Deep learning for prestack strong scattered noise suppression

Dawei Liu and Mauricio D. Sacchi

UNIVERSITY OF ALBERTA

SIGNAL ANALYSIS & IMAGING GROUP

OUTLINE

- Introduction
- Model and network training
- Instances on Synthetic data and real Data
- Conclusion
- Acknowledgement

INTRODUCTION

Benefits of Prestack Data

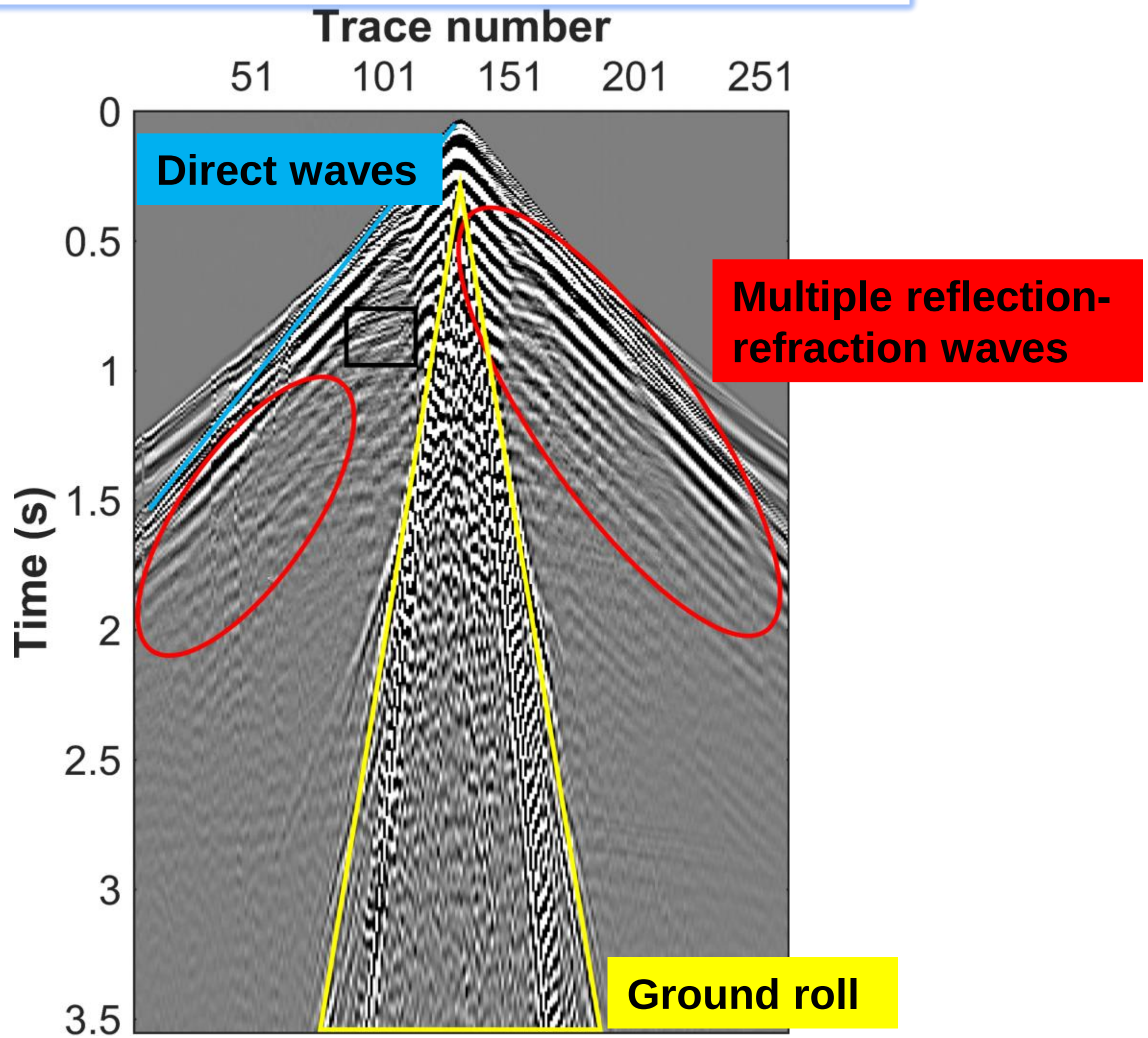
- Velocity analysis
- AVO analysis
- Fracture prediction
- lithology prediction
- ...

INTRODUCTION

Benefits of Prestack Data

- Velocity analysis
- AVO analysis
- Fracture prediction
- lithology prediction
- ...

Problem: Low SNR



A prestack Shot gather

CONVENTIONAL METHODS

Prior knowledge

CONVENTIONAL METHODS

Prior knowledge

Signal

Noise

CONVENTIONAL METHODS

- **Filtering methods**
  - ☐ High-pass and f–k filtering
    (Embree et al., 1963; Gelisli and Karsli, 1988; Treitel et al., 1967)
  - ☐ prediction filtering and f-x decon
    (Gulunay 1986; Canales, 1984)
  - ☐ Wavelet Transform Filtering
    (Deighan and Watts, 1997; Zhang and Ulrych, 2003)
  - ☐ S and x-f-k transforms (Askari and Siahkoohi, 2008)
- **Sparsity representation**
  - ☐ Wavelet transform (Chen et al., 2017)
  - ☐ Curvelet transform
    (Yarham et al., 2006; Yarham and Hermann et al., 2008; Naghizadeh and Sacchi, 2018)
  - ☐ Ridgelet transform
    (Chen et al., 2007)
  - ......

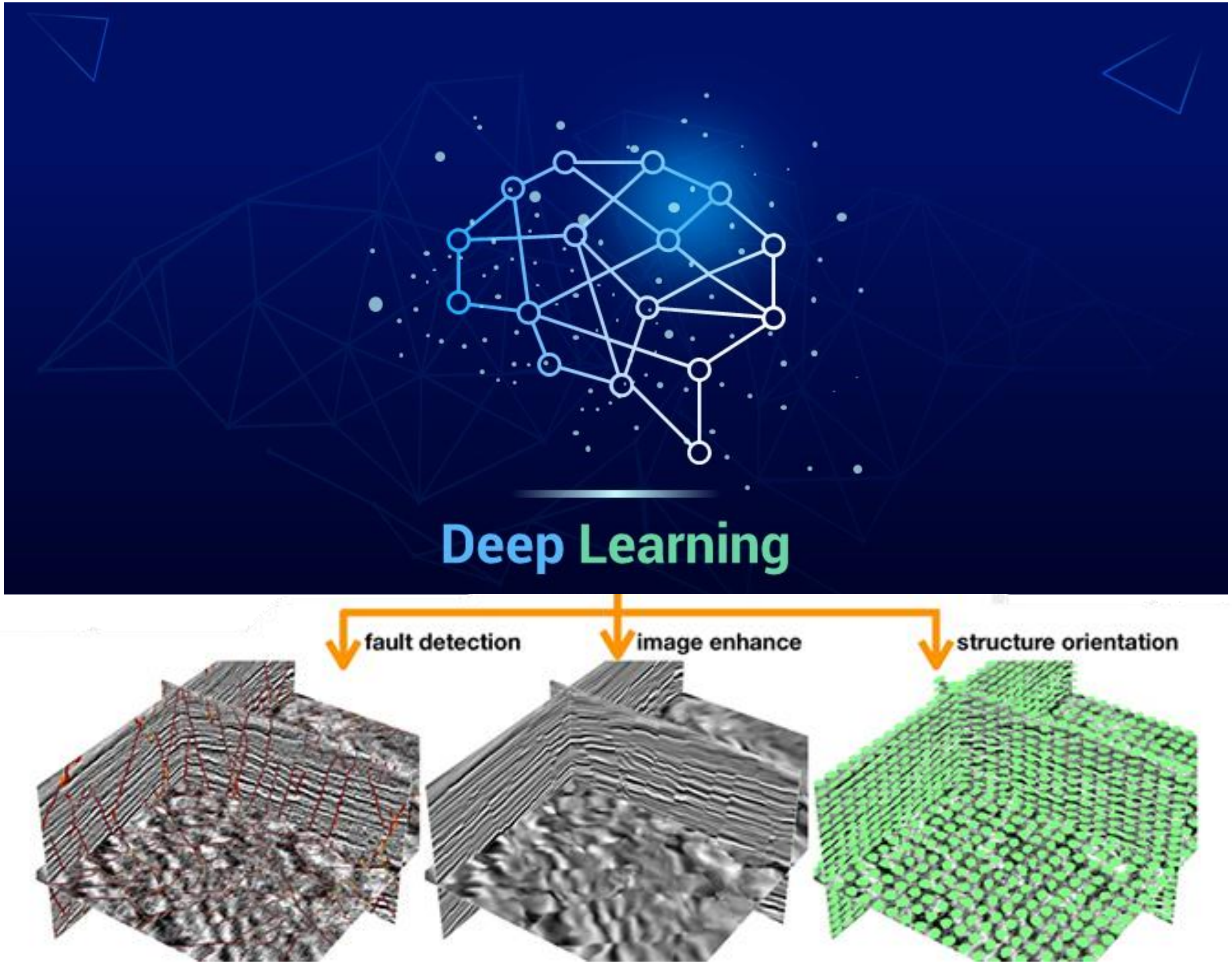
CONVENTIONAL METHODS

- Filtering methods
- Sparsity representation
- Low-rank representation
  - ☐ SVD (Trickett, 2002; Cary and Zhang, 2009; da Silva et al., 2016)
  - ☐ Multichannel Singular Spectrum Analysis (Chiu, 2013)
  - ☐ Nuclear norm minimization (Kreimer et al., 2013; Li et al., 2017)

  ……

DEEP LEARNING-BASED METHODS

DEEP LEARNING-BASED METHODS



DEEP LEARNING-BASED METHODS

- Unsupervised learning

Prior knowledge

Network structure

DEEP LEARNING-BASED METHODS

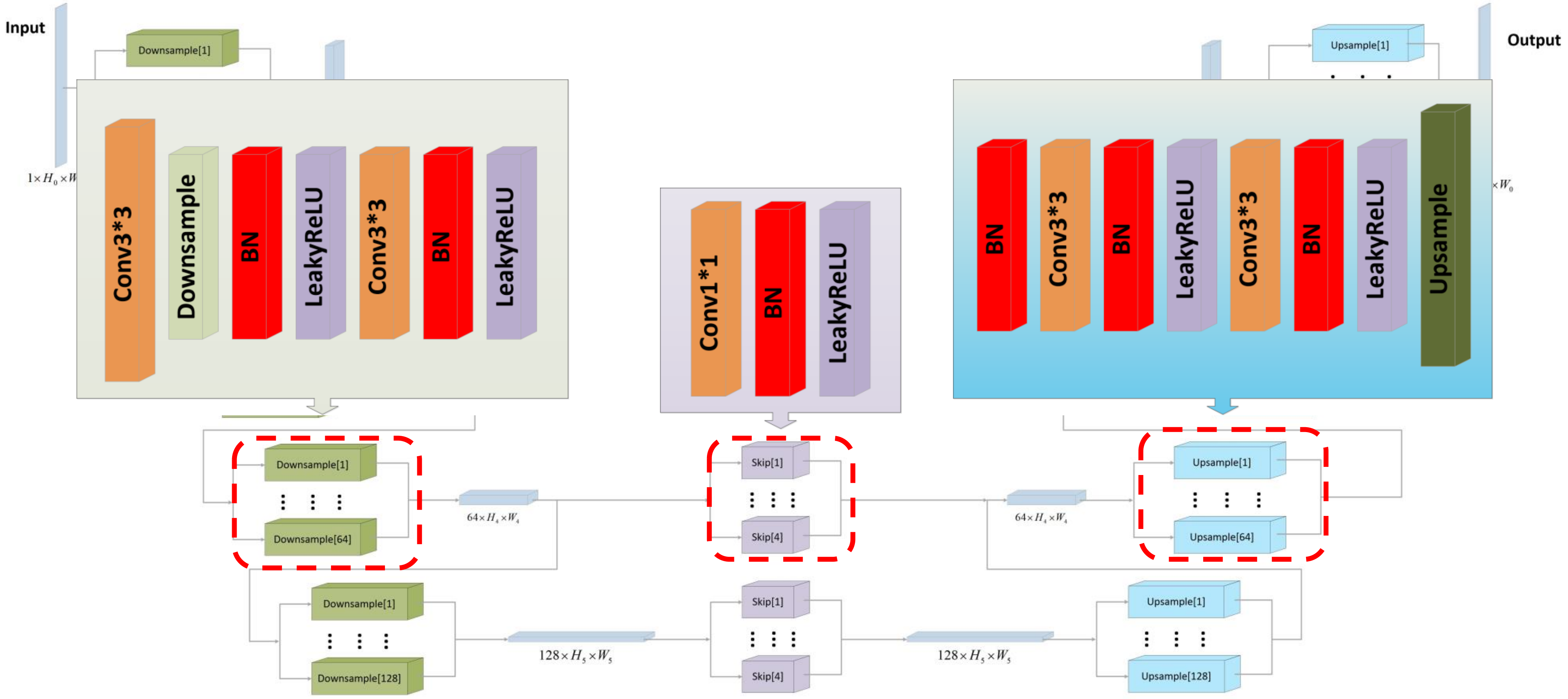
- Unsupervised learning
  - ☐ Sparse autoencoder (Zhang et al., 2019)
  - ☐ Deep prior (Liu et al., 2020)
  - ☐ Deep skip autoencoder (Yang et al., 2021;)

    ……

Prior knowledge

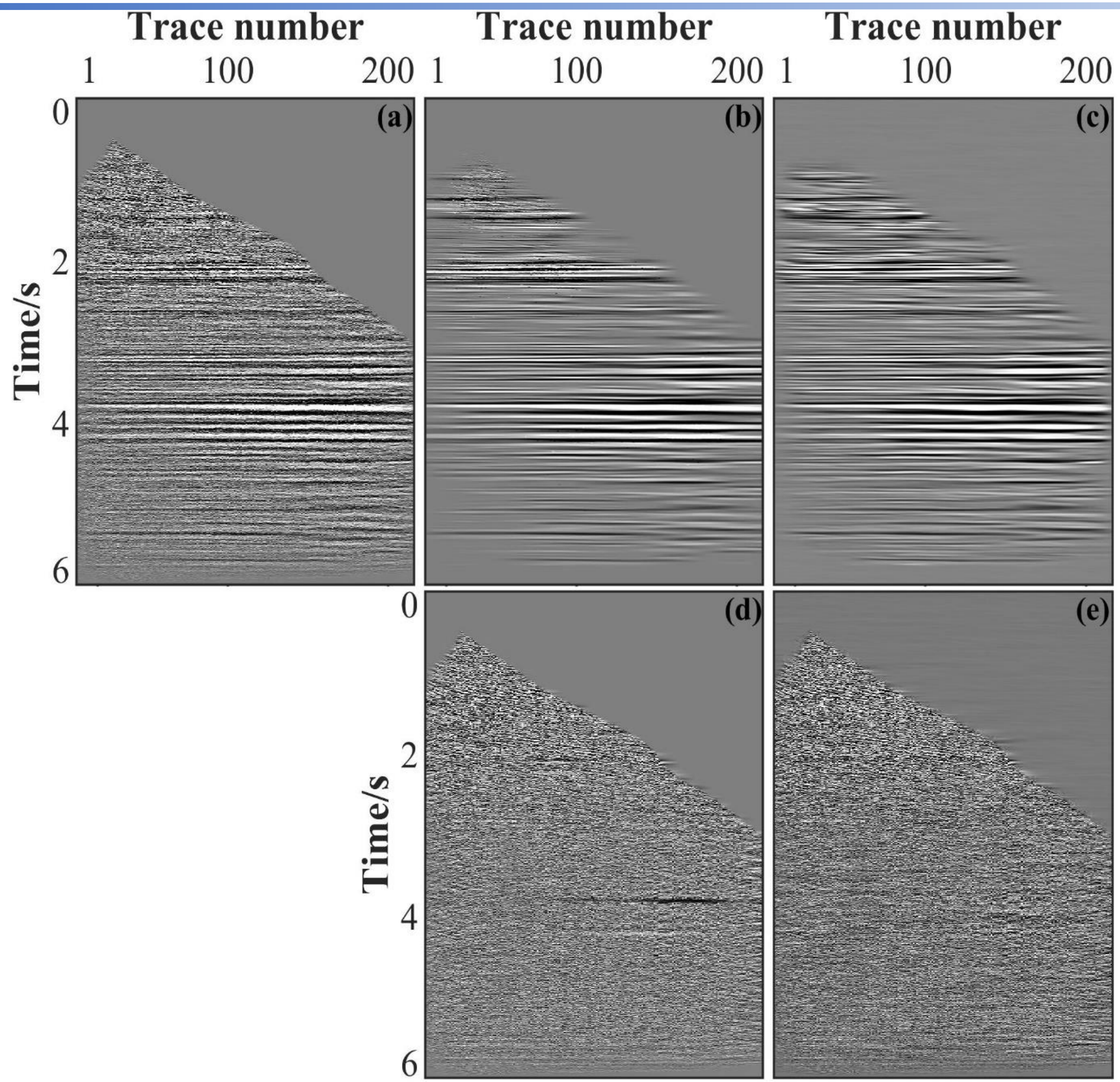
Network structure

DEEP LEARNING-BASED METHODS



Network architecture

DEEP LEARNING-BASED METHODS



Comparison of denoising results of actual seismic data.

(a) Noisy CRP gather.

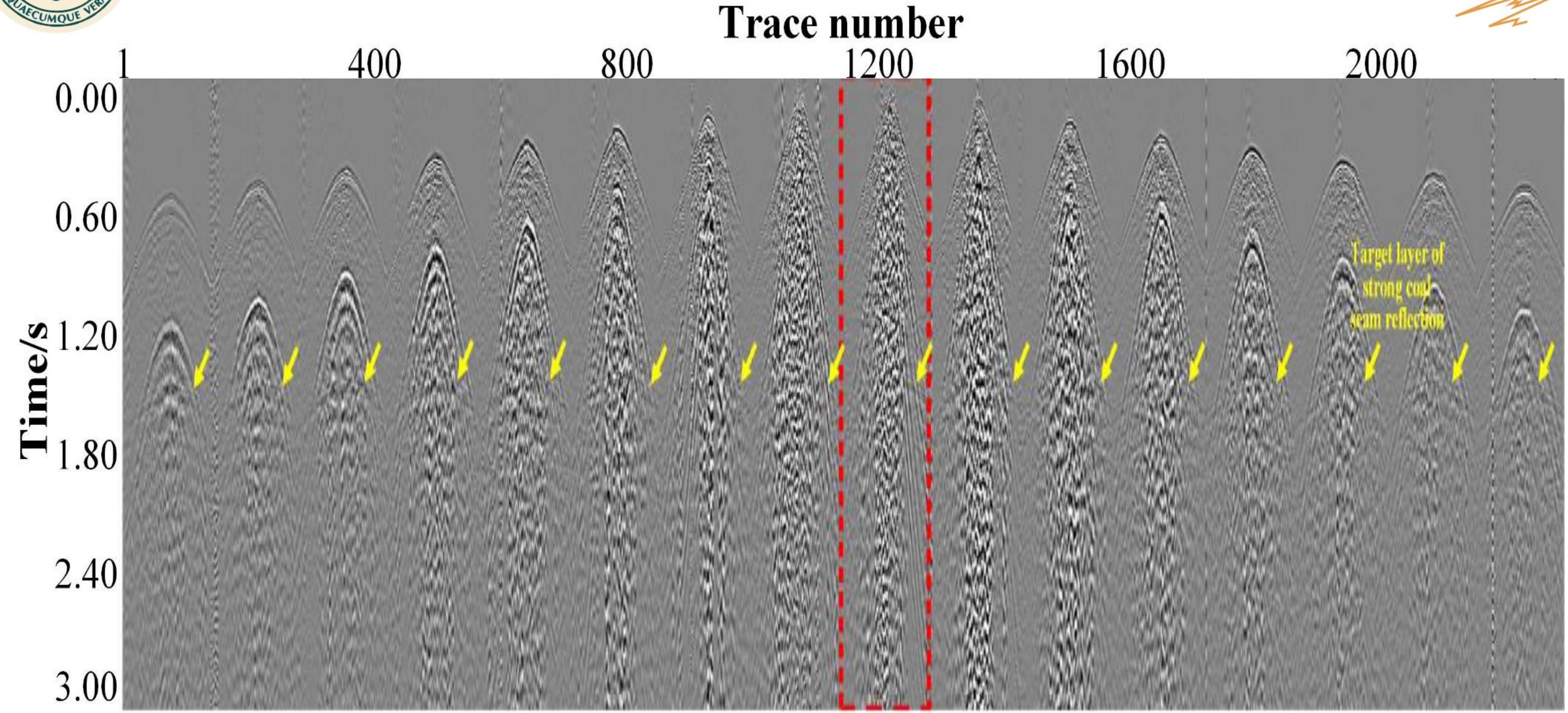
(b) Results obtained by DDTF.

(c) Results obtained by our method.

(d) Removed noise by DDTF.

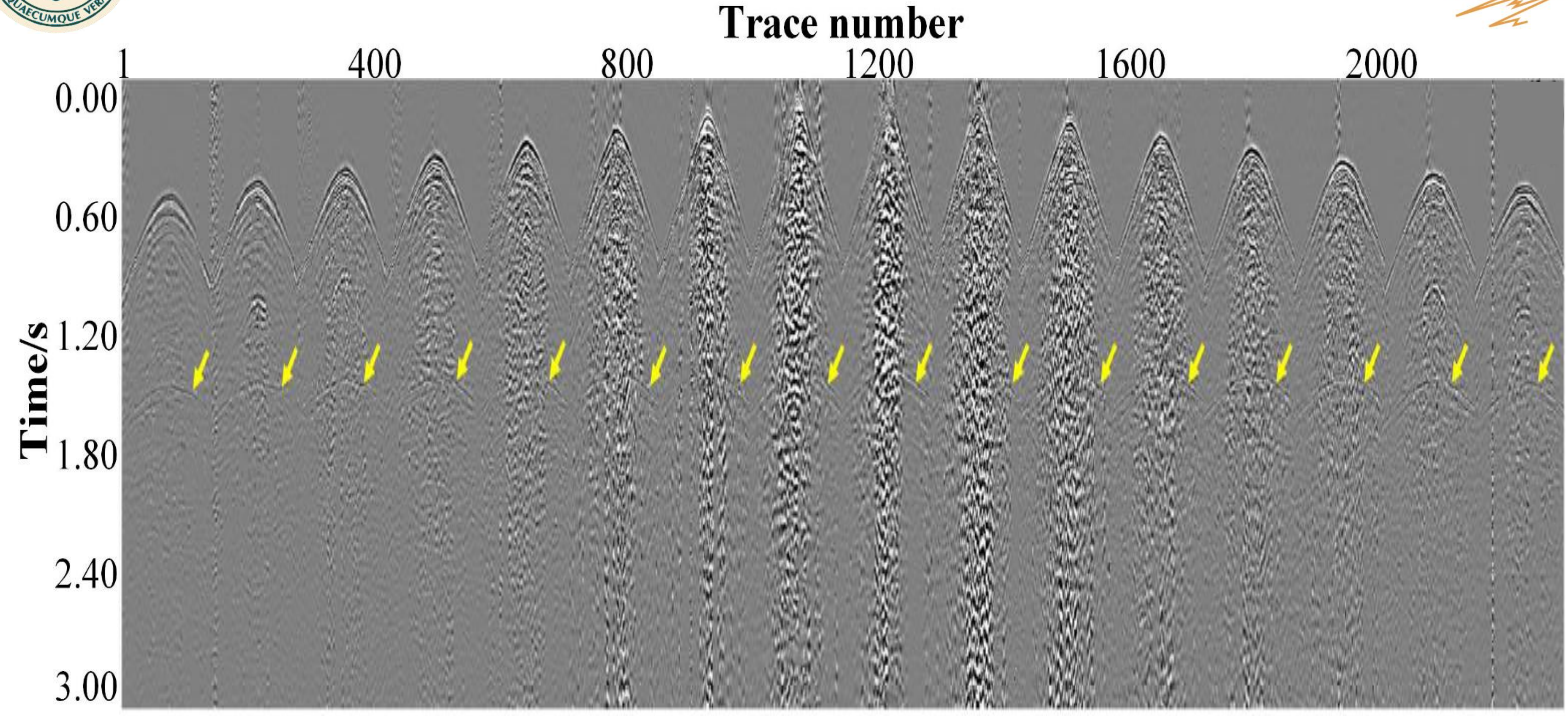
(e) Removed noise by our method.

DEEP LEARNING-BASED METHODS



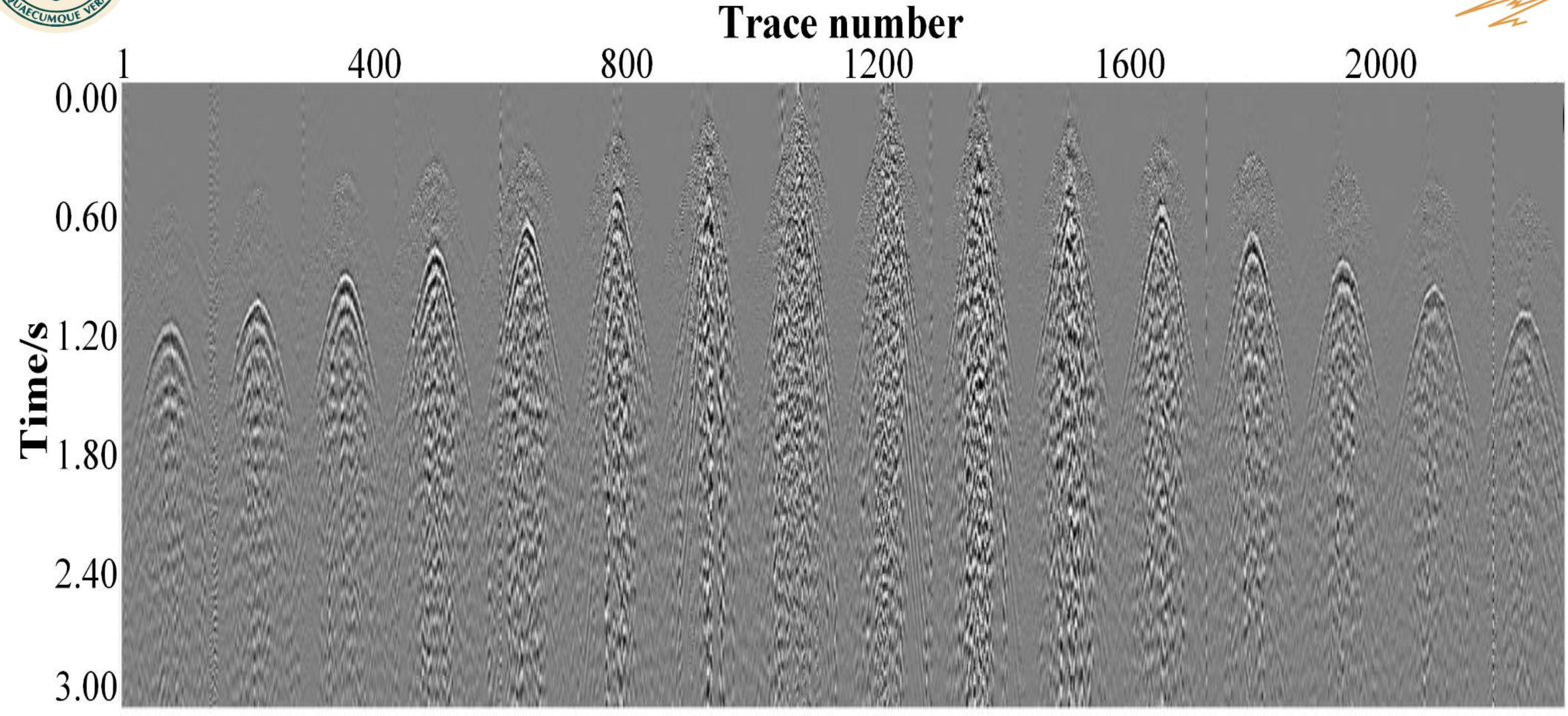
Noisy data

DEEP LEARNING-BASED METHODS



Denoised by the unsupervised method

DEEP LEARNING-BASED METHODS



Removed ground roll and scattered noise.

