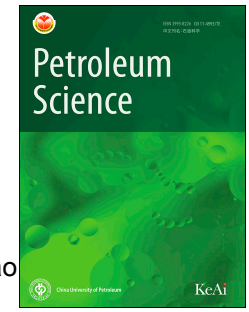

STISR: A Stacked Tucker Implicit Seismic Reparameterization Framework for 3D Seismic Data Denoising

Qingfang Wang, Dawei Liu, Mauricio D. Sacchi, Xiaokai Wang, Deyu Meng, Wenchao Chen

PII: S1995-8226(26)00173-1

DOI: https://doi.org/10.1016/j.petsci.2026.03.048

Reference: PETSCI 1502

To appear in: *Petroleum Science*

Received Date: 27 August 2025

Revised Date: 28 January 2026

Accepted Date: 23 March 2026

Please cite this article as: Wang, Q., Liu, D., Sacchi, M.D., Wang, X., Meng, D., Chen, W., STISR: A Stacked Tucker Implicit Seismic Reparameterization Framework for 3D Seismic Data Denoising, *Petroleum Science*, https://doi.org/10.1016/j.petsci.2026.03.048.

Highlights

STISR: A Stacked Tucker Implicit Seismic Reparameterization Framework for 3D Seismic Data Denoising

Qingfang Wang, Dawei Liu, Mauricio D. Sacchi, Xiaokai Wang, Deyu Meng, Wenchao Chen

- A nonlinear reparameterization using INR models Tucker factors, combining low-rank constraints with nonlinear approximation for denoising.
- A progressive re-decomposition refines the Tucker core across scales, balancing low-rank stability and representational capacity.
- An adaptive l_1 regularization adjusts sparsity by residual noise, improving robustness to heterogeneous field seismic noise.

STISR: A Stacked Tucker Implicit Seismic Reparameterization Framework for 3D Seismic Data Denoising

Qingfang Wang[a], Dawei Liu[a,*], Mauricio D. Sacchi[b], Xiaokai Wang[a], Deyu Meng[c], Wenchao Chen[a]

[a] *The School of Information and Communication Engineering, Xi'an Jiaotong University, Xi'an, 710049, Shaanxi, China*
[b] *The Department of Physics, University of Alberta, Edmonton, T6G 2R3, Alberta, Canada*
[c] *The School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, 710049, Shaanxi, China*

Abstract

High-quality seismic data are critical for characterizing complex geological reservoirs, yet persistent noise contamination remains challenging. Parameterization methods separate signals from noise by expressing seismic data as mathematical models. While linear approaches like Tucker decomposition effectively impose low-rank constraints to isolate structured seismic reflections, they lack the nonlinear expressiveness required for complex stratigraphic features. Conversely, like implicit neural representations (INR), nonlinear parameterization achieves enhanced expressiveness through continuous nonlinear mappings, leveraging spectral bias to suppress high-frequency noise. However, this strength paradoxically becomes a limitation in high-frequency regimes: without explicit structural guidance, INRs sacrifice structural fidelity and struggle to distinguish subtle geological features (e.g., fault edges, pinch-outs) from spectrally overlapping noise, resulting in over-smoothed structures or amplified high-frequency artifacts. Recent advances in reparameterization methods demonstrate promising noise suppression through enhanced model expressiveness. To resolve this expressiveness-stability tradeoff, we propose Stacked Tucker Implicit Seismic Reparameterization (STISR), a hybrid framework that synergizes Tucker's low-rank structural anchors with INR's high-expressiveness nonlinear approximation. Tucker decomposition in STISR provides a noise-reduced, structured initialization to guide INR optimization, effectively regularizing the neural representation to maintain coherent reflection structures. Then, the neural network nonlinearly reparameterizes the Tucker decomposition, further enhancing its expressiveness and recovering subtle features beyond linear subspace constraints. A progressive hierarchical re-decomposition strategy applies linear reparameterization to the Tucker core tensor, iteratively optimizing it across scales and

*Corresponding author. Email: 409791715@qq.com

reinforcing low-rank stability while adaptively allocating expressiveness to resolve fine-scale features. To address the heterogeneity of noise in field seismic data, we introduce l_1 regularization, which adjusts the sparsity threshold based on residual noise, enabling targeted handling of diverse noise distributions. Validation on synthetic and field pre-stack datasets confirms STISR's superiority in balancing computational efficiency, structural fidelity, and noise rejection compared to conventional tensor decomposition or pure neural network approaches.

Keywords: 3-D, Seismic data denoising, Tensor decomposition, Implicit neural representation (INR)

1. Introduction

Seismic processing techniques play a critical role in identifying, localizing, and analyzing geological reservoirs. However, as conventional reserves are rapidly depleted, exploration targets have shifted toward increasingly complex geological formations requiring advanced extraction technologies, leading to higher demands on high signal-to-noise ratio (SNR) seismic data [1]. Therefore, noise suppression has become increasingly important in seismic data processing [2, 3].

Parameterization methods [4, 5, 6] describe the characteristics of seismic data through mathematical models or specific parameter sets, thereby enabling the effective separation of signal and noise. Parameterization methods can be divided into linear and nonlinear categories.

Low-rank decomposition is a classic linear parameterization method that exploits the low-rank nature of clean signals, as additive noise increases data rank, facilitating noise reduction. Unlike traditional matrix-based low-rank methods [7, 8], tensor-based low-rank methods extend this concept to multidimensional data [9, 10]. For example, Tucker decomposition [11, 12] factorizes higher-order tensors into a core tensor and factor matrices, capturing key data features while removing noise. However, as a linear representation method, it has certain limitations when dealing with data that contains complex nonlinear features (such as intricate faults, fractures, and pore structures) and strong noise interference. Enhancing Tucker decomposition's nonlinear expressiveness and robustness remains a key challenge.

