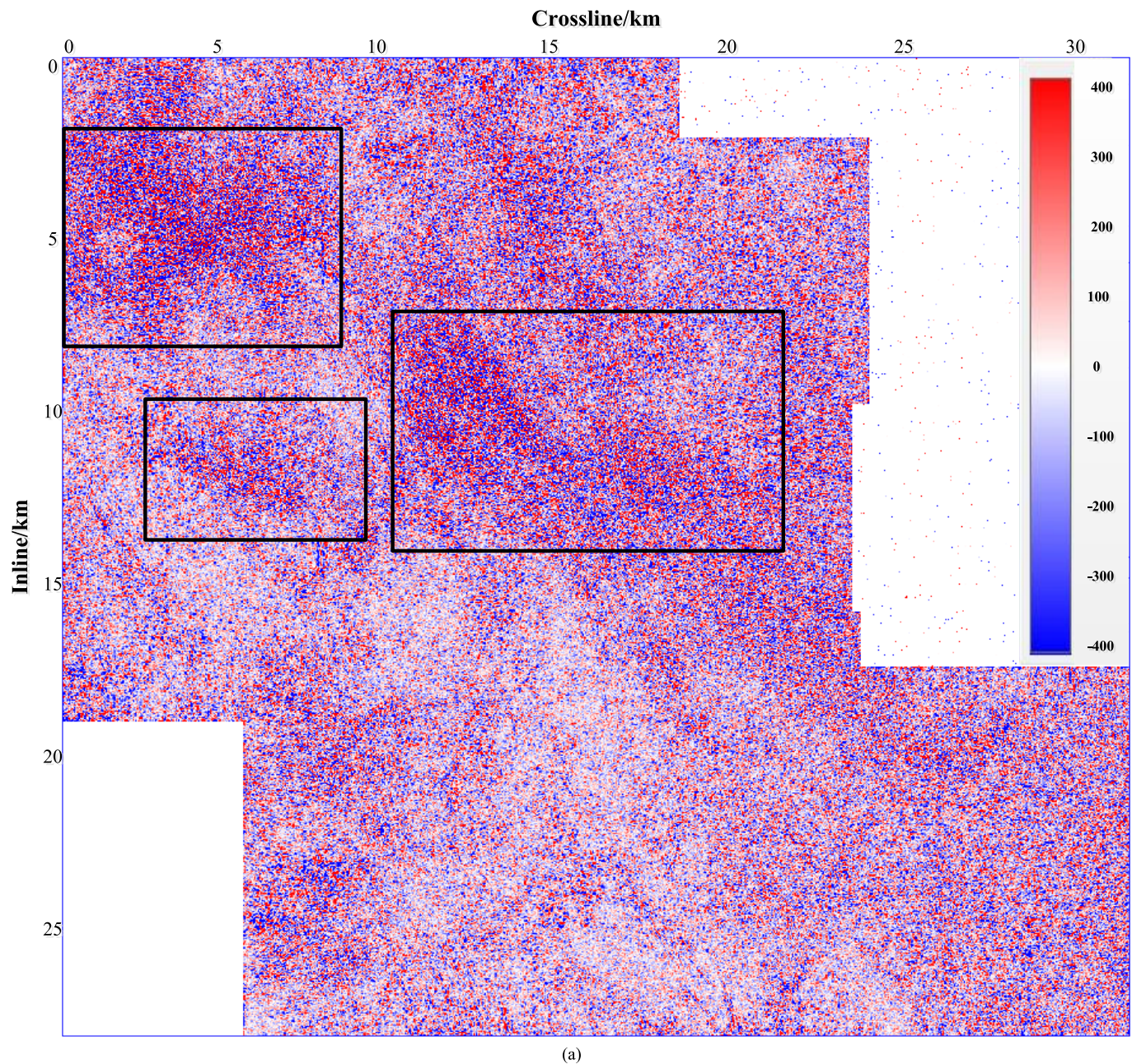


Fig. 12. (a) Noise removed by the conventional method. (b) Noise removed by the 3-D-DnCNN network. (c) Noise removed by the 2-D-DnCNN network (inline cascade crossline). (d) Noise removed by the 2-D-DnCNN network (crossline cascade inline).

method and BN are adopted to accelerate the training process and boost the denoising performance because 3-D networks have more parameters and are easier to overfit. By incorporating convolution with a nonlinearity activation function ReLU, the 3-D-DnCNN can gradually separate a useful signal structure from the noisy seismic data through the hidden layers, as shown in Fig. 2 (top).

2) *Reducing Boundary Artifacts*: In the seismic data processing field, it is a basic requirement that the output signal size should stay the same as the input signal size. However, this may lead to boundary artifacts mainly because the length of the input signal is finite and the number of overlapped denoising windows at the boundary is lower than

that in the middle regions. To overcome the above-mentioned drawback, several methods of extending the input signal have been proposed, such as zero padding, symmetric boundary padding, antisymmetric padding, and periodic padding [73]. Among them, the computational cost of zero padding is the lowest because multiplying zero by any number is always zero. This advantage in computation will increase as the number of feature cubes c and the convolution filter dimensions increase, which causes more multiplication at the boundary. Therefore, we adopt zero padding in each layer to reduce the number of boundary artifacts and to ensure that the size of each feature cube generated by the middle layers is the same as the size of the input signal. By doing this, the sizes of the

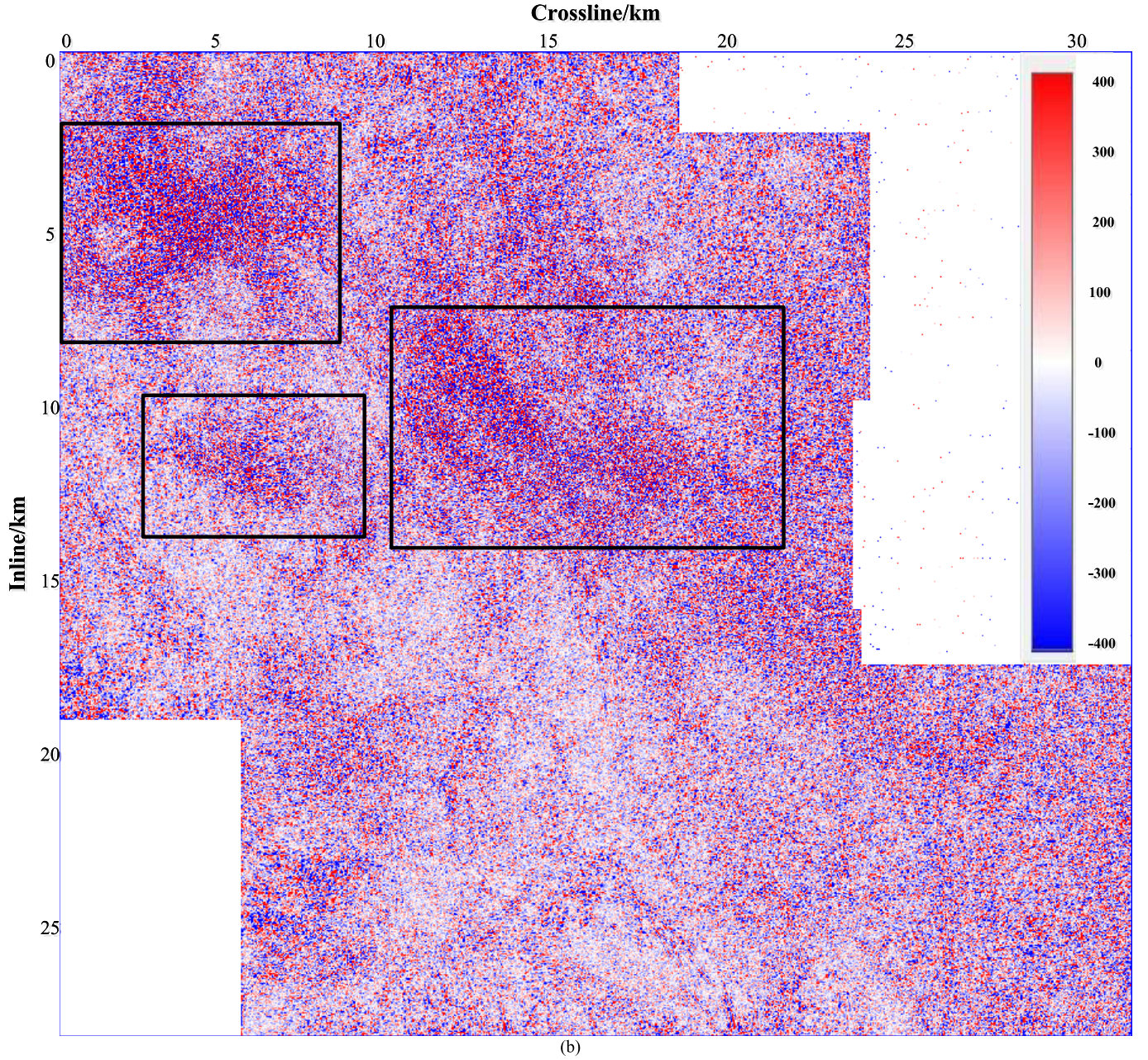


Fig. 12. (Continued.) (a) Noise removed by the conventional method. (b) Noise removed by the 3-D-DnCNN network. (c) Noise removed by the 2-D-DnCNN network (inline cascade crossline). (d) Noise removed by the 2-D-DnCNN network (crossline cascade inline).

feature cubes generated by different layers are equal, i.e., $n^{(1)} = n^{(2)} = \dots = n^{(D)}$. The number of zero paddings at each dimension P can be calculated as follows:

$$P = \frac{(O - 1) \times S + K - W}{2}, \quad (8)$$

where O indicates the edge length of the output cube, W indicates the edge length of the input cube, K indicates the edge length of the convolutional filter, and S indicates the stride of the convolutional filter. Later, both the synthetic seismic data denoising experiment and field seismic data denoising experiment verify that a simple zero-padding strategy can overcome boundary artifacts. This good property is a powerful aspect of the 3-D-DnCNN, which performs better than conventional methods.

C. Integration Batch Normalization and Residual Learning for 3-D Seismic Denoising

BN is proposed for alleviating internal covariate shift problems. Using BN, the distribution of output signals to be processed by the next layer is approximately fixed during training processing, and this speeds up the training. In this study, BN is conducted at every activation function in the 3-D-DnCNN. If the k th convolutional layer has c^k output feature cubes of size $n^{(k)} \times n^{(k)} \times n^{(k)}$, then the j th feature cube output with BN and ReLU can be formulated as

$$[\hat{\mathbf{y}}^k]^j = \frac{[\mathbf{y}^k]^j - E([\mathbf{y}^k]^j)}{\text{Var}([\mathbf{y}^k]^j)} \quad (9)$$

$$[\tilde{\mathbf{y}}^k]^j = [\gamma^k]^j \times [\hat{\mathbf{y}}^k]^j + [\beta^k]^j \quad (10)$$

$$[\mathbf{y}^k]^j = R([\tilde{\mathbf{y}}^k]^j) \quad (11)$$

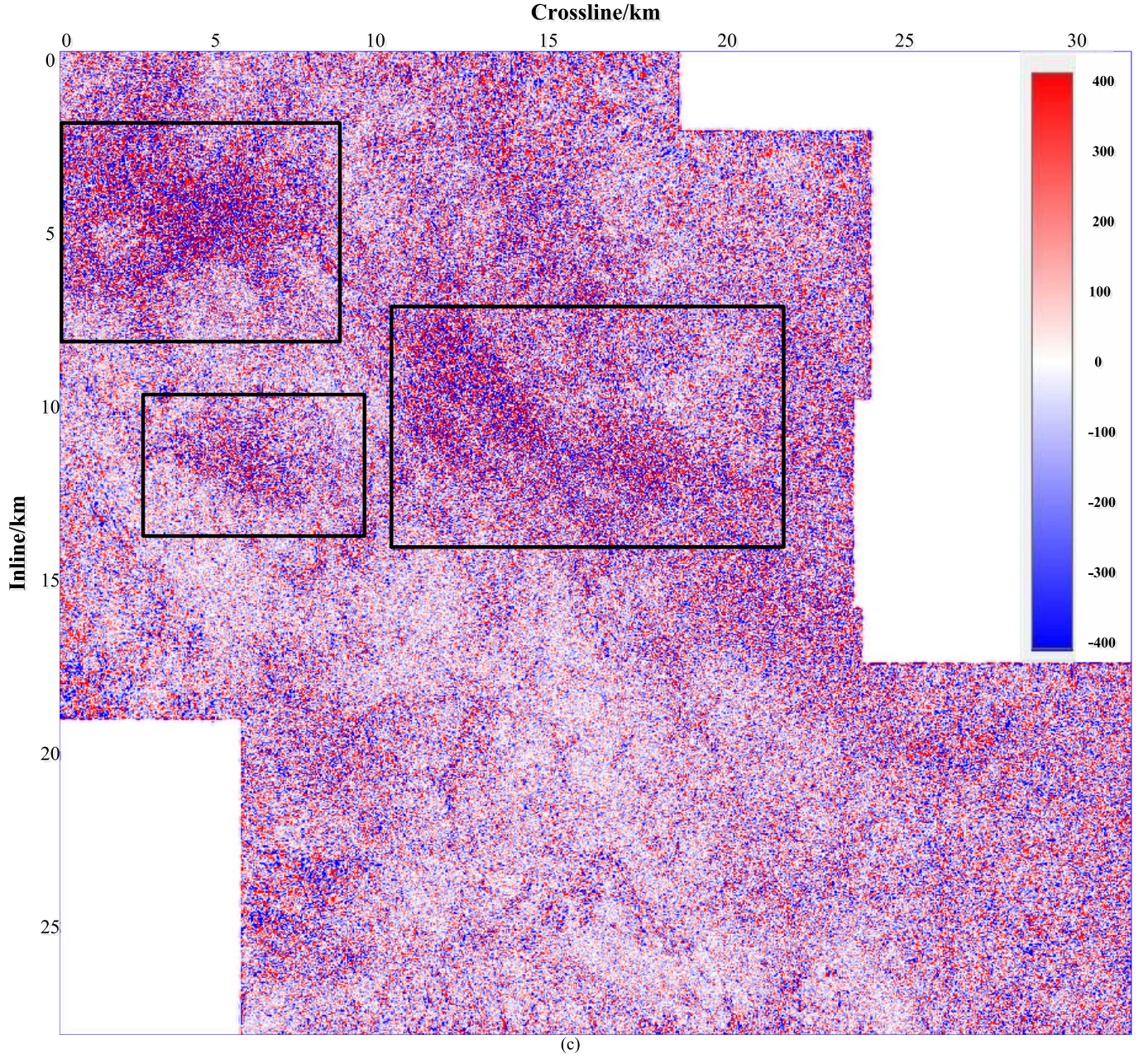


Fig. 12. (Continued.) (a) Noise removed by the conventional method. (b) Noise removed by the 3-D-DnCNN network. (c) Noise removed by the 2-D-DnCNN network (inline cascade crossline). (d) Noise removed by the 2-D-DnCNN network (crossline cascade inline).

where $[\mathbf{y}^k]^j \in \mathbb{R}^{n^{(k)} \times n^{(k)} \times n^{(k)}}$ indicates the output j th feature cube of the k th convolutional layer, $E(\cdot)$ and $\text{Var}(\cdot)$ indicate the expectation and variance computed over the feature cube, respectively, and $R(\cdot)$ indicates the ReLU activation function that sets negative values to zero without changing the positive values. $\hat{\mathbf{y}}^k$ with a mean of zero and variance of 1 is obtained by normalizing \mathbf{y}^k . Using parameters γ and β , which are updated during the training processing, $\tilde{\mathbf{y}}^k$ can be calculated from $\hat{\mathbf{y}}^k$ by a linear transform. Thus, the input of the activation function ReLU is normalized by BN, and the output of ReLU is likely to have a stable distribution.