DEEP LEARNING-BASED METHODS

- **Unsupervised learning**
- **Supervised learning**

Prior knowledge

Training data

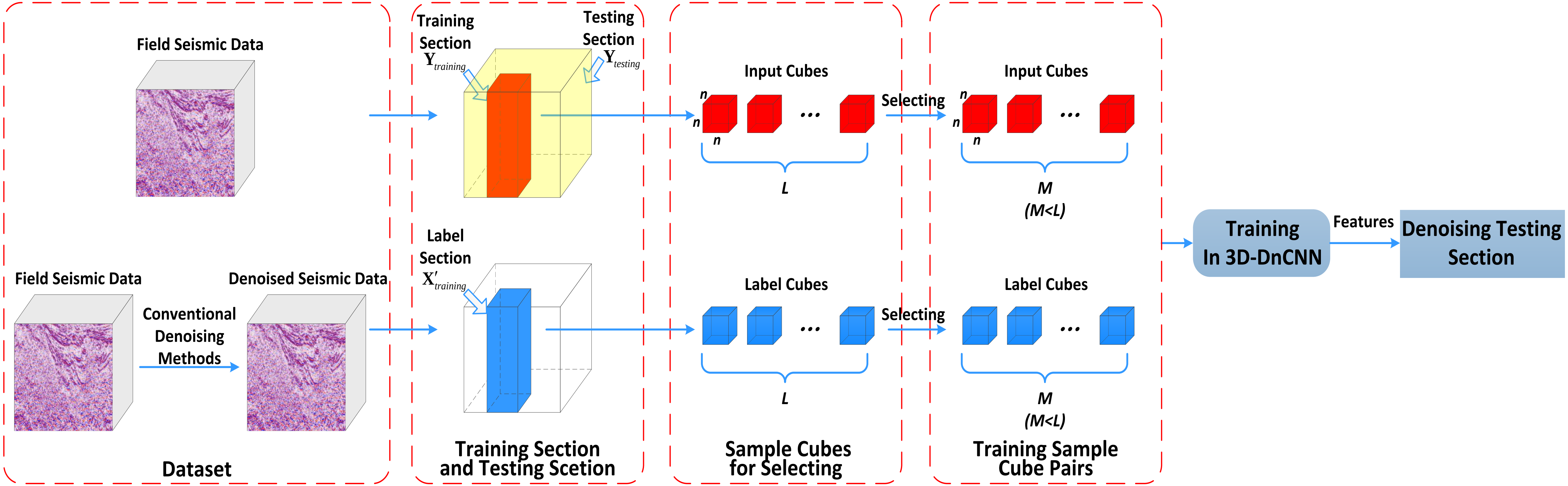
DEEP LEARNING-BASED METHODS

- Unsupervised learning
- Supervised learning
  - ☐ DnCNN (Li et al., 2019;Liu et al., 2019)
  - ☐ U-net (Sun et al., 2020; Wang et al., 2021)
  - ☐ Generative adversarial network (Kaur et al., 2019; Yu et al., 2019; Yuan et al., 2020)

Prior knowledge

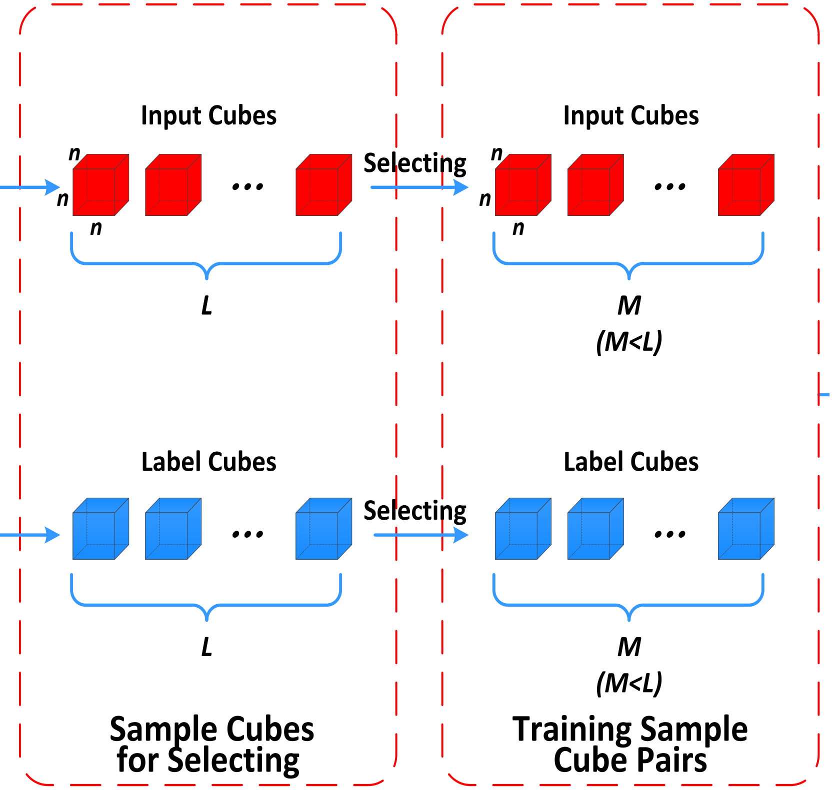
Training data

DEEP LEARNING-BASED METHODS

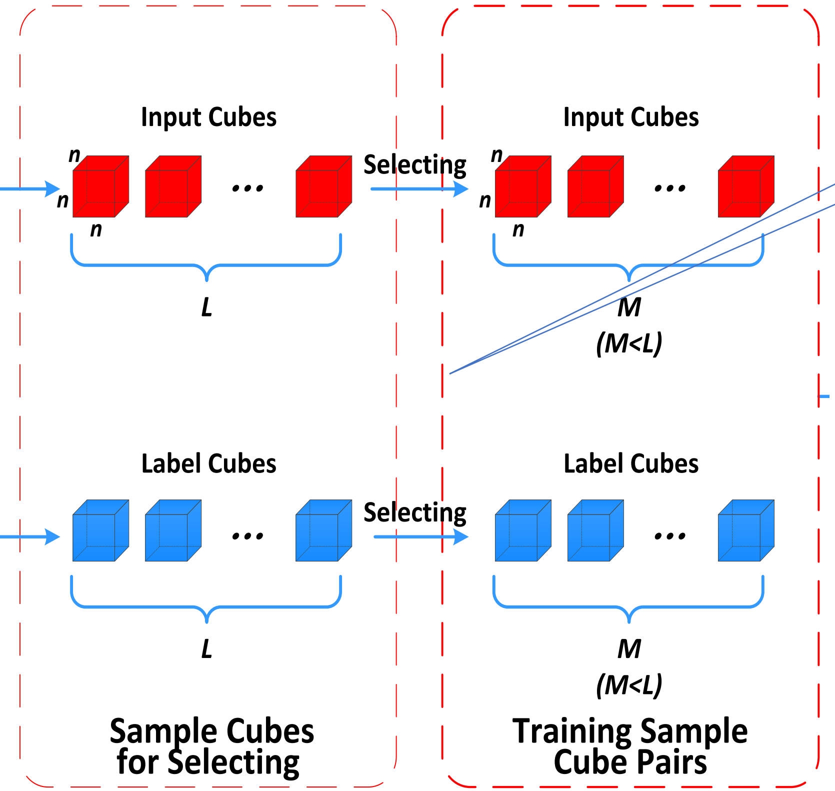


Flow chart of constructing training samples.

DEEP LEARNING-BASED METHODS



DEEP LEARNING-BASED METHODS



Fault confidence

Hilberta transform $\hat{\mathbf{X}}'_{\text{training}} = \mathrm{H}\left[\mathbf{X}'_{\text{training}}\right] = \mathbf{X}'_{\text{training}} * \frac{1}{\pi t}$

Instantaneous phase $\psi = \arctan \frac{\hat{\mathbf{X}}'_{\text{training}}}{\mathbf{X}'_{\text{training}}}$

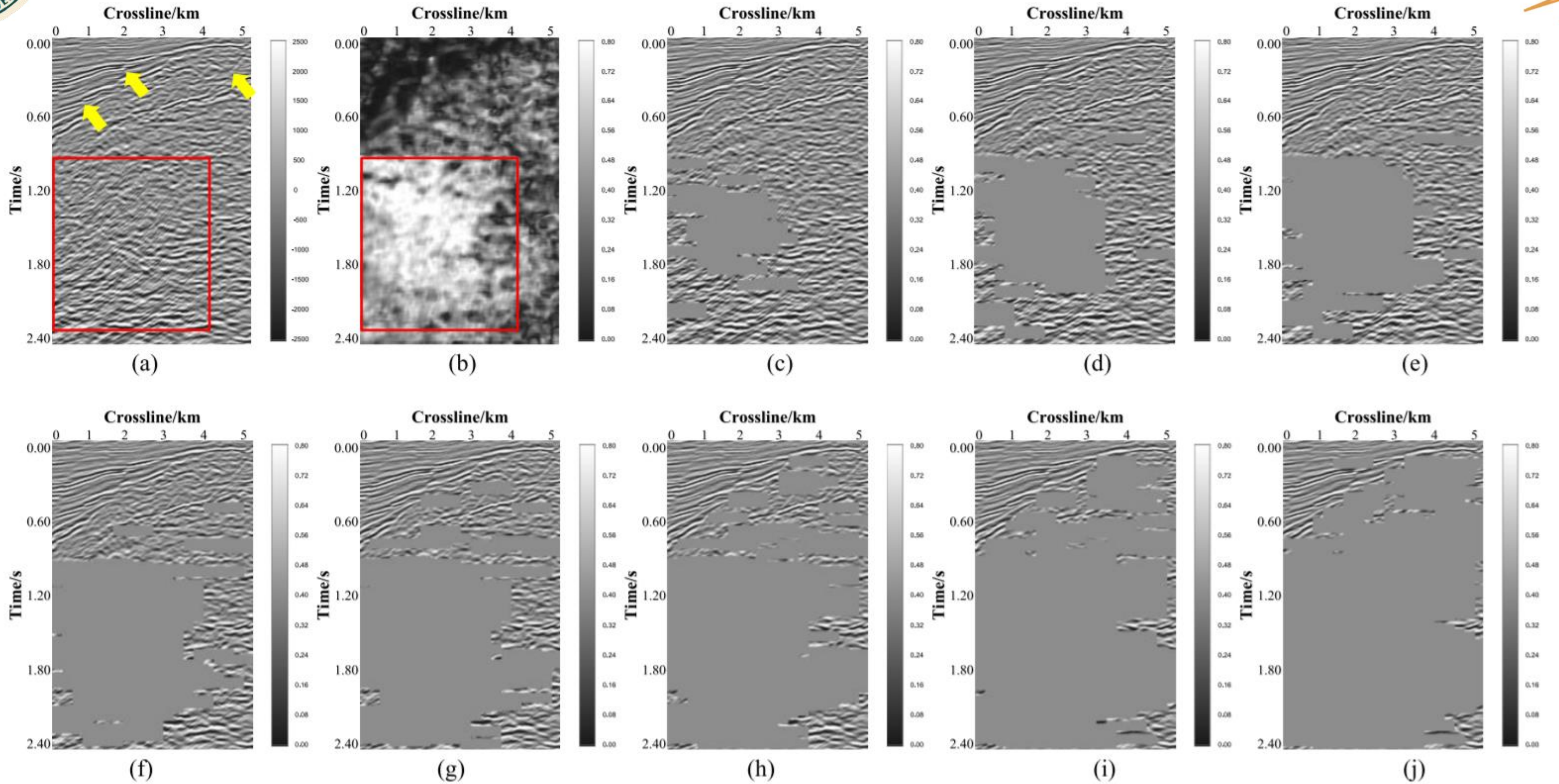
Gradient structure tensor

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

SVD $C = \frac{2\lambda_2(\lambda_2 - \lambda_3)}{(\lambda_1 + \lambda_2)(\lambda_2 + \lambda_3)}$

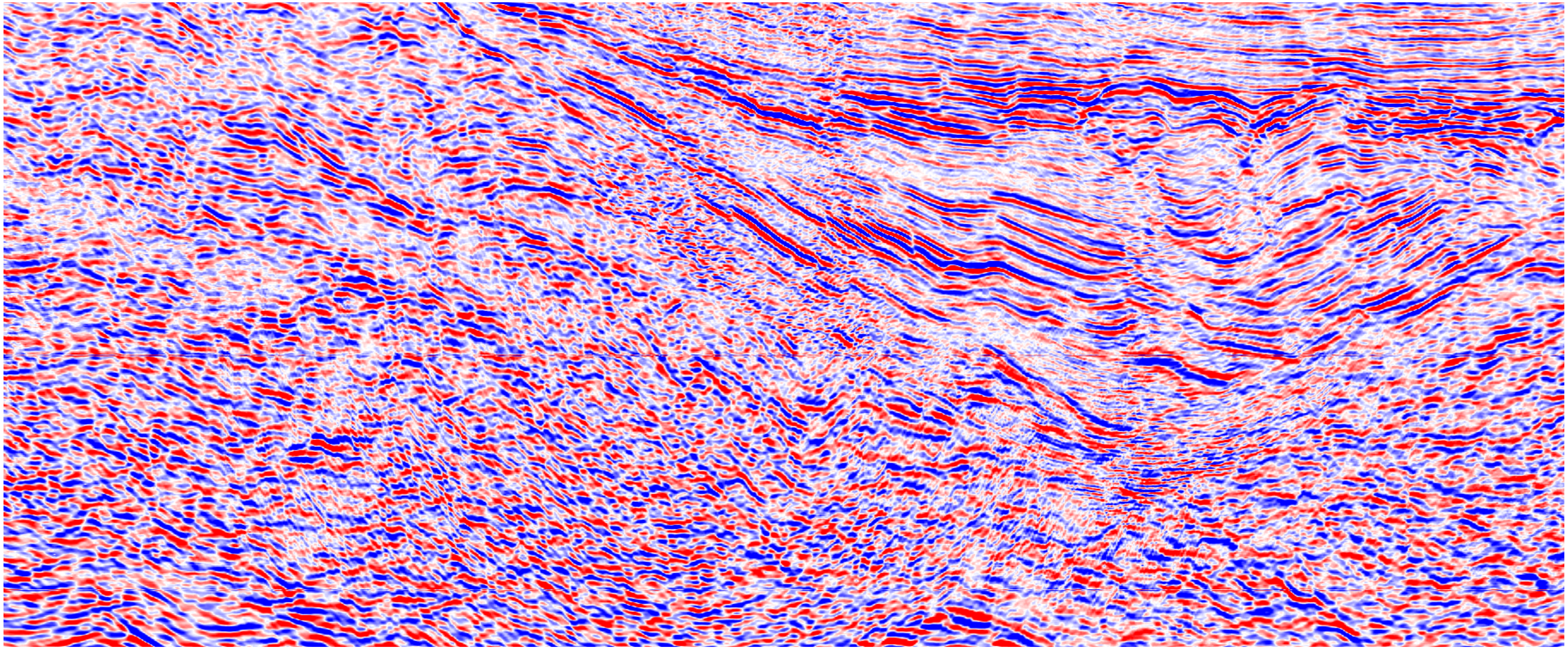
Flow chart of selecting training samples.

DEEP LEARNING-BASED METHODS



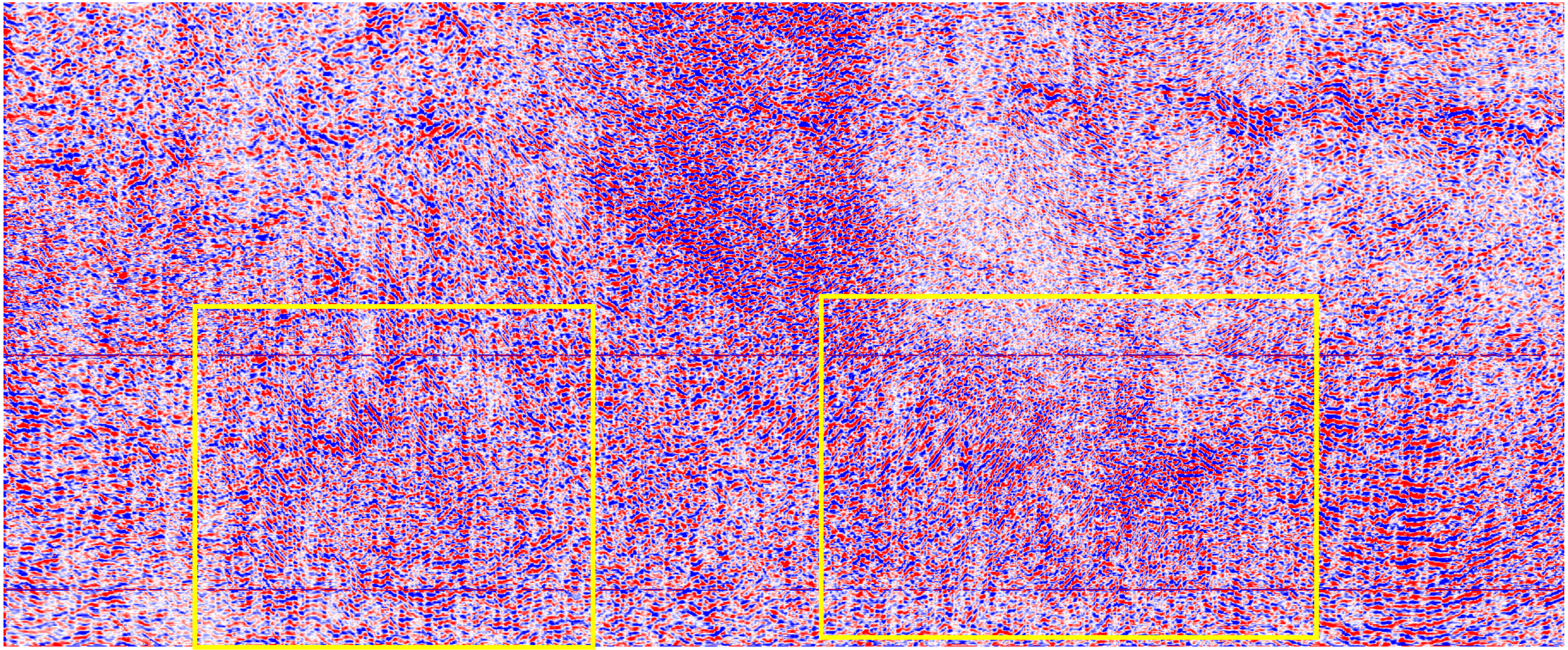
Seismic sections obtained by different fault confidence thresholds.

DEEP LEARNING-BASED METHODS



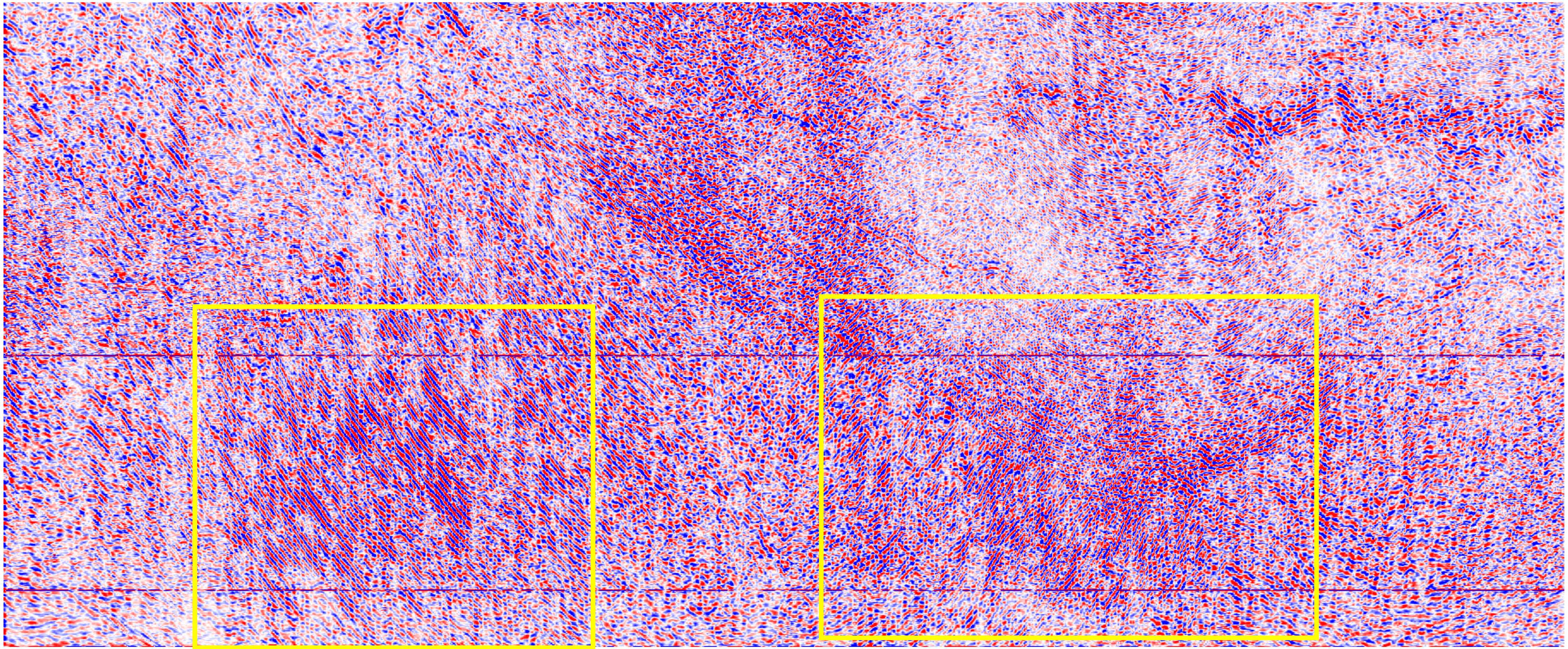
Original Inline section.

DEEP LEARNING-BASED METHODS



Removed arc-like imaging noise by the conventional method.

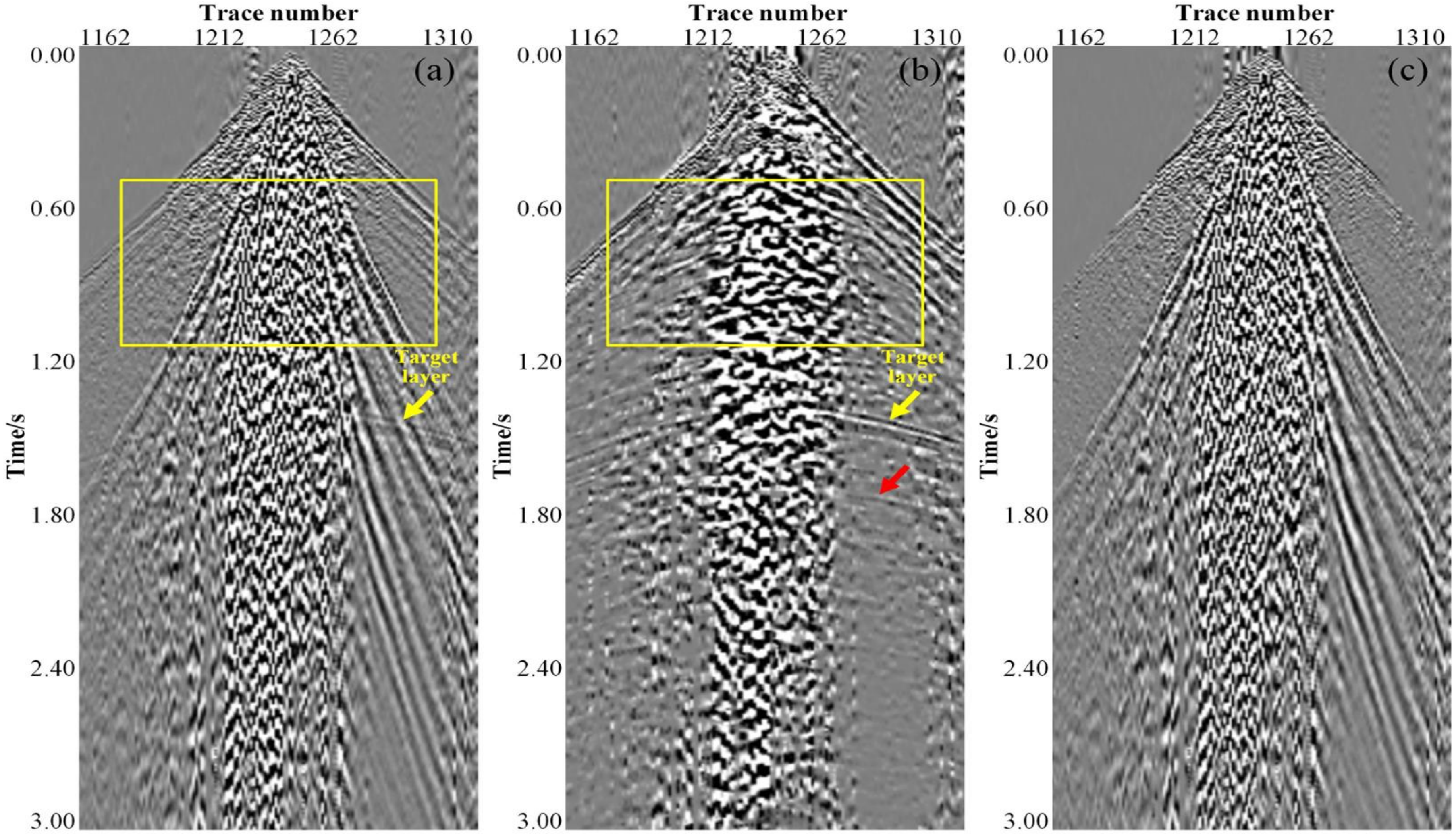
DEEP LEARNING-BASED METHODS



Removed arc-like imaging noise by our method.

Cited by Yilmaz, Öz in his new book: Land seismic case studies for near-surface modeling and subsurface imaging, 2021.

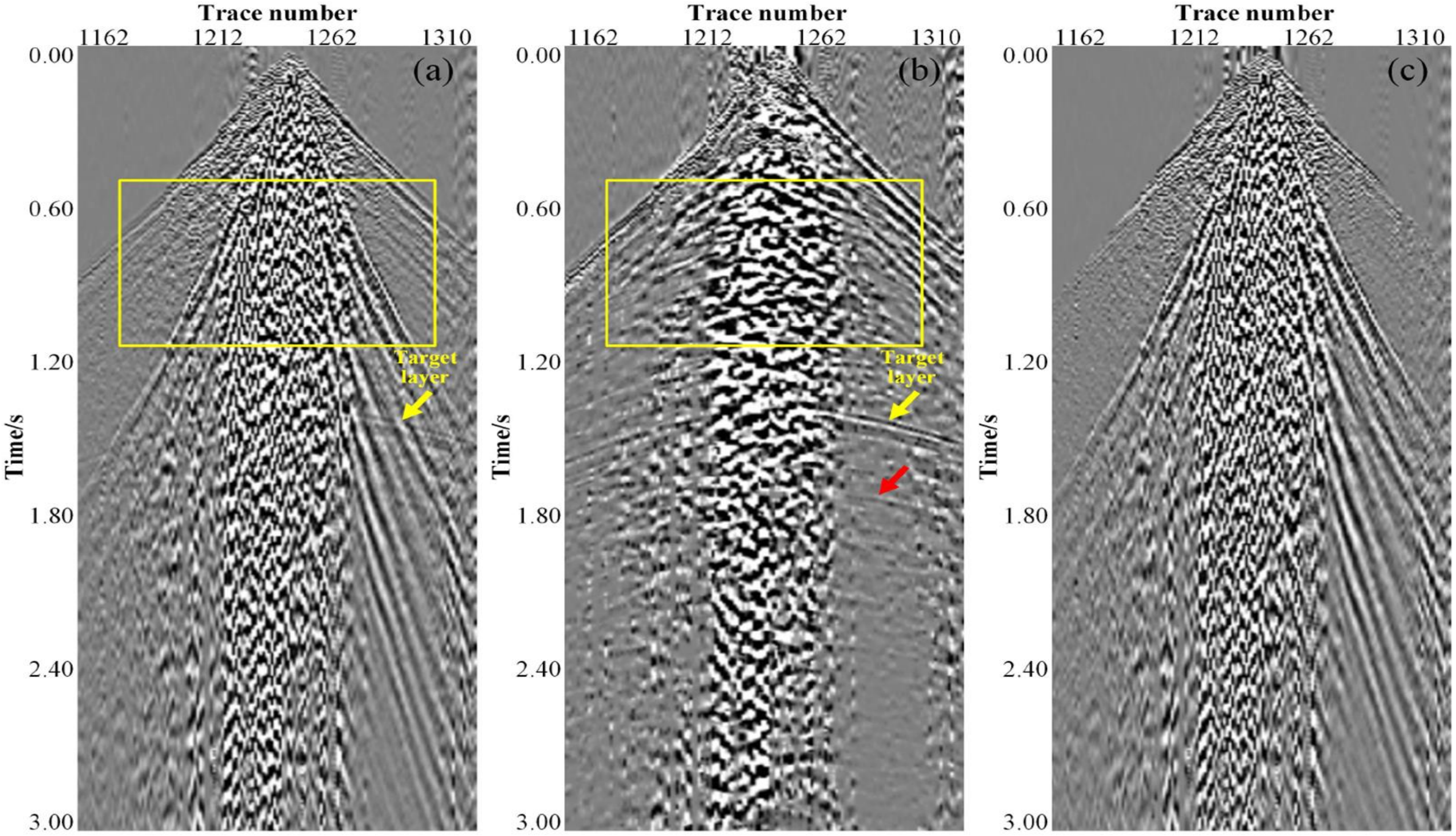
Our concerns



Prestack data is hard to train using networks.