In contrast, nonlinear parameterization methods achieve denoising by reconstructing the underlying clean signal via nonlinear representations. These methods have been widely applied in deep learning [13, 14] and rely on activation functions [15, 16], such as ReLU, Sigmoid, Tanh, and Sine, to help neural networks learn nonlinear patterns in complex tasks. Nonlinear parameterization compensates for the limitations of linear methods in fitting nonlinear features. However, without explicit structural guidance, nonlinear parameterization may sacrifice structural fidelity in high-frequency cases, making it difficult to distinguish complex geological features (e.g., fault edges, pinch-outs) from overlapping noise.

Recently, to address the challenge that weak seismic signals are easily overwhelmed by noise under complex geological conditions, several denoising and enhancement methods have been proposed. For instance, an adaptive high-dimensional progressive denoising approach enhances weak reflection events while gradually suppressing noise in high-dimensional seismic data [17]. Moreover, physically constrained unsupervised seismic data registration methods preserve structural consistency without labeled data, offering new perspectives for complex seismic data processing [18].

Implicit parameterization methods, such as Implicit Neural Representations (INR) [19], have also shown great success in seismic data denoising [20] and interpolation [21, 22]. INR leverages positional encoding and implicit functions to approximate data details while preserving the intrinsic properties of high-dimensional seismic wavefields. Its coordinate encoding mechanism enables flexible scalability to 3D and higher-dimensional data, avoiding the storage challenges of explicit tensor representations. Additionally, its continuously differentiable nature allows integration with traditional regularization methods. However, it still suffers from the potential drawbacks of nonlinear parameterization, which may lead to the over-smoothing of structures or the amplification of high-frequency artifacts.

A core issue in current research is how to precisely characterize complex geological features while preserving the overall structure of seismic data. In addition to designing new network architectures, reparameterization is a widely used technique in machine learning [23, 24]. By mapping the original problem to a new parameter space or introducing new constraints, reparameterization methods not only simplify the optimization process but also improve the quality of model outputs [25, 26]. Reparameterization methods can also be divided into linear and nonlinear types.

Sparse reparameterization [27, 28] is a form of linear reparameterization that enhances denoising by leveraging noise sparsity. In seismic data processing, certain types of noise (such as irregular noise and partially coherent noise) exhibit sparse characteristics, which can be effectively removed using sparse representation techniques, such as l_1 norm [29] and Robust Principal Component Analysis (Robust PCA) [30]. Unlike traditional PCA, Robust PCA introduces a sparse term to separate data into low-rank components (representing useful signals) and sparse components (representing noise or outliers), enabling precise signal extraction in complex and sparse noise environments. Low-rank reparameterization [31, 32] is another linear approach that relies on the low-rank characteristics of parameters, reducing redundant information through matrix (such as PCA [33], SVD [34]) or tensor (CP [35], Tucker [36]) decomposition techniques, thus effectively compressing the model size and improving its robustness. For example, applying an additional Tucker decomposition to the core tensor [37] improves reconstruction performance compared to single-layer decomposition, demonstrating that multi-layer tensor decomposition networks not only enhance model interpretability but also more effectively capture the intrinsic structural features within the data [38, 39].

Nonlinear reparameterization methods enhance data feature representation

by applying further nonlinear transformations to the parameters. Common techniques include diffusion-based reparameterization [40] and gradient reparameterization [41, 42]. These methods allow the model to learn more complex representations from the latent space, enhancing their generalization and nonlinear expression ability, particularly when facing unknown or complex data.

Recent studies have explored integrating low-rank structures with continuous function representations to balance expressive power and stability. By introducing low-rank tensor function representations, Tucker decomposition is combined with INR for implicit modeling of high-dimensional data, reducing parameter redundancy while improving multidimensional data recovery performance [43, 44]. These studies indicate that coupling low-rank constraints with functional representations can effectively alleviate the lack of structural regularization in purely nonlinear models.

Motivated by this, we propose a hybrid framework, Stacked Tucker Implicit Seismic Reparameterization (STISR), to resolve the trade-off between expressiveness and stability. This method applies nonlinear reparameterization to the factor matrices of Tucker decomposition using an INR with a sine activation function while using linear reparameterization for the core tensor, as shown in Fig. 1. Tucker decomposition in STISR provides a noise-reduced, structured initialization to guide INR optimization, effectively regularizing the neural representation to maintain coherent reflection structures. Subsequently, the neural network refines the Tucker expressiveness by recovering subtle features beyond the constraints of linear subspaces. Additionally, statistical analysis over the entire dataset indicates that random noise in the field CRP gathers exhibits mixed distributions, with both sparse and approximately Gaussian components; Fig. 2 provides a representative example. Inspired by Robust PCA, we introduced the l_1 regularization mechanism to improve the denoising ability of the model. Validation on synthetic and field pre-stack datasets confirms STISR's superiority in balancing computational efficiency, structural fidelity, and noise rejection compared to conventional tensor decomposition or pure neural network approaches. The main contributions of this paper are as follows:

1. A nonlinear reparameterization is proposed, using INR to represent the Tucker factor matrices, combining the low-rank constraints of Tucker decomposition with the high-expressiveness nonlinear approximation of INR to achieve efficient denoising and feature recovery for seismic data.
2. A progressive re-decomposition strategy is proposed, which applies linear reparameterization to the Tucker core tensor and optimizes it across scales to balance low-rank stability and expressiveness.
3. An l_1 regularization mechanism is introduced to adjust the sparsity threshold based on residual noise characteristics, enhancing the model's adaptability to heterogeneous noise in field seismic data.