Residual learning [39] is proposed to solve degradation problems, i.e., stacking more layers will degrade accuracy. In general, there are two forms to asymptotically approximate the

desired clean label when using CNNs for denoising seismic data. One is building the original mapping $\Phi(\mathbf{y})$ to predict \mathbf{x} . The other is building residual mapping $\mathfrak{R}(\mathbf{y})$ to predict \mathbf{v} , which is also called residual learning. Residual learning has a faster convergence speed when the first form is more like an identity mapping. This is because learning the residual mapping with reference to the inputs is much easier than learning unreferenced mapping, which learns the label as a new one if the label is closer to the inputs than to zero. Note that the amplitude of the useful signal in seismic data is almost ten times the amplitude of the noise, and $\Phi(\mathbf{y})$ is much closer to an identity mapping than $\mathfrak{R}(\mathbf{y})$. Therefore, we adopt residual learning to solve the degradation problem and to accelerate the training processing in the 3-D-DnCNN by adding a single shortcut connection between the input and output.

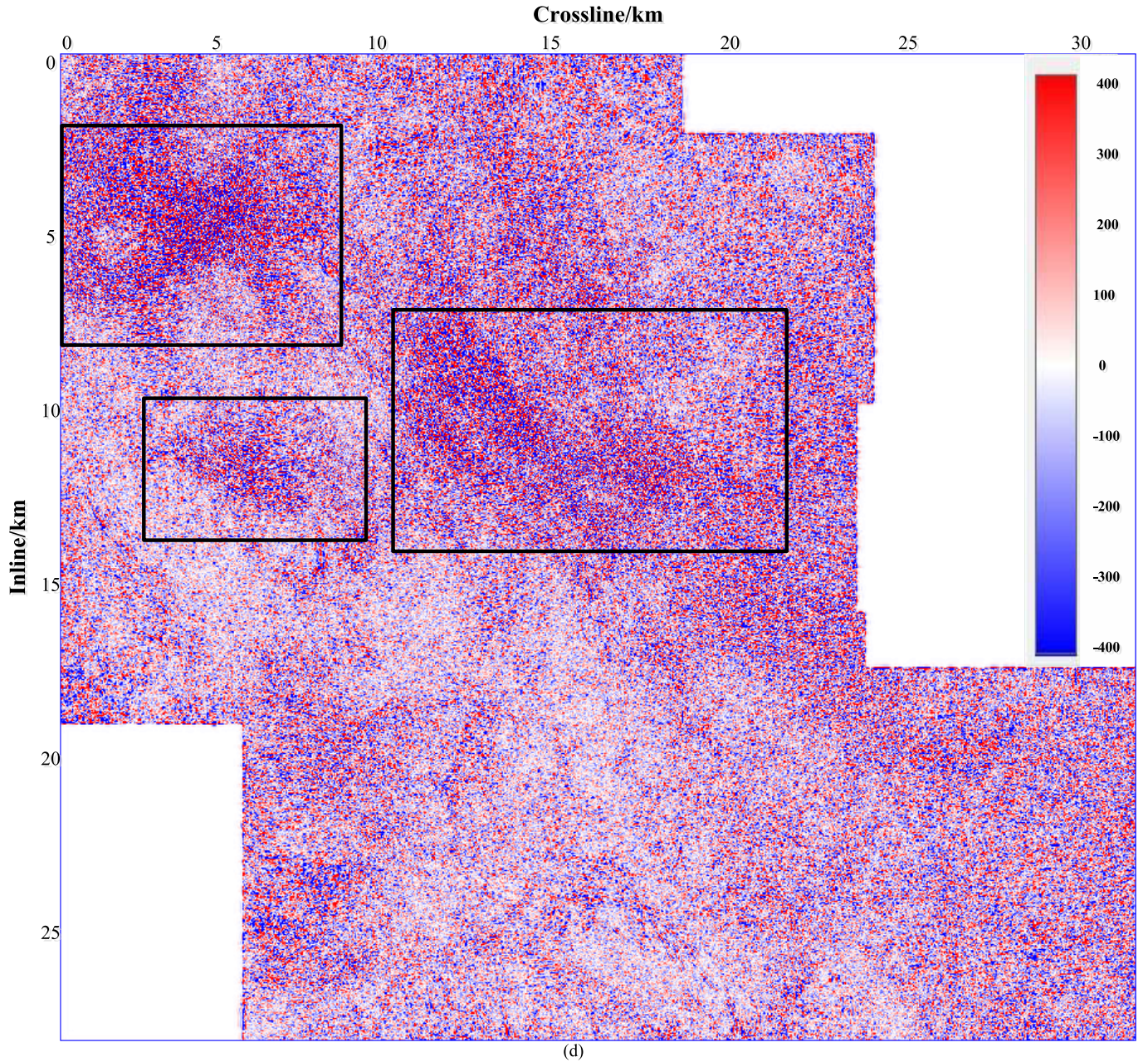


Fig. 12. (Continued.) (a) Noise removed by the conventional method. (b) Noise removed by the 3-D-DnCNN network. (c) Noise removed by the 2-D-DnCNN network (inline cascade crossline). (d) Noise removed by the 2-D-DnCNN network (crossline cascade inline).

III. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, the performance of the proposed 3-D-DnCNN is evaluated with synthetic seismic data and field seismic data. The field seismic data are poststack seismic data from an area of approximately 700 km² in Daqing Oilfield. The shallow-layer structures and deep-layer structures of the field seismic data in this study are basically very similar in the whole work area. One time slice of the field seismic data is shown in Fig. 3. All experiments are implemented in Python and Tensorflow. Tensorflow using data flow graphs is the most popular deep learning framework. All experiments are implemented on Ubuntu 14.04, 48-core Intel E5-2650 CPUs with 128 GB of memory and an Nvidia Quadro M4000 GPU.

A. Experimental Setting

1) *Training and Testing Data:* For Gaussian denoising with a known specific noise level in synthetic seismic data with a size of $300 \times 300 \times 300$, we select a block of size $180 \times 180 \times 180$ as a training block. This is indicated by the red cube in Fig. 1. We found that changing the position of the training block causes only a slight improvement in the denoising performance. We select the rest block of synthetic seismic data volume as the testing block. Following the principle in [58], we set the receptive field size of the 3-D-DnCNN to $35 \times 35 \times 35$ with a corresponding depth of 17. We set the size of the training samples generated from the training block to $40 \times 40 \times 40$ and crop 6×10000 cubic patches to train

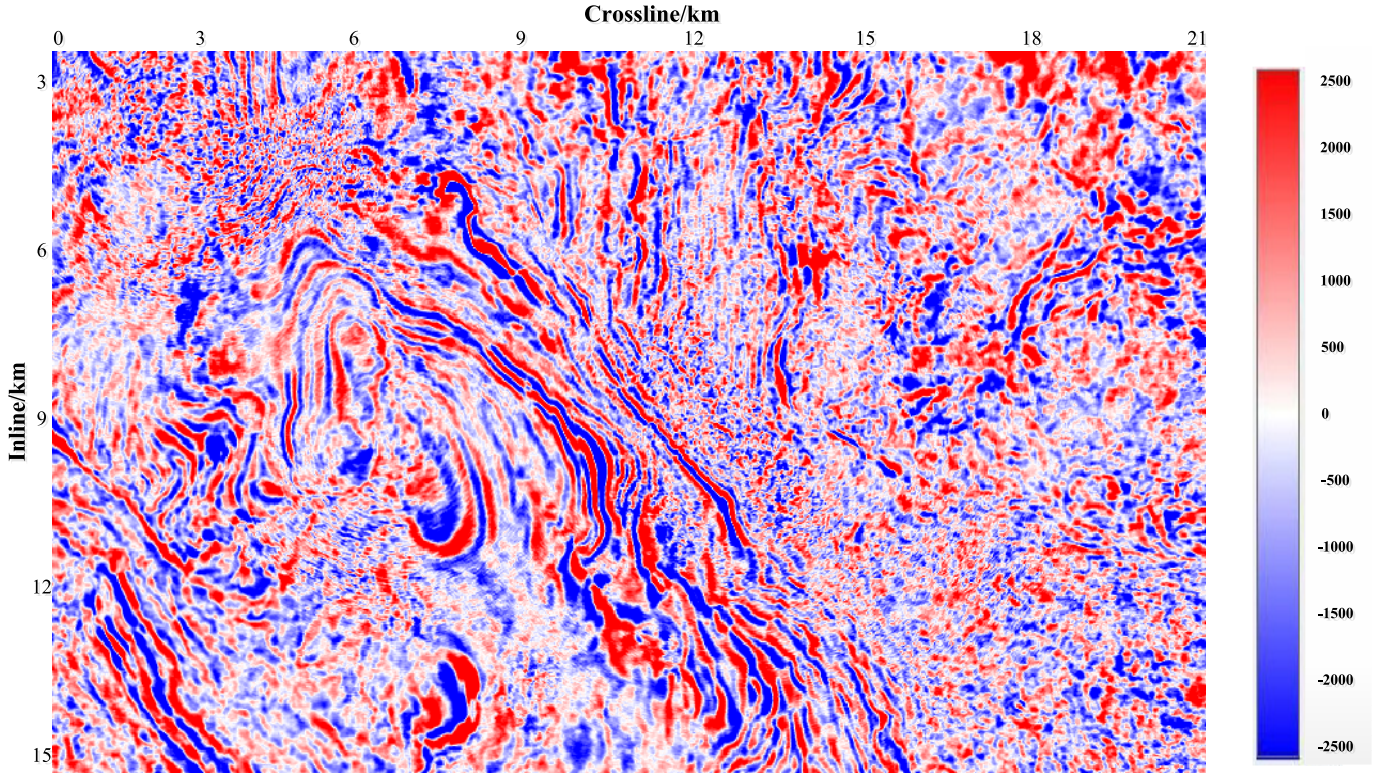


Fig. 13. Magnify of the black box in a time slice.

the model. We consider an arbitrary noise level, i.e., $\sigma = 30$. It can be seen that the random noise has obvious interference with the useful signals in Figs. 6(a), 7(a), and 8(a). Due to memory size limitations, we set the size of the testing cubes to $180 \times 180 \times 180$. Adjacent testing cubes do not overlap or intersect, and cubes at the boundary are padded by zero when generated from the testing block.

For field seismic data denoising, we first set the parameters of the training-sample selection method. In order to calculate fault confidence C , we set the size of \mathbf{z} to $5 \times 5 \times 25$. Another important parameter of the sample selection is the fault confidence threshold T , which is used to further select the training sample pairs. Random noise in the field seismic signals of this study also contains arc-like imaging noise that was generated during the process of seismic imaging. We want to use T to filter out sample cubes with arc-like imaging noise while preserving the sample cubes with faults and continuous events as much as possible. We conducted an experiment to find the proper fault confidence threshold T .

The experiment was conducted on a section of field seismic data that was full of arc-like imaging noise and also contained faults. These are indicated by the red box and yellow arrows, respectively, in Fig. 4(a). As shown in Fig. 4(c)–(j), the regions containing arc-like imaging noise have a higher C and can successfully be dropped by thresholding C below T . Meanwhile, the regions containing faults have a medium C and are successfully reserved with a well-chosen threshold T . Here, we made a compromise between filtering out arc-like imaging noise regions and reserving fault regions. We eventually set T to 0.65. Second, we selected approximately 66 km^2 of

coverage for training block $\mathbf{Y}_{\text{training}}$ with a relatively high SNR from field seismic data, as indicated by the red box in Fig. 3. The whole work area is a combination of many small work areas. Except for one small work area, the number of stacking fold is 150, and the others are 84 stacks. Our training block $\mathbf{Y}_{\text{training}}$ is selected from the 150-stack area, which has a relatively high SNR. The rest areas beyond the training block $\mathbf{Y}_{\text{training}}$ are selected as the testing block \mathbf{Y}_{test} . Third, we applied the state-of-the-art conventional method with cascading Cadzow filtering and edge-preserving filtering to denoise the seismic data in the training block in order to obtain $\mathbf{X}'_{\text{training}}$. Finally, we generated training-sample cube pairs $\{(\mathbf{y}_i, \mathbf{x}'_i)\}_{i=1}^M$ with size $40 \times 40 \times 40$ and stride of 10 from $\mathbf{X}'_{\text{training}}$ using fault confidence C . Testing cubes were generated in the same way as those of the synthetic data. After denoising, we restored the testing block by stitching the testing cubes according to the original order.