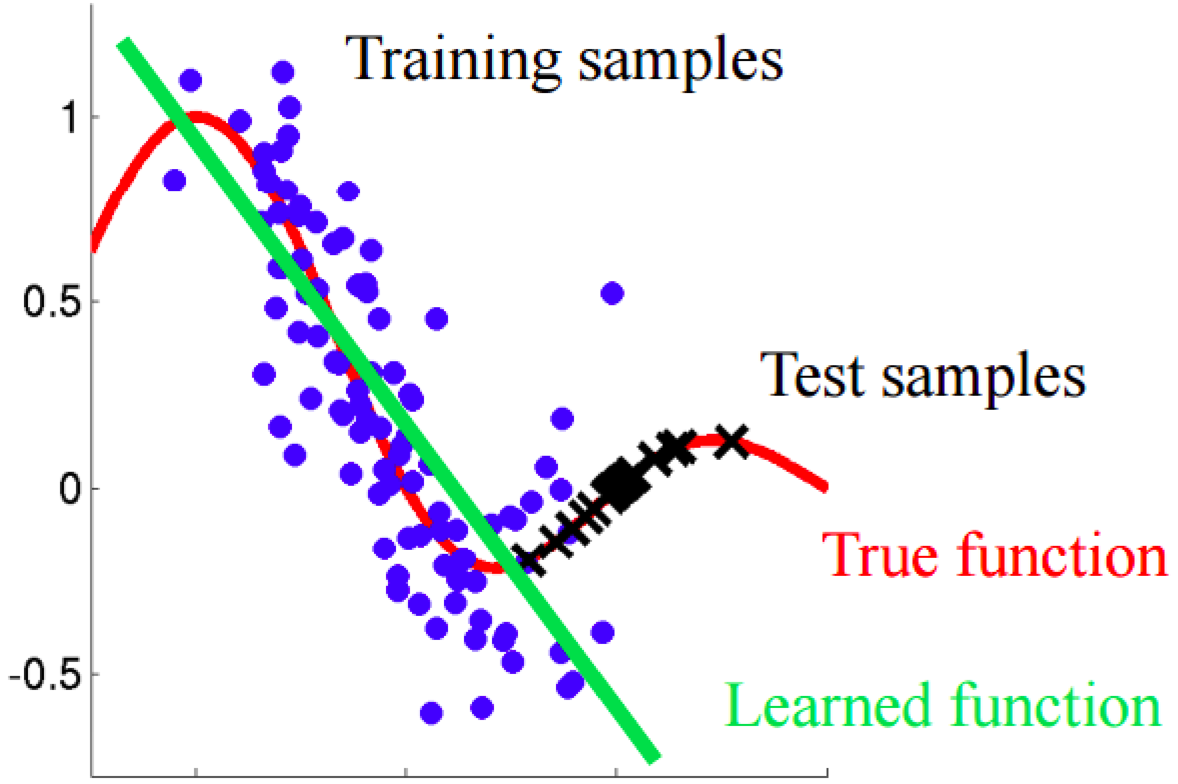
Our concerns

•**Can deep learning be applied to prestack strong scattered noise suppression? Especially in near-offset land seismic data.**



Our concerns

•**Can deep learning be applied to prestack strong scattered noise suppression? Especially in near-offset land seismic data.**

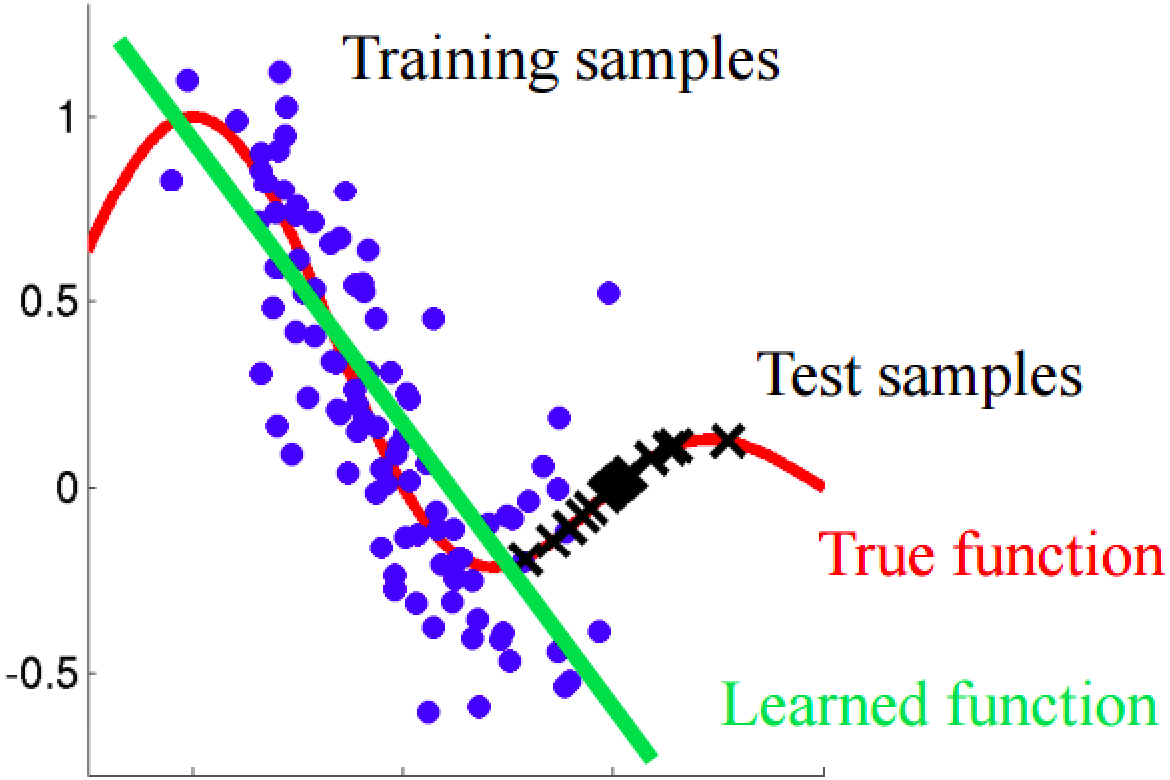


Covariate Shift

Our concerns

•**Can deep learning be applied to prestack strong scattered noise suppression? Especially in near-offset land seismic data.**

•**What domain is better for prestack data using deep learning?**



Covariate Shift

Our concerns

•Can deep learning be applied to prestack strong scattered noise suppression? Especially in near-offset land seismic data.

•What domain is better for deep learning?

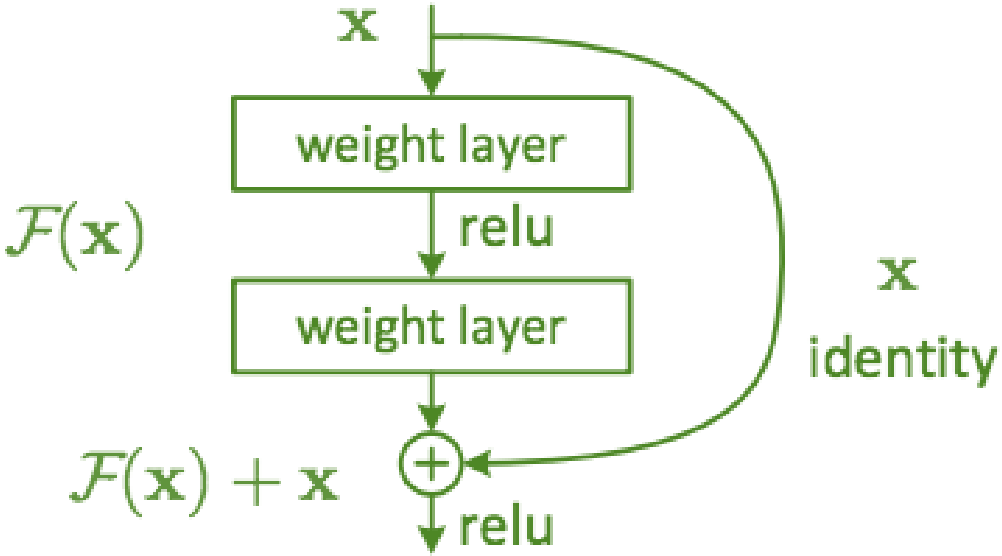
•Learning noise or useful signals with the network?

Our concerns

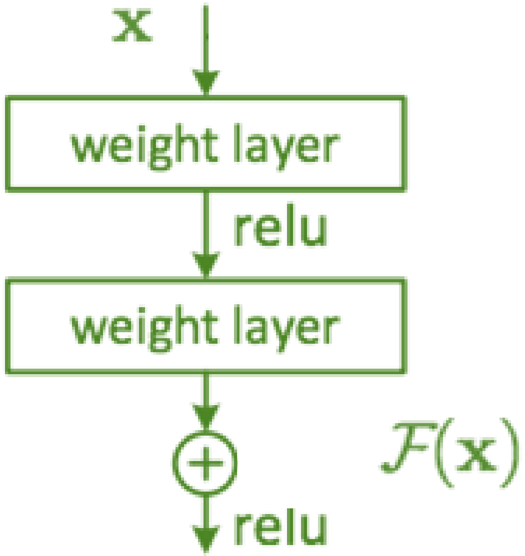
- Can deep learning be applied to prestack strong scattered noise suppression? Especially in near-offset land seismic data.

- What domain is better for deep learning?

- Learning noise or useful signals with the network?



Residual learning.



Direct signal learning.

OUTLINE

- Introduction
- Model and network training
- Instances on Synthetic data and real Data
- Conclusion
- Acknowledgement

MODEL FORMULATION

We model the seismic data, denoted by a vector $\mathbf{y} \in \psi = \mathbb{R}^m$, as a superposition of reflections and noise:

$$\mathbf{y} = \mathbf{x} + \mathbf{n},$$

where $\mathbf{x} \in \chi = \mathbb{R}^m$ represents useful signals and $\mathbf{n}$ represents scattered and random noise.

MODEL FORMULATION

We model the seismic data, denoted by a vector $\mathbf{y} \in \psi = \mathbb{R}^m$, as a superposition of reflections and noise:

$$\mathbf{y} = \mathbf{x} + \mathbf{n},$$

where $\mathbf{x} \in \chi = \mathbb{R}^m$ represents useful signals and $\mathbf{n}$ represents scattered and random noise.

Given a large number of training sample pairs containing noisy input and clean labels

$$\left(\mathrm{y}_i, \mathrm{x}_i\right) \sim \left(\mathrm{Y}, \mathrm{X}\right) = \left(\mathrm{X} + \mathrm{N}, \mathrm{X}\right), \quad i = 1, \ldots, K$$

MODEL FORMULATION

We model the seismic data, denoted by a vector $\mathbf{y} \in \psi = \mathbb{R}^m$, as a superposition of reflections and noise:

$$\mathbf{y} = \mathbf{x} + \mathbf{n},$$

where $\mathbf{x} \in \chi = \mathbb{R}^m$ represents useful signals and $\mathbf{n}$ represents scattered and random noise.

Given a large number of training sample pairs containing noisy input and clean labels

$$\left(\mathrm{y}_i, \mathrm{x}_i\right) \sim \left(\mathrm{Y}, \mathrm{X}\right) = \left(\mathrm{X} + \mathrm{N}, \mathrm{X}\right), \quad i = 1, \ldots, K$$

or $$\left(\mathrm{y}_i, \mathrm{x}_i\right) \sim \left(\mathrm{Y}, \mathrm{X}\right) = \left(\mathrm{X} + \mathrm{N}, \mathrm{N}\right), \quad i = 1, \ldots, K$$

MODEL FORMULATION

We model the seismic data, denoted by a vector $\mathrm{y} \in \psi = \mathbb{R}^m$, as a superposition of reflections and noise:

$$\mathbf{y} = \mathbf{x} + \mathbf{n},$$

where $\mathbf{x} \in \chi = \mathbb{R}^m$ represents useful signals and $\mathbf{n}$ represents scattered and random noise.

Given a large number of training sample pairs containing noisy input and clean labels

$$\left(\mathrm{y}_i, \mathrm{x}_i\right) \sim \left(\mathrm{Y}, \mathrm{X}\right) = \left(\mathrm{X} + \mathrm{N}, \mathrm{X}\right), \quad i = 1, \ldots, K$$

or $$\left(\mathrm{y}_i, \mathrm{x}_i\right) \sim \left(\mathrm{Y}, \mathrm{X}\right) = \left(\mathrm{X} + \mathrm{N}, \mathrm{N}\right), \quad i = 1, \ldots, K$$

By network training, deep learning aim to find the regression function

$$h^* = \underset{h}{\mathrm{argmin}}\ \mathbb{E}_{\mathrm{X},\mathrm{N}}\left\{L\left(h(\mathrm{X}+\mathrm{N}), \mathrm{X}\right)\right\},$$

or

$$h^* = \underset{h}{\mathrm{argmin}}\ \mathbb{E}_{\mathrm{X},\mathrm{N}}\left\{L\left(h(\mathrm{X}+\mathrm{N}), \mathrm{N}\right)\right\}.$$

MODEL FORMULATION

We model the seismic data, denoted by a vector $\mathbf{y} \in \psi = \mathbb{R}^m$, as a superposition of reflections and noise:

$$\mathbf{y} = \mathbf{x} + \mathbf{n},$$

where $\mathbf{x} \in \chi = \mathbb{R}^m$ represents useful signals and $\mathbf{n}$ represents scattered and random noise.

Given a large number of training sample pairs containing noisy input and clean labels

$$\left(\mathrm{y}_i, \mathrm{x}_i\right) \sim \left(\mathrm{Y}, \mathrm{X}\right) = \left(\mathrm{X} + \mathrm{N}, \mathrm{X}\right), \quad i = 1, \ldots, K$$

or $$\left(\mathrm{y}_i, \mathrm{x}_i\right) \sim \left(\mathrm{Y}, \mathrm{X}\right) = \left(\mathrm{X} + \mathrm{N}, \mathrm{N}\right), \quad i = 1, \ldots, K$$

By network training, deep learning aim to find the regression function

$$h^* = \underset{h}{\operatorname{argmin}}\, \mathbb{E}_{\mathrm{X},\mathrm{N}} \left\{ L\left(h(\mathrm{X} + \mathrm{N}), \mathrm{X} \right) \right\},$$

or

$$h^* = \underset{h}{\operatorname{argmin}}\, \mathbb{E}_{\mathrm{X},\mathrm{N}} \left\{ L\left(h(\mathrm{X} + \mathrm{N}), \mathrm{N} \right) \right\}.$$

Pixel-wise mean square error is the loss function

$$L\left(h(\mathrm{X} + \mathrm{N}), \mathrm{X} \right) = \| h(\mathrm{X} + \mathrm{N}) - \mathrm{X} \|_2^2,$$

or

$$L\left(h(\mathrm{X} + \mathrm{N}), \mathrm{X} \right) = \| h(\mathrm{X} + \mathrm{N}) - \mathrm{N} \|_2^2 .$$

MODEL FORMULATION

Since the joint distribution function is unknown, the expectation is estimated by the empirical risk as follows

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| f_\theta\left(\mathrm{y}_i\right) - \mathrm{x}_i \right\|_2^2,$$

or

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| f_\theta\left(\mathrm{y}_i\right) - \left(\mathrm{y}_i - \mathrm{x}_i\right) \right\|_2^2.$$

MODEL FORMULATION

Since the joint distribution function is unknown, the expectation is estimated by the empirical risk as follows

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| f_\theta(\mathrm{y}_i) - \mathrm{x}_i \right\|_2^2,$$

or

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| f_\theta(\mathrm{y}_i) - (\mathrm{y}_i - \mathrm{x}_i) \right\|_2^2.$$

Training a network is to minimize the CNN paremeterized mapping

$$f_\theta : \psi \to \chi,$$

or

$$f_\theta : \psi \to \eta.$$

MODEL FORMULATION

Since the joint distribution function is unknown, the expectation is estimated by the empirical risk as follows

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| f_\theta(\mathrm{y}_i) - \mathrm{x}_i \right\|_2^2,$$

or

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| f_\theta(\mathrm{y}_i) - (\mathrm{y}_i - \mathrm{x}_i) \right\|_2^2.$$

Training a network is to minimize the CNN paremeterized mapping

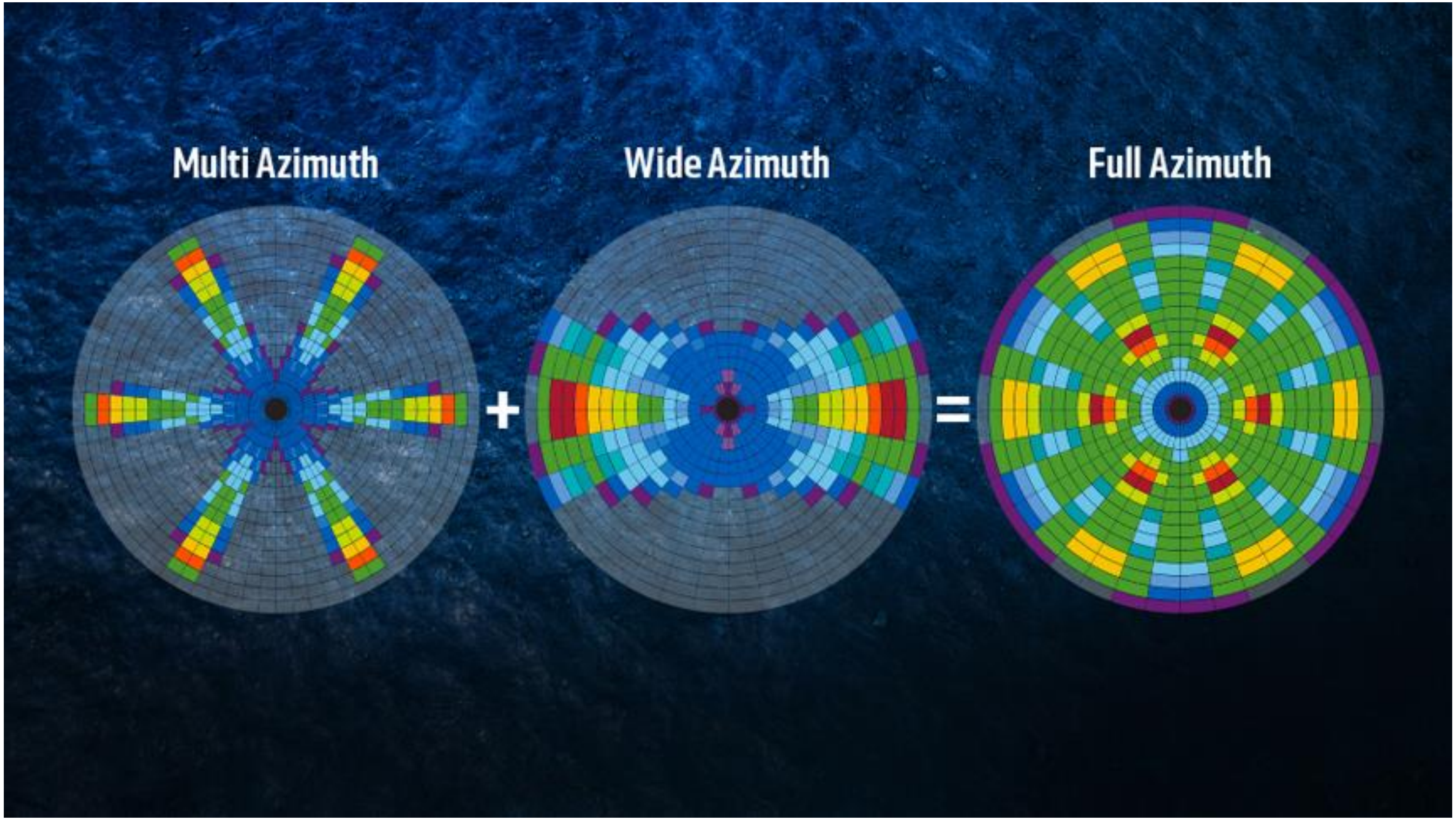
$$f_\theta : \psi \to \chi,$$

or

$$f_\theta : \psi \to \eta.$$

Can we find a better data sorting type for network training with fixed reflection distribution or noise distribution?

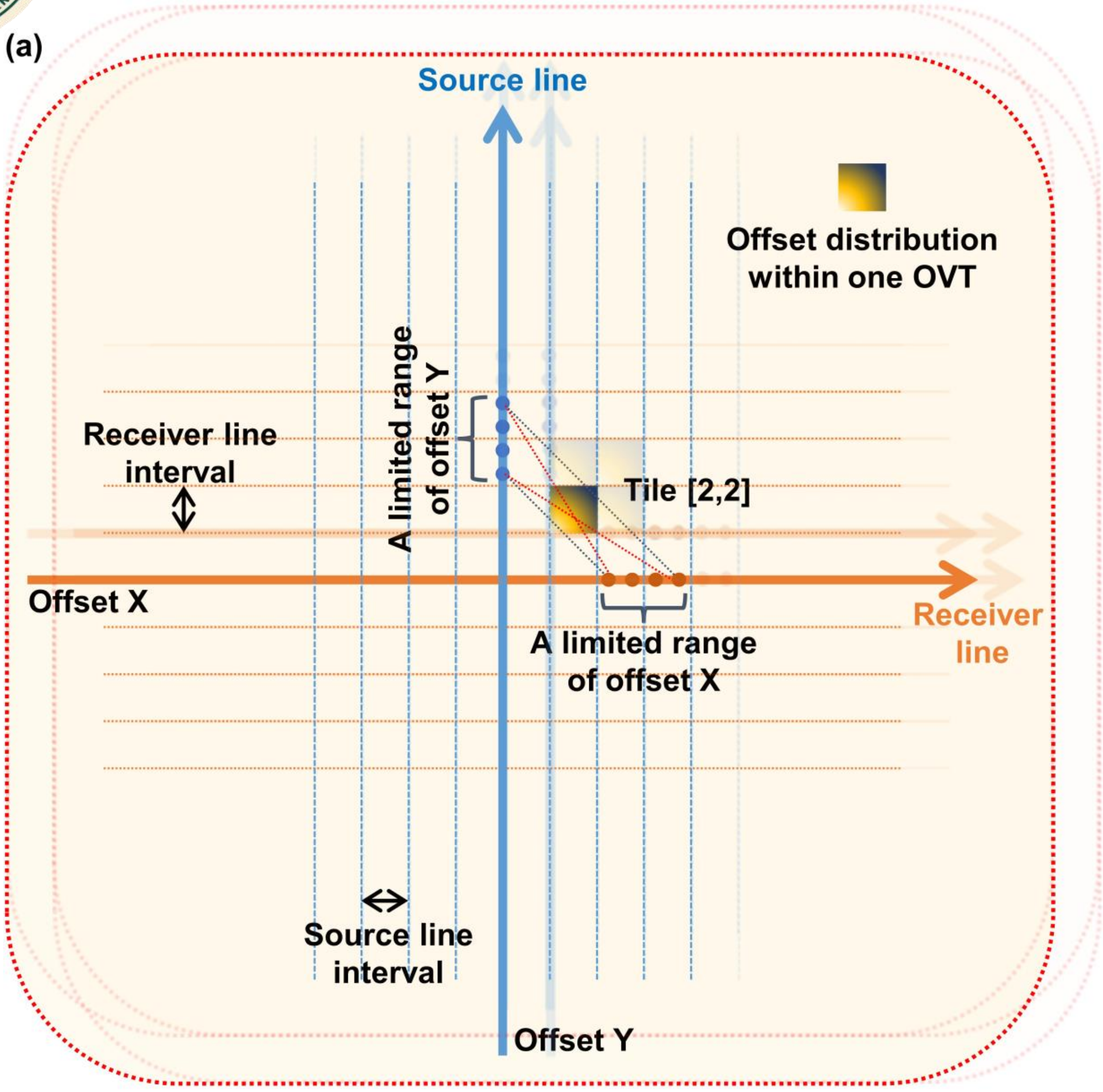
OFFSET VECTOR TILE



Cited from PGS

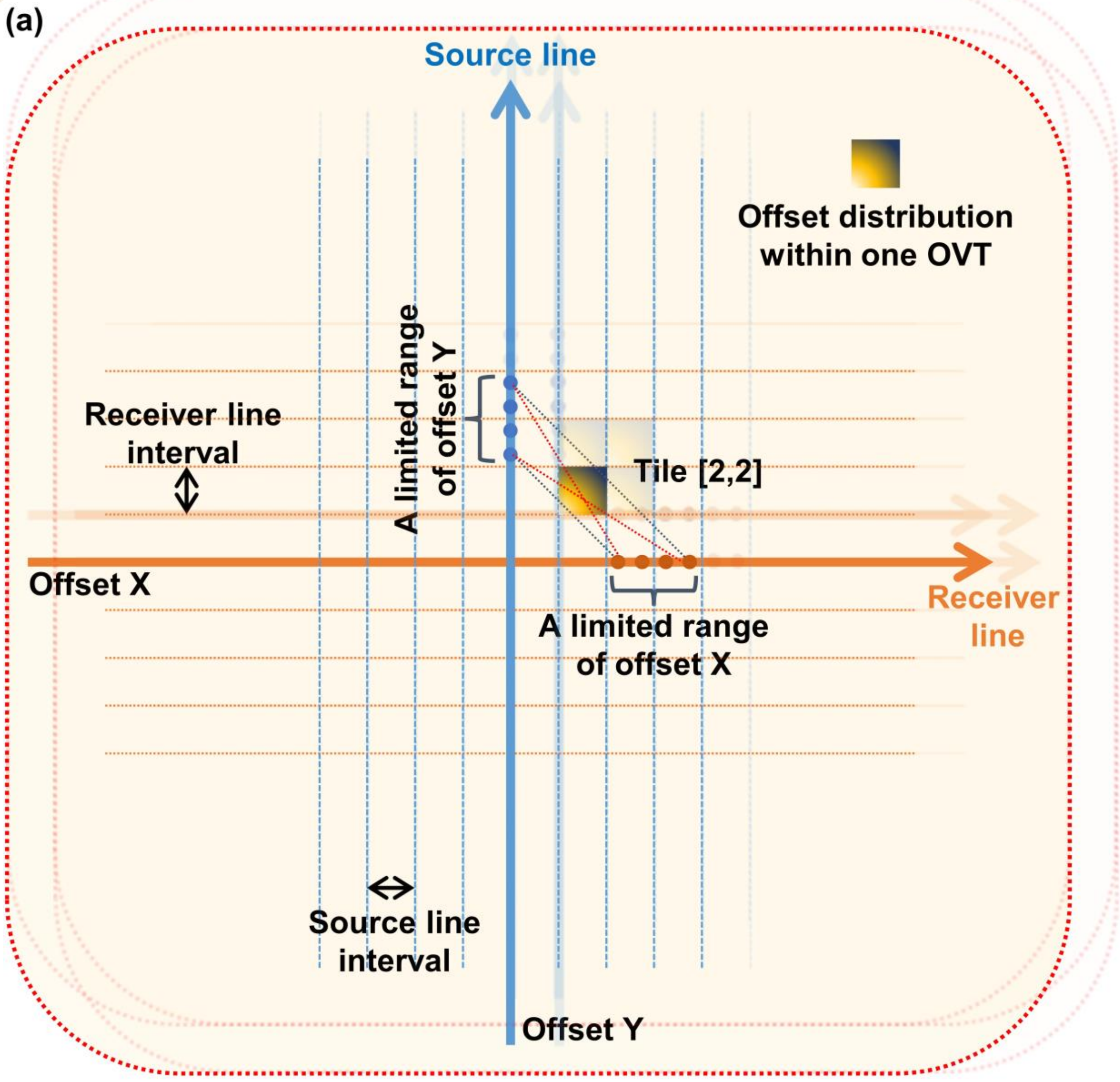
COMMON OFFSET VECTOR VOLUME

(a)



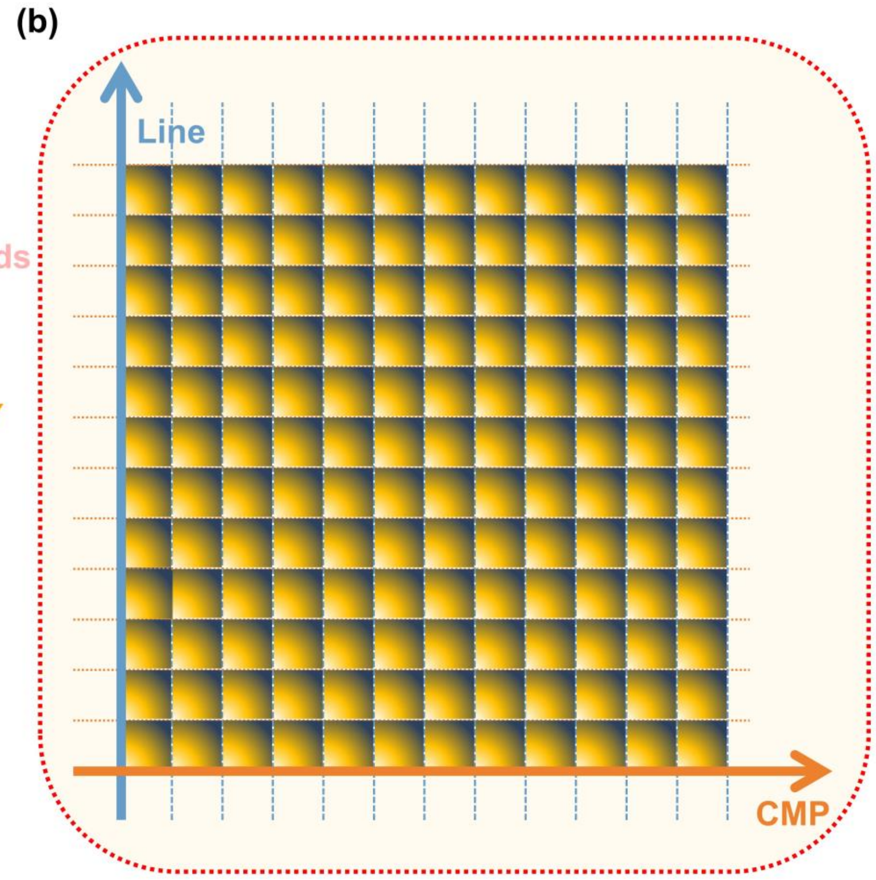
A cross spread

COMMON OFFSET VECTOR VOLUME



Different
cross-spreads

Patching
together



OVT gather with Tiles [2,2]
from all cross-spreads

A cross spread

Common offset vector volume

ADVANTAGES BASED ON OVT PROCESSING

- **Offsets and azimuths are relatively constant in the OVT domain, which is conducive to regularization and immigration processing.**