The structure of this paper is arranged as follows: Section II first introduces the tensor basics and notation used in this paper, followed by a detailed explanation of the Tucker-INR and STISR methods, and outlines the optimization

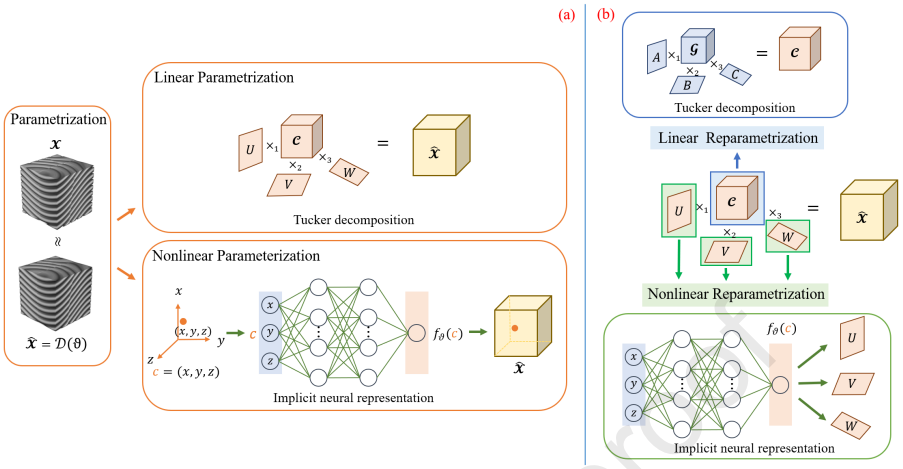

Figure 1: Linear and Nonlinear Parameterization and Reparameterization Methods. (a) Linear (Tucker) and nonlinear (INR) parameterization methods. (b) Linear (Tucker) reparameterization is applied to the core tensor, and nonlinear (INR) reparameterization is applied to the factor matrices of Tucker decomposition.

process of the STISR method. Section III presents the reconstruction experimental results on synthetic and field seismic data, verifying the superiority of the proposed method in terms of performance and computational efficiency. Finally, Section IV summarizes the main contributions of this paper and proposes future research directions.

2. Method

We propose an unsupervised 3D seismic data denoising method that integrates Tucker decomposition [45], and INR [46, 47, 48] to construct the Stacked Tucker Implicit Seismic Reparameterization (STISR) network.

Lowercase and uppercase bold letters (e.g., $\mathbf{x}$ and $\mathbf{X}$) are used to represent vectors and matrices. In contrast, uppercase bold calligraphic letters (e.g., $\boldsymbol{\mathcal{X}}$) denote higher-order tensors. For a tensor $\boldsymbol{\mathcal{X}}$, $\boldsymbol{\mathcal{X}}_p$ represents the p-mode unfolding of the tensor, and $\boldsymbol{\mathcal{X}} \times_p \mathbf{A}$ denotes the p-mode product of tensor $\boldsymbol{\mathcal{X}}$ with matrix $\mathbf{A}$. The Frobenius norm is denoted as $\|\cdot\|_F$, and the l_1 norm is defined as $\|\mathbf{X}\|_{l_1} = \sum_{ij} |\mathbf{X}_{ij}|$. Additionally, $\mathbb{R}$ represents the real field.

2.1. Pre-stack noise attenuation with parametrization

The 3D seismic data is typically modeled as the superposition of useful signals and noise, which can generally be categorized into coherent and random noise [49]. Recently, research has shown that combining the l_2 and l_1 norms can more effectively remove noise [50, 51]. The l_1 norm handles sparse noise, while the l_2 norm helps smooth the signal and remove gaussian random noise. For the

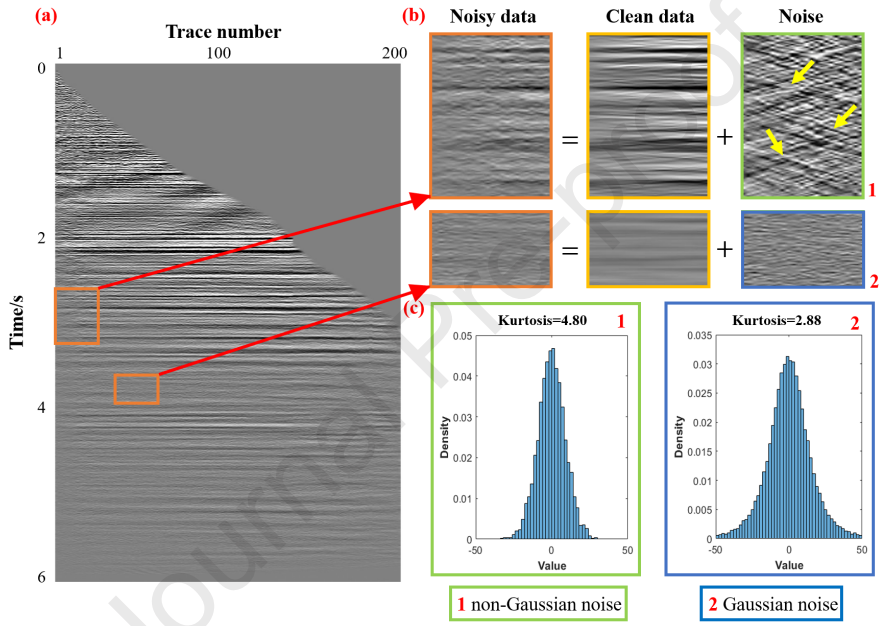

Figure 2: Noise statistical analysis is conducted over the entire field CPR dataset, with this figure shown as a representative example. The clean data in this figure are obtained using continuous wavelet denoising to facilitate more reliable statistical analysis. (a) Field seismic data; (b) a zoomed-in view of the orange box in (a); and (c) the probability density distribution of the noise in (b). The results show that part of the random noise has a kurtosis of 4.80, indicating a super-Gaussian distribution (kurtosis $>$ 3) and sparsity, whereas another part exhibits a kurtosis of 2.88, approximately following a Gaussian distribution (kurtosis $\approx$ 3). The yellow arrows indicate random interference.