2) *Network Training*: We adopted the loss function in (7) to predict the residual \mathbf{v} by learning the residual mapping $\mathfrak{R}(\mathbf{y}; \Theta)$. The weight Θ was initialized by the method in [74] and updated by minibatch stochastic gradient descent as follows:

$$\Theta' = \Theta - \frac{\alpha}{m} \sum_{i=1}^m \frac{\partial l(\mathbf{y}_i, \Theta)}{\partial \Theta} \quad (12)$$

where α indicates the learning rate and m indicates the minibatch size. α is decayed from the initial value $1e-3$ to $1e-4$ after five epochs. We set m to 6 and trained 20 epochs for our 3-D-DnCNN model. The training settings

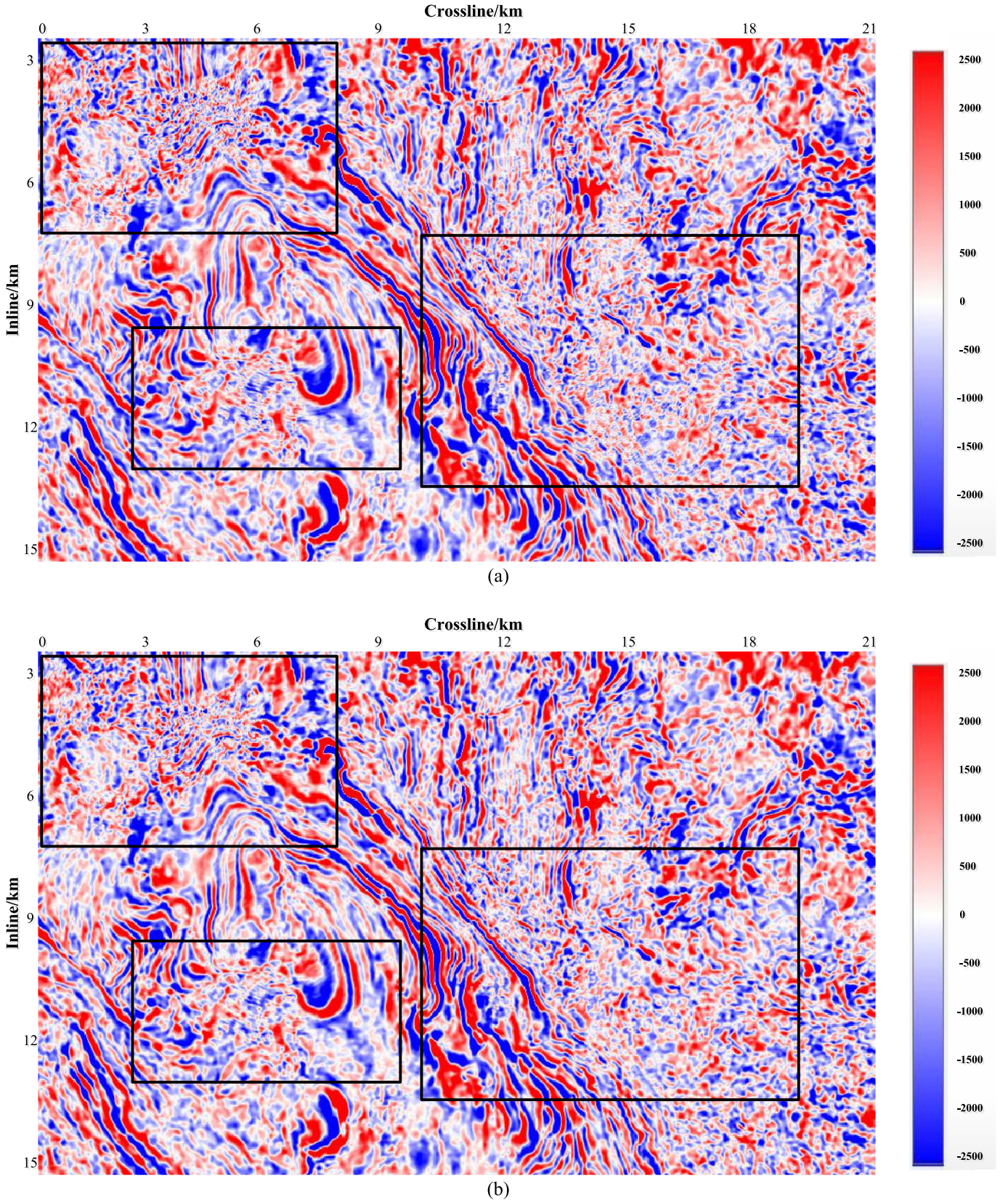


Fig. 14. Denoising results of the magnified part in the time slice. (a) Denoised by the conventional method. (b) Denoised by the 3-D-DnCNN network. (c) Denoised by the 2-D-DnCNN network (inline cascade crossline). (d) Denoised by the 2-D-DnCNN network (crossline cascade inline).

of the synthetic seismic data and the field seismic data were the same for verifying the generalization of the 3-D-DnCNN.

B. Denoising Performance of Synthetic Seismic Data

Fig. 5 illustrates two vertical profiles and one time slice of the original noisy synthetic seismic data. The positions

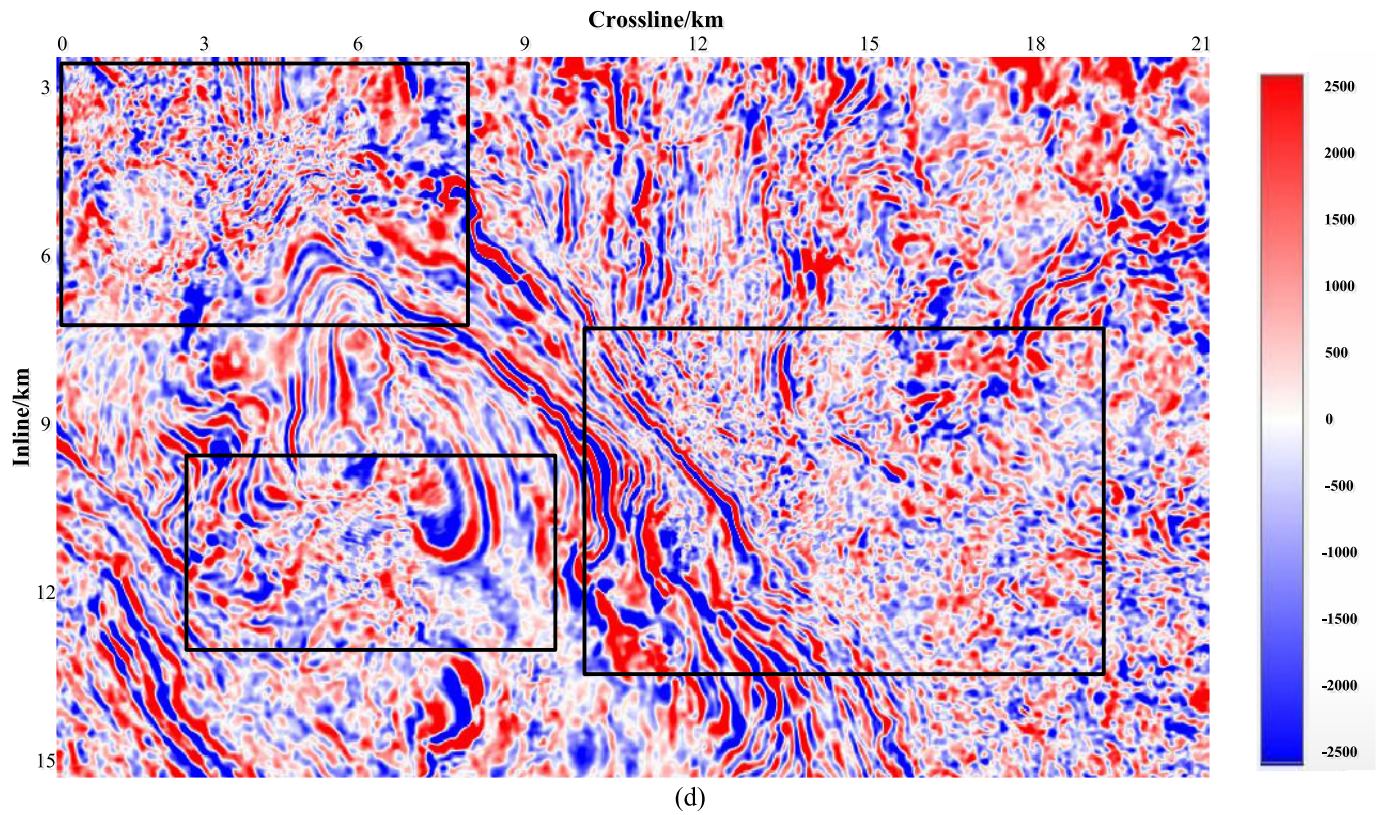
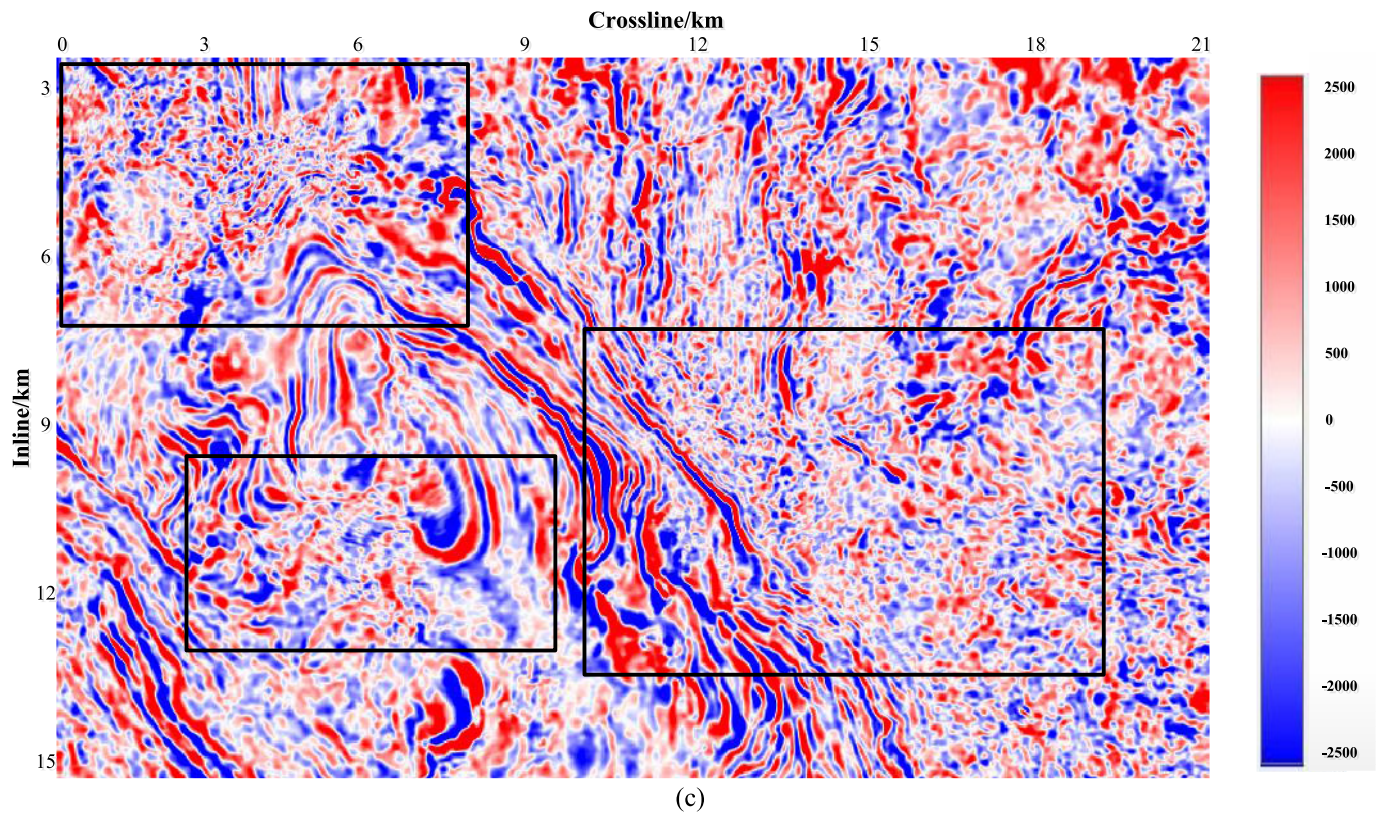


Fig. 14. (Continued.) Denoising results of the magnified part in the time slice. (a) Denoised by the conventional method. (b) Denoised by the 3-D-DnCNN network. (c) Denoised by the 2-D-DnCNN network (inline cascade crossline). (d) Denoised by the 2-D-DnCNN network (crossline cascade inline).

of the vertical profiles and time slice shown in Fig. 5 are at the 100th sampling point in the direction of the inline, crossline, and time. It can be seen that there is an obliquely curved fault structure in the synthetic seismic data. The cubic

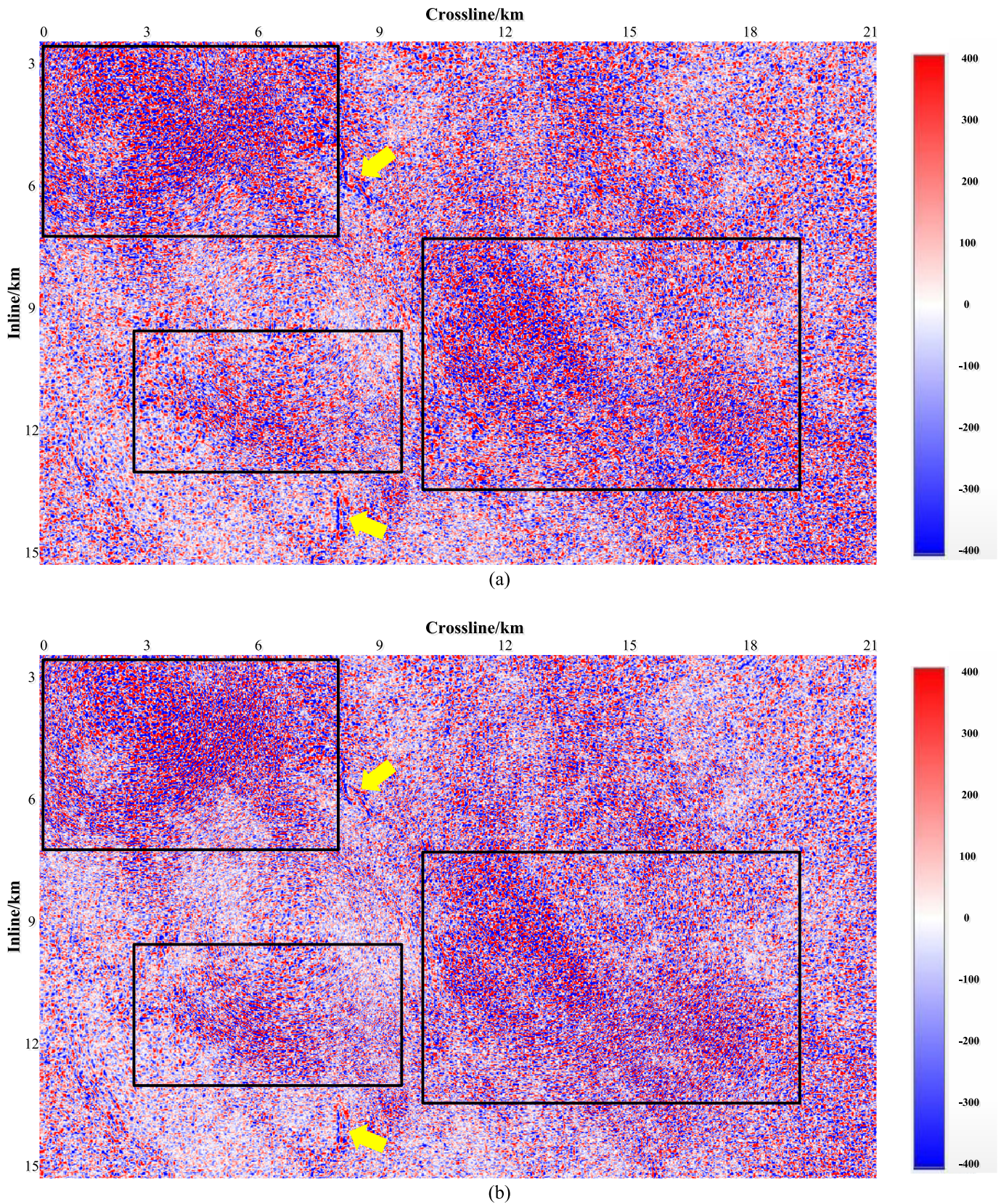


Fig. 15. Removed noise of the magnified part in the time slice. (a) Noise removed by the conventional method. (b) Noise removed by the 3-D-DnCNN network. (c) Noise removed by the 2-D-DnCNN network (inline cascade crossline). (d) Noise removed by the 2-D-DnCNN network (crossline cascade inline).