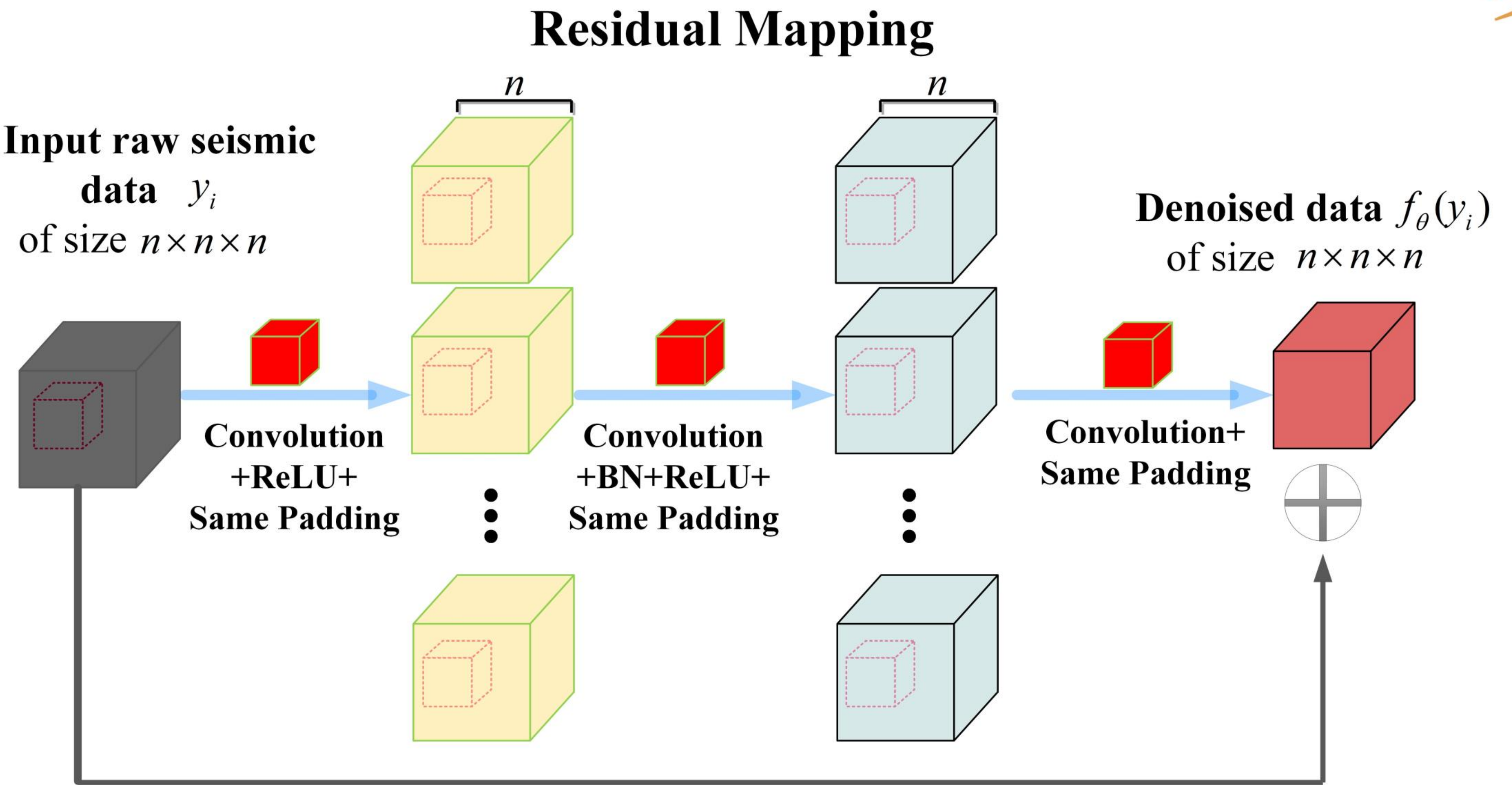
ADVANTAGES BASED ON OVT PROCESSING

- **Offsets and azimuths are relatively constant in the OVT domain, which is conducive to regularization and immigration processing.**

- **OVT is single fold coverage of the entire survey area with similar offsets and azimuths, thereby reducing the spatial discontinuity and laying the data foundation for network learning.**

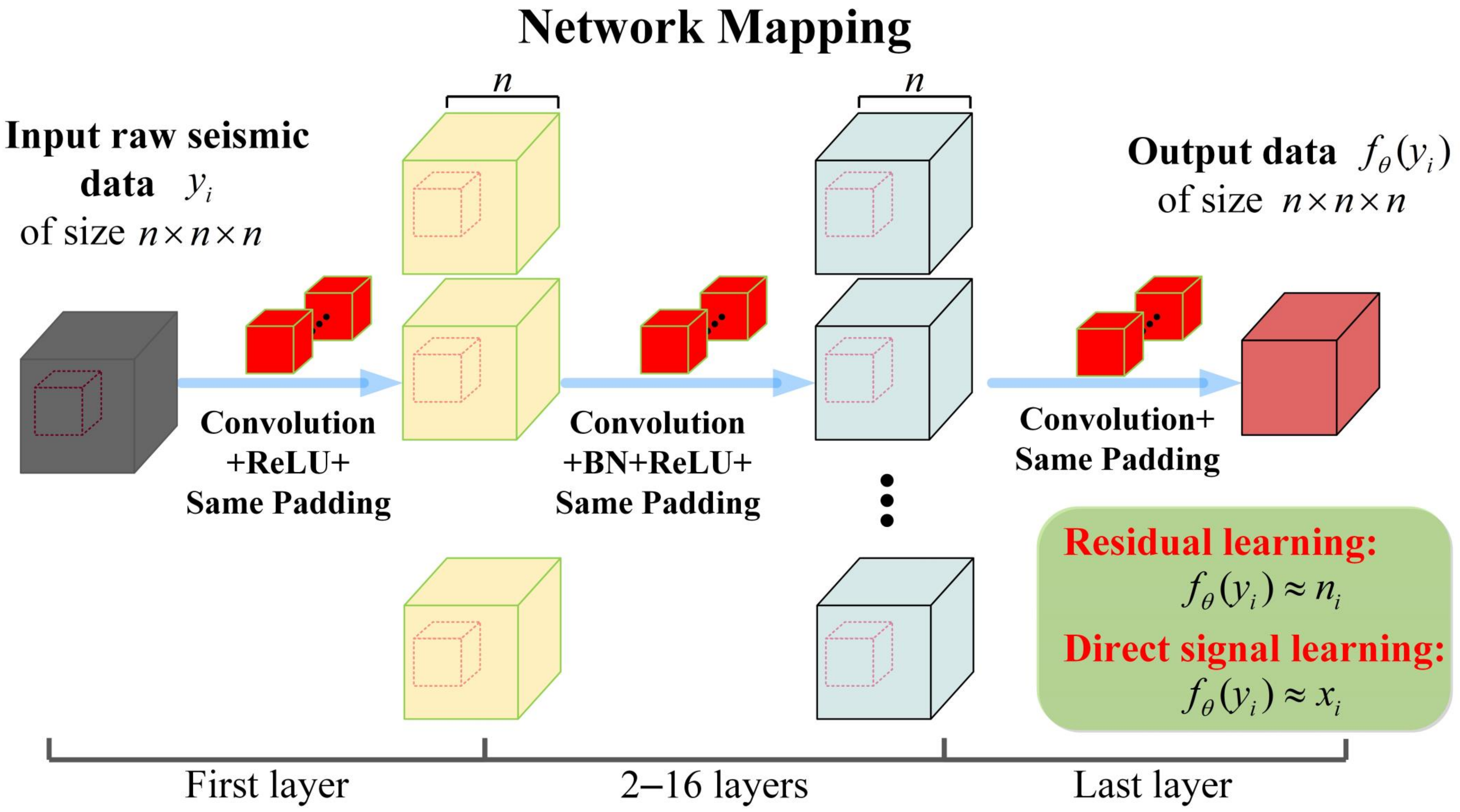
NETWORK ARCHITECTURE

NETWORK ARCHITECTURE



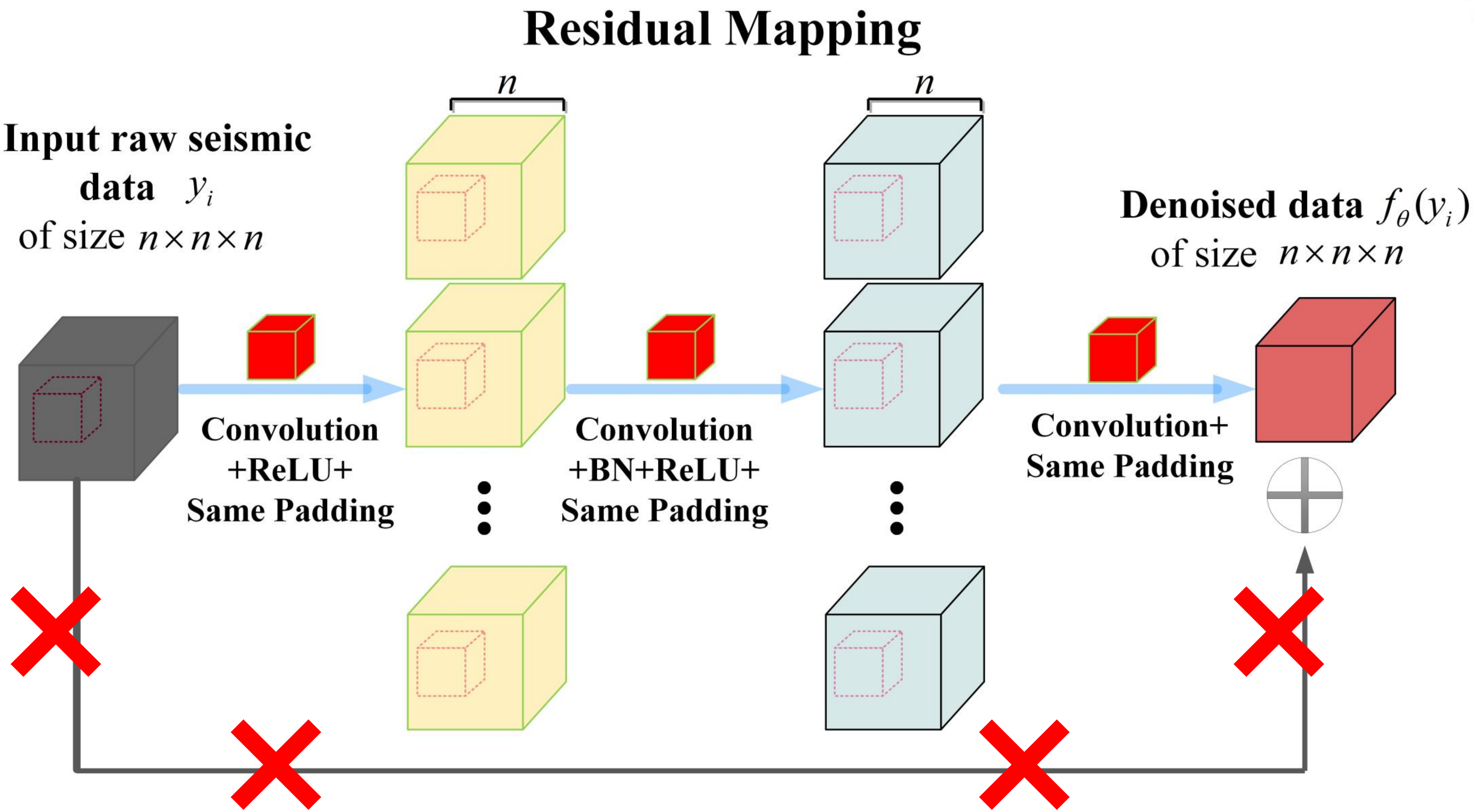
3D-DnCNN

NETWORK ARCHITECTURE



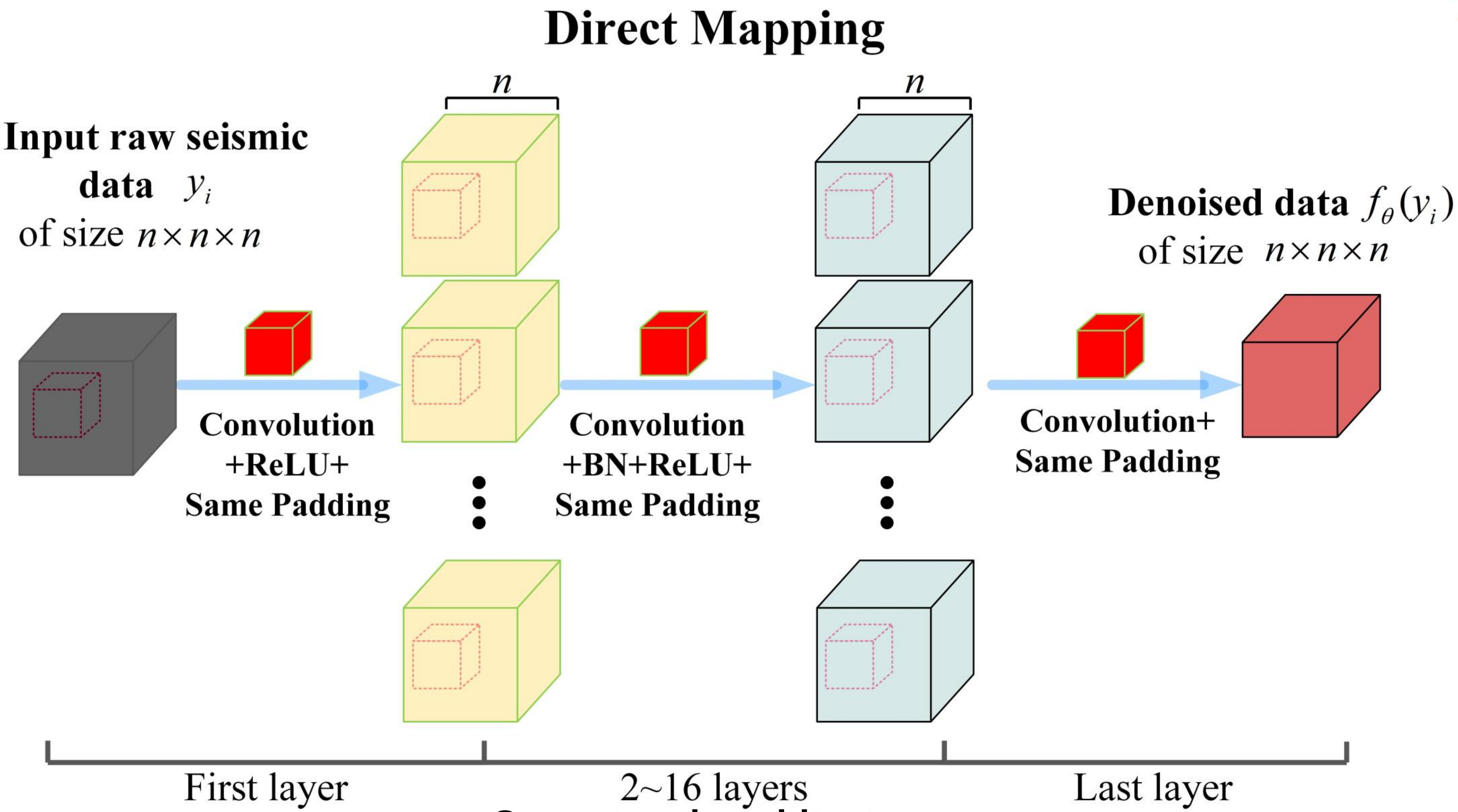
3D-CNN

NETWORK ARCHITECTURE



3D-DnCNN

NETWORK ARCHITECTURE



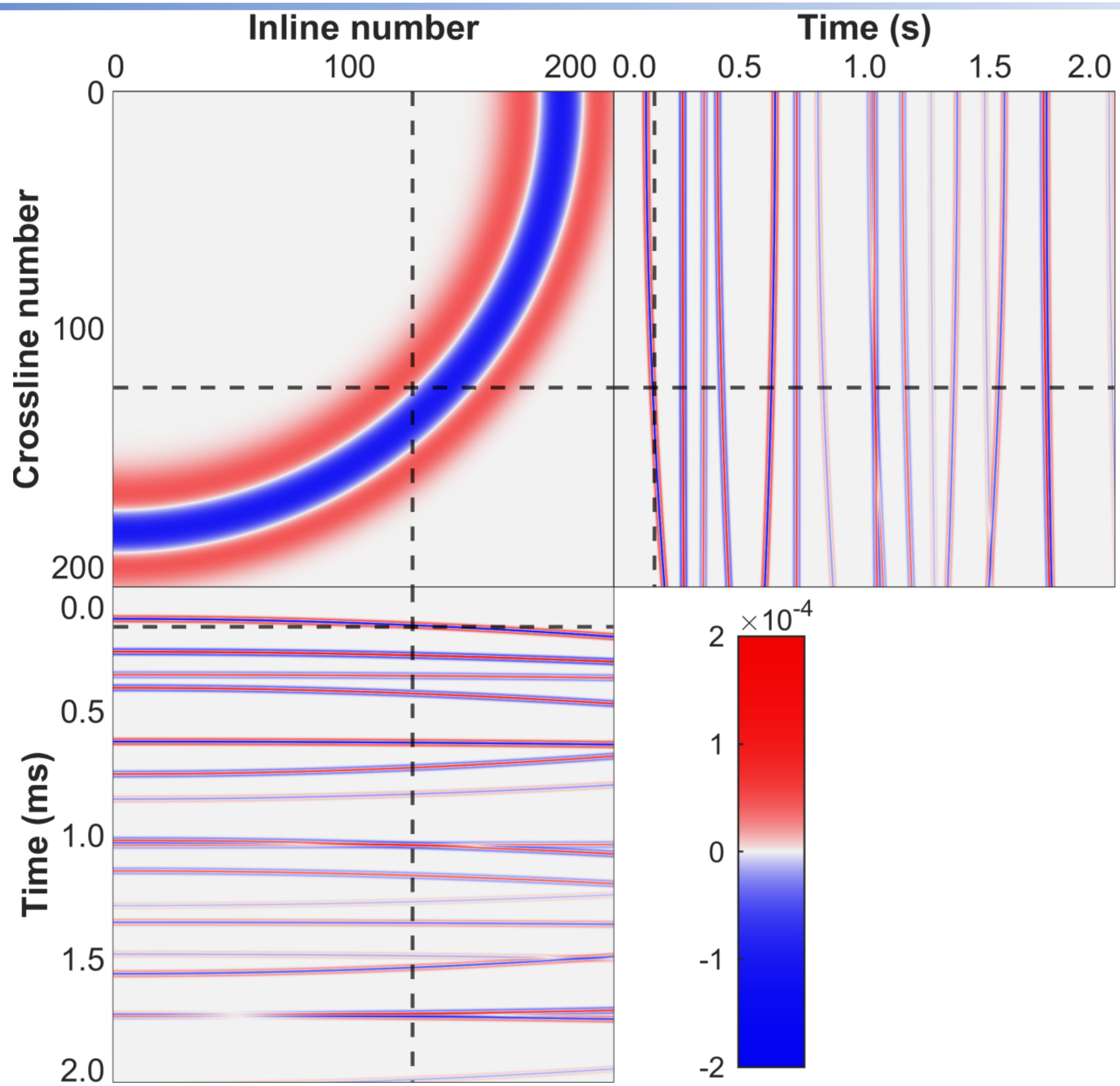
Our network architecture

OUTLINE

- Introduction
- Model and network training
- Instances on Synthetic data and real Data
- Conclusion
- Acknowledgement