given CRP gather, shown in Fig. 2, part of the random noise exhibits a kurtosis of 4.80, indicating a highly sparse super-Gaussian distribution, whereas another part shows a kurtosis of 2.88, which is close to that of a Gaussian distribution. Therefore, we model the 3D seismic data $\mathcal{Y} \in \mathbb{R}^{N_x \times N_y \times N_z}$ as:

$$\mathcal{Y} = \mathcal{X} + \mathcal{E} + \mathcal{N} \tag{1}$$

where N_x, N_y and N_z represent the number of sampling points along x, y and z dimensions, $\mathcal{X} \in \mathbb{R}^{N_x \times N_y \times N_z}$ represents the useful signal, $\mathcal{E} \in \mathbb{R}^{N_x \times N_y \times N_z}$ denotes random noise with sparsity, and $\mathcal{N} \in \mathbb{R}^{N_x \times N_y \times N_z}$ corresponds to Gaussian random noise. To extract the useful signal $\mathcal{X}$, we parameterize it in an optimizable form:

$$\mathcal{X} = \mathcal{D}(\vartheta) \tag{2}$$

where $\mathcal{D}(\vartheta)$ represents the surrogate model that captures essential features of clean data, and ϑ represents the model parameters, including weights and biases. By adjusting the parameters ϑ in the parameterized space, the model can better approximate the clean seismic data $\mathcal{X}$, thereby achieving reconstruction.

Substituting the parameterized expression (2) into the original model equation (1) yields:

$$\mathcal{Y} = \mathcal{D}(\vartheta) + \mathcal{E} + \mathcal{N} \tag{3}$$

The parameter optimization can be achieved by minimizing the following objective function:

$$\min_{\vartheta} \|\mathcal{Y} - \mathcal{D}(\vartheta) - \mathcal{E}\|_F^2 + \gamma_1 \|\mathcal{E}\|_{l_1} \tag{4}$$

where $\|\mathcal{E}\|_{l_1}$ is a sparsity constraint term for sparse noise [30], and γ_1 is hyperparameter for noise constraints. As Fig. 2 shows, the part of random noise in the given CRP gather exhibits sparsity. Inspired by Robust PCA, we introduce an l_1 regularization mechanism to adjust the sparsity threshold based on the characteristics of heterogeneous noise, thereby enhancing the suppression of seismic noise.

Parameterization methods can be classified into linear and nonlinear categories. Taking 3D data as an example, as shown in Fig. 1a.

2.1.1. Linear Parameterization

Linear parameterization approaches like Tucker decomposition effectively impose low-rank constraints to isolate structured seismic reflections. The useful seismic signals in specific pre-stack gathers, such as CRP, typically exhibit high correlation and low-rank characteristics, whereas the noise components are mostly unstructured [52]. Tucker decomposition leverages the low-rank nature of signals by constraining the rank of the core tensor to represent the signal, as shown in Fig. 1a, while noise increases the rank of the data. Specifically, the data reconstruction using Tucker decomposition is:

$$\mathcal{D}(\vartheta) = \hat{\mathcal{X}} = \mathcal{C} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W} \tag{5}$$

Table 1: Core tensors size with Rank (A_x, A_y, A_z)

Tucker-INR	STISR
$A_x \times A_y \times A_z$	$A_z/2 \times A_y/2 \times A_z/2 + \sum_{i\in\{x,y,z\}} A_i/2 \times A_i$

where ϑ represents the optimization parameters, including the core tensor $\mathcal{C} \in \mathbb{R}^{R_x \times R_y \times R_z}$ and three factor matrices $\mathbf{U} \in \mathbb{R}^{N_x \times R_x}$, $\mathbf{V} \in \mathbb{R}^{N_y \times R_y}$, and $\mathbf{W} \in \mathbb{R}^{N_z \times R_z}$. Here, R_x, R_y and R_z denote the ranks of $\hat{\mathcal{X}}$. This method reduces dimensionality and computational complexity through low-rank representation, enabling efficient convergence. However, traditional Tucker decomposition involves repeatedly computing the SVD of unfolded matrices at each mode, resulting in high computational cost and slower convergence when handling large, high-dimensional data[9, 53]. In addition, it lacks the nonlinear expressiveness required for complex geological features.

2.1.2. Nonlinear Parameterization

Nonlinear implicit neural representations (INR) achieve enhanced expressiveness through continuous nonlinear mappings, leveraging spectral bias to suppress high-frequency noise. It models data coordinates directly with a Multilayer Perceptron (MLP), mapping continuous space coordinates to seismic data values [48], as shown in Fig. 1b. Specifically, for an input $\mathbf{c} \in \mathbb{R}^d$, the network f_ϑ maps it to the output $f_\vartheta(\mathbf{c}) \in \mathbb{R}^m$, which can be expressed as:

$$f_\vartheta(\mathbf{c}) = \sigma(\mathbf{W}_L \cdot \sigma(\mathbf{W}_{L-1} \cdots \sigma(\mathbf{W}_1 \cdot \mathbf{c}))) \tag{6}$$

where d denotes the dimensionality of the input vector $\mathbf{c}$, m represents the dimensionality of the output vector $f_\vartheta(\mathbf{c})$, $\{W_l\}_{l=1}^{L}$ are the weight matrices for each layer, and σ is the activation function. INR can adaptively learn complex non-stationary signals and reconstruct the data structure through this process. However, INR's enhanced expressiveness can be a limitation in high-frequency regions: without explicit structural guidance, it may sacrifice structural fidelity and struggle to distinguish subtle geological features (e.g., fault edges, pinchouts) from spectrally overlapping noise, resulting in over-smoothed structures or amplified high-frequency artifacts [46].