patches presented in Section III-A are fed to the 3-D-DnCNN network. After approximately 80 h of training with a single

GPU, the loss function of the 3-D-DnCNN is stable, which means that the network training has been completed. Then,

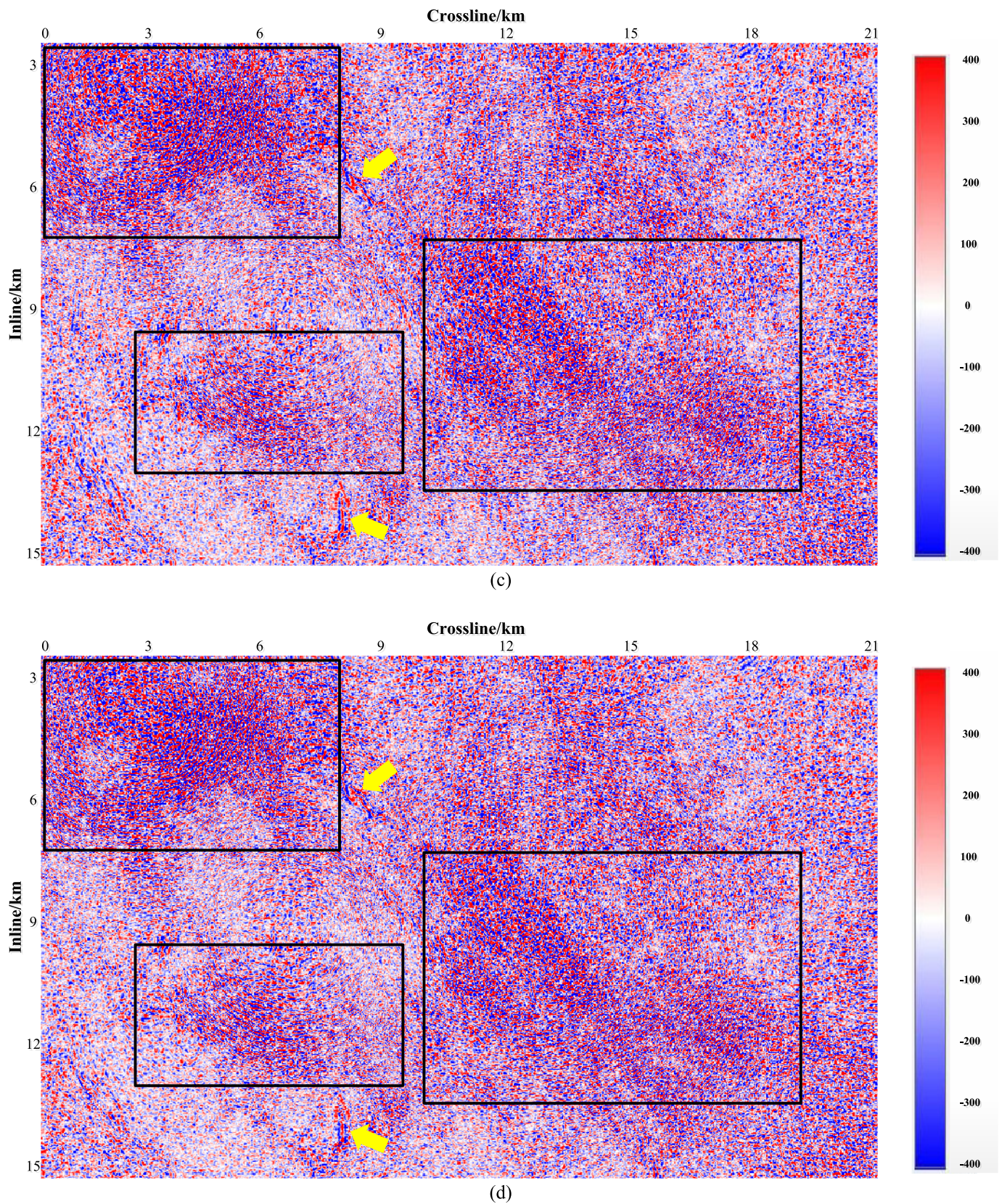


Fig. 15. (Continued.) Removed noise of the magnified part in the time slice. (a) Noise removed by the conventional method. (b) Noise removed by the 3-D-DnCNN network. (c) Noise removed by the 2-D-DnCNN network (inline cascade crossline). (d) Noise removed by the 2-D-DnCNN network (crossline cascade inline).

we use the well-trained network to suppress the random noise of all synthetic data.

The SNR results for our proposed method on the synthetic seismic data are listed in Table I. As one can see, the SNR

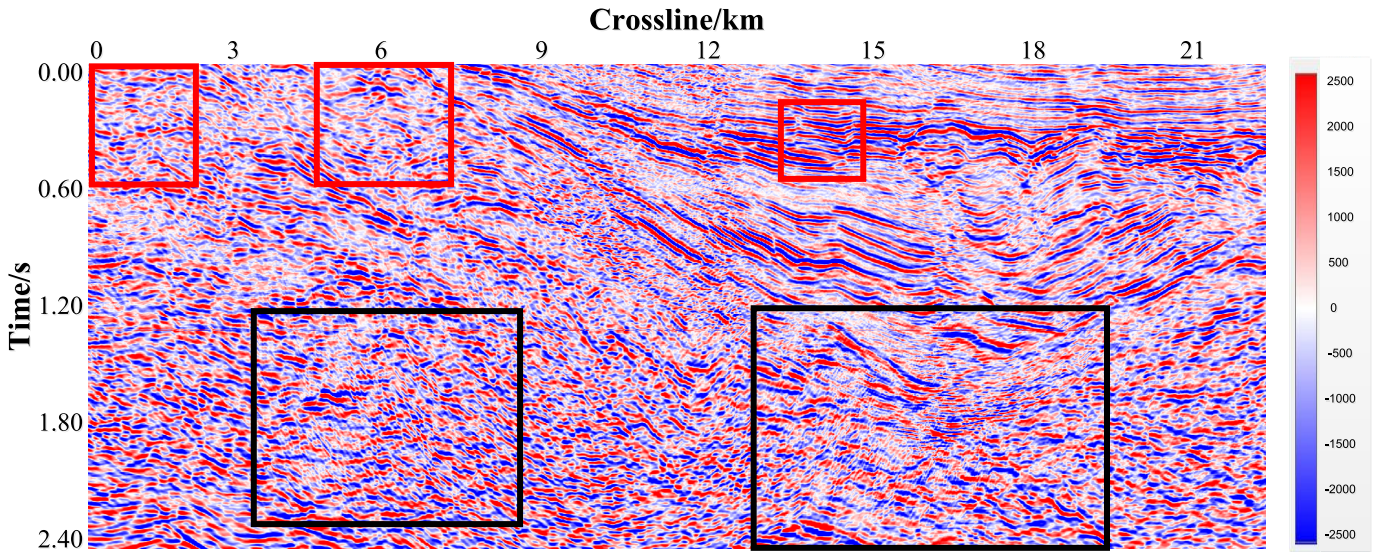


Fig. 16. Noisy inline profile.

TABLE I
DENOISING RESULTS OF 3-D-DnCNN

| SNR of noisy synthetic seismic data | SNR of denoised synthetic seismic data | SNR of denoised training block in synthetic seismic data | SNR of denoised testing block in synthetic seismic data |
|-------------------------------------|--|--|---|
| 7.45 dB | 27.93 dB | 34.00 dB | 25.03 dB |

is greatly improved after suppressing random noise using the 3-D-DnCNN network. Figs. 6–8 show the results of suppressing random noise on the time slice and vertical profiles. Compared with the original noisy synthetic data in Fig. 5, we find that the 3-D-DnCNN network has successfully suppressed most of the random noise, and the continuity and consistency of the reflection events have been significantly improved, while the fault structure has been effectively maintained. To further illustrate the feasibility and effectiveness of the proposed method, a statistical analysis is performed on the random noise removed by the network. Fig. 9 shows the probability density distribution of the original noise and noise removed by the network. It can be seen that both the original noise and the noise removed by the network are significant Gaussian distributions. The variance and kurtosis values of the original noise and the noise removed by the network in Table II are close, and the kurtosis values of approximately zero also reflect the Gaussian distribution characteristics when the noise is removed. To sum up, by learning the training samples constructed by the training block of the synthetic seismic data, the 3-D-DnCNN network successfully extracted the features of the random noise in the synthetic seismic data and used the extracted features to suppress the random noise in all of the seismic data. Therefore, it is feasible to suppress the random noise of the Gaussian distributions in seismic data by using a 3-D-DnCNN. More importantly, with good training samples for training, the noise removed by a well-trained 3-D-DnCNN does not contain an obvious effective signal structure. That is to say, deep learning automatically learns the ability to remove

noise from samples, and its denoising ability has strong fidelity to the fault regions in the seismic data.

C. Denoising Performance of Field Seismic Data

The training-sample cube pairs generated by the method in Section II-A were fed to the 3-D-DnCNN. After approximately 170 h of iterative training with a single GPU, the network training was completed. As shown in Fig. 10, we randomly selected two profiles from the inline and crossline directions to show our training data according to the positions of the two blue lines in Fig. 3. Note that the training time for the field seismic data was longer than that of the synthetic data because of the larger number of training samples within an epoch. We used the well-trained 3-D-DnCNN network to suppress random noise in all of the field seismic data. The noise suppression time was approximately 90 min with a single GPU. The suppression time was far below the 24 h required for the conventional suppression method with a single CPU node, which was used to generate the training label $\mathbf{X}'_{\text{training}}$. The denoising results of our proposed methods and the specific state-of-the-art conventional method used to generate $\mathbf{X}'_{\text{training}}$ are shown from different directions.

In addition, most existing deep learning methods for 3-D volume denoising typically convert 3-D data into 2-D slices or 2-D multichannel representations [75] and then training in 2-D CNNs. While these architectures of 2-D networks might be successful in some problems, they are suboptimal in their use of available 3-D information. To demonstrate it, we compare

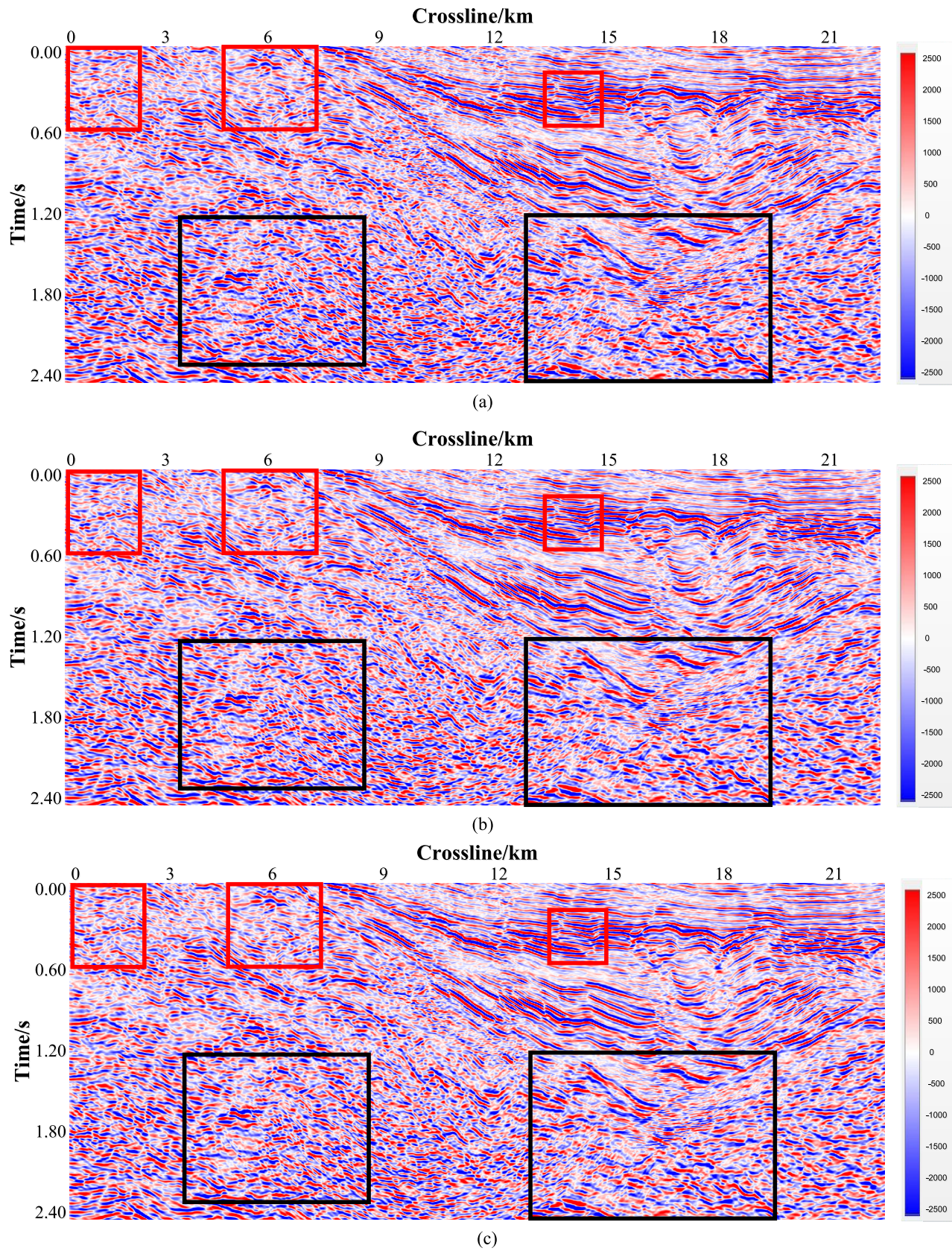


Fig. 17. Denoising results of inline profile. (a) Denoised by the conventional method. (b) Denoised by the 3-D-DnCNN network. (c) Denoised by the 2-D-DnCNN network (inline cascade crossline). (d) Denoised by the 2-D-DnCNN network (crossline cascade inline).