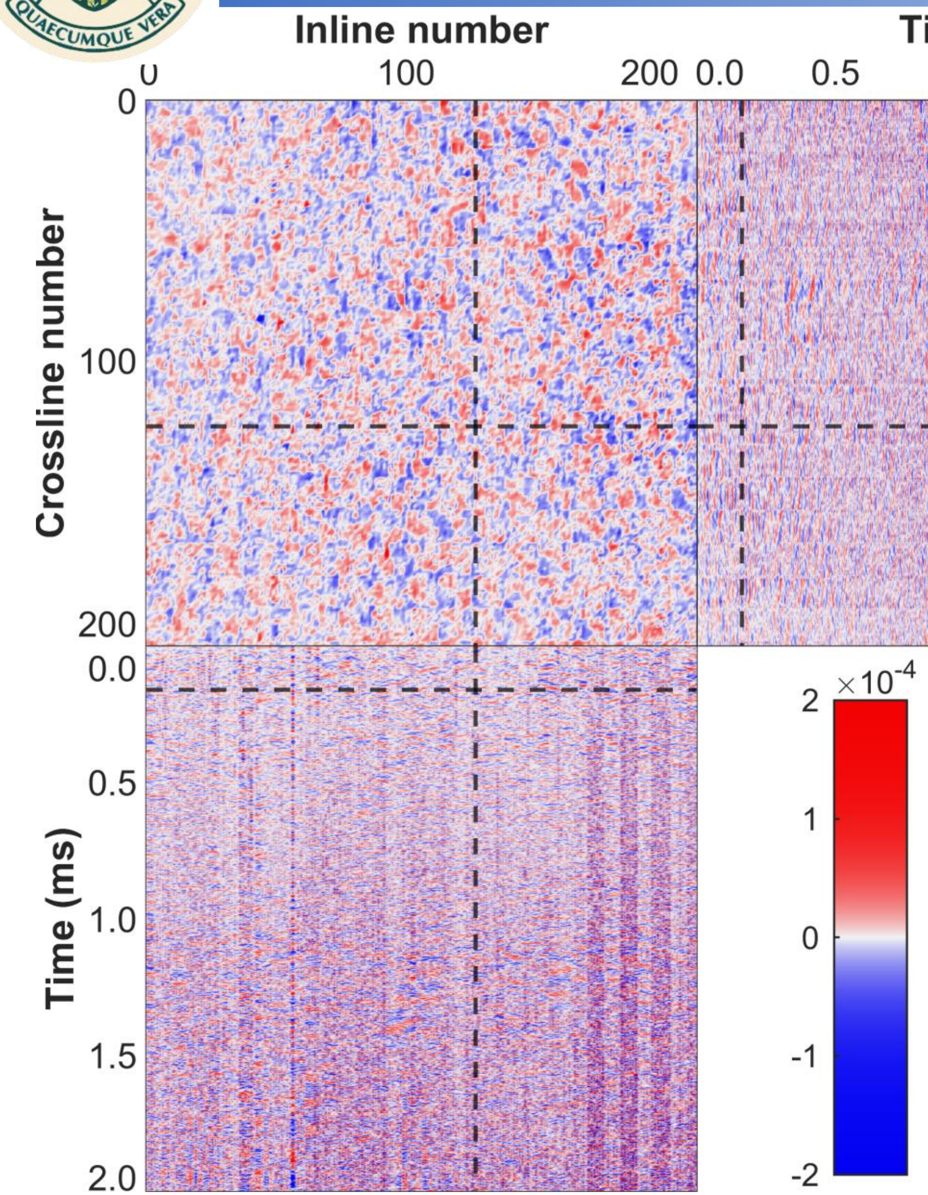
SYNTHETIC SEISMIC DATA

SYNTHETIC SEISMIC DATA



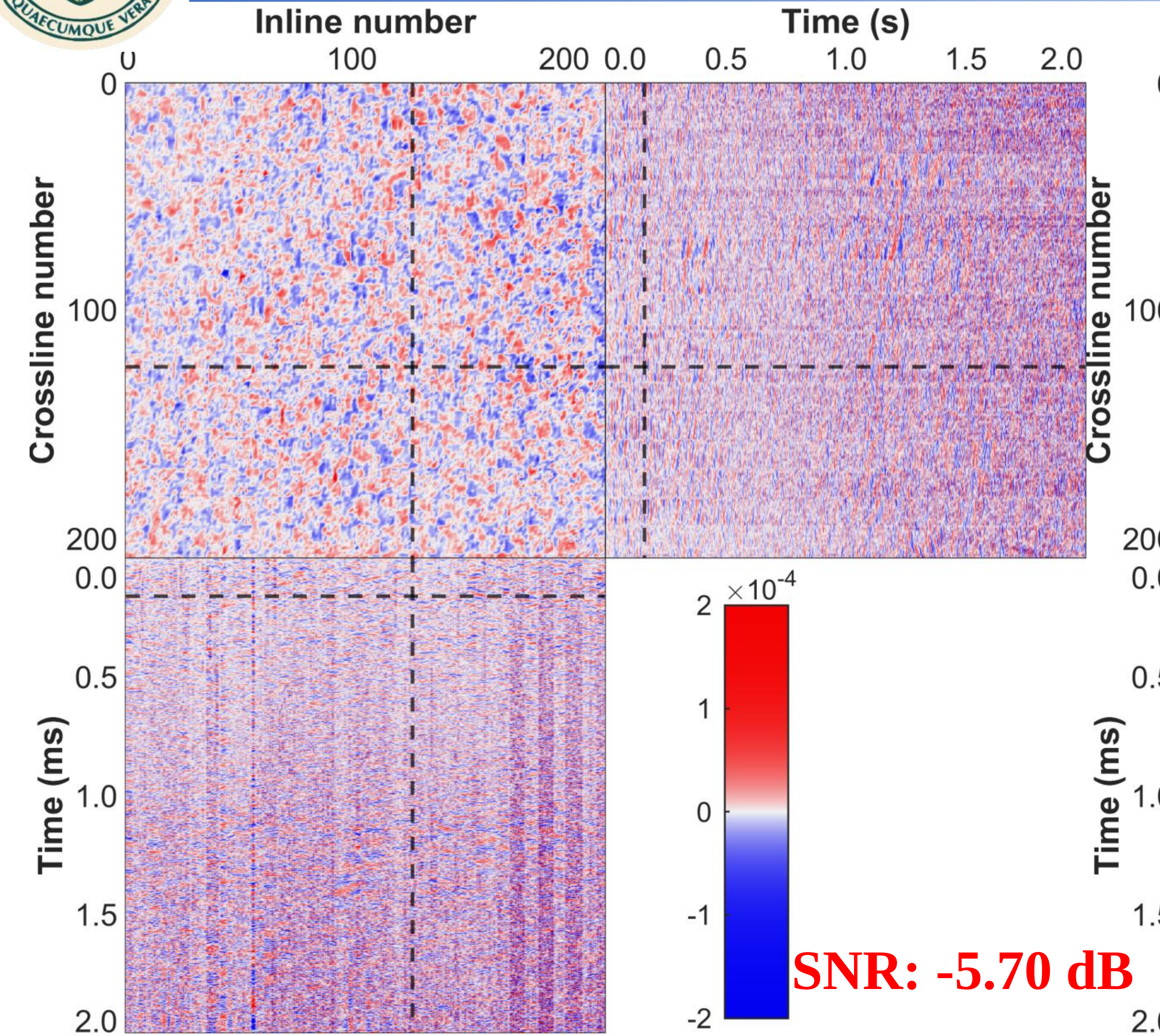
Training labels with offset of 4991

SYNTHETIC SEISMIC DATA

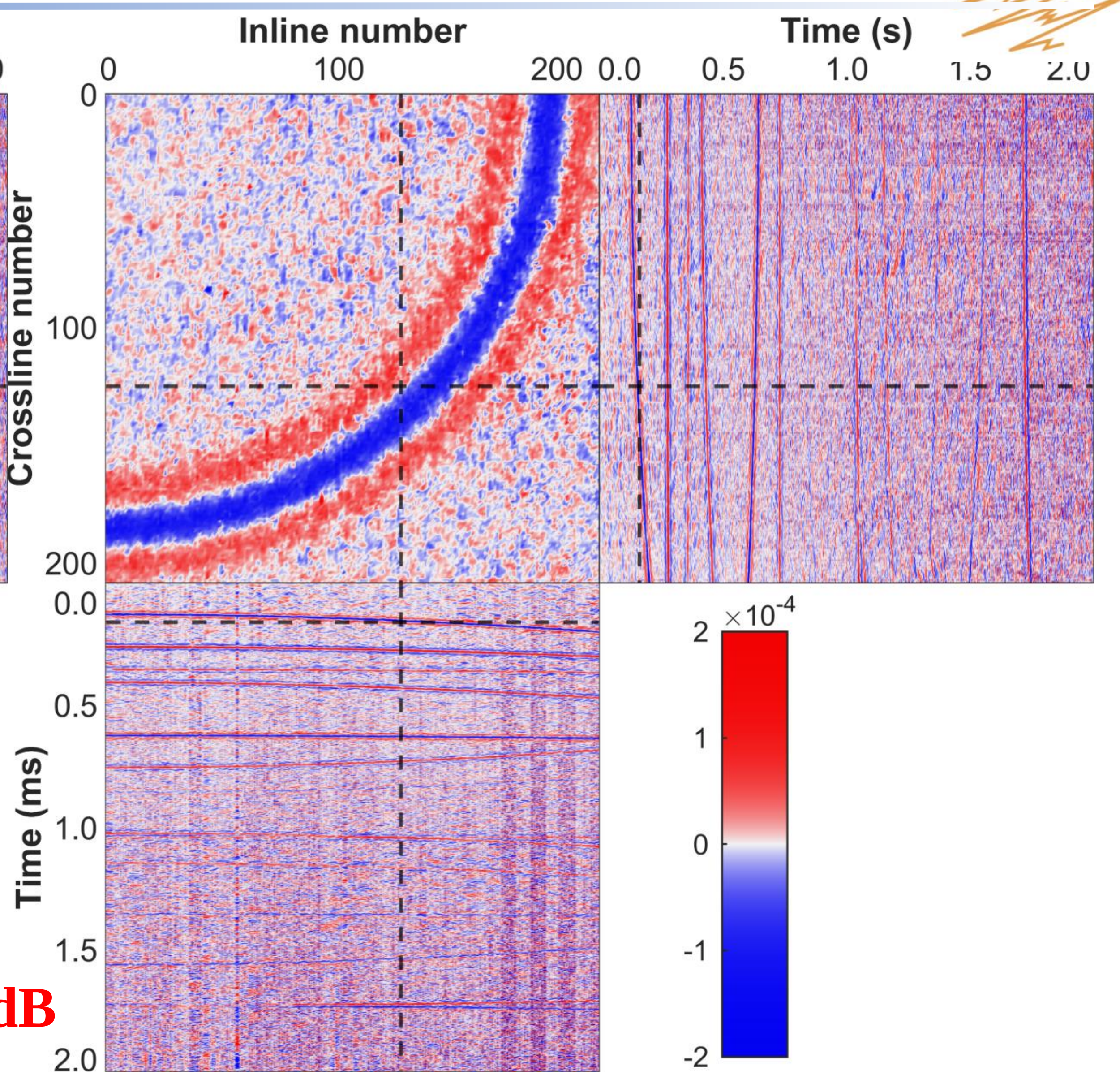


Field seismic noise with offset of 4991

SYNTHETIC SEISMIC DATA

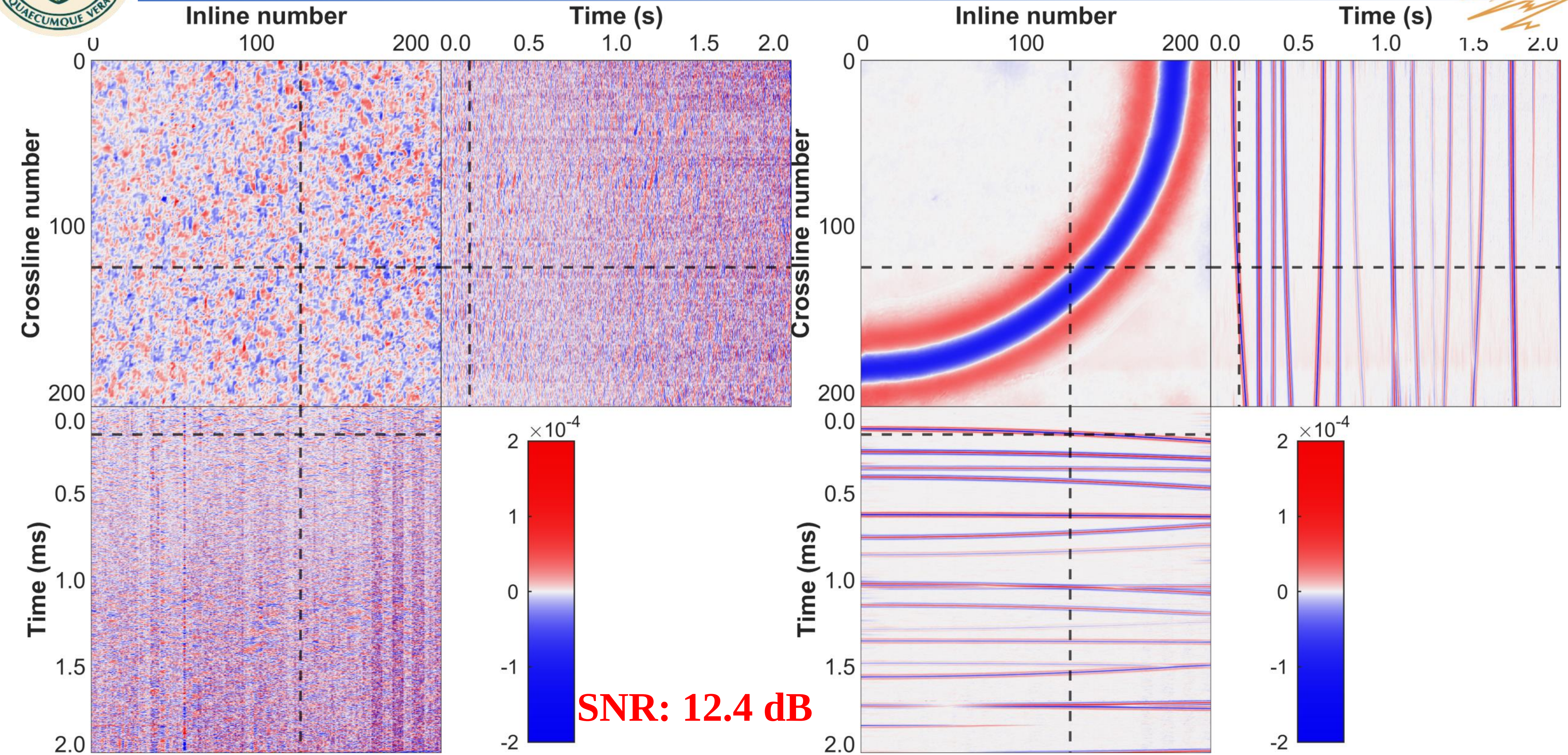


Field seismic noise with offset of 4991



Noisy seismic noise with offset of 4991

SYNTHETIC SEISMIC DATA

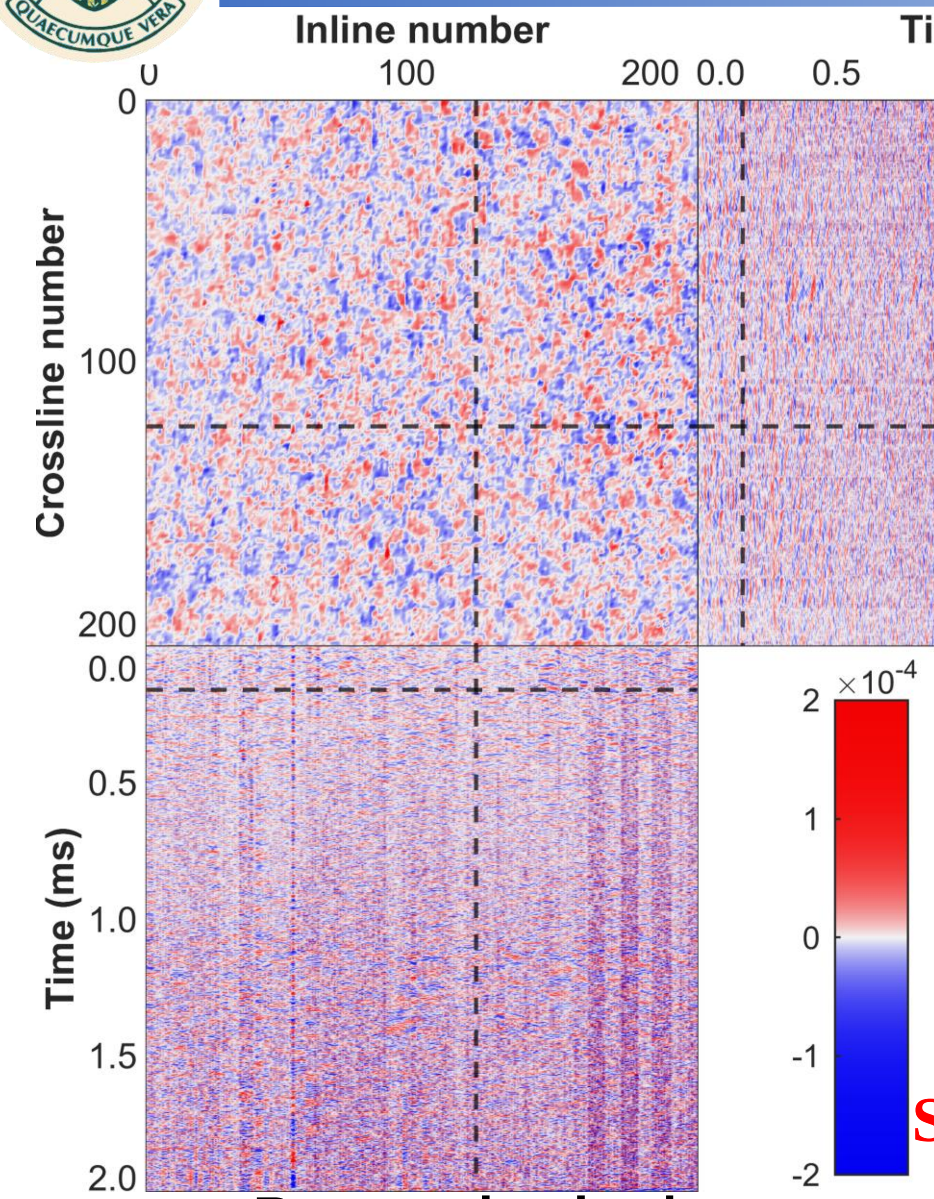


SNR: 12.4 dB

Removed noise by residual learning

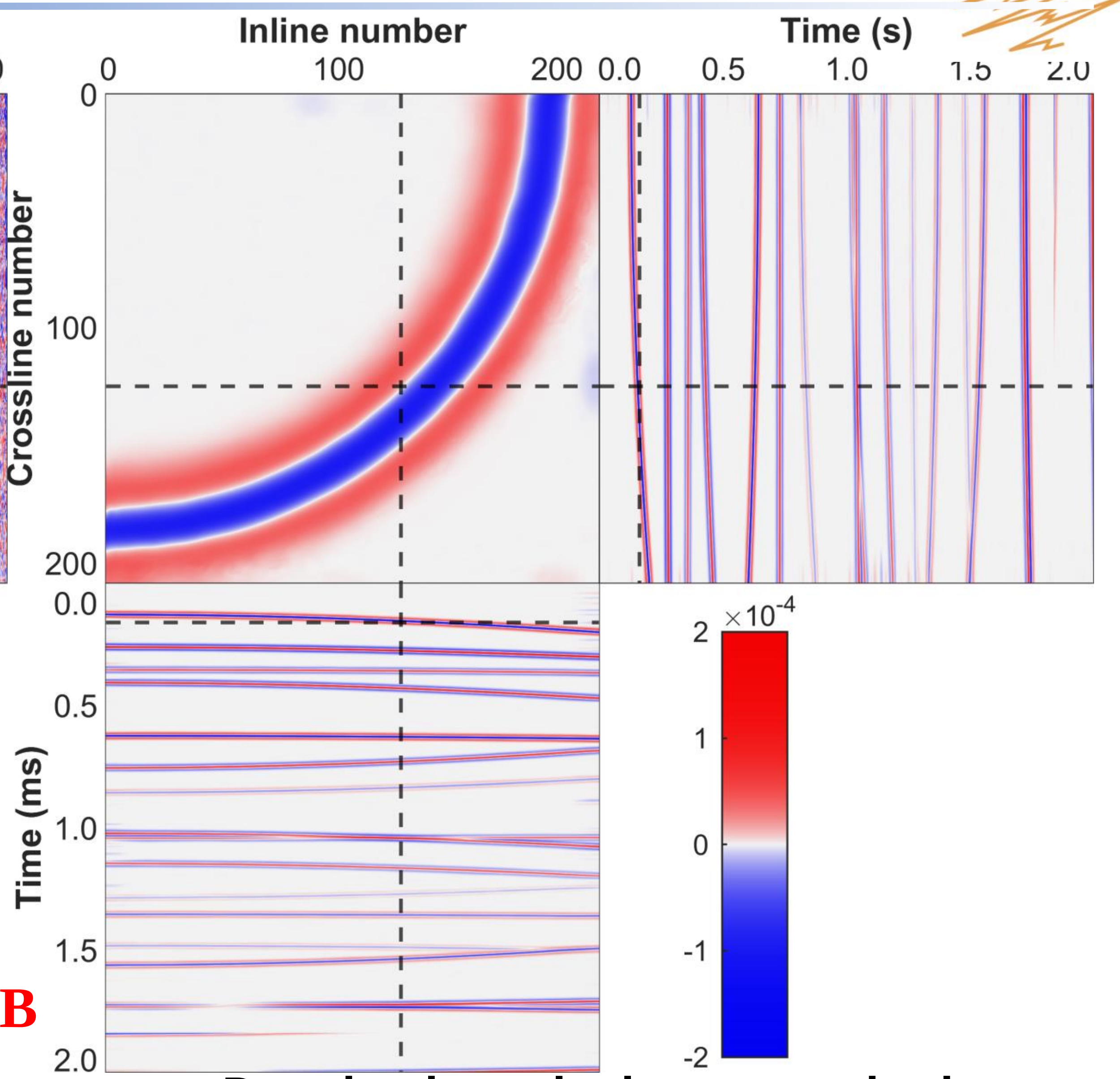
Denoised results by residual learning

SYNTHETIC SEISMIC DATA



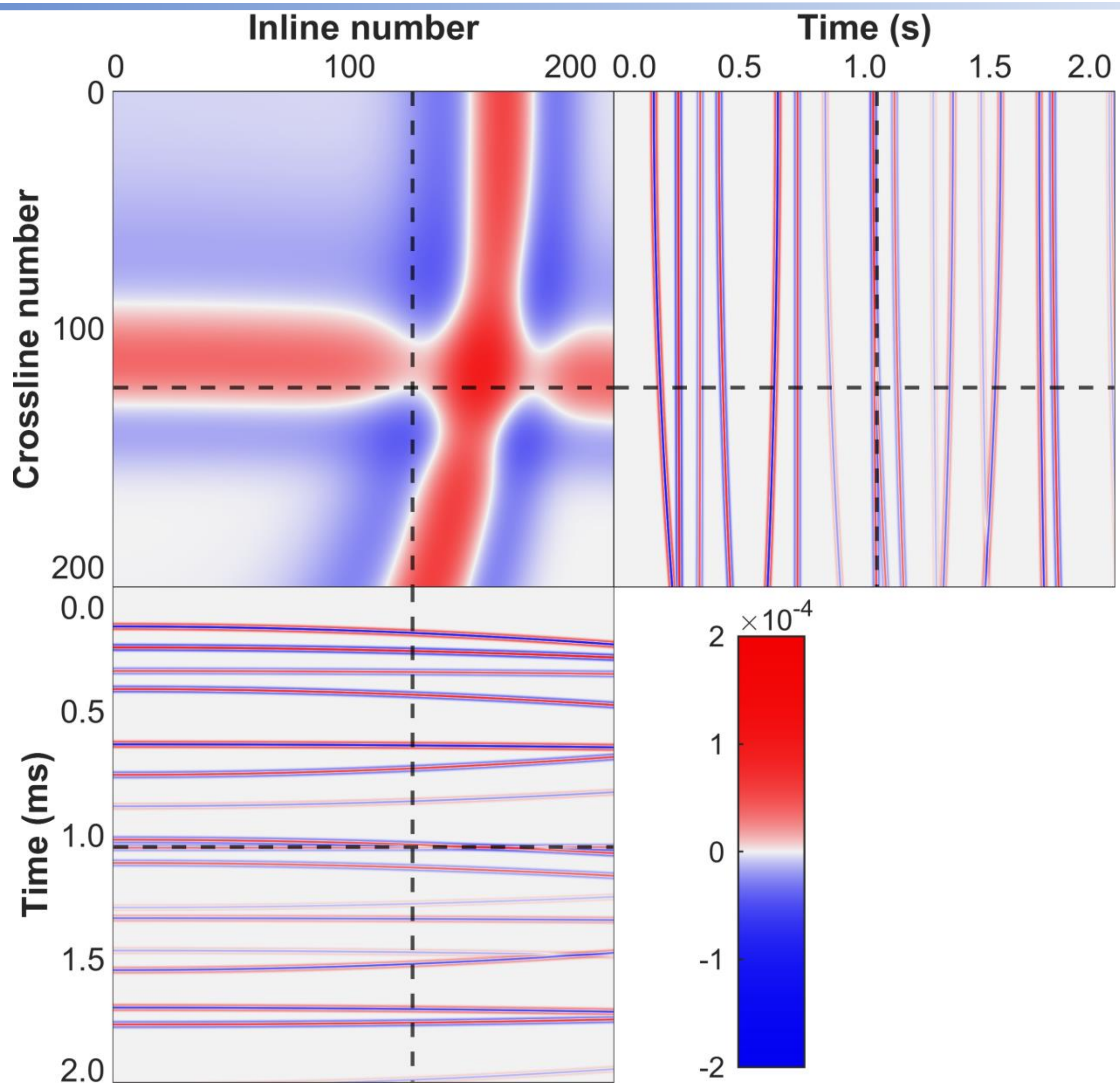
SNR: 13.2 dB

Removed noise by our method



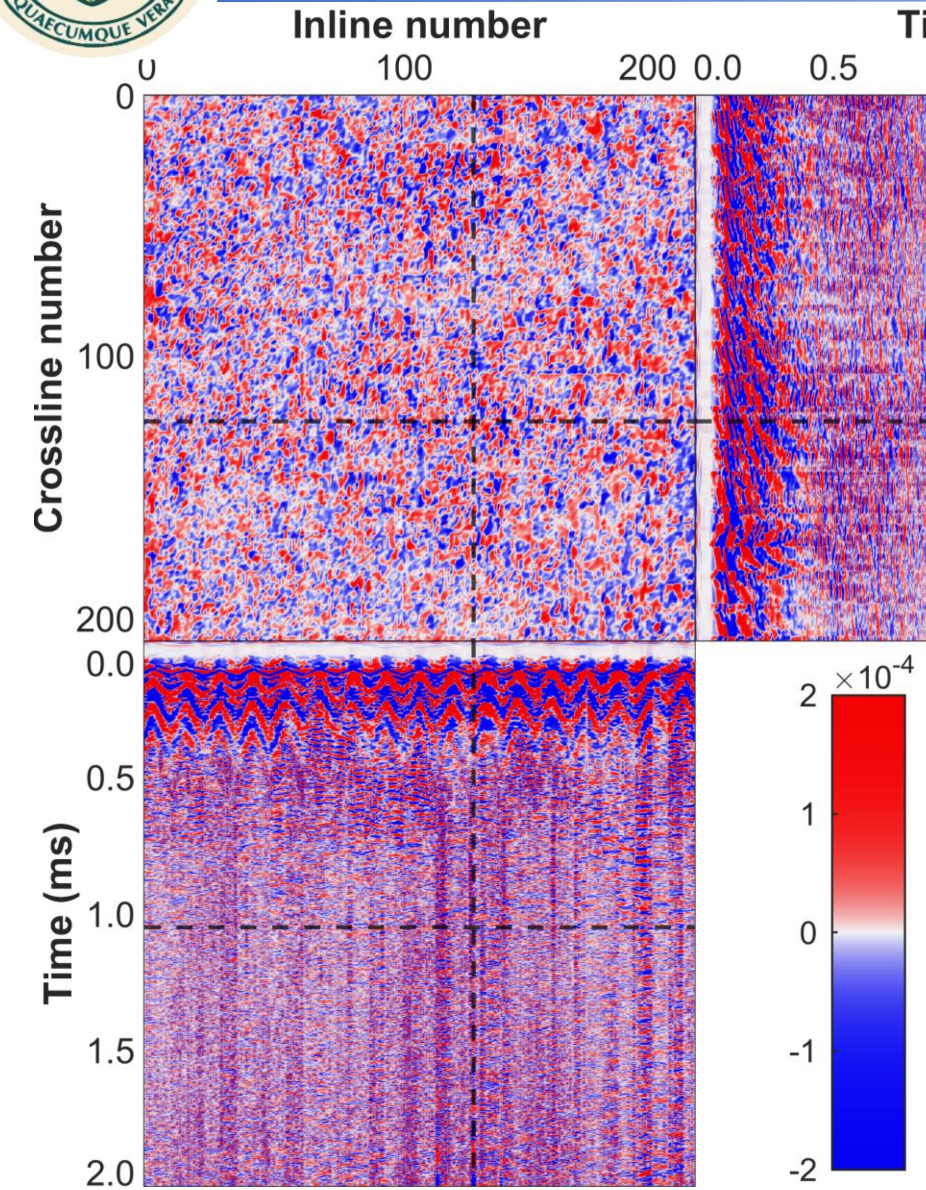
Denoised results by our method

SYNTHETIC SEISMIC DATA



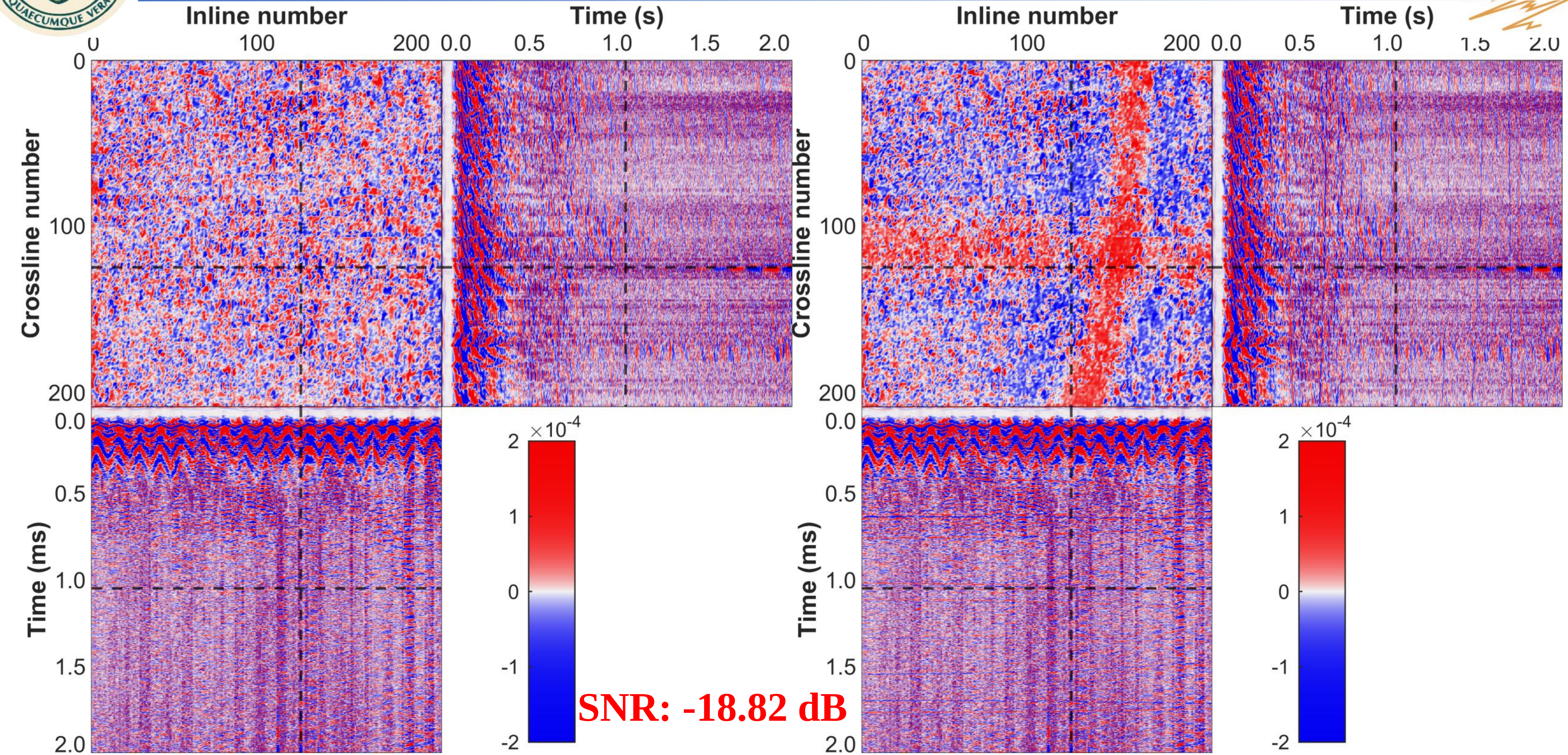
Training labels with offset of 5001

SYNTHETIC SEISMIC DATA



Field seismic noise with offset of 5001

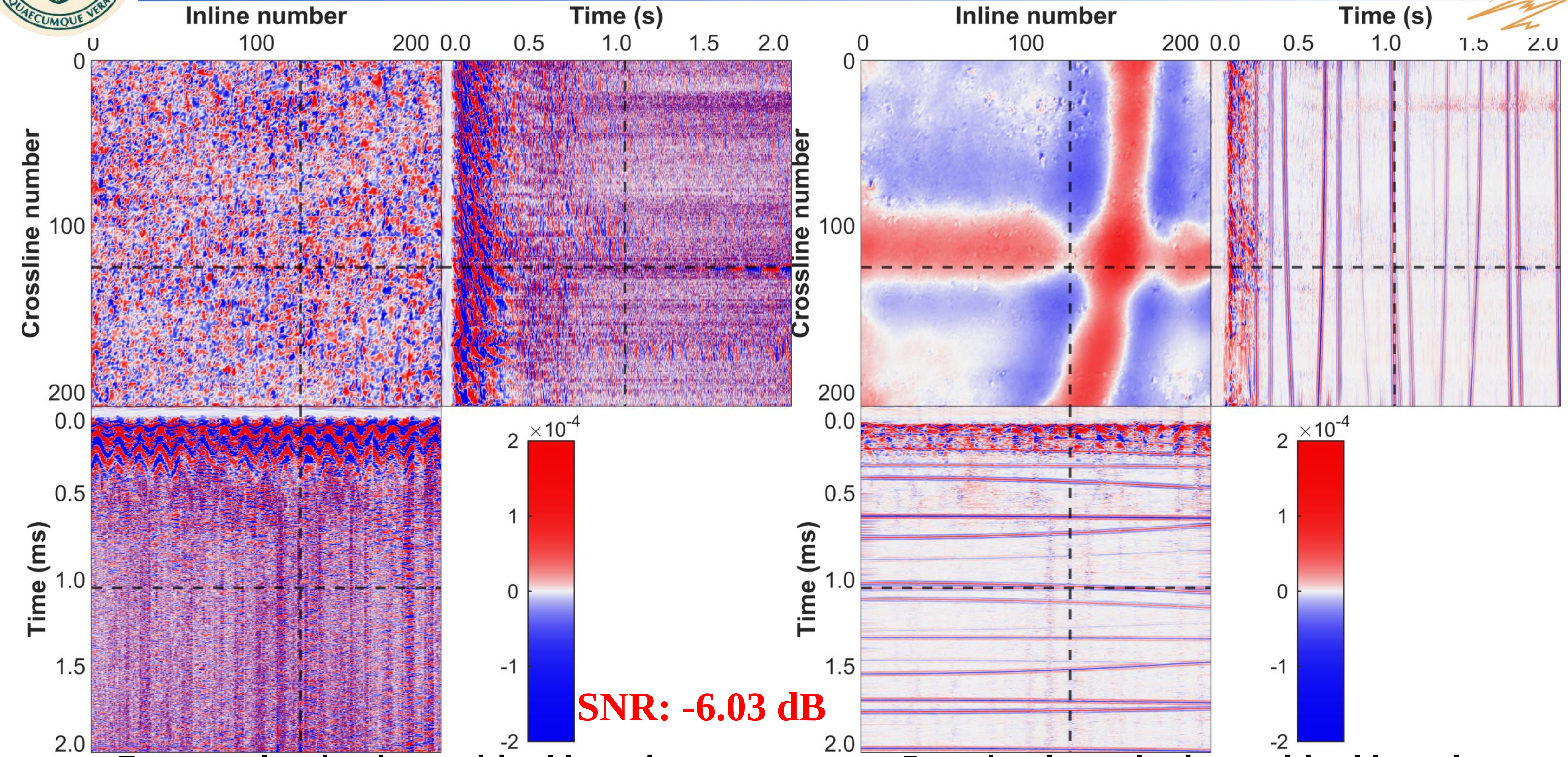
SYNTHETIC SEISMIC DATA



Field seismic noise with offset of 5001

Noisy seismic noise with offset of 5001

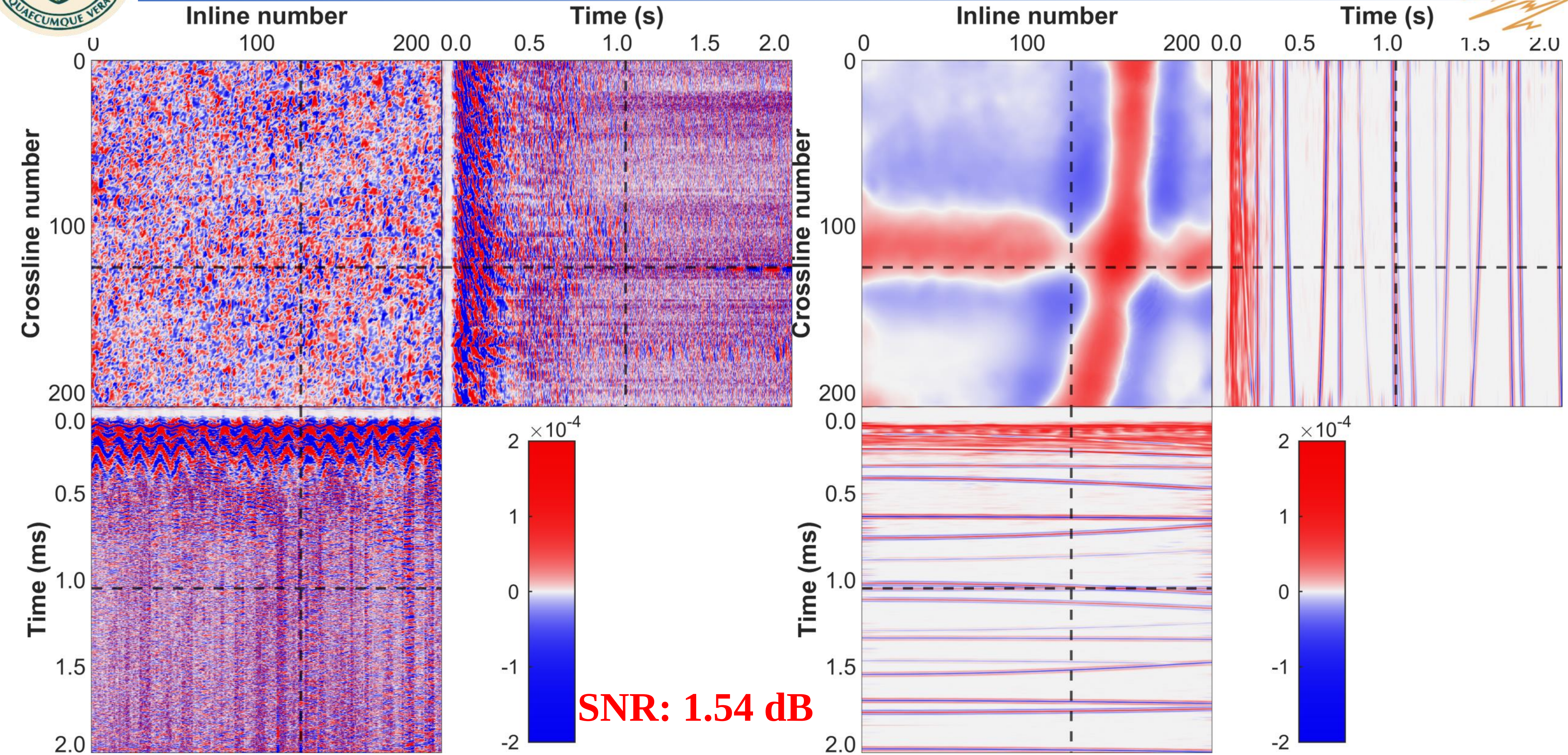
SYNTHETIC SEISMIC DATA



Removed noise by residual learning

Denoised results by residual learning

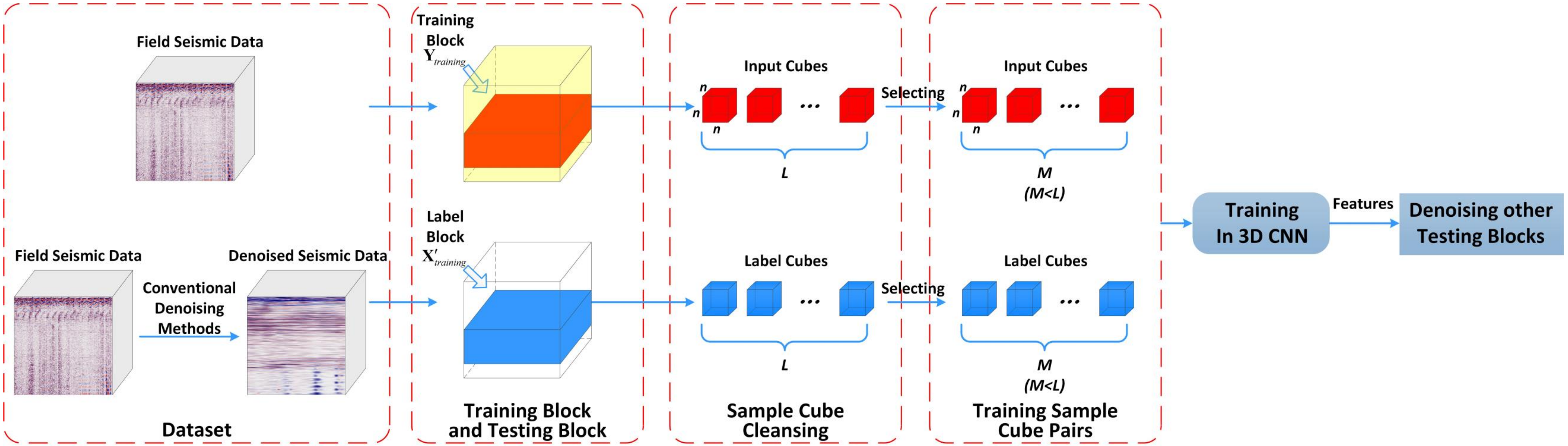
SYNTHETIC SEISMIC DATA



Removed noise by our method

Denoised results by our method