2.2. STISR method

To address the trade-off between expressive power and stability, we use Tucker decomposition as the main framework, combined with linear and nonlinear reparameterization, as shown in Fig. 1b. Specifically, the factor matrices are nonlinear reparameterized through an MLP, following Tucker-INR formulations [43], as:

$$\mathcal{D}(\vartheta) = \mathcal{X}_{\text{Tucker-INR}} = \mathcal{C} \times_1 f_{\theta_x}(\mathbf{v}_x) \times_2 f_{\theta_y}(\mathbf{v}_y) \times_3 f_{\theta_z}(\mathbf{v}_z) \tag{7}$$

where $\mathcal{X}_{\text{Tucker-INR}}$ represents the reconstructed useful signal using Tucker-INR, $\mathbf{v}_x$, $\mathbf{v}_y$, $\mathbf{v}_z$ are the coordinate sets of the tensor along the x, y, z directions,

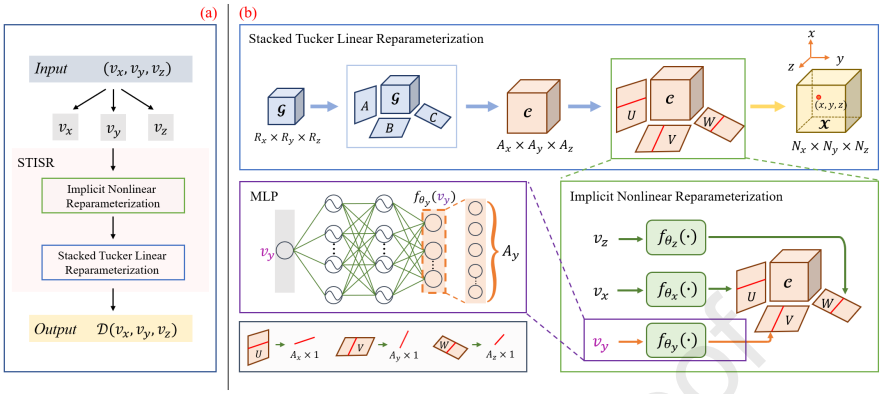

Figure 3: Network architecture for the proposed STISR method. (a) The schematic illustration of the overall workflow. (b) Detailed architectural components of the proposed model.

and f_{θ_x}, f_{θ_y}, f_{θ_z} are MLP networks reparameterizing the factor matrices. The parameter set ϑ includes the core tensor $\mathcal{C}$ and the parameters θ_x, θ_y, and θ_z.

The core tensor in Tucker decomposition effectively models the intrinsic structure of the original data through its interaction with the factor matrices. Its elements can be viewed as the weights and correlations between features across different dimensions. However, as the data scale increases, the parameter count of the core tensor and multiple factor matrices also increases, leading to a significant computational burden that impacts both training efficiency and model accuracy. To enhance the expressive power for complex data modeling while reducing the number of network parameters, we build upon the Tucker-INR framework [43] and introduce Stacked Tucker decomposition [54] to further reparameterize the core tensor. This method iteratively optimizes the Tucker core tensor across scales, reinforcing low-rank stability while adaptively allocating expressiveness to resolve fine-scale features. The core tensor is expressed as the product of a smaller core tensor and multiple factor matrices, as shown in Fig. 3. Specifically, the core tensor $\mathcal{C}$ is further decomposed as:

$$\mathcal{C} = \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} \tag{8}$$

where $\mathcal{G}$ is the new core tensor, and $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are the factor matrices. The rank of $\mathcal{G}$ is set to half of the rank of $\mathcal{C}$. Core tensors size for the different methods in Table 1.

Substituting Eq. (8) into Eq. (7), the data reconstruction process is further represented as:

$$\begin{aligned} \mathcal{D}(\vartheta) = \mathcal{X}_{\text{STISR}} = (\mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}) \\ \times_1 f_{\theta_x}(\mathbf{v}_x) \times_2 f_{\theta_y}(\mathbf{v}_y) \times_3 f_{\theta_z}(\mathbf{v}_z) \end{aligned} \tag{9}$$

where $\mathcal{X}_{\text{STISR}}$ represents the reconstruction result of our proposed method, and ϑ includes the new core tensor $\mathcal{G}$, factor matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$, and parameters θ_x, θ_y, and θ_z.

Based on the proposed STISR model, we construct the following optimization problem:

$$\min_{\vartheta} \|\mathcal{Y} - \mathcal{X}_{\text{STISR}} - \mathcal{E}\|_F^2 + \gamma_1 \|\mathcal{E}\|_{l_1} + \gamma_2 \|\mathcal{X}_{\text{STISR}}\|_{\text{TV}} \tag{10}$$

where $\mathcal{X}_{\text{STISR}}$ is defined as in Eq. (9). For conventional 3D seismic data, we employ the classic anisotropic total variation (TV) regularization [55], which is expressed as:

$$\begin{aligned} \|\mathcal{X}\|_{\text{TV}} = & \sum_{i=1}^{n_1-1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} |\mathcal{X}_{(i+1,j,k)} - \mathcal{X}_{(i,j,k)}| \\ & + \sum_{i=1}^{n_1} \sum_{j=1}^{n_2-1} \sum_{k=1}^{n_3} |\mathcal{X}_{(i,j+1,k)} - \mathcal{X}_{(i,j,k)}| \\ & + \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3-1} |\mathcal{X}_{(i,j,k+1)} - \mathcal{X}_{(i,j,k)}| \end{aligned} \tag{11}$$

This regularization term enhances the structural continuity of the reconstructed signal by minimizing the differences between adjacent pixels, thereby improving the denoising effect. For specific seismic data, as shown in Fig. 2, the spatial relationship between the time and inline profiles is stronger, exhibiting horizontal reflection events. At the same time, the correlation with the crossline is weaker. Therefore, we utilize structural priors to construct a 2D anisotropic TV regularization. The formula is as follows:

$$\begin{aligned} \|\mathcal{X}\|_{\text{TV}} = & \sum_{k=1}^{n_3} (\sum_{i=1}^{n_1-1} \sum_{j=1}^{n_2} |\mathcal{X}_{(i+1,j,k)} - \mathcal{X}_{(i,j,k)}| \\ & + \sum_{i=1}^{n_1} \sum_{j=1}^{n_2-1} |\mathcal{X}_{(i,j+1,k)} - \mathcal{X}_{(i,j,k)}|)) \end{aligned} \tag{12}$$