the denoising performance in a 2-D network and 3-D network for field seismic data, given similar architecture and

same training sample construction method. For 2-D networks, we use 2-D-by-2-D learning that is usually implemented in

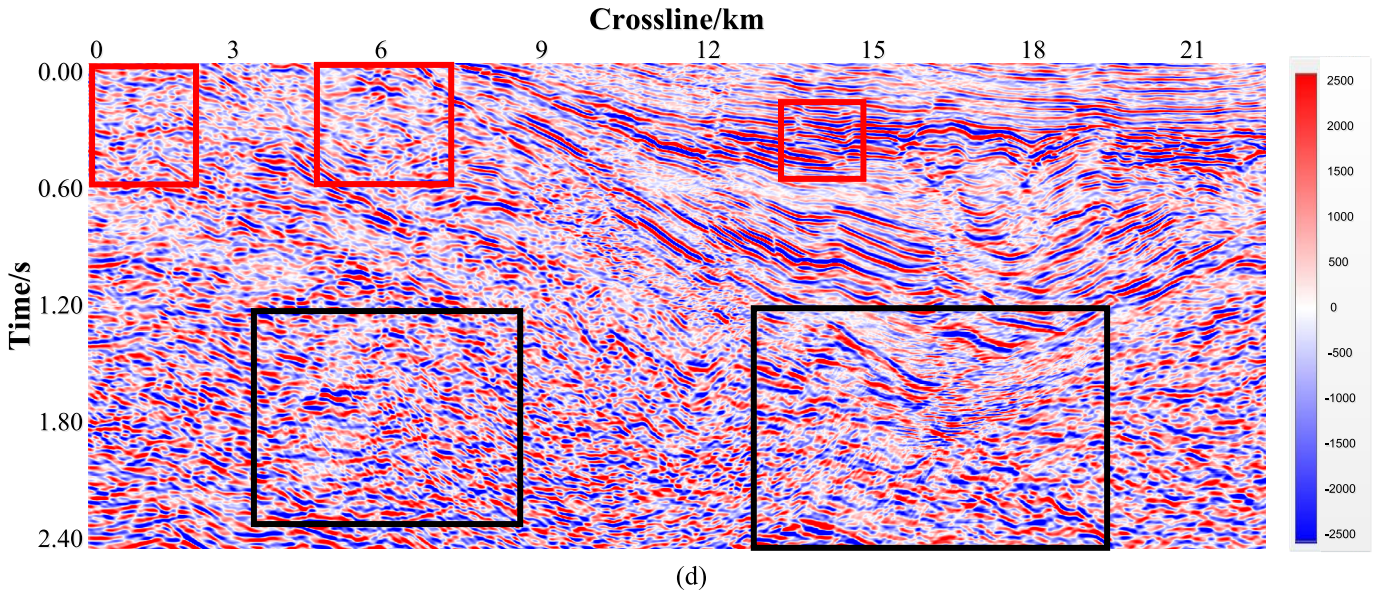


Fig. 17. (Continued.) Denoising results of inline profile. (a) Denoised by the conventional method. (b) Denoised by the 3-D-DnCNN network. (c) Denoised by the 2-D-DnCNN network (inline cascade crossline). (d) Denoised by the 2-D-DnCNN network (crossline cascade inline).

TABLE II
STATISTICAL CHARACTERISTICS OF NOISE FROM SYNTHETIC SEISMIC DATA

| Noise type | Average value | variance | kurtosis |
|---------------------------------------|---------------|----------|----------|
| Noise added to synthetic seismic data | -0.0008 | 900.1 | 0.0002 |
| Noise removed by 3D-DnCNN network | -0.3180 | 886.7 | -0.0057 |

two ways. One way is to learn the inline direction first and then learn the crossline direction. We call it the inline–cascade–crossline (ICC) denoising method. Conversely, it is called the crossline–cascade–inline (CCI) denoising method when the network learns the crossline direction first and then learns the inline direction. With the same computing resources, the training and testing time of 2-D network in our experiment is about a quarter of that of 3-D network, but 2-D network needs to be trained and tested twice.

1) *Denoising Results in Time Slice*: First, we compare the conventional denoising results and the 3-D-DnCNN network denoising results on an arbitrarily selected time slice. The denoising results of the conventional method are shown in Fig. 11(a). It can be found that the conventional method has effectively suppressed the random noise. The denoising results of the 3-D-DnCNN network on the same time slice are shown in Fig. 11(b). It can be observed that the 3-D-DnCNN network also successfully suppressed the random noise. Further observing the areas indicated by the three black boxes in Fig. 11(a) and (b), the field seismic data acquired in these areas are seriously corrupted because only low-dose explosives can be used for exploration in these areas due to the dense distribution of villages. By comparing the denoising results, it can be found that the structure of the useful signal in the denoising results obtained by the 3-D-DnCNN network is clearer. This is because the network uses training samples

selected from the area that has the best denoising performance of the conventional method. The 3-D-DnCNN network learned the good noise suppression ability of the conventional method in the training area during the training process and then applied this good noise-suppression ability to suppress the random noise of all 3-D seismic data. Fig. 11(c) and (d) shows the ICC and CCI denoising results, respectively. It can be seen that both 2-D networks can also effectively remove random noise. However, such a cascaded two-step denoising method suffers from the drawback that the perturbation error of the first step would be amplified at the second step, which may cause more damages to the useful signals. Subsequent denoising results prove that the damage caused by 2-D networks to the useful signal is greater than that of the 3-D network.

To further illustrate the differences between these methods, we observe the noise removed by the conventional method, the 3-D-DnCNN network, and ICC and CCI denoising methods for the same time slice. Fig. 12(a) shows the random noise removed by the conventional method. It can be seen that the noise removed by the conventional method is randomly distributed over the entire time slice and has no obvious structural features. Fig. 12(b) shows the noise removed by the 3-D-DnCNN network for the same time slice. Compared with Fig. 12(a), we note the following two facts. First, there is no obvious structure in the noise removed by the 3-D-DnCNN network and the noise removed by the conventional method,

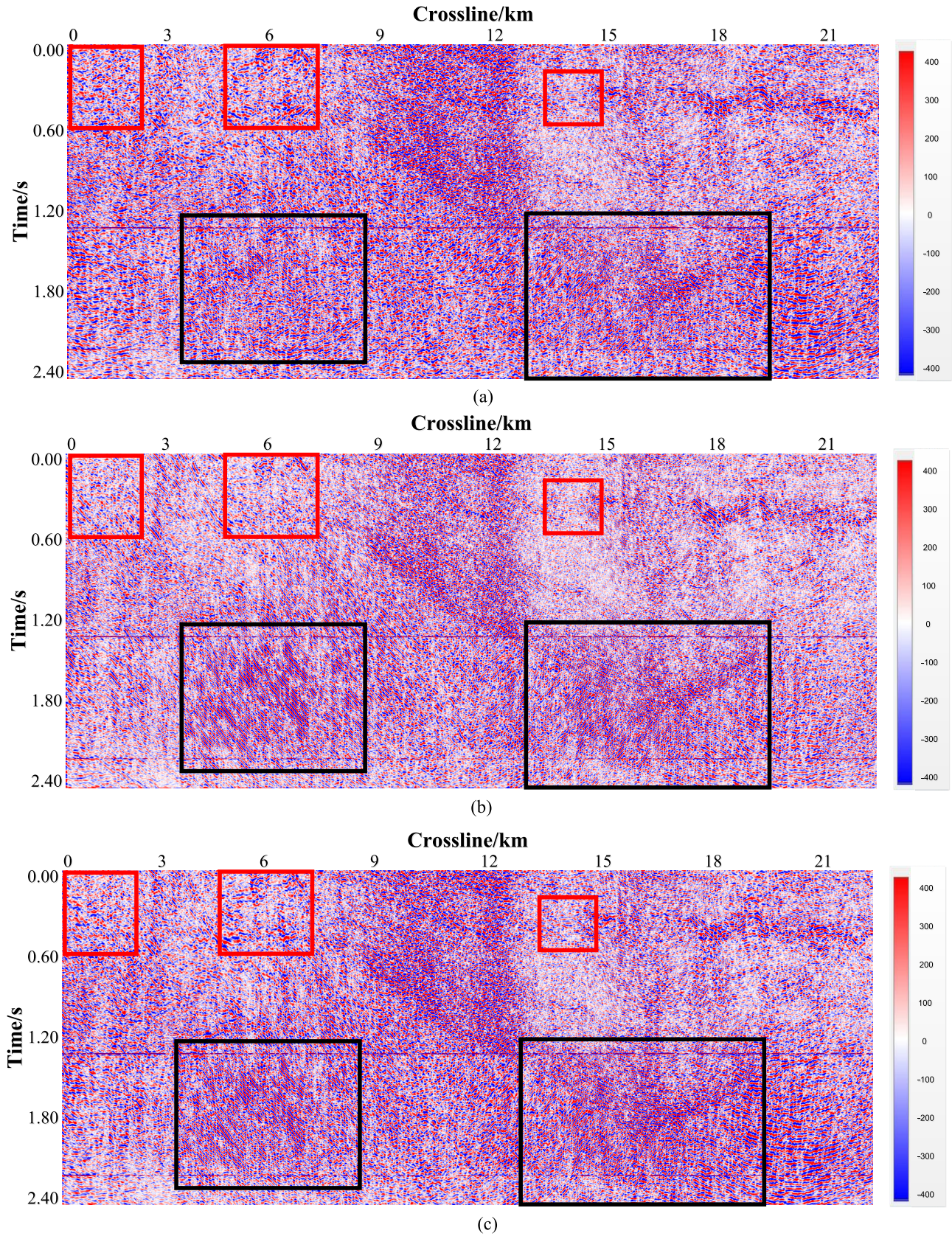


Fig. 18. Removed noise of inline profile. (a) Noise removed by the conventional method. (b) Noise removed by the 3-D-DnCNN network. (c) Noise removed by the 2-D-DnCNN network (inline cascade crossline). (d) Noise removed by the 2-D-DnCNN network (crossline cascade inline).

indicating that the two methods for suppressing random noise are effective. Second, in the area indicated by the three black

boxes, the SNR is relatively low, and the noise energy removed by the 3-D-DnCNN network is strong. However, in the training

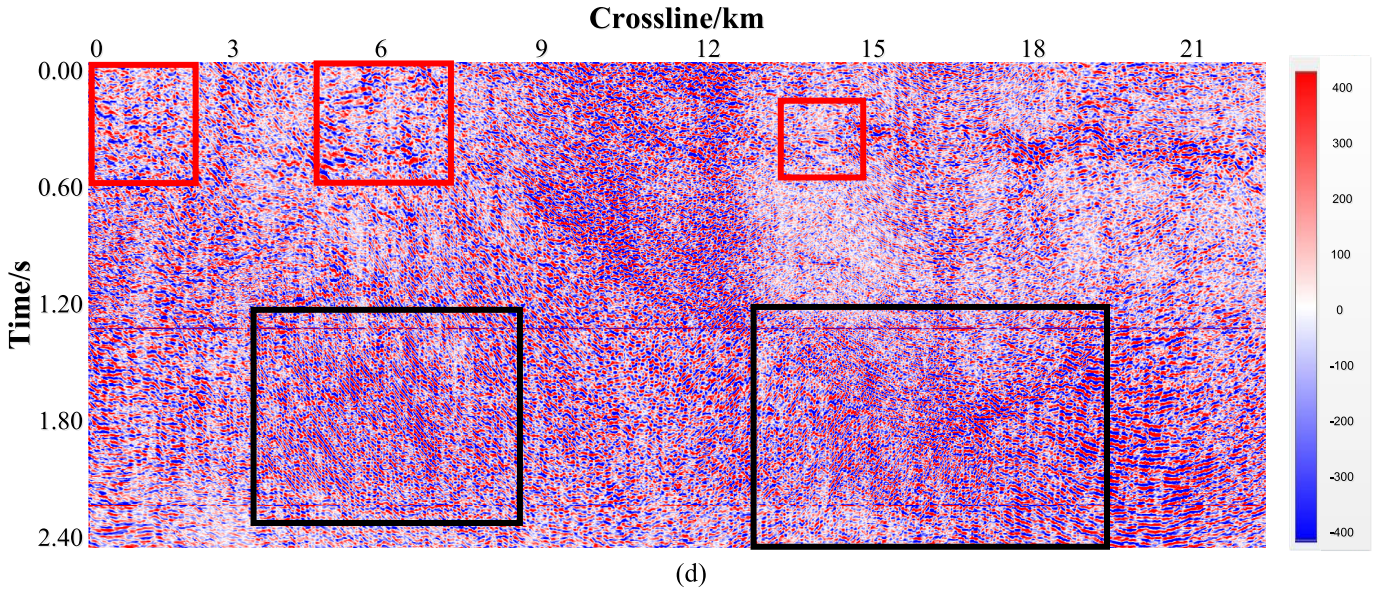


Fig. 18. (Continued.) Removed noise of inline profile. (a) Noise removed by the conventional method. (b) Noise removed by the 3-D-DnCNN network. (c) Noise removed by the 2-D-DnCNN network (inline cascade crossline). (d) Noise removed by the 2-D-DnCNN network (crossline cascade inline).