FIELD SEISMIC DATA



Our network processing procedure

CONVENTIONAL DENOISING METHOD

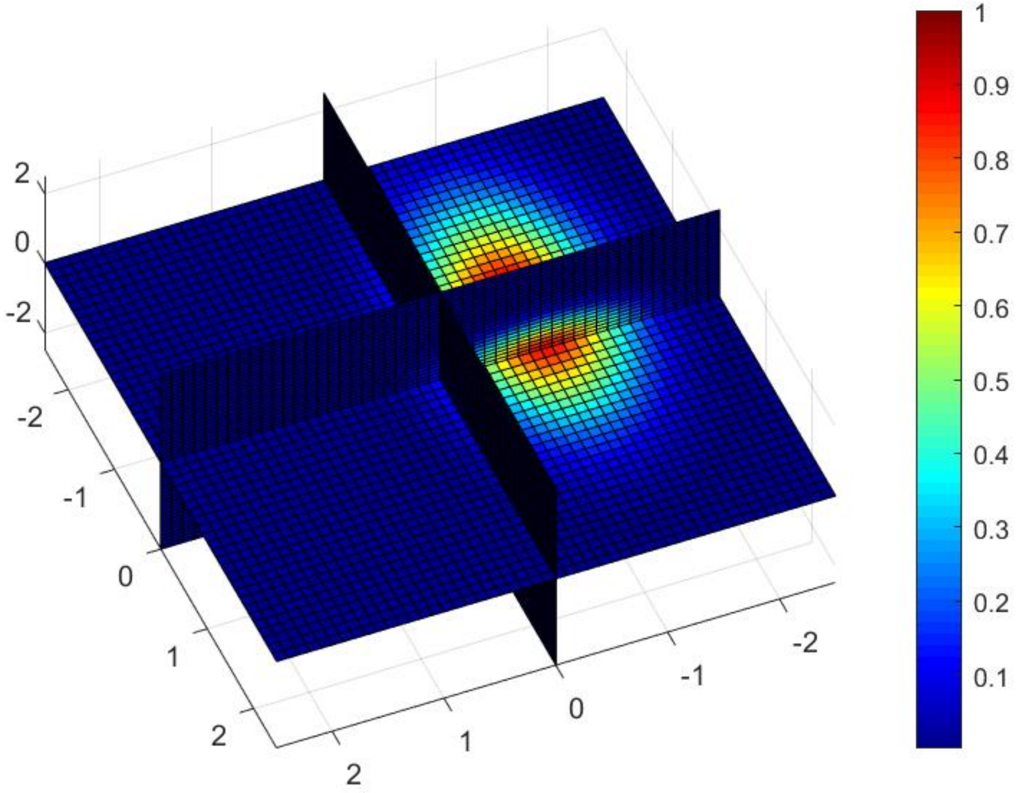
To make full use of dip information, we choose 3D-Morlet wavelet to construct labels.

CONVENTIONAL DENOISING METHOD

To make full use of dip information, we choose 3D-Morlet wavelet to construct labels.

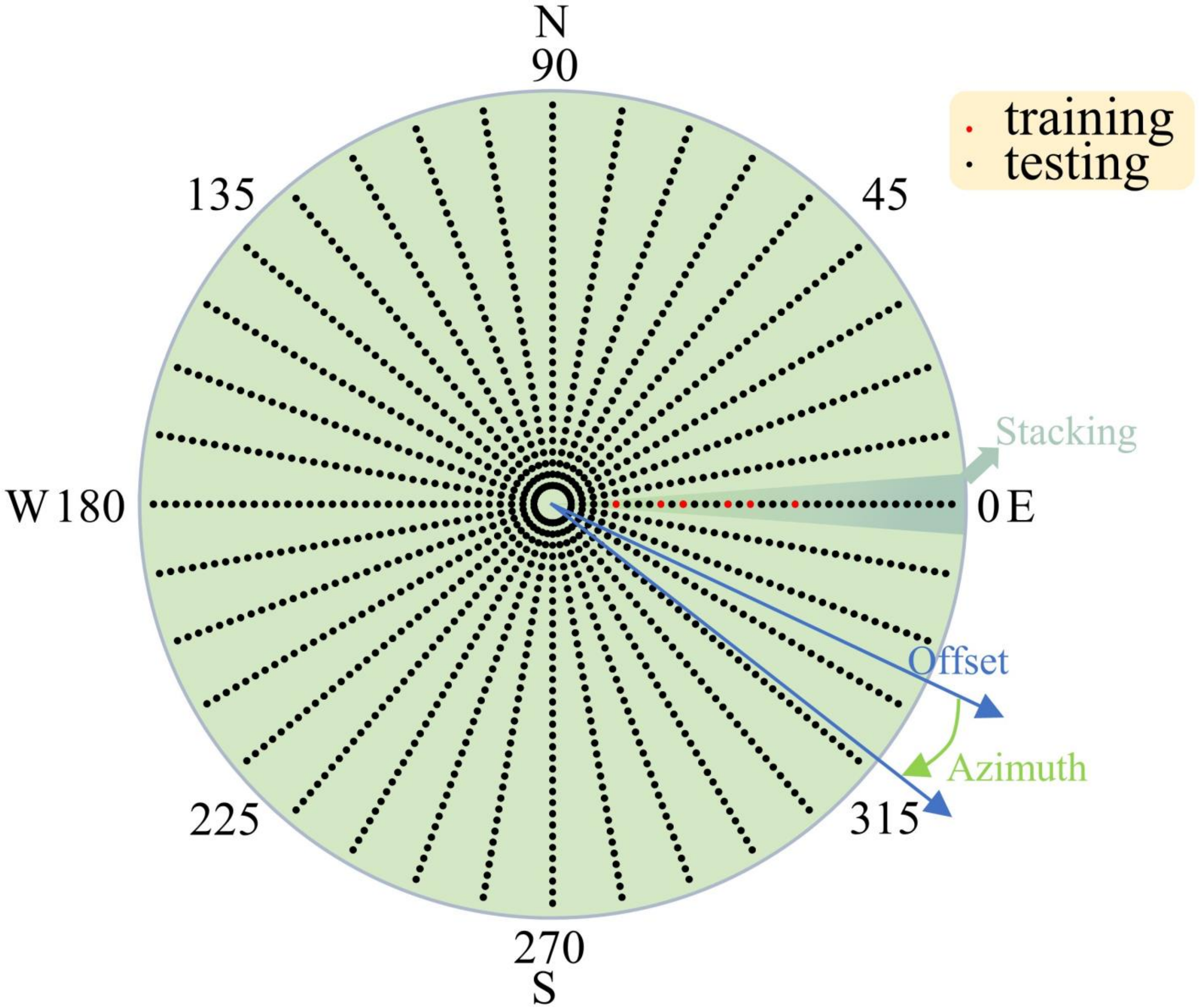
The definition of 3D-CWT is

$$
\begin{aligned}
& CWT(f;\mathbf{b},a,\rho,\varphi)=\left\langle f,\psi_{\mathbf{b},a,(\rho,\varphi)}\right\rangle \\
& =\frac{1}{a^3}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(\mathbf{m})\psi^*\left(\frac{1}{a}\Re_{\rho,\varphi}(\mathbf{m}-\mathbf{b})\right)dxdydz \\
& =\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \hat{f}(\mathbf{k})\hat{\psi}^*\left(a\Re_{\rho,\varphi}(\mathbf{k})\right)e^{j\mathbf{bk}}dk_x dk_y dk_z
\end{aligned}
$$



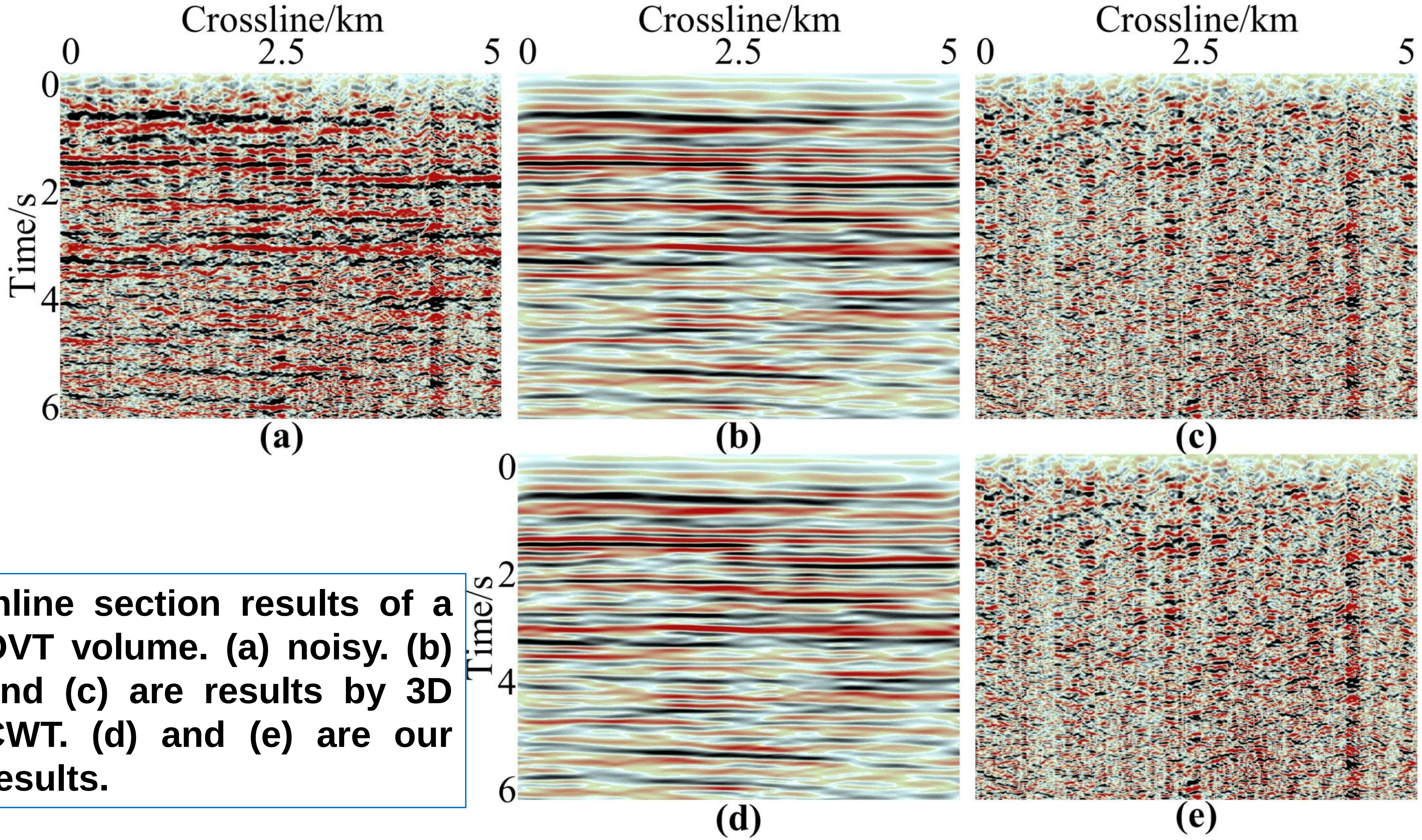
Space slice of a 3d-morlet wavelet

LOCATIONS



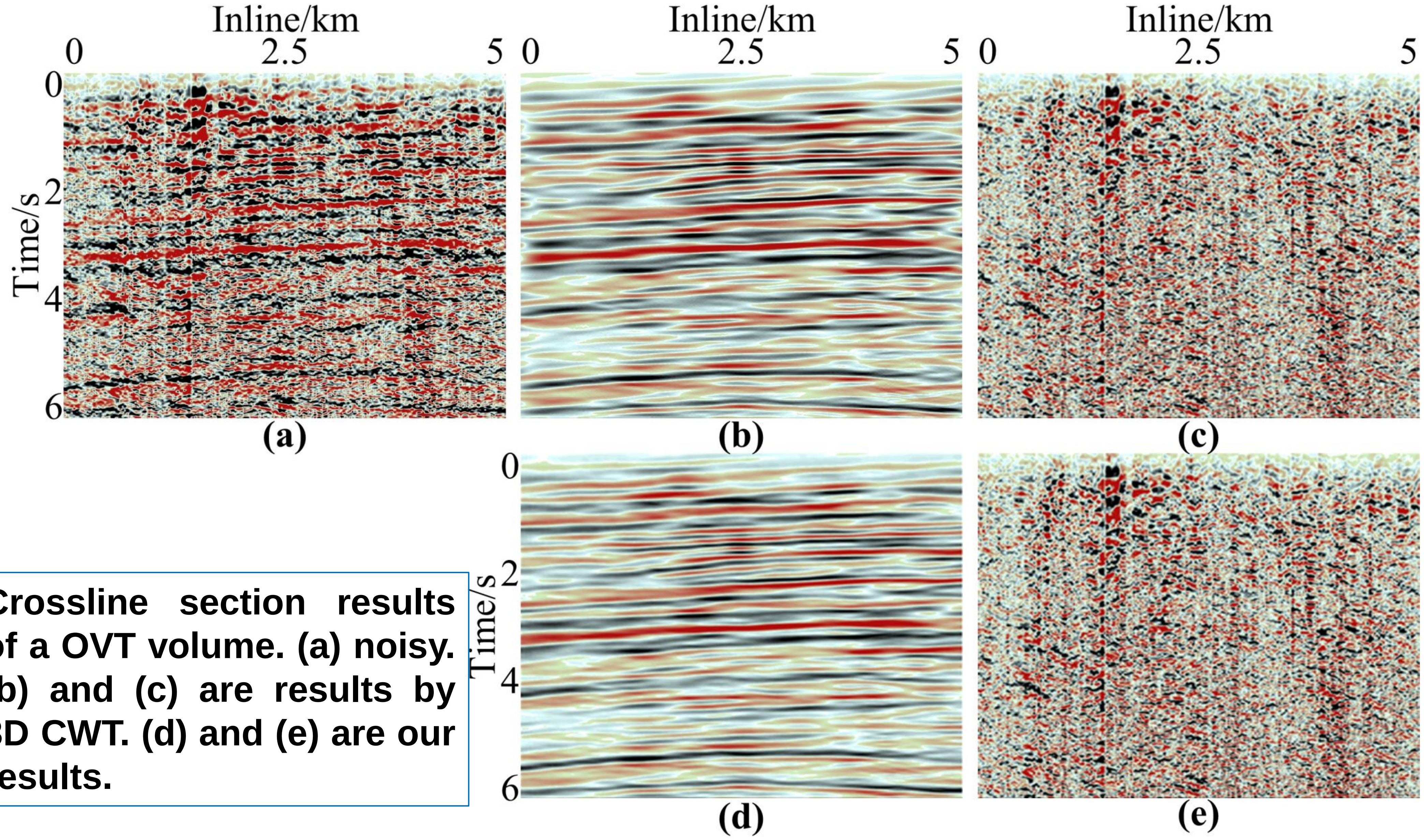
Locations of OVT volumes in offset-azimuth polar coordinates

OVT RESULTS



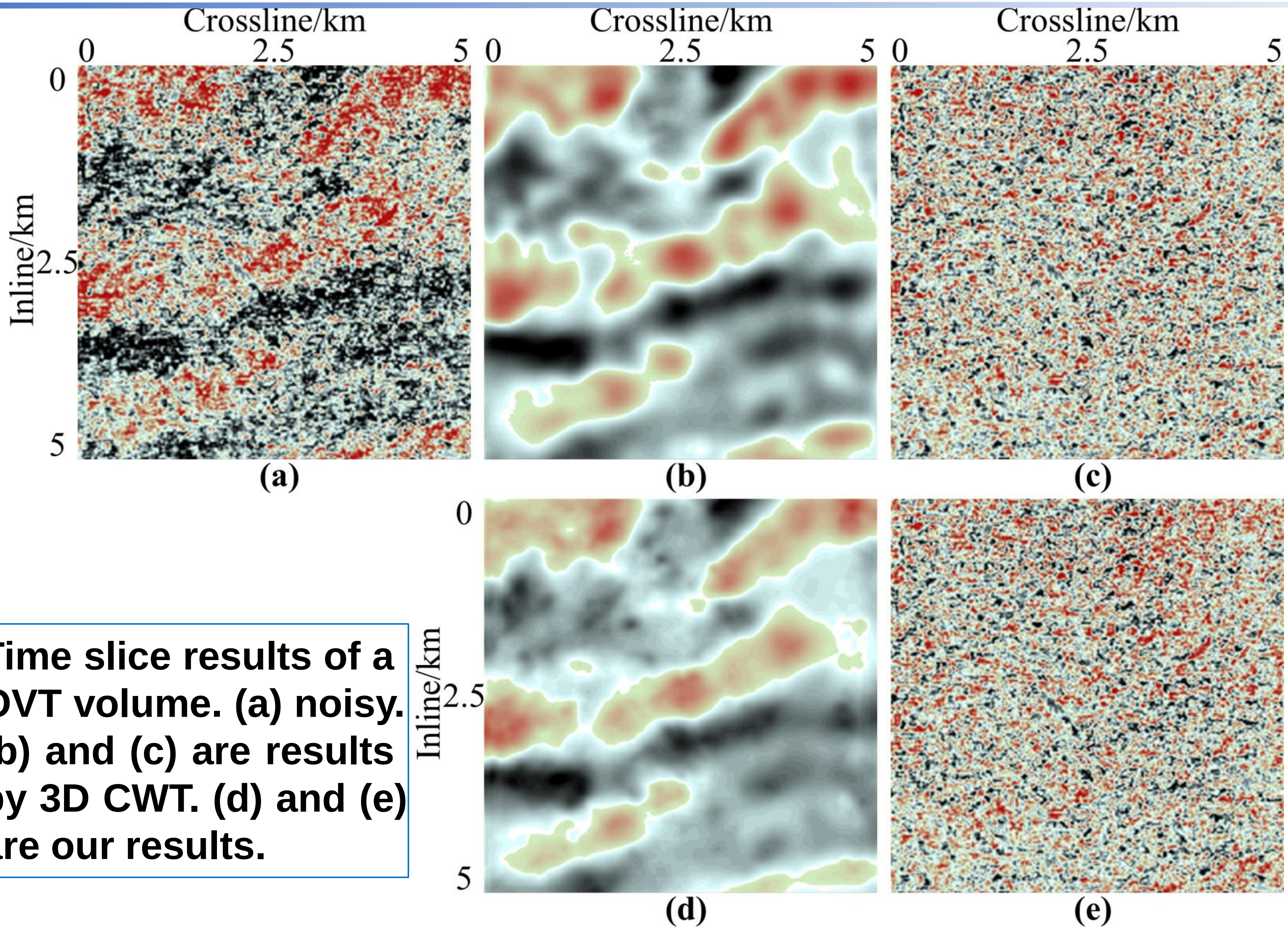
Inline section results of a OVT volume. (a) noisy. (b) and (c) are results by 3D CWT. (d) and (e) are our results.

OVT RESULTS



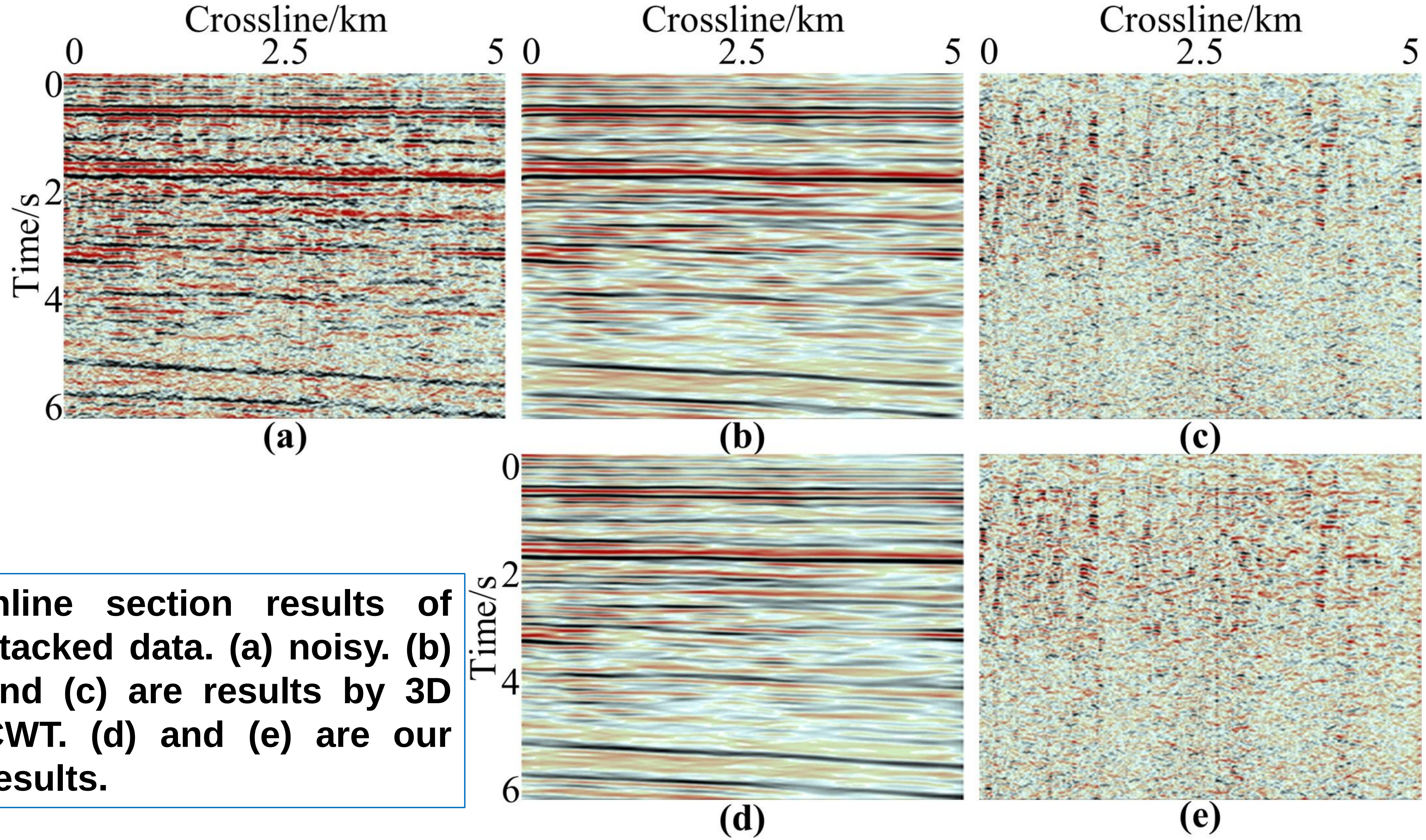
Crossline section results of a OVT volume. (a) noisy. (b) and (c) are results by 3D CWT. (d) and (e) are our results.