2.3. Algorithm Optimization

To address the denoising problem, we adopt the Alternating Minimization algorithm [56], a widely used optimization strategy that is particularly effective for multivariable decomposition problems. The algorithm iteratively approaches the global optimal solution by alternately fixing certain variables and updating the others. Specifically, the alternating optimization process can be divided into the following two subproblems:

$$\min_{\vartheta} \left\|\mathcal{Y} - \mathcal{X}_{\text{STISR}} - \mathcal{E}^t\right\|_F^2 + \gamma_2 \|\mathcal{X}_{\text{STISR}}\|_{\text{TV}} \tag{13a}$$

$$\min_{\mathcal{E}} \left\|\mathcal{Y}^t - \mathcal{X}_{\text{STISR}}^t - \mathcal{E}\right\|_F^2 + \gamma_1 \|\mathcal{E}\|_{l_1} \tag{13b}$$

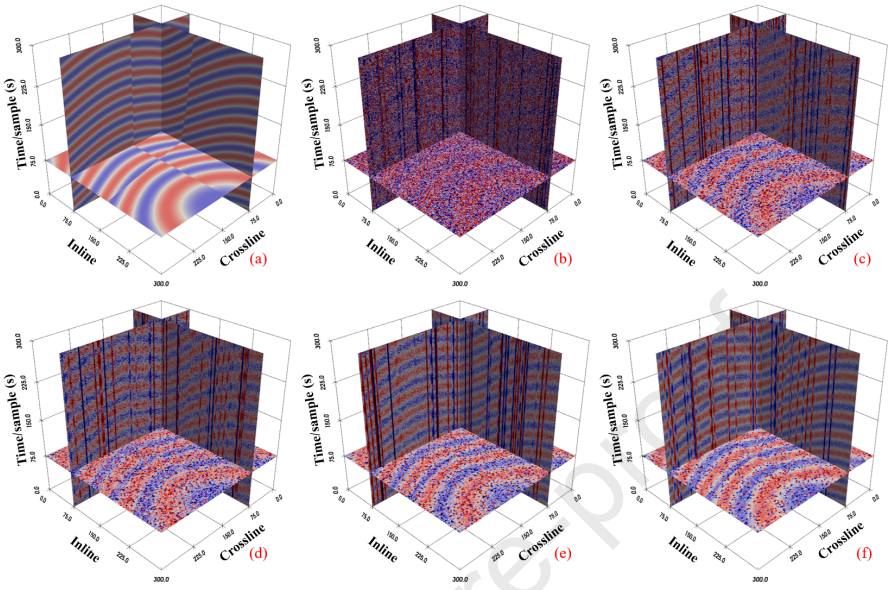

Figure 4: Synthetic data at different noise levels. (a) Clean data. (b) Noisy data with $\sigma = 0.1$. (c) Noisy data with $\sigma = 0.5$. (d) Noisy data with $\sigma = 1$. (e) Noisy data with $\sigma = 2$. (f) Noisy data with $\sigma = 3$.

2.3.1. Optimizing Parameter Set ϑ and Reconstructed Tensor $\mathcal{X}_{STISR}$

In Subproblem 13a, we fix the noise term $\mathcal{E}^t$ and update the parameter set ϑ and the reconstructed tensor $\mathcal{X}_{\text{STISR}}$.

Step 1: Updating Parameter Set ϑ

The parameter set $\vartheta = \{\mathcal{G}, \mathbf{A}, \mathbf{B}, \mathbf{C}, \theta_x, \theta_y, \theta_z\}$ can be updated using gradient descent. The update rule is expressed as:

$$\vartheta^{(t+1)} = \vartheta^{(t)} - \eta \nabla_{\vartheta} \mathcal{D}(\vartheta^{(t)}) \tag{14}$$

where ϑ^t represents the model parameters at iteration t, η is the learning rate, and $\mathcal{D}(\vartheta)$ is the network model. Common gradient-based optimization algorithms such as gradient descent or Adam [57] can be used for updating these parameters.

Step 2: Updating Reconstructed Tensor $\mathcal{X}_{STISR}$

The reconstructed tensor $\mathcal{X}_{\text{STISR}}$ is updated based on the current estimation of the core tensor and factor matrices. The updating formula is as follows:

$$\begin{aligned} \mathcal{X}_{\text{STISR}} = \mathcal{D}(\vartheta^t) = (\mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}) \\ \times_1 f_{\theta_x}(v_x) \times_2 f_{\theta_y}(v_y) \times_3 f_{\theta_z}(v_z) \end{aligned} \tag{15}$$

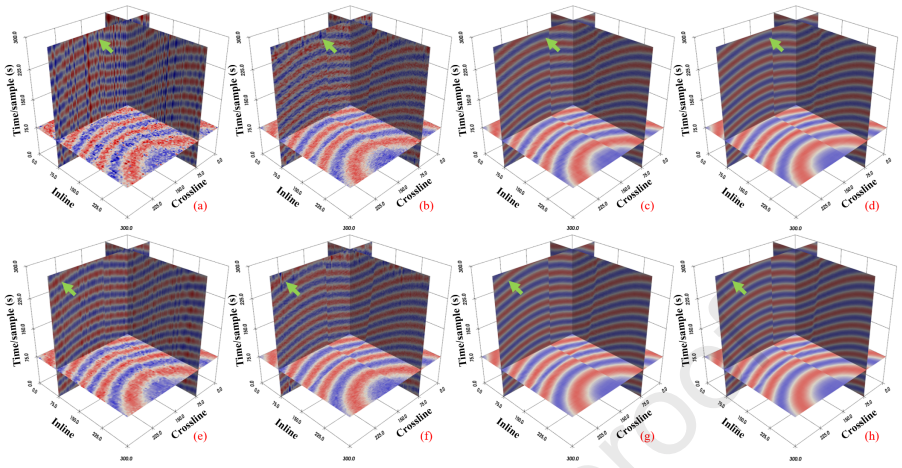

Figure 5: Denoising results at different noise levels. (a)-(d) show results for $\sigma = 0.5$ using (a) DIP, (b) MSSA, (c) Tucker-INR, and (d) STISR. (e)-(h) shows results for $\sigma = 2$ using (e) DIP, (f) MSSA, (g) Tucker-INR, and (h) STISR. The areas indicated by the green arrows retain residual erratic noise in DIP and MSSA, while no such remnants are present in Tucker-INR and STISR.