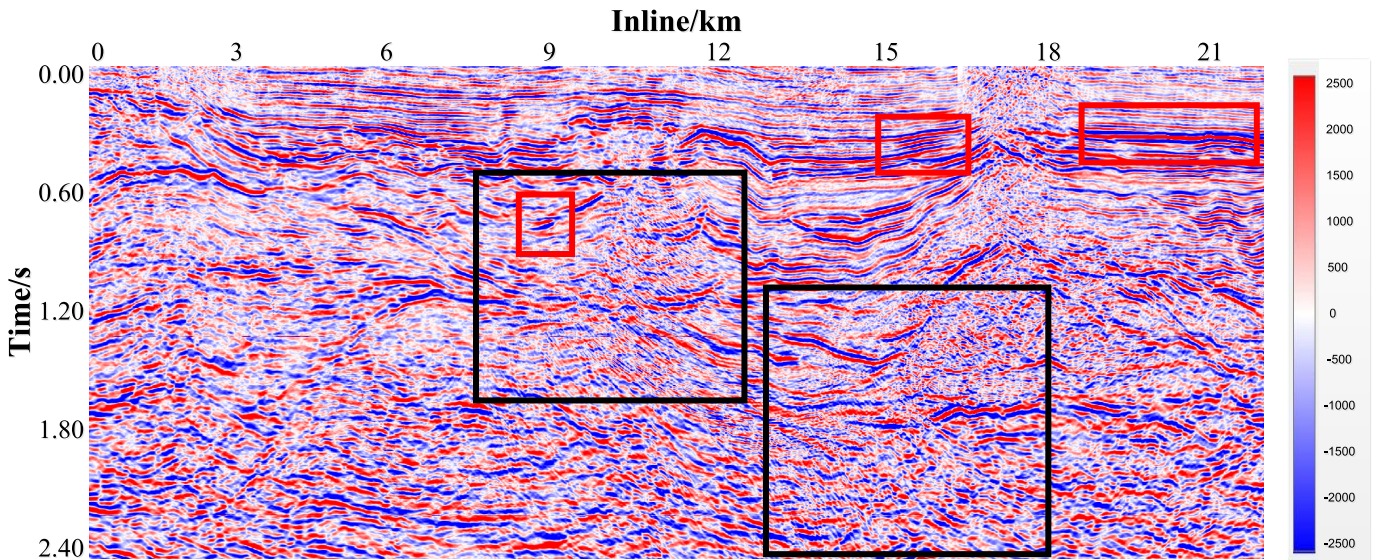


Fig. 19. Noisy crossline profile.

area where the seismic data have a high SNR, the noise energy removed by the network is weak. Comparing Fig. 12(b) with Fig. 12(c) and (d), we can observe that the 2-D networks cause more damage to the useful signal than the 3-D network does. This is because no matter which direction the 2-D network denoises first, it will interfere with the other direction so that the final result will cause more damage to the useful signal. This is similar to the conventional 2-D denoising method.

2) *Magnifying Denoising Results in Time Slice:* For a clearer view of the above-mentioned facts, we magnify the black box of the time slice in Fig. 3, as shown in Fig. 13. The corresponding denoising results and removed noise are shown in Figs. 14 and 15, respectively. Obviously, we can draw the following conclusions. The 3-D-DnCNN network

has learned the ability of the conventional method to suppress random noise from the training samples. However, the ability of the 3-D-DnCNN network to suppress random noise is not simply a copy of the conventional method. The noise energy suppressed by the 3-D-DnCNN network has the characteristics of weak energy in the region with high SNR and strong energy in the region with low SNR. That is to say, the 3-D-DnCNN network denoising method is more adaptive than the conventional method. At the same time, comparing the two regions indicated by two yellow arrows in Fig. 15(b) with that in Fig. 15(a), (c), and (d), it can be observed that the noise removed by the 3-D network contains very few useful signals than the noise removed by the other methods.

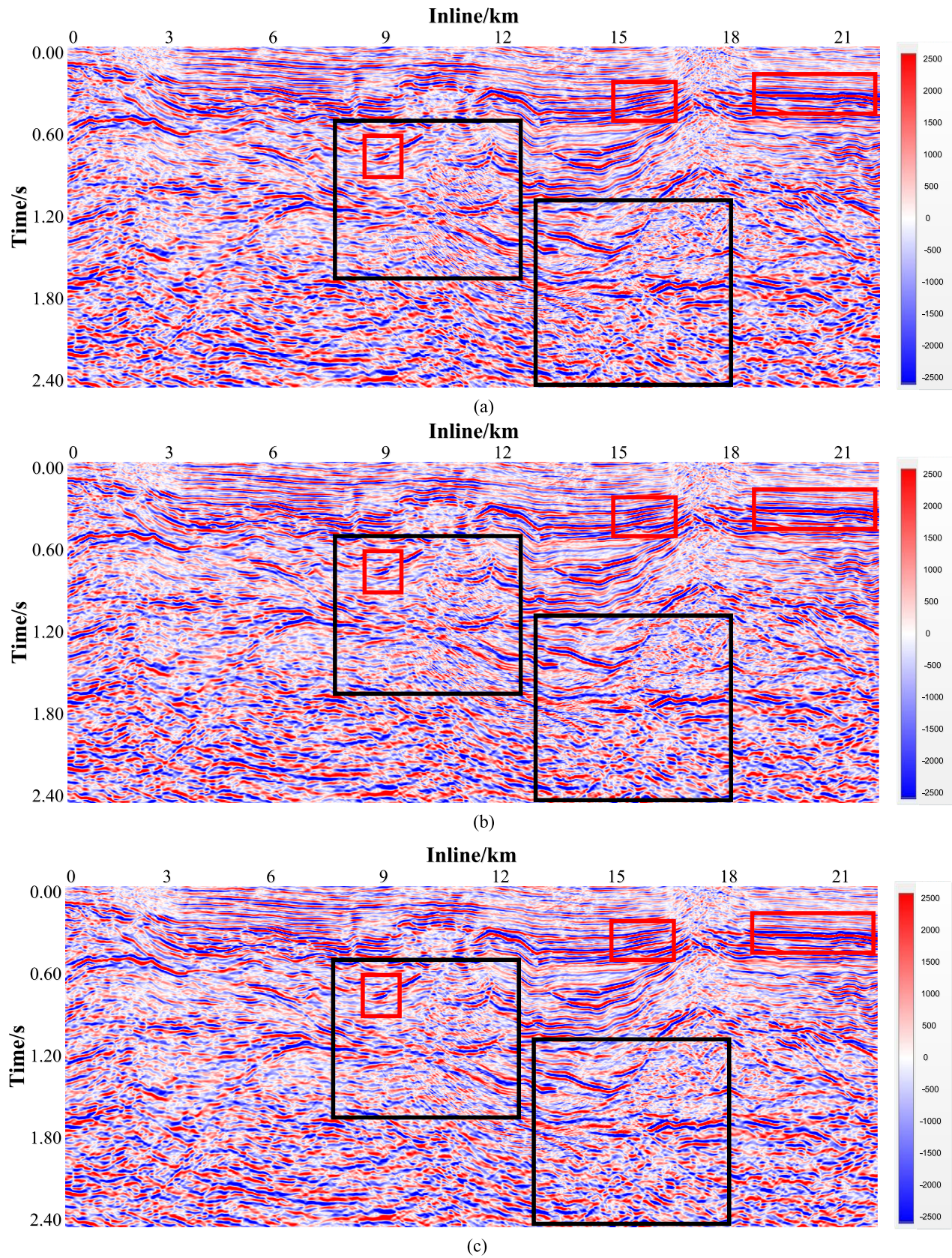


Fig. 20. Denoising results of the crossline profile. (a) Denoised by the conventional method. (b) Denoised by the 3-D-DnCNN network. (c) Denoised by the 2-D-DnCNN network (inline cascade crossline). (d) Denoised by the 2-D-DnCNN network (crossline cascade inline).

3) *Denoising Results in Profiles*: In order to make a clearer comparison of the difference between the networks and the

conventional method, this article presents two vertical profiles according to the positions of the two yellow lines in Fig. 3.

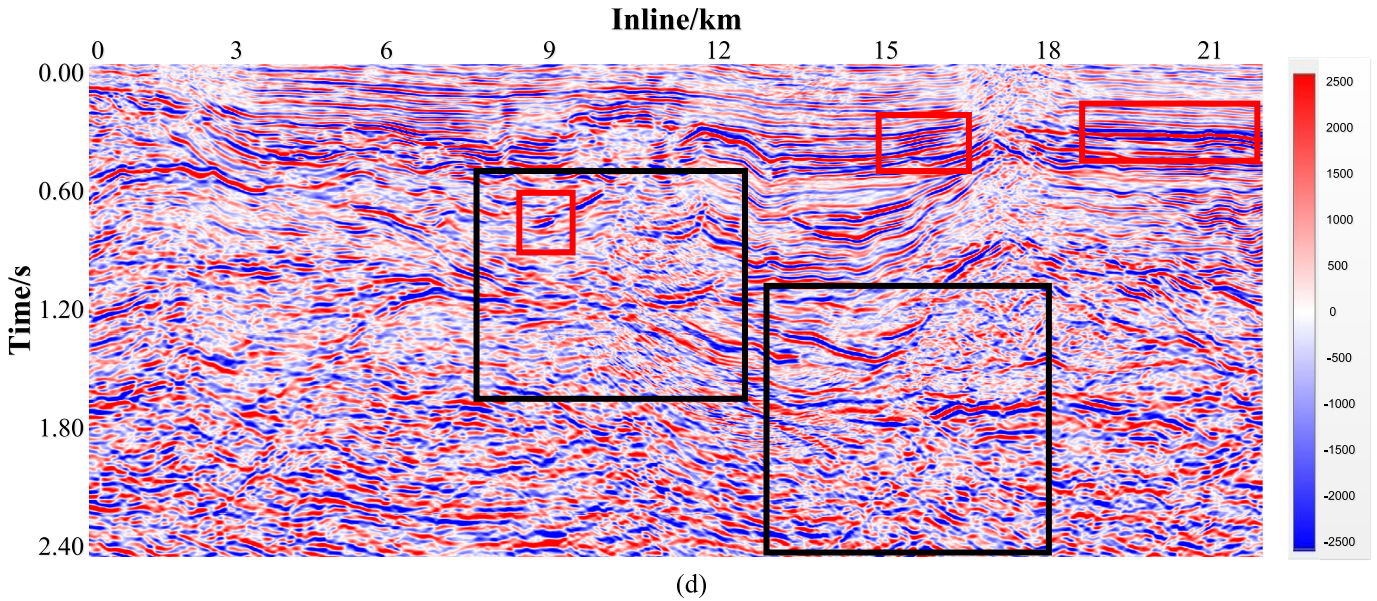


Fig. 20. (Continued.) Denoising results of the crossline profile. (a) Denoised by the conventional method. (b) Denoised by the 3-D-DnCNN network. (c) Denoised by the 2-D-DnCNN network (inline cascade crossline). (d) Denoised by the 2-D-DnCNN network (crossline cascade inline).

The noisy inline profile is shown in Fig. 16. The denoising results of the inline profile are shown in Fig. 17. To further illustrate the differences among the four methods, we observe the noise removed by the conventional method, the 3-D-DnCNN network, and ICC and CCI denoising methods. The noise removed by the conventional method on the inline profile is shown in Fig. 18(a). It can be seen that the noise suppressed by the conventional method is distributed randomly over the profile, and the conventional method has successfully suppressed the random noise. The noise removed by the 3-D-DnCNN network on the inline profile is shown in Fig. 18(b).

Compared with Fig. 18(a), we find that the 3-D-DnCNN network also successfully suppressed the random noise. The random noise removed by the 3-D-DnCNN network and the conventional method is generally similar across the entire profile. That is to say, the 3-D-DnCNN network obtains a random noise suppression capability for all of the 3-D seismic data by learning training-sample pairs generated from the small training block. In addition, comparing the regions indicated by the black rectangle in the two figures, it can be found that the ability of the network to suppress random noise is not simply a copy of the conventional method. The noise removed by the 3-D-DnCNN network in the black rectangular region has a more obvious structure than that by the conventional method. Looking at Fig. 18(c) and (d), we can also observe that the noise with structure is suppressed well by 2-D-DnCNN.

The structural noise that is difficult to remove by the conventional method is 3-D arc-like imaging noise produced during the imaging process. Therefore, this denoising result also shows that the CNN-based denoising methods have a more powerful denoising ability toward arc-like imaging noise. There are two main reasons for the improvement of arc-like imaging noise removal ability. First, most conventional denoising methods, including the conventional method used in this article, denoise the whole 3-D seismic data with 2-D

or 3-D sliding windows. Taking the conventional method used in this article as an example and considering the complexity of seismic data, a local edge-preserving filter is adopted to make the filter adaptive to the complex changes of dip angle and stratum orientation. The denoising process of different patches is independent. However, seismic data in the same work area have similarity in both useful signal and noise distribution. Most conventional methods are local filtering methods, which cannot make full use of self-similarity of the seismic data. On the contrary, the CNN-based denoising method is a nonlocal denoising method. Our loss function is to minimize the average errors between network outputs and labels. Although CNNs are locally perceived in a single patch, they share network parameters during the whole training process. In this way, our network can learn many repetitive features in the training process and make full use of the self-similarity of seismic data. That is to say, both 3-D-DnCNN and 2-D-DnCNN can learn useful signals and noises from a global point of view, so as to better remove the arc-like imaging noise. Second, there is little arc-like imaging noise in the labels of the training-sample pairs fed to the network as most of the arc-like imaging noise has been dropped after the sample selection. After sample selection, the 3-D-DnCNN network can accurately identify the useful signal masked by the arc-like imaging noise and then improve the ability to suppress the arc-like imaging noise.

Although the 2-D network has good noise suppression capability, the denoising performance of 3-D network for field seismic data is better than that of 2-D learning. The damage caused by the 2-D-by-2-D learning to the useful signals is significantly greater than that of 3-D learning, as can be seen from red boxes in Fig. 18.