OVT RESULTS



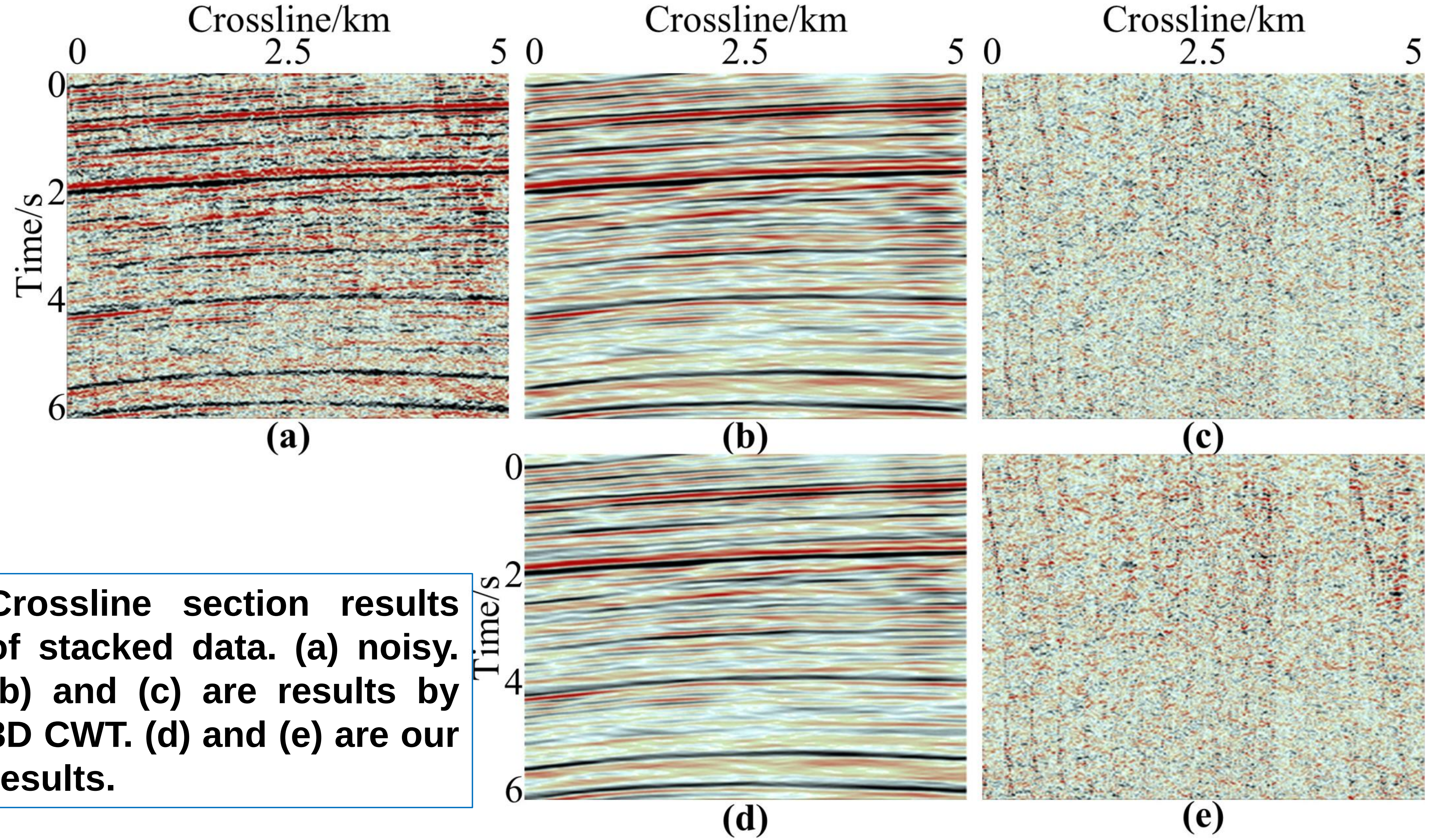
Time slice results of a OVT volume. (a) noisy. (b) and (c) are results by 3D CWT. (d) and (e) are our results.

STACKED RESULTS



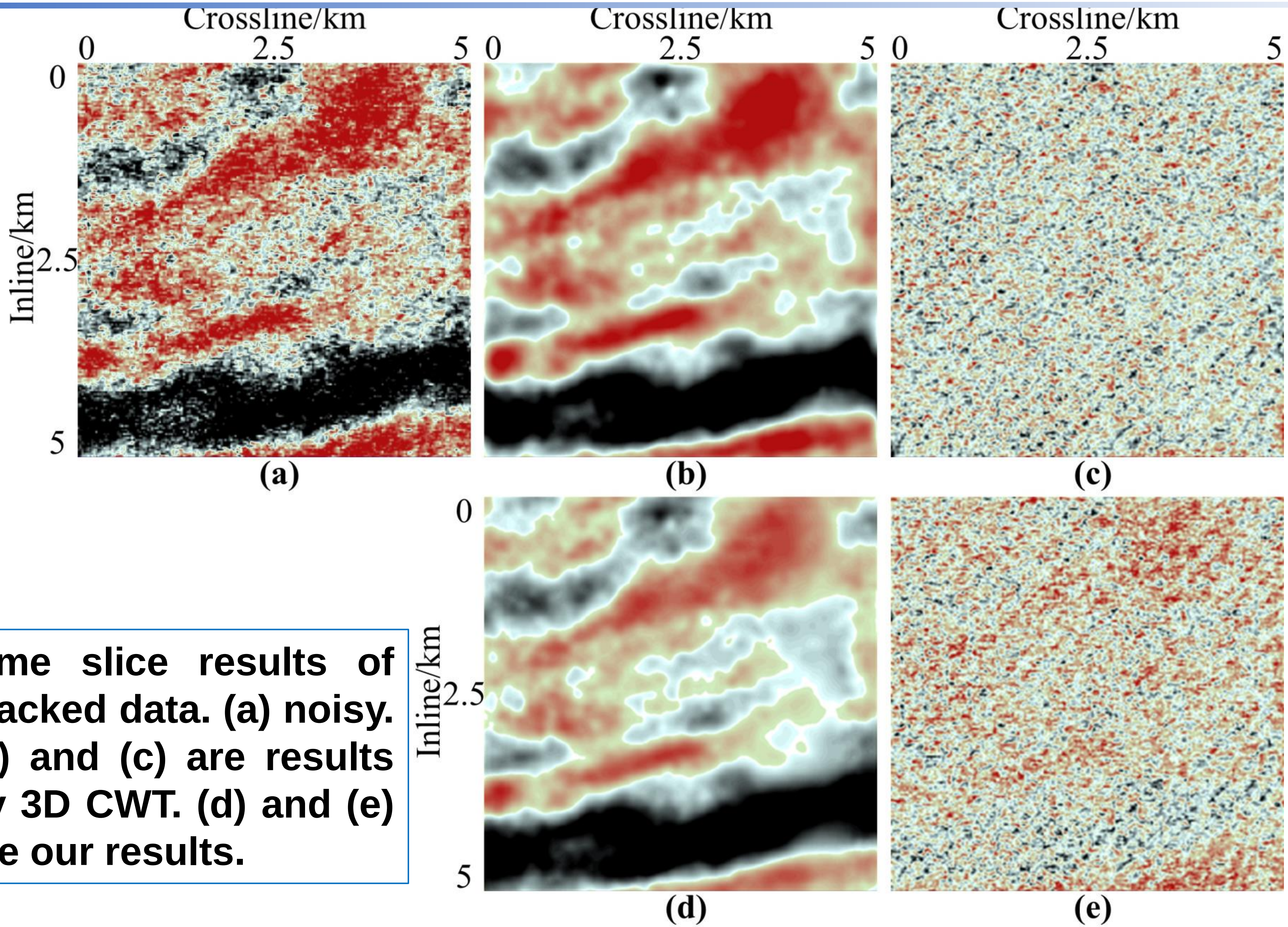
Inline section results of stacked data. (a) noisy. (b) and (c) are results by 3D CWT. (d) and (e) are our results.

STACKED RESULTS



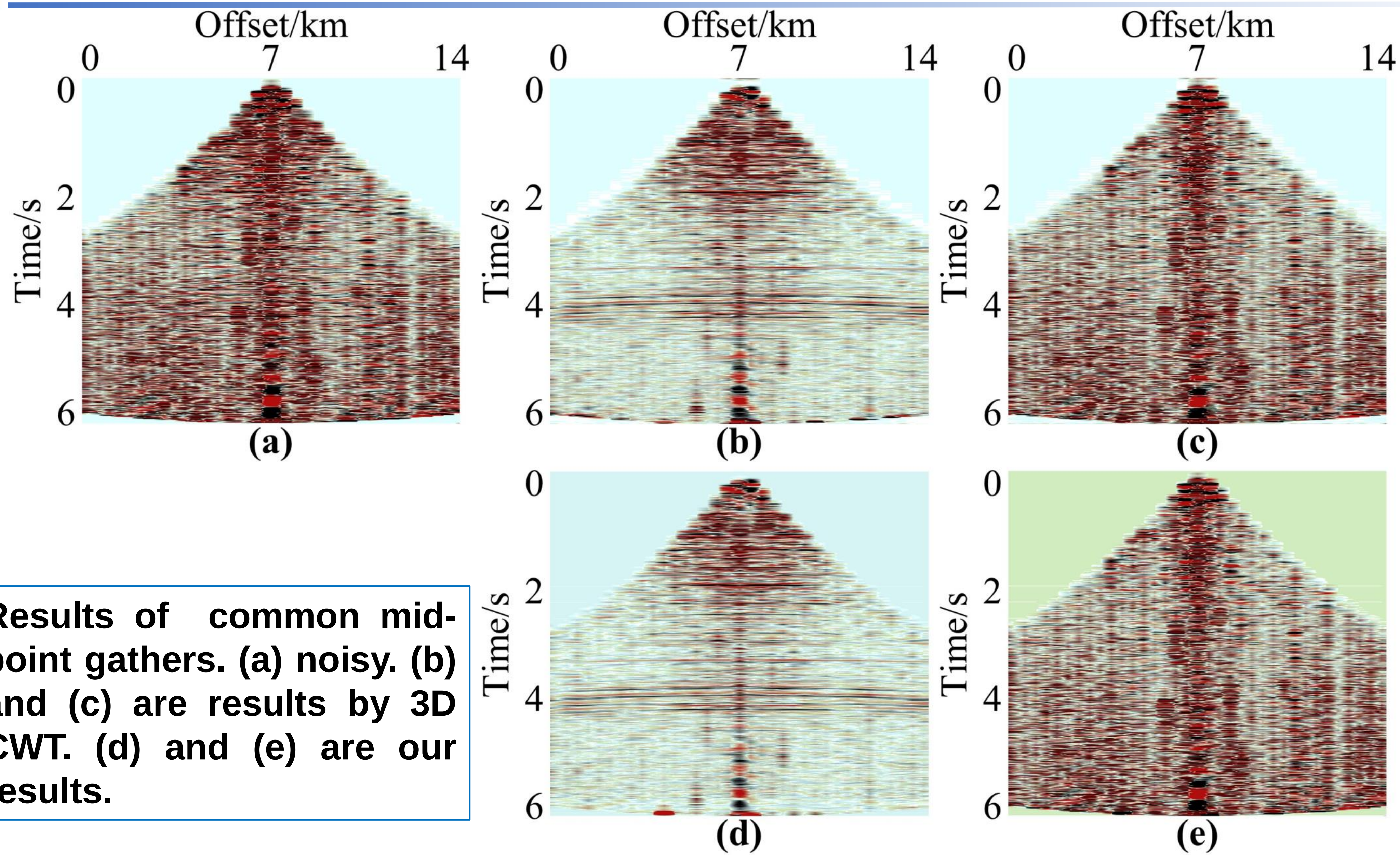
Crossline section results of stacked data. (a) noisy. (b) and (c) are results by 3D CWT. (d) and (e) are our results.

STACKED RESULTS



Time slice results of stacked data. (a) noisy. (b) and (c) are results by 3D CWT. (d) and (e) are our results.

CMP RESULTS



Results of common mid-point gathers. (a) noisy. (b) and (c) are results by 3D CWT. (d) and (e) are our results.

Cost time comparison

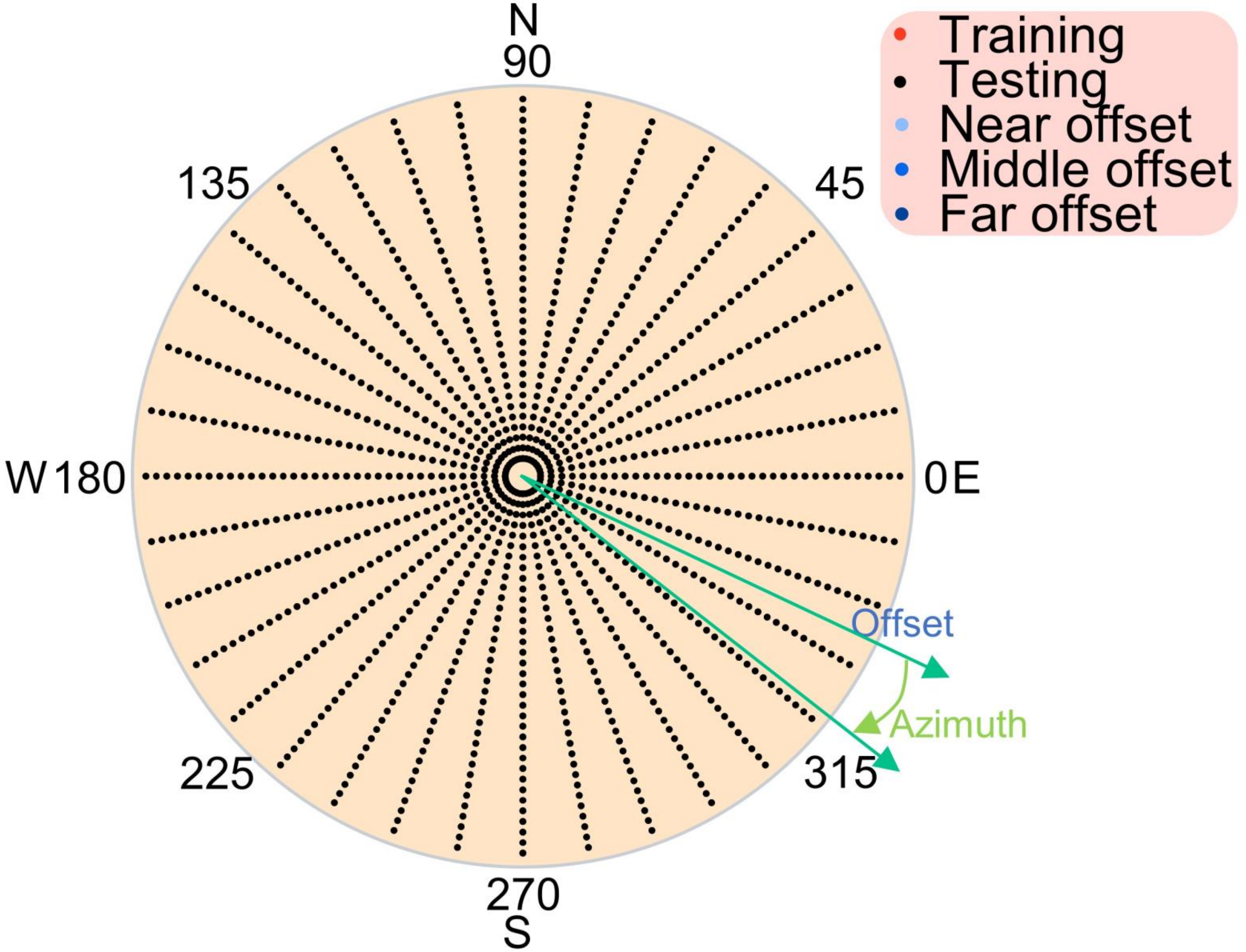
Table: Time consumption comparison.

	Training time	Test time for one OVT	Total time for 1260 OVTs
3D CWT	–	68.5 minutes	61 days
Our method	2 days	5.9 minutes	4.1 days

Middle-offset selection strategy

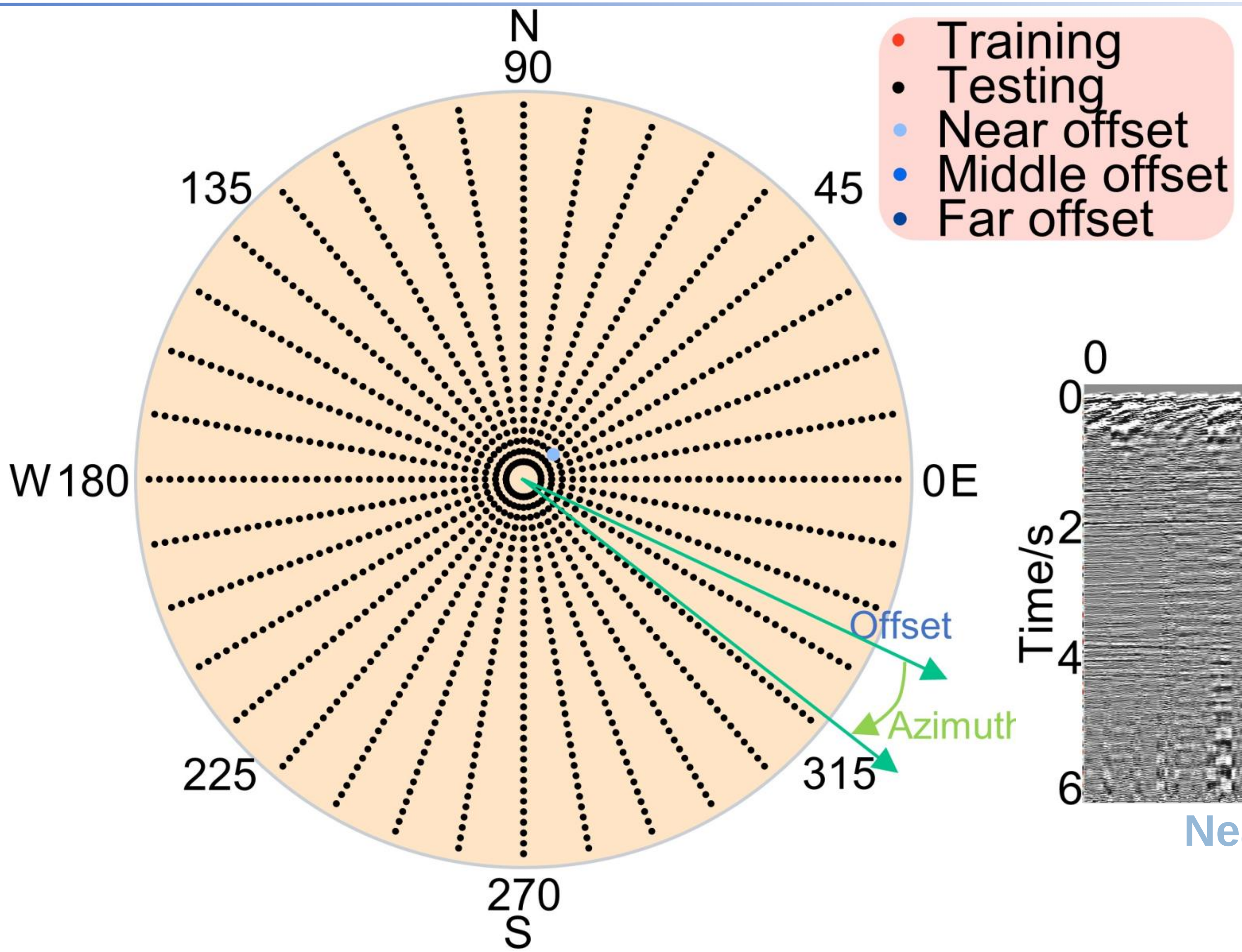
Can we further improve the denoising results?

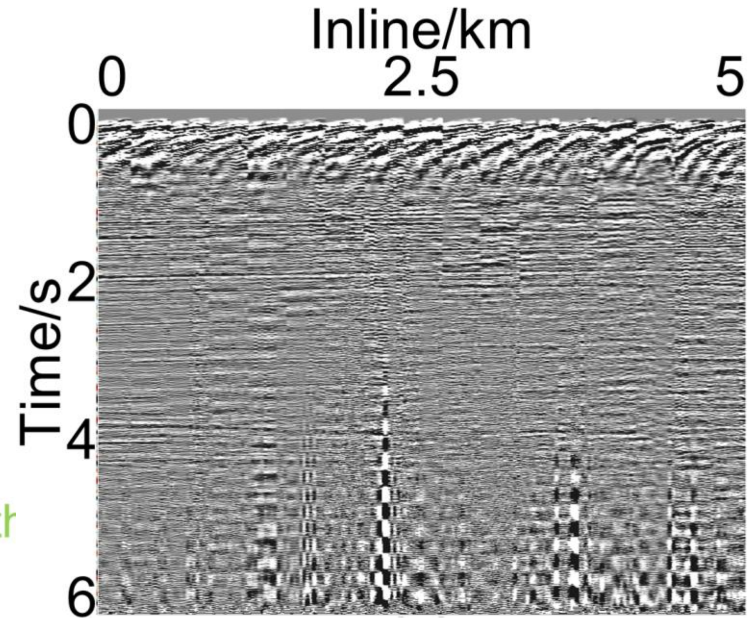
Middle-offset selection strategy



Locations of OVT volumes in offset-azimuth polar coordinates

Middle-offset selection strategy

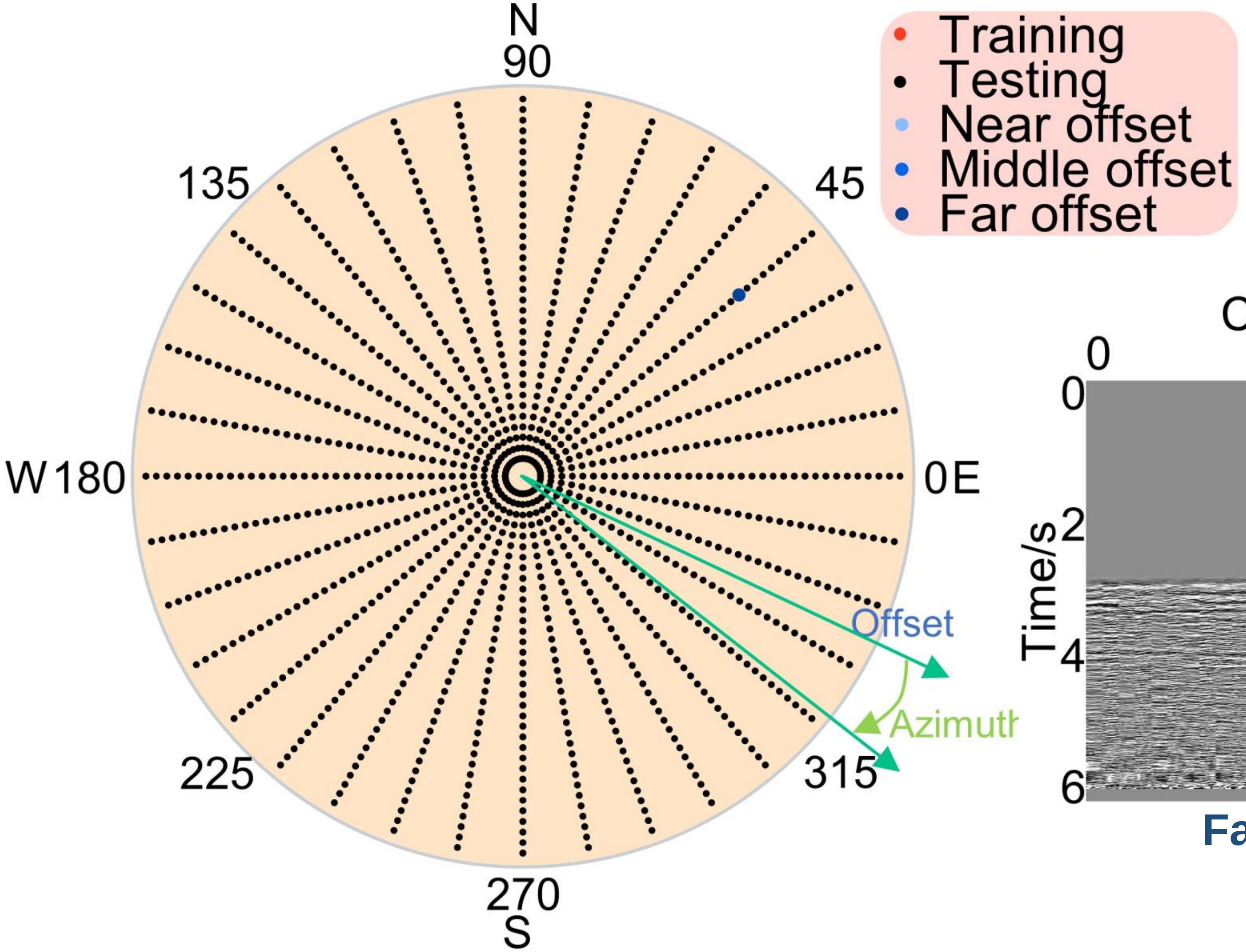




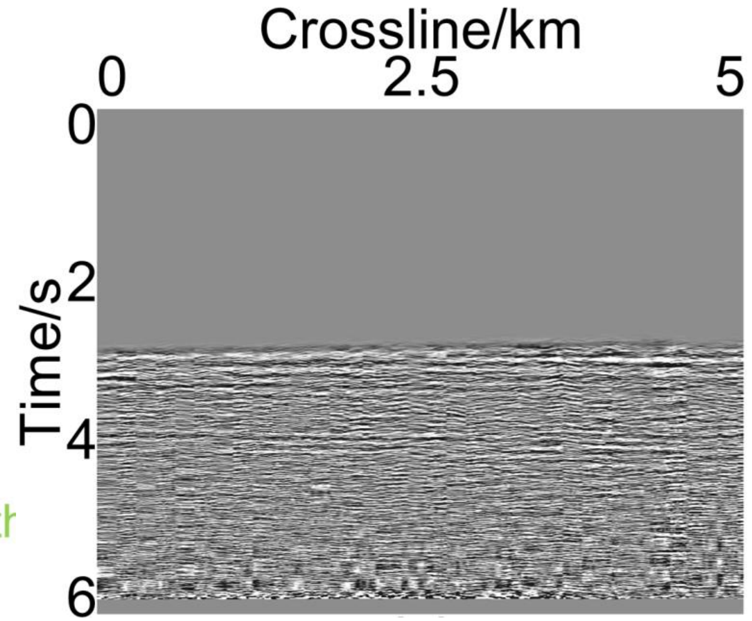
Near-offset data

Locations of OVT volumes in offset-azimuth polar coordinates

Middle-offset selection strategy

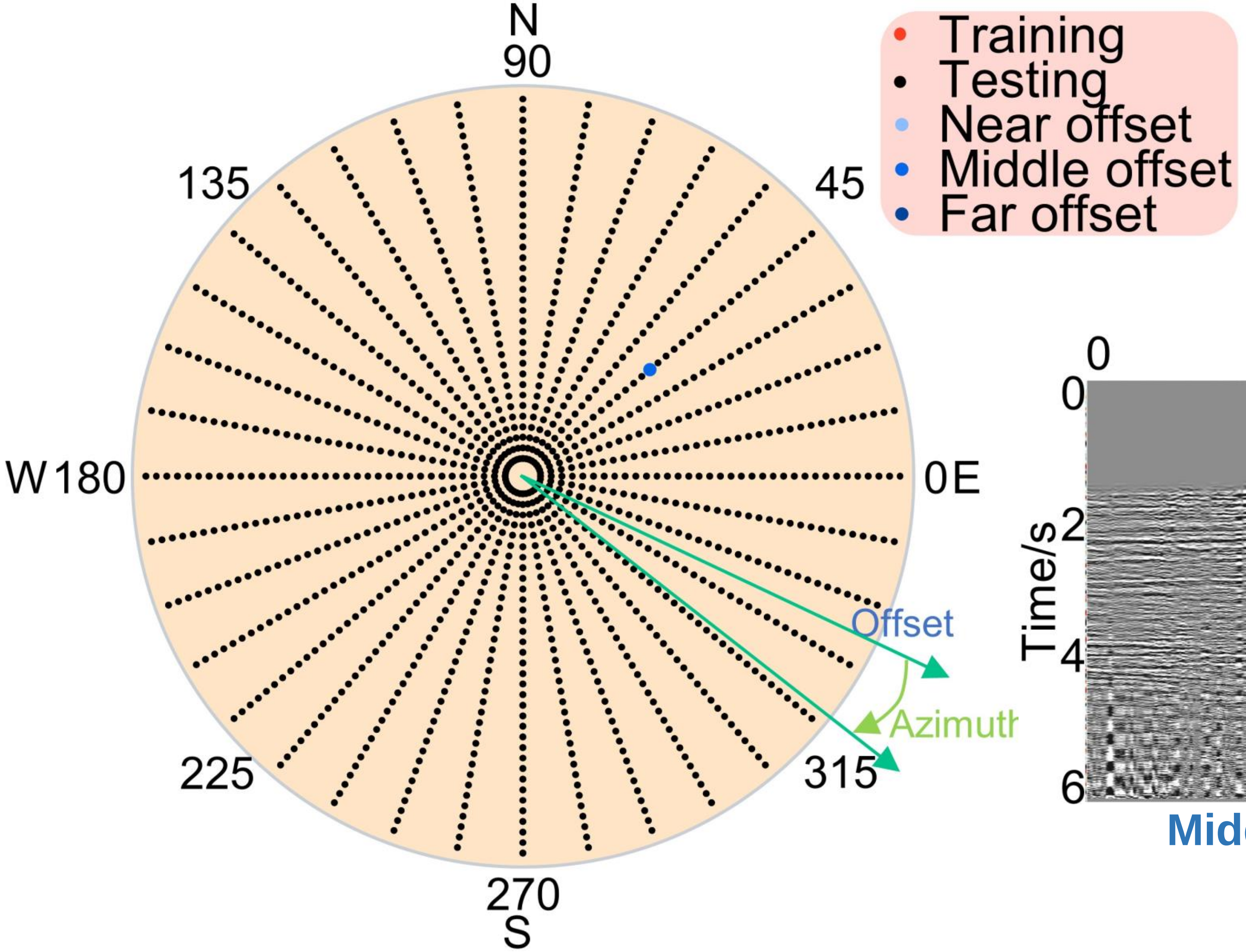


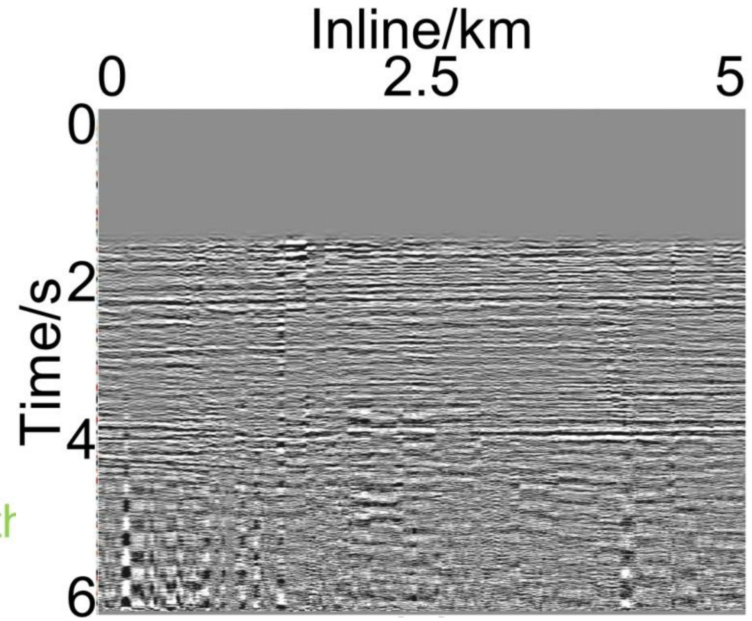
Locations of OVT volumes in offset-azimuth polar coordinates