2.3.2. Optimizing the Noise Term $\boldsymbol{\mathcal{E}}$

In Subproblem 13b, we fix the tensor $\boldsymbol{\mathcal{X}}_{\text{STISR}}$ and parameter set ϑ^t, then update the noise term $\boldsymbol{\mathcal{E}}$.

The noise term is constrained by l_1 regularization. This optimization problem can be solved using common sparse optimization algorithms such as the Coordinate Descent method [58] or the Iterative Shrinkage-Thresholding Algorithm (ISTA) [59]. Specifically, the update for the noise term can be expressed as:

$$\boldsymbol{\mathcal{E}} = \mathcal{S}(\boldsymbol{\mathcal{Y}}^t - \boldsymbol{\mathcal{X}}^t_{\text{STISR}}, \gamma_1) \tag{16}$$

where $\mathcal{S}(\cdot, \gamma_1)$ denotes the soft-thresholding operation, defined as:

$$\mathcal{S}(\cdot, \gamma_1) = \text{sign}(\cdot)\max(|\cdot| - \gamma_1, 0) \tag{17}$$

2.4. Network Architecture

The study adopts the classical MLP used in INR, as shown in Fig. 3. With strong approximation capabilities, MLPs continuously represent seismic wavefields. In tensor decomposition, they model factor matrix mappings, adapting dynamically to seismic data features and improving performance in handling high-dimensional, sparse, and noisy data. Specifically, three MLP networks,θ_x, θ_y, and θ_z, parameterize factor matrices. The formulation is:

$$f_\theta(c) = \mathbf{W}_L\sigma\left(\mathbf{W}_{L-1}\cdots\sigma(\mathbf{W}_1 c)\right) \tag{18}$$

Table 2: Denoising performance evaluation of different methods at different noise levels σ (unit dB)

	$\sigma = 0.1$		$\sigma = 0.5$		$\sigma = 1$		$\sigma = 2$		$\sigma = 3$	
Noisy data	-11.80		-7.79		-6.25		-5.67		-2.91	
Rank	Tucker-INR	STISR	Tucker-INR	STISR	Tucker-INR	STISR	Tucker-INR	STISR	Tucker-INR	STISR
10	**16.87**	16.63	**19.01**	17.18	**19.52**	16.35	**19.55**	16.44	20.19	17.26
20	16.86	**18.18**	18.92	**19.88**	19.21	19.52	19.49	**20.49**	**20.47**	19.75
30	15.83	17.71	18.23	19.19	18.57	**20.14**	19.38	20.19	20.46	**21.91**
40	15.11	17.44	17.54	18.86	17.95	19.26	18.39	19.75	19.45	21.10
50	14.48	17.29	16.97	18.56	17.27	18.94	17.75	19.72	18.83	20.64
60	13.91	16.87	16.50	18.21	16.65	18.46	17.04	19.11	18.29	19.91
70	13.60	16.12	16.19	17.88	16.18	18.15	16.62	18.64	17.64	19.56
80	13.25	15.54	15.79	17.60	15.80	17.71	16.19	18.08	17.13	19.24
90	13.00	15.36	15.54	17.21	15.52	17.40	15.83	17.74	16.79	19.02
100	12.49	15.10	15.37	16.98	15.22	17.09	15.40	17.39	16.15	18.74

where $\sigma(\cdot)$ is the nonlinear activation function, $\{\mathbf{W}_l\}_{l=1}^{L}$ are the learnable weight matrices of the MLP network, and c represents the coordinates along one of the directions (x, y, or z) of the tensor.

Like other deep learning networks, INR exhibits a noticeable low-frequency spectral shift [60, 61]. This means the network tends to prioritize learning self-similarity and low-frequency features. When leveraged appropriately, this bias makes INR effective in signal component separation. However, the limited resolution of the narrow frequency range restricts broader applications. To mitigate this spectral bias and expand the output frequency range, approaches like Fourier mapping layers [62] and sine activations [63] have been proposed.

We adopt MLP networks with sinusoidal activation functions, which are formulated as:

$$f_\theta(c) = \mathbf{W}_L \mathbf{z}_{L-1} + \mathbf{b}_L \tag{19}$$

where:

$$\begin{cases} \mathbf{z}_0 = \sin\left(\omega_0 \left(\mathbf{W}_0 c + \mathbf{b}_0\right)\right) \\ \mathbf{z}_l = \sin\left(\mathbf{W}_l \mathbf{z}_{l-1} + \mathbf{b}_l\right), \quad l = 1, \ldots, L-1 \end{cases} \tag{20}$$

where, c is the input spatial coordinate, $\mathbf{z}_0$ is the output of the first layer, and $\mathbf{z}_l$ is the output after the l-th layer of the MLP network. Each layer is defined by its respective weights $\mathbf{W}_l$ and offsets $\mathbf{b}_l$, and the sine activation function is applied element-wise at each layer for $l = 1, \ldots, L-1$. The constant ω_0 is used to rescale the initial parameters, with larger values helping the network capture higher-frequency signals, thus effectively mitigating the low-frequency spectral shift.