Due to the lack of the ability to extract features in 3-D space simultaneously, the denoising result of 2-D-by-2-D learning can only guarantee the denoising result in one direction is

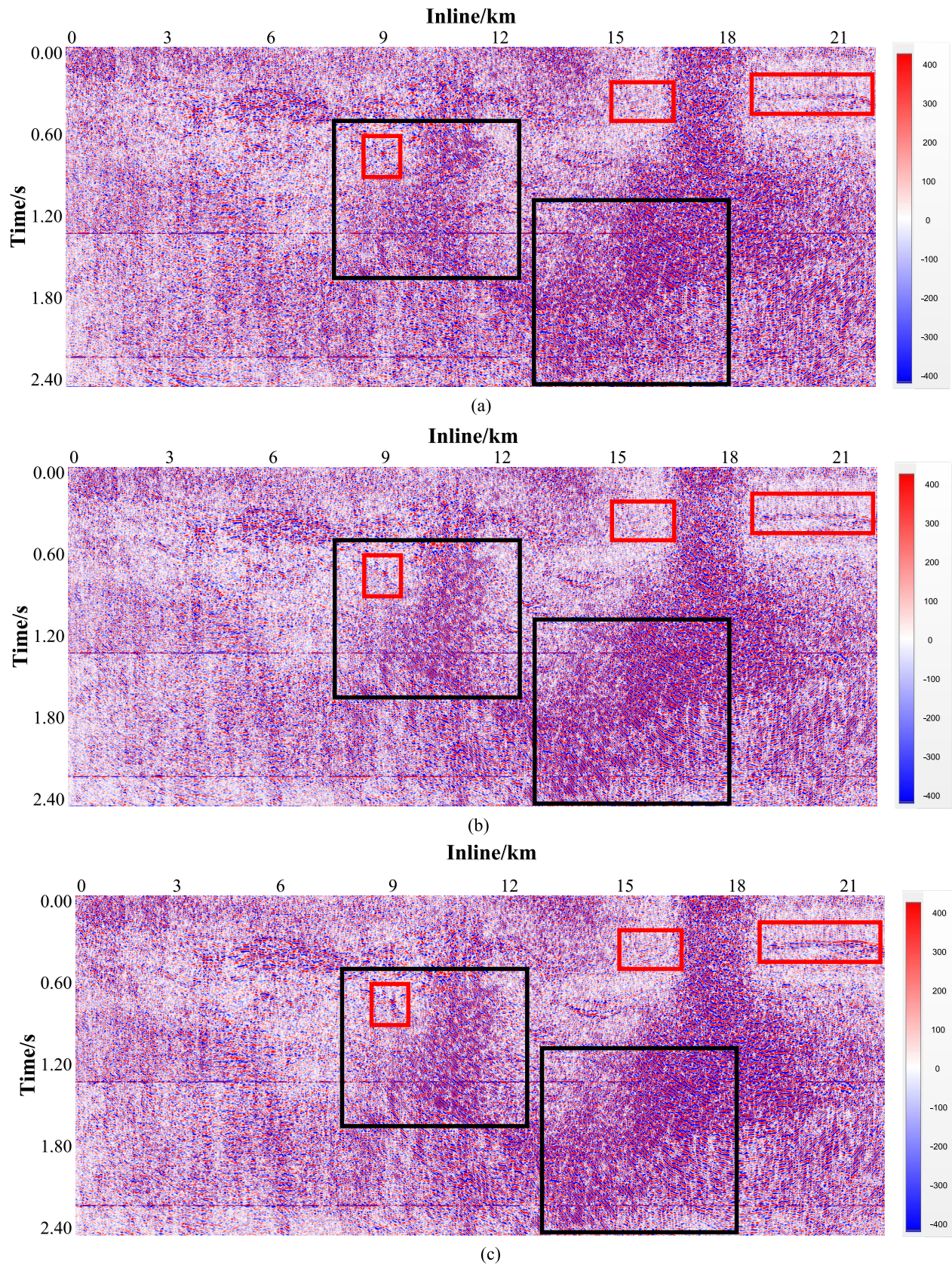


Fig. 21. Removed noise of crossline profile. (a) Noise removed by the conventional method. (b) Noise removed by the 3-D-DnCNN network. (c) Noise removed by the 2-D-DnCNN network (inline cascade crossline). (d) Noise removed by the 2-D-DnCNN network (crossline cascade inline).

good, but the denoising result in another direction is relatively poor. Taking inline direction denoising as an example, the CCI

denoising method causes greater damage to useful signals than the ICC denoising method does. The unbalanced denoising

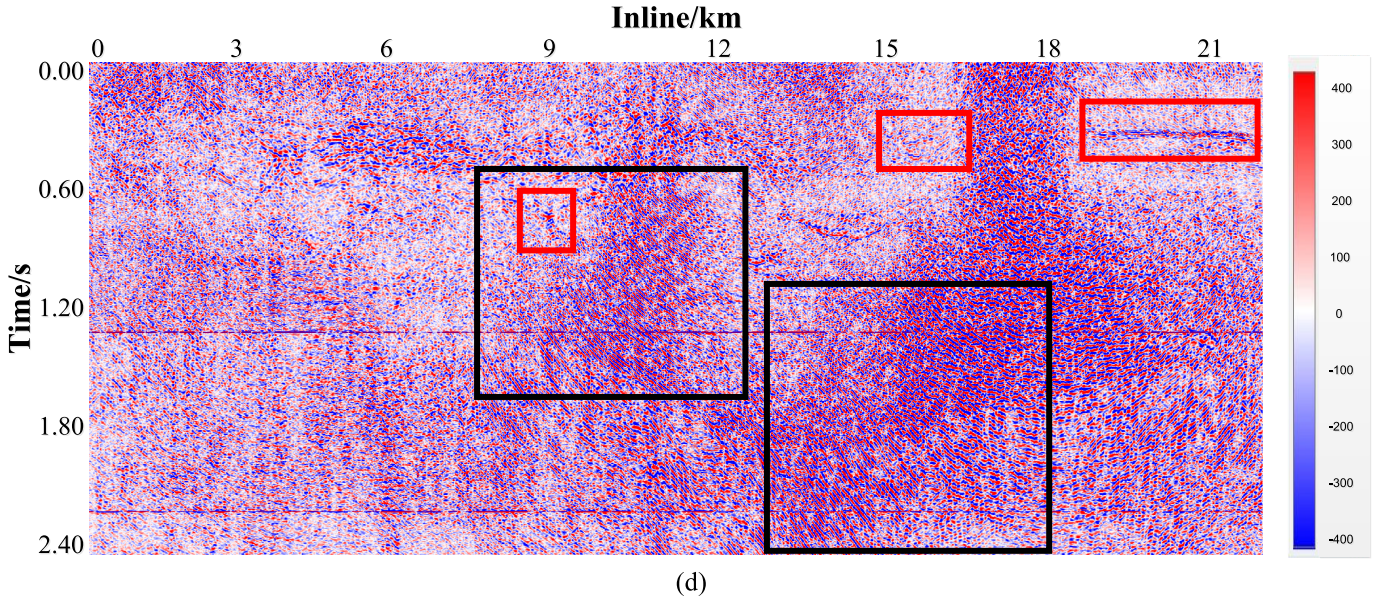


Fig. 21. (Continued.) Removed noise of crossline profile. (a) Noise removed by the conventional method. (b) Noise removed by the 3-D-DnCNN network. (c) Noise removed by the 2-D-DnCNN network (inline cascade crossline). (d) Noise removed by the 2-D-DnCNN network (crossline cascade inline).

TABLE III
STATISTICAL CHARACTERISTICS OF THE REMOVED NOISE FROM FIELD SEISMIC DATA

| Denoising Method | Average value | variance | kurtosis |
|---------------------|---------------|----------|----------|
| Conventional method | 0.0006 | 45162 | 1.9505 |
| 3D-DnCNN network | 4.2082 | 57287 | 2.3422 |

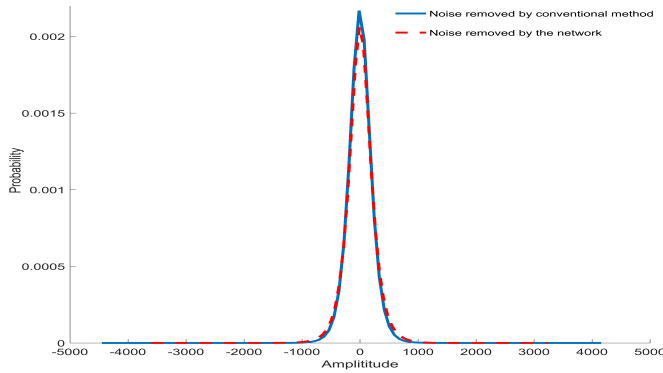


Fig. 22. Probability density distribution of noise removed by the conventional method and noise removed by the network.

performance makes 2-D learning denoising inferior to that of 3-D learning. However, 3-D learning can capture the full range of spatial variation expected from the seismic data and obtain good denoising results in all directions.

We further compared the noise removed on the crossline profile shown in Fig. 19. The denoising results of the crossline profile are shown in Fig. 20. Fig. 21(a) shows the noise removed by the conventional method, and Fig. 21(b) shows the noise removed by the 3-D-DnCNN network. Comparing Fig. 21(a) with Fig. 21(b), the denoising results also confirm our above-mentioned conclusions. Fig. 21(c) and (d) shows the

noise removed by ICC and CCI denoising methods, respectively. Similarly, we can observe red boxes from Fig. 21 that the damage to the useful signal caused by the 2-D network is greater than that by the 3-D network.

Fig. 22 shows the probability density distribution of the noise removed by the conventional method and the noise removed by the network. By comparison, it can be found that the probability density distributions of the noise removed by the two methods are basically the same. This also proves that the network has successfully learned the random noise suppression ability for 3-D seismic data. To further compare the differences between the two denoising methods, this study calculates the statistical characteristics of the noise removed by the two denoising methods, as shown in Table III.

Comparing the data in Table III, we find the following facts. First, the average values of the noise removed by the two denoising methods are close. Second, the kurtosis values are also close, that is, the number of extreme values of the noise removed by the network is basically equal to that removed by the conventional method. In addition, the kurtosis values are all greater than 0, indicating that the probability density distribution of the noise is subject to a super-Gaussian distribution. Finally, the variance of the noise removed by the network is significantly greater than that removed by the conventional method, that is, the degree of arc-like imaging noise removed by the network is greater than that of the

conventional method. This is consistent with the fact observed in Fig. 12(a) and (b) that the energy of noise removed by the network in the high-SNR area is weaker than that in the low-SNR area.

In summary, we chose a small high-SNR block that had no obvious arc-like imaging noise as a training area. With sample selection, relatively ideal training-sample pairs whose labels contained almost no arc-like noise were fed to the 3-D-DnCNN. Then, the 3-D-DnCNN was utilized to denoise all of the field seismic data. In the figures of denoising results we show, the 3-D-DnCNN can suppress random noise, including arc-like imaging noise, effectively. The denoising ability of the 3-D-DnCNN network is obtained by learning training samples generated by sample selection. Although the process of our sample selection is very complex, the 3-D-DnCNN also learns the inherent knowledge of this complex process and gains a strong ability to denoise all of the seismic data in the whole work area. However, this ability is not a complete copy of the conventional denoising method because the training samples were selected by the method in Section II-A. This also proves that the network can learn its denoising ability from the more ideal training samples so that the denoising results have better fidelity and generalization. Moreover, because the 3-D network can fully capture 3-D spatial structure information, the damage of 3-D network denoising to useful signals is less than that of the 2-D network.

IV. CONCLUSION

In this article, a novel deep CNN called 3-D-DnCNN was proposed for seismic data denoising, inspired by the fact that the poststack seismic data has a 3-D spatial structure. The denoising ability of a 3-D-DnCNN can be automatically learned from the given training-sample pairs. We adopted different strategies to build training-sample pairs for synthetic seismic data and field seismic data. In experiments with synthetic seismic data, we built training samples by adding the Gaussian random noise to the clean synthetic seismic data. With these good labels, the 3-D-DnCNN exhibited an impressive performance in denoising noisy synthetic seismic data with fault reservation. In the experiments with field seismic data, the random noise was super-Gaussian.

We applied a training-sample selection method that contained an excellent conventional method to obtain the comparatively ideal training-sample pairs. Although the process of training sample selection is complex, the 3-D-DnCNN learns the denoising ability for field seismic data directly from the training samples. The denoising results for the synthetic seismic data and field seismic data demonstrated that our method has general denoising ability on both the Gaussian random noise and the super-Gaussian random noise while reserving the useful signals. Moreover, when the energy of arc-like imaging noise was relatively strong compared with the useful signal, it could be found that the conventional method was difficult to suppress the arc-like imaging noise, but the network had obvious suppression ability on the arc-like imaging noise.

ACKNOWLEDGMENT

The authors would like to thank the associate editor and two reviewers for their valuable comments and suggestions that improved the manuscript significantly.

REFERENCES

- [1] D. C. Hagen, "The application of principal components analysis to seismic data sets," *Geoexploration*, vol. 20, nos. 1–2, pp. 93–111, 1982.
- [2] G.-C. Liu, X.-H. Chen, J. Du, and J.-W. Song, "Seismic noise attenuation using nonstationary polynomial fitting," *Appl. Geophys.*, vol. 8, no. 1, pp. 18–26, 2011.
- [3] M. Bekara and M. Van der Baan, "Local singular value decomposition for signal enhancement of seismic data," *Geophysics*, vol. 72, no. 2, pp. V59–V65, 2007.
- [4] W. Lu, "Adaptive noise attenuation of seismic images based on singular value decomposition and texture direction detection," *J. Geophys. Eng.*, vol. 3, no. 1, pp. 28–34, 2006.
- [5] B. Ursin and Y. Zheng, "Identification of seismic reflections using singular value decomposition," *Geophys. Prospecting*, vol. 33, no. 6, pp. 773–799, 1985.
- [6] I. Jones and S. Levy, "Signal-to-noise ratio enhancement in multichannel seismic data via the Karhunen-Loève transform," *Geophys. prospecting*, vol. 35, no. 1, pp. 12–32, 1987.
- [7] K. M. Al-Yahya, "Application of the partial Karhunen-Loève transform to suppress random noise in seismic sections," *Geophys. prospecting*, vol. 39, no. 1, pp. 77–93, 1991.
- [8] T. Yanan, L. Yue, W. Bo, L. Yanping, and Z. Tie, "A novel convolutive ICA for seismic data denoising," in *Recent Advances in Computer Science and Information Engineering*, vol. 128. Berlin, Germany: Springer, 2012, pp. 95–101.
- [9] M. Bekara and M. Van Der Baan, "Local SVD/ICA for signal enhancement of pre-stack seismic data," in *Proc. 68th EAGE Conf. Exhib. Incorporating SPE EUROPEC*, 2006, pp. 1–4.
- [10] M. D. Sacchi and H. Kuehl, "ARMA formulation of FX prediction error filters and projection filters," *J. Seismic Explor.*, vol. 9, no. 3, pp. 185–198, 2001.
- [11] R. Abma and J. Claerbout, "Lateral prediction for noise attenuation by tx and fx techniques," *Geophysics*, vol. 60, no. 6, pp. 1887–1896, 1995.
- [12] M. K. Chase, "Random noise reduction by FXY prediction filtering," *Explor. Geophys.*, vol. 23, no. 2, pp. 51–56, 1992.
- [13] S. Trickett, "F-xy Cadzow noise suppression," in *Proc. 78th Annu. Int. Meeting, SEG, Expanded Abstr.*, 2008, pp. 2586–2590.
- [14] S. Yuan and S. Wang, "A local f-x Cadzow method for noise reduction of seismic data obtained in complex formations," *Petroleum Sci.*, vol. 8, no. 3, pp. 269–277, 2011.
- [15] J. H. Zhang, L. Ning, L. Y. Tian, L. U. Wen-Zhi, and L. Zhong, "An overview of the methods and techniques for seismic data noise attenuation," *Prog. Geophys.*, vol. 21, no. 2, pp. 546–553, 2006.
- [16] W.-K. Lu, "Seismic random noise suppression based on the discrete cosine transform," *Oil Geophys. Prospecting*, vol. 46, no. 2, pp. 202–206, 2011.
- [17] K. Ye, Y. Chengye, and J. Wo, "A study on noise-suppression method in fx domain," *Oil Geophys. Prospecting*, vol. 38, no. 2, pp. 136–138, 2003.
- [18] D. Gabor, "Theory of communication. Part 1: The analysis of information," *J. Inst. Elect. Eng.*, vol. 93, no. 26, pp. 429–441, Jul. 1946.
- [19] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, Jul. 1989.
- [20] S. Sinha, P. S. Routh, P. D. Anno, and J. P. Castagna, "Spectral decomposition of seismic data with continuous-wavelet transform," *Geophysics*, vol. 70, no. 6, pp. 19–25, 2008.
- [21] M. Jian, G. Jinghuai, and C. Wenchao, "On the denoising method of prestack seismic data in wavelet domain," in *Proc. 76th Annu. Int. Meet., SEG, Expanded Abstr.*, 2006, pp. 2852–2856.
- [22] E. J. Candès and D. L. Donoho, "Continuous curvelet transform: I. Resolution of the wavefront set," *Appl. Comput. Harmon. Anal.*, vol. 19, no. 2, pp. 162–197, 2005.
- [23] G. Hennenfent and F. J. Herrmann, "Seismic denoising with nonuniformly sampled curvelets," *Comput. Sci. Eng.*, vol. 8, no. 3, pp. 16–25, May 2006.
- [24] R. Neelamani, A. I. Baumstein, D. G. Gillard, M. T. Hadidi, and W. L. Soroka, "Coherent and random noise attenuation using the curvelet transform," *Lead. Edge*, vol. 27, no. 2, pp. 240–248, 2008.