Far-offset data

Middle-offset selection strategy

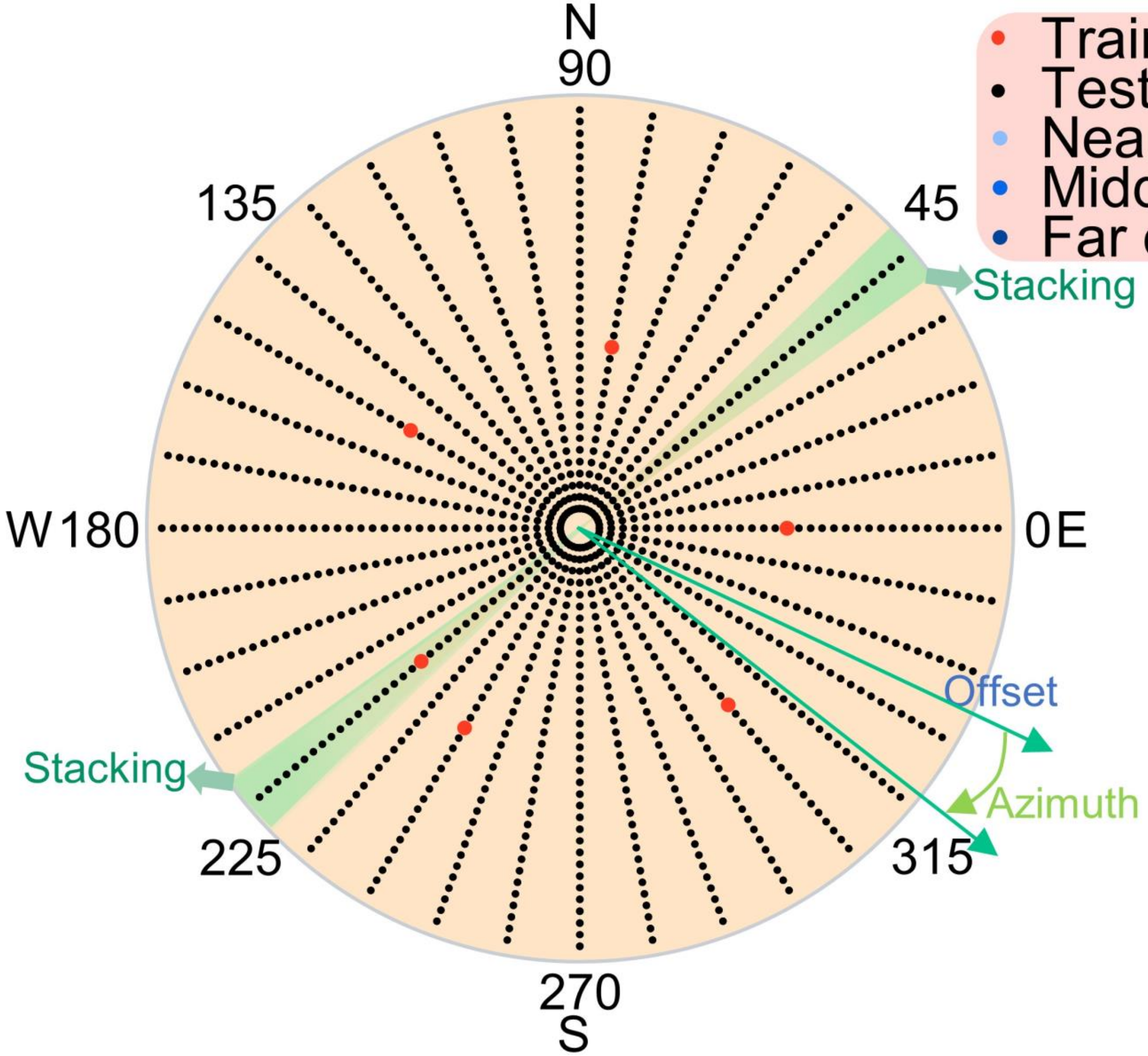




Middle-offset data

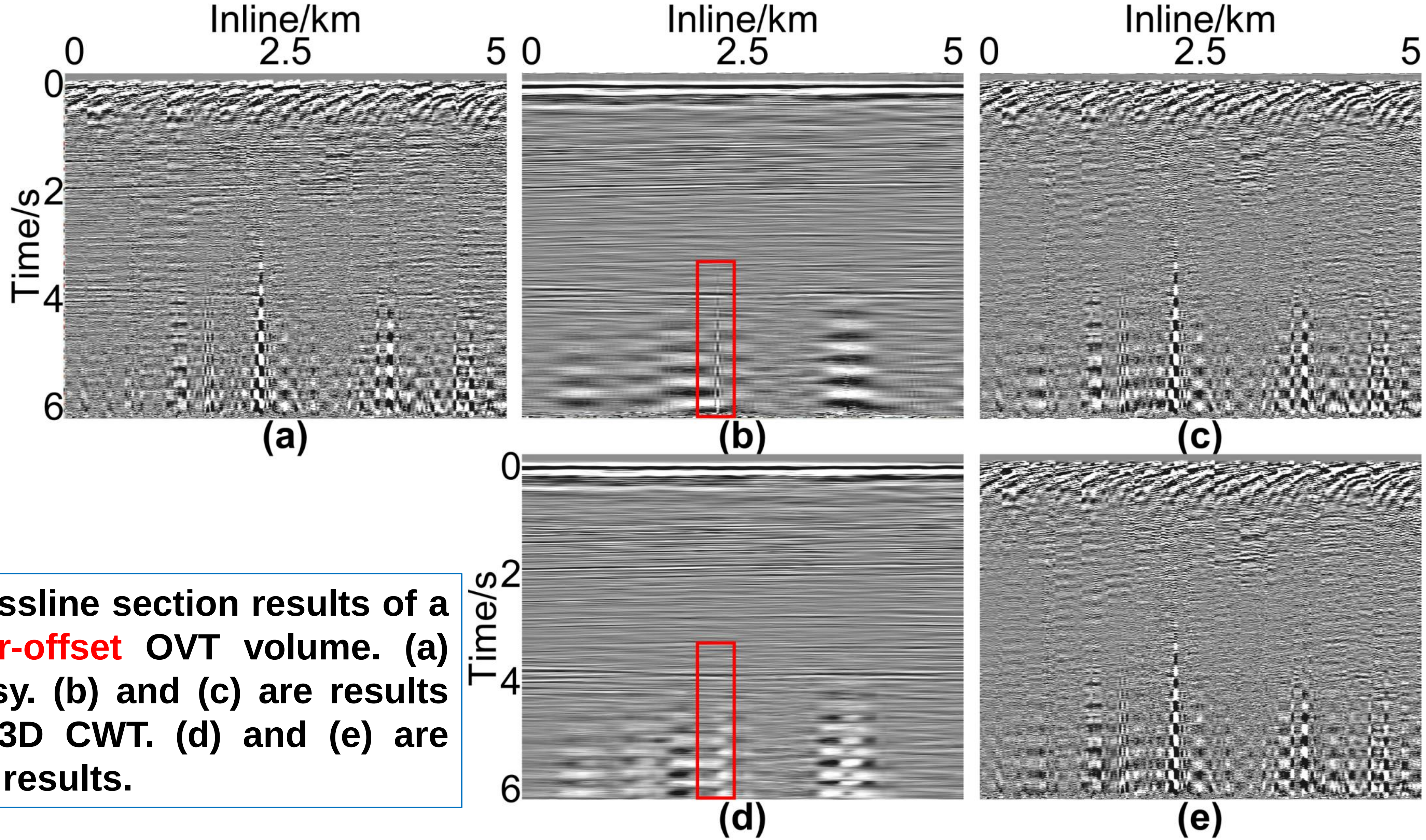
Locations of OVT volumes in offset-azimuth polar coordinates

Middle-offset selection strategy



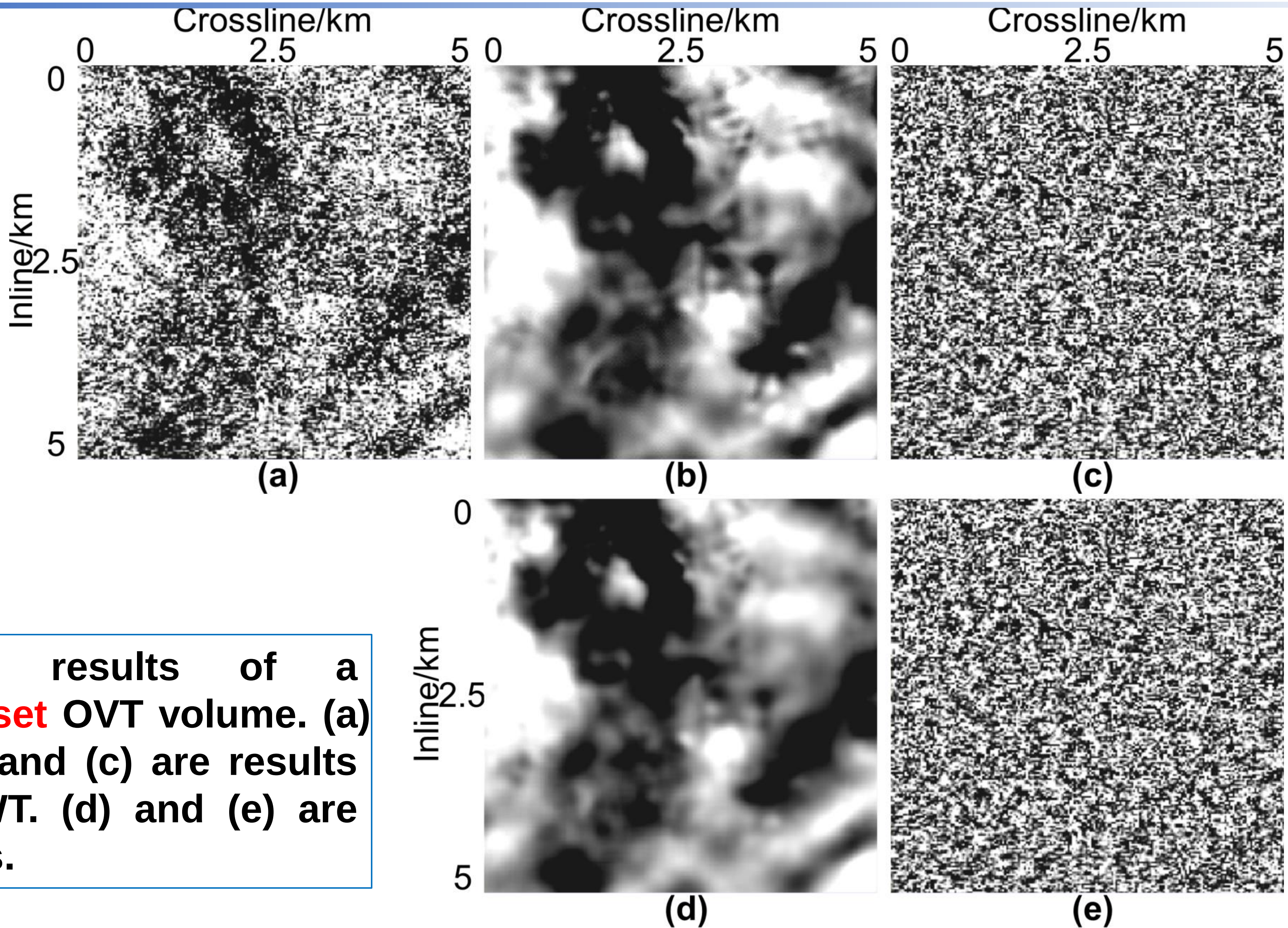
Locations of OVT volumes in offset-azimuth polar coordinates

OVT RESULTS



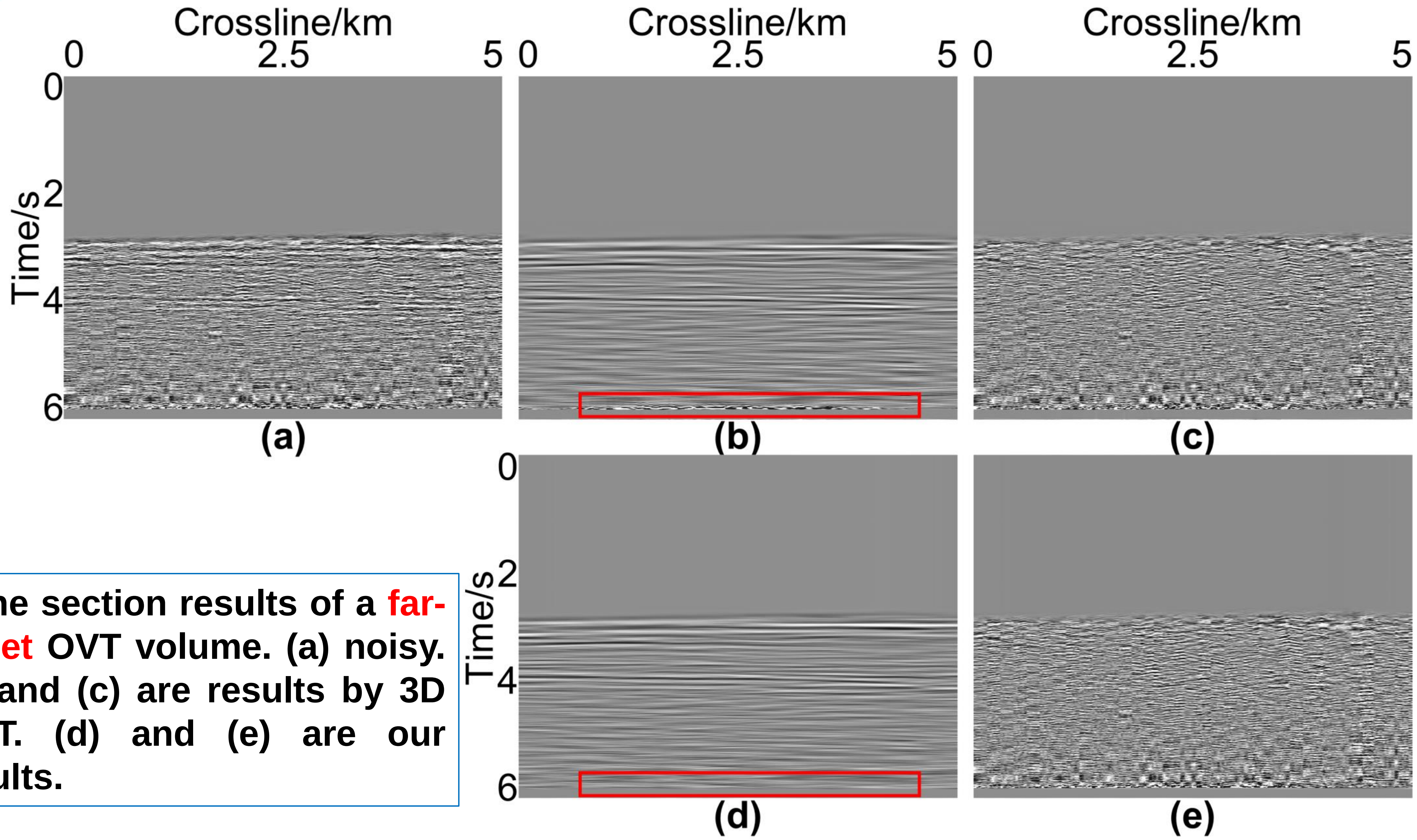
Crossline section results of a near-offset OVT volume. (a) noisy. (b) and (c) are results by 3D CWT. (d) and (e) are our results.

OVT RESULTS



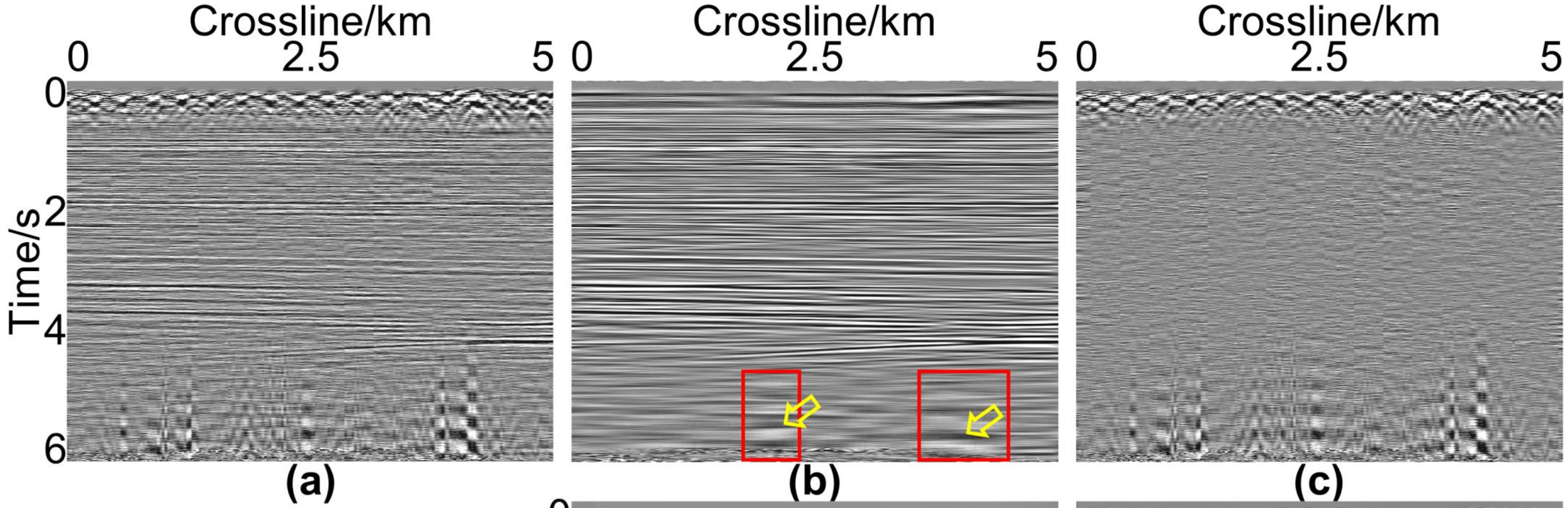
Time-slice results of a middle-offset OVT volume. (a) noisy. (b) and (c) are results by 3D CWT. (d) and (e) are our results.

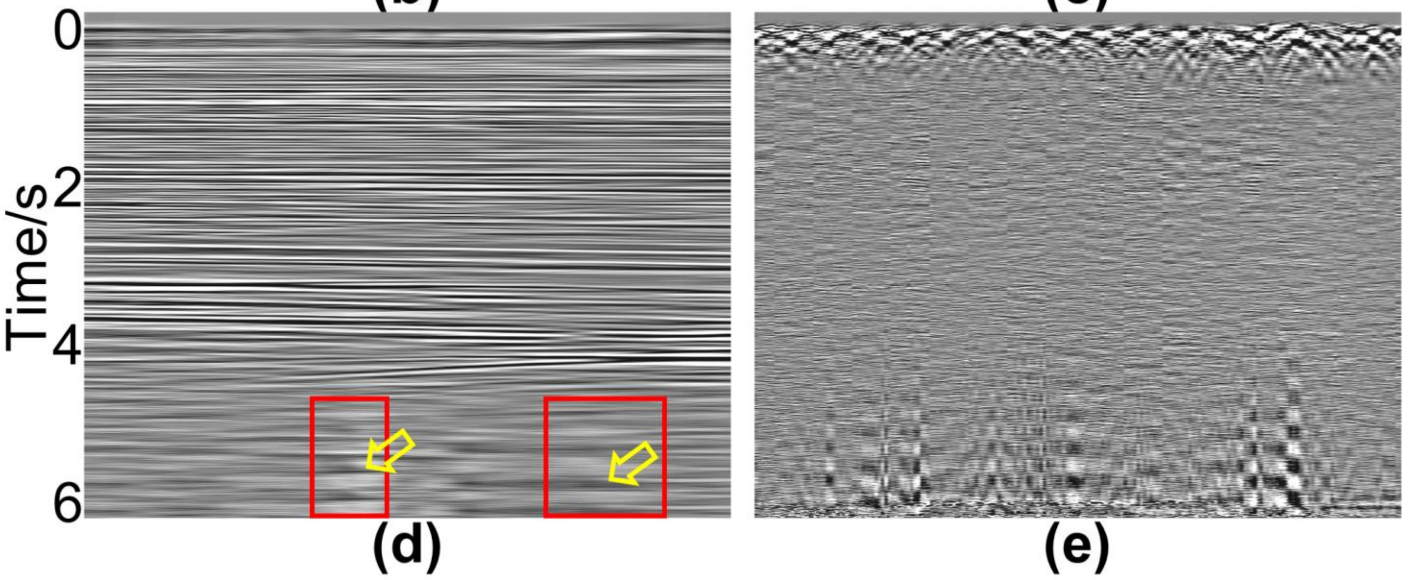
OVT RESULTS



Inline section results of a far-offset OVT volume. (a) noisy. (b) and (c) are results by 3D CWT. (d) and (e) are our results.

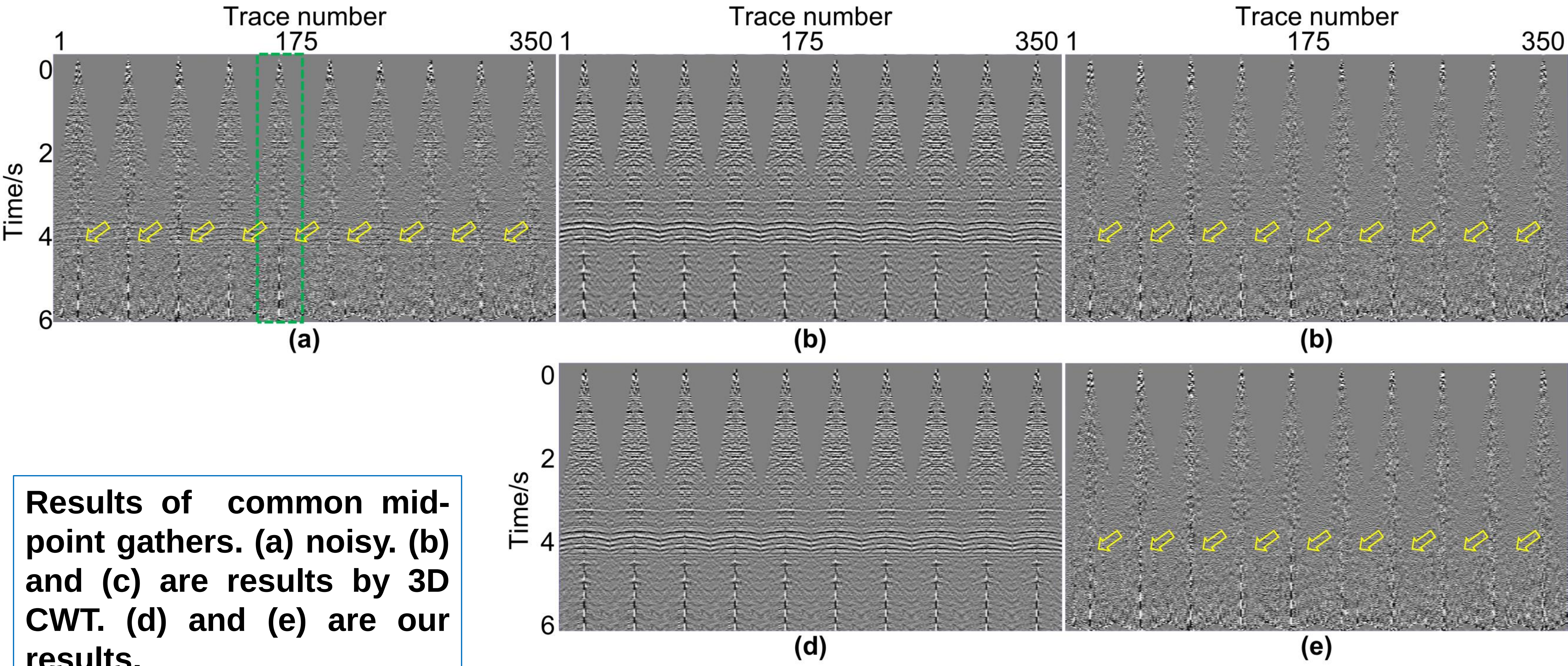
STACKED RESULTS





Inline section results of stacked data. (a) noisy. (b) and (c) are results by 3D CWT. (d) and (e) are our results.

CMP RESULTS



Results of common mid-point gathers. (a) noisy. (b) and (c) are results by 3D CWT. (d) and (e) are our results.

OUTLINE

- Introduction
- Model and network training
- Instances on Synthetic data and real Data
- Conclusion
- Acknowledgement

Conclusion

•We propose a prestack denoising method by combining the merits of deep learning and OVT partitioning techniques. In the OVT domain, the wavefield continuity and data consistency provide a conducive signal learning environment for the network. The massive amount of OVT gathers can make full use of the high computational efficiency of deep learning.

Conclusion

- We propose a prestack denoising method by combining the merits of deep learning and OVT partitioning techniques. In the OVT domain, the wavefield continuity and data consistency provide a conducive signal learning environment for the network. The massive amount of OVT gathers can make full use of the high computational efficiency of deep learning.

- The direct signal learning is better than residual learning for strong scattered noise attenuation. The field results demonstrate that only a minimal number of OVT volumes can make the network obtain the ability to suppress the whole survey's noise.

Conclusion

•We propose a prestack denoising method by combining the merits of deep learning and OVT partitioning techniques. In the OVT domain, the wavefield continuity and data consistency provide a conducive signal learning environment for the network. The massive amount of OVT gathers can make full use of the high computational efficiency of deep learning.

•The direct signal learning is better than residual learning for strong scattered noise attenuation. The field results demonstrate that only a minimal number of OVT volumes can make the network obtain the ability to suppress the whole survey's noise.

•Randomly selecting the middle-offset OVT gathers as the training volumes can get better results.

OUTLINE

- Introduction
- Model and network training
- Instances on Synthetic data and real Data
- Conclusion
- Acknowledgement

Acknowledgement

- **All the sponsors of SAIG for supporting our work.**

Contact Us

Thanks for listening!

Contact me by email: dawei3@ualberta.ca
Homepage: davidliu-code.github.io