- [25] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.
- [26] M. Elad and M. Aharon, "Image denoising via learned dictionaries and sparse representation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, Jun. 2006, pp. 895–900.
- [27] G. Tang, J.-W. Ma, and H.-Z. Yang, "Seismic data denoising based on learning-type overcomplete dictionaries," *Appl. Geophys.*, vol. 9, no. 1, pp. 27–32, 2012.
- [28] J. Ristau and W. M. Moon, "Adaptive filtering of random noise in 2-D geophysical data," *Geophysics*, vol. 66, no. 1, pp. 342–349, 2001.
- [29] Y. Jeng, Y.-W. Li, C.-S. Chen, and H.-Y. Chien, "Adaptive filtering of random noise in near-surface seismic and ground-penetrating radar data," *J. Appl. Geophys.*, vol. 68, no. 1, pp. 36–46, 2009.
- [30] N. M. AlBinHassan, Y. Luo, and M. N. Al-Faraj, "3D edge-preserving smoothing and applications," *Geophysics*, vol. 71, no. 4, pp. P5–P11, 2006.
- [31] A. Y. Anagaw and M. D. Sacchi, "Edge-preserving seismic imaging using the total variation method," *J. Geophys. Eng.*, vol. 9, no. 2, p. 138, 2012.
- [32] M. Bekara and M. Van der Baan, "Random and coherent noise attenuation by empirical mode decomposition," *Geophysics*, vol. 74, no. 5, pp. V89–V98, 2009.
- [33] J. Ferahtia, K. Baddari, N. Djarfour, and A. K. Kassouri, "Incorporation of a non-linear image filtering technique for noise reduction in seismic data," *Pure Appl. Geophys.*, vol. 167, no. 11, pp. 1389–1404, 2010.
- [34] Y. Liu, S. Fomel, and G. Liu, "Nonlinear structure-enhancing filtering using plane-wave prediction," *Geophys. Prospecting*, vol. 58, no. 3, pp. 415–427, 2010.
- [35] K. Baddari, J. Ferahtia, T. Aïfa, and N. Djarfour, "Seismic noise attenuation by means of an anisotropic non-linear diffusion filter," *Comput. Geosci.*, vol. 37, no. 4, pp. 456–463, 2011.
- [36] S. Yuan, S. Wang, and G. Li, "Random noise reduction using Bayesian inversion," *J. Geophys. Eng.*, vol. 9, no. 1, pp. 60–68, 2011.
- [37] D. Bonar and M. Sacchi, "Denoising seismic data using the nonlocal means algorithm," *Geophysics*, vol. 77, no. 1, pp. A5–A8, 2012.
- [38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: <https://arxiv.org/abs/1409.1556>
- [39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [40] R. Socher, Y. Bengio, and C. D. Manning, "Deep learning for NLP (without magic)," in *Proc. Tutorial Abstr. ACL*, Jul. 2012, p. 5.
- [41] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "DeepDriving: Learning affordance for direct perception in autonomous driving," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 2722–2730.
- [42] A. P. Valentine and J. Trampert, "Data space reduction, quality assessment and searching of seismograms: Autoencoder networks for waveform data," *Geophys. J. Int.*, vol. 189, no. 2, pp. 1183–1202, 2012.
- [43] A. P. Valentine, L. M. Kalnins, and J. Trampert, "Discovery and analysis of topographic features using learning algorithms: A seamount case study," *Geophys. Res. Lett.*, vol. 40, no. 12, pp. 3048–3054, 2013.
- [44] Q. Liu, X. Hu, M. Ye, X. Cheng, and F. Li, "Gas recognition under sensor drift by using deep learning," *Int. J. Intell. Syst.*, vol. 30, no. 8, pp. 907–922, 2015.
- [45] T. Dahlke, M. Araya-Polo, C. Zhang, and C. Frogner, "Predicting geological features in 3D Seismic Data," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–4.
- [46] M. Araya-Polo, T. Dahlke, C. Frogner, C. Zhang, T. Poggio, and D. Hohl, "Automated fault detection without seismic processing," *Lead. Edge*, vol. 36, no. 3, pp. 208–214, 2017.
- [47] K. Vikraman, "A deep neural network to identify foreshocks in real time," 2016, *arXiv:1611.08655*. [Online]. Available: <https://arxiv.org/abs/1611.08655>
- [48] M. Korjani, A. Popa, E. Grijalva, S. Cassidy, and I. Ershaghi, "A new approach to reservoir characterization using deep learning neural networks," presented at the SPE Western Regional Meeting, Anchorage, AK, USA, May 2016, pp. 1–15, Paper SPE-180359-MS.
- [49] S. Wu and J. Cao, "Lithology identification method based on continuous restricted Boltzmann machine and support vector machine," *Prog. Geophys.*, vol. 31, no. 2, 2016, pp. 821–828, 2016.
- [50] P. M. DeVries, T. B. Thompson, and B. J. Meade, "Enabling large-scale viscoelastic calculations via neural network acceleration," *Geophys. Res. Lett.*, vol. 44, no. 6, pp. 2662–2669, 2017.
- [51] J. Cao, "Deep learning and its application in deep gas reservoir prediction," *Comput. Techn. Geophys. Geochem. Explor.*, vol. 39, no. 6, pp. 775–782, 2017.
- [52] X. Wu, Y. Shi, S. Fomel, and L. Liang, "Convolutional neural networks for fault interpretation in seismic images," in *Proc. 88th Annu. Int. Meeting, SEG, Expanded Abstr.*, 2018, pp. 1946–1950.
- [53] Y. Ma, X. Ji, N. M. BenHassan, and Y. Luo, "A deep learning method for automatic fault detection," in *Proc. 88th Annu. Int. Meeting, SEG, Expanded Abstr.*, 2018, pp. 1941–1945.
- [54] O. Gramstad and M. Nickel, "Automated interpretation of top and base salt using deep convolutional networks," in *Proc. 88th Annu. Int. Meeting, SEG, Expanded Abstracts*, 2018, pp. 1956–1960.
- [55] H. Li, W. Yang, and X. Yong, "Deep learning for ground-roll noise attenuation," in *Proc. 88th Annu. Int. Meeting, SEG, Expanded Abstracts*, 2018, pp. 1981–1985.
- [56] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2392–2399.
- [57] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 341–349.
- [58] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian Denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.
- [59] H. Wu, B. Zhang, T. Lin, F. Li, and N. Liu, "White noise attenuation of seismic trace by integrating variational mode decomposition with convolutional neural network," *Geophysics*, vol. 84, no. 5, pp. V315–V325, 2019.
- [60] S. Yu, J. Ma, and W. Wang, "Deep learning for denoising," *Geophysics*, vol. 84, no. 6, pp. 1–107, 2019. [Online]. Available: <https://library.seg.org/doi/abs/10.1190/geo2018-0668.1>
- [61] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.
- [62] J. Lehtinen *et al.*, "Noise2Noise: Learning image restoration without clean data," 2018, *arXiv:1803.04189*. [Online]. Available: <https://arxiv.org/abs/1803.04189>
- [63] W. Wang, X. Wang, H. Zeng, and Q. Liang, "Preconditioning point-source/point-receiver high-density 3D seismic data for lacustrine shale characterization in a loess mountain area," *Interpretation*, vol. 5, no. 2, pp. SF177–SF188, 2017.
- [64] G. C. Fehmers and C. F. Höcker, "Fast structural interpretation with structure-oriented filtering," *Geophysics*, vol. 68, no. 4, pp. 1286–1293, 2003.
- [65] P. Bakker, "Image structure analysis for seismic interpretation," M.S. thesis, Delft Univ. Technol., Delft, The Netherlands, 2002.
- [66] X. Huang, J. Shan, and V. Vaidya, "Lung nodule detection in CT using 3D convolutional neural networks," in *Proc. IEEE 14th Int. Symp. Biomed. Imag.*, Apr. 2017, pp. 379–383.
- [67] Q. Dou *et al.*, "Automatic detection of cerebral microbleeds from MR images via 3D convolutional neural networks," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1182–1195, May 2016.
- [68] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and imagenet?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6546–6555.
- [69] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral-spatial residual network for hyperspectral image classification: A 3-D deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847–858, Feb. 2018.
- [70] M. Lai, "Deep learning for medical image segmentation," 2015, *arXiv:1505.02000*. [Online]. Available: <https://arxiv.org/abs/1505.02000>
- [71] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, 2012, pp. 1106–1114.
- [72] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
- [73] R. Ogden, "Wavelets made easy," *J. Amer. Stat. Assoc.*, vol. 95, no. 451, p. 1007, 2000.
- [74] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1026–1034.
- [75] D. Jiang, W. Dou, L. Vosters, X. Xu, Y. Sun, and T. Tan, "Denoising of 3D magnetic resonance images with multi-channel residual learning of convolutional neural network," *Jpn. J. Radiol.*, vol. 36, no. 9, pp. 566–574, Sep. 2018.



Dawei Liu received the bachelor's degree in communication engineering from Chang'an University, Xi'an, China, in 2013, and the master's degree in electronic and communication engineering from Xi'an Jiaotong University (XJTU), Xi'an, in 2015, where he is currently pursuing the Ph.D. degree with the School of Information and Communication Engineering.

His research interests include deep learning, and time–frequency analysis and their applications in the seismic data processing.



Cheng Wang was born in January 1981. He received the master's degree from the China University of Petroleum, Beijing, China, in 2007.

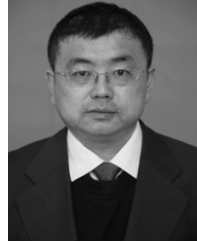
He is currently a Senior Engineer with the Exploration and Development Research Institute, Daqing Oilfield, Daqing, China, where he has been mainly engaged in seismic data high-resolution processing and complex wave field imaging methods.



Wei Wang received the bachelor's degree in information engineering and the Ph.D. degree in communication engineering from Xi'an Jiaotong University (XJTU), Xi'an, China, in 2006 and 2012, respectively.

From 2012 to 2016, he was a Senior Engineer Researcher with the Institute of Shaanxi Yanchang Petroleum (Group) Company, Ltd., Xi'an, where he was in charge of integrated geophysical data analysis. He was a Visiting Scholar of the Bureau of Economic Geology at The University of Texas

at Austin, Austin, TX, USA, from 2014 to 2016. He then joined DataYes, Inc., Shanghai, China, as an Algorithm Expert. His research interests include statistical machine learning, deep neural networks, and big data analysis and their applications in asset management and investment consulting.



Jiangyun Pei was born in February 1963. He received the Ph.D. degree from the Chinese Academy of Sciences, Beijing, China, in 2002.

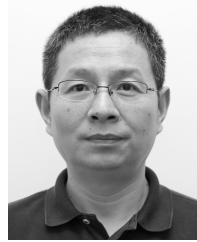
He is currently a professor-level Senior Engineer with the Exploration and Development Research Institute, Daqing Oilfield, Daqing, China. He has long been engaged in high-resolution seismic processing and method research.



Xiaokai Wang (M'17) received the bachelor's degree in information engineering and the Ph.D. degree in communication engineering from Xi'an Jiaotong University (XJTU), Xi'an, China, in 2006 and 2012, respectively.

From 2012 to 2014, he was a Post-Doctoral Fellow with the Institute of Geology and Geophysics, Chinese Academy of Sciences, Beijing, China. He joined the Department of Computational Geophysics, XJTU in 2015 where he was transferred to the School of Information and Communication

Engineering in 2018. He was a Visitor of the Bureau of Economic Geology at The University of Texas at Austin, Austin, TX, USA, from 2016 to 2017. His research interests include seismic attributes extraction, and time–frequency analysis and their applications in seismic data processing.



Wenchao Chen received the B.S. and M.S. degrees in seismic exploration and information technology from Chang'an University, Xi'an, China, in 1993 and 1996, respectively, and the Ph.D. degree in electromagnetic field and microwave technology from Xi'an Jiaotong University, Xi'an, in 2000.

From 2000 to 2002, he was a Post-Doctoral Fellow with the Department of Computation Science, Northwestern Polytechnical University, Xi'an. Since 2002, he has been with the Institute of Wave and Information, Xi'an Jiaotong University, where

he is currently a Professor. His research interests include seismic and ground-penetrating radar (GPR) signal processing, blind signal processing, and sparse representation.

Dr. Chen has served as a member of the Society of Exploration Geophysicists (SEG) and the Chinese Geophysical Society. He was also a Referee for several journals, including the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, *Geophysics*, *Interpretation*, and the *Journal of Applied Geophysics*.