Notably, as shown in Fig. 3, for a single-point coordinate v_y in the y-direction, after inputting into the MLP, the network outputs a vector of length A_y. Similarly, the coordinates in the v_x and v_z directions correspond to vectors of lengths A_x and A_z, respectively. These outputs correspond to the red lines in the factor matrices. Subsequently, these red lines are mode-multiplied with the core tensor $\mathcal{C}$, and the corresponding values for the coordinates (x, y, z) in the 3D data $\mathcal{X}$ are obtained.

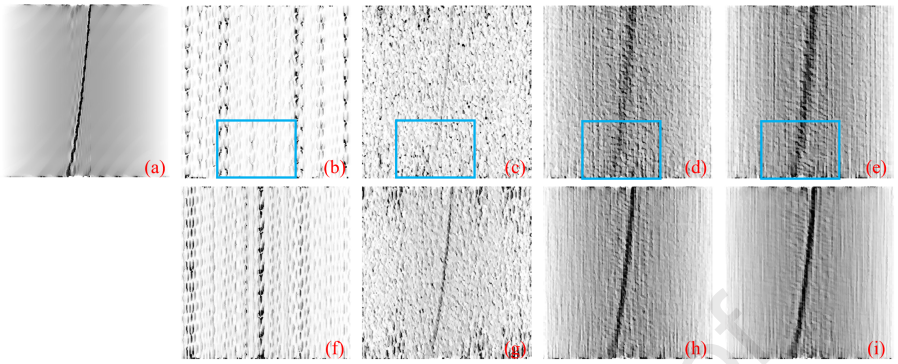

Figure 6: Fault recognition results for different methods on the crossline=100 profile. (a) represents the results for clean data, (b)-(e) show results for $\sigma = 0.5$ using (b) DIP, (c) MSSA, (d) Tucker-INR, and (e) STISR. (f)-(i) show results for $\sigma = 2$ using (f) DIP, (g) MSSA, (h) Tucker-INR, and (i) STISR. DIP fails to recover the fault structure at the locations indicated by the blue boxes, while MSSA and Tucker-INR perform worse than STISR, resulting in blurred identification.

2.5. Computational Complexity Analysis

For a 3D seismic data tensor $\mathcal{Y} \in \mathbb{R}^{N_x \times N_y \times N_z}$, the computational complexity of the proposed framework consists of:

1. Re-decomposition of the Core Tensor: The new core tensor $\mathcal{G} \in \mathbb{R}^{R_x \times R_y \times R_z}$ is multiplied with three-factor matrices $\mathbf{A} \in \mathbb{R}^{R_x \times A_x}$, $\mathbf{B} \in \mathbb{R}^{R_y \times A_y}$, and $\mathbf{C} \in \mathbb{R}^{R_z \times A_z}$, resulting in a complexity of $O(R_x R_y R_z \sum_{i \in \{x,y,z\}} A_i + \sum_{i \in \{x,y,z\}} R_i A_i)$.

2. Update of INR Factor Matrices: INR generates three-factor matrices using inputs of size $N_x \times 1$, $N_y \times 1$, and $N_z \times 1$, leading to a complexity of $O(md \sum_{i \in \{x,y,z\}} A_i N_i)$, where m represents the number of hidden units of the MLP and d is the depth.

3. Tucker Decomposition: The core tensor $\mathcal{C} \in \mathbb{R}^{A_x \times A_y \times A_z}$ is multiplied with three factor matrices $\mathbf{U} \in \mathbb{R}^{A_x \times N_x}$, $\mathbf{V} \in \mathbb{R}^{A_y \times N_y}$, and $\mathbf{W} \in \mathbb{R}^{A_z \times N_z}$, leading to a complexity of $O(A_x A_y A_z \sum_{i \in \{x,y,z\}} N_i)$.

Thus, the overall computational complexity of the algorithm is $O(R_x R_y R_z \sum_{i \in \{x,y,z\}} A_i + \sum_{i \in \{x,y,z\}} R_i A_i + md \sum_{i \in \{x,y,z\}} A_i N_i + A_x A_y A_z \sum_{i \in \{x,y,z\}} N_i)$. The overall computational complexity of Tucker-INR is $O(md \sum_{i \in \{x,y,z\}} A_i N_i + A_x A_y A_z \sum_{i \in \{x,y,z\}} N_i)$. Although our method slightly increases complexity compared to Tucker-INR, it reduces parameter count and maintains a comparable runtime (see the experimental section). More importantly, it offers superior stability and performance, making it more effective in handling complex tasks and noisy environments.

3. Experiments

We evaluate the performance of the proposed method on synthetic and field data and compare it with two advanced seismic data denoising methods, DIP [64], and MSSA [65]. Our method is implemented in PyTorch 1.7.0 and runs on an NVIDIA RTX 3090 GPU (12GB GPU memory). A learning rate of 10^{-4} is applied during the optimization process.

To quantify the performance of the method, we use the SNR on the synthetic data as the evaluation metric, defined as follows:

$$\text{SNR (dB)} = 20\log_{10}\left(\frac{A_{\text{signal}}}{A_{\text{noise}}}\right) \tag{21}$$

where A_{signal} represents the amplitude of signal, and A_{noise} represents amplitude of noise.

3.1. Analysis of denoising performance under different ranks

To evaluate the effectiveness of the proposed method, we generate a synthetic seismic dataset (size: $300 \times 300 \times 300$) with fault structures, as shown in Fig 4a. Noise with varying σ values $\sigma = 0.1, 0.5, 1, 2, 3$ and 10% erratic noise are added to the data, shown in Figs 4b to 4f. We then apply the Tucker-INR and STISR methods for denoising at different noise levels.

Both methods are tested with the same hyperparameters to ensure fairness, using the Adam optimizer with weight decay of 1, $\omega_0 = 2$, a maximum iteration count of 3001, $\gamma_1 = 0.11$, and $\gamma_2 = 1.8 \times 10^{-5}$. Additionally, the ranks R of the core tensors $\mathbf{C}$ for both methods are set to be the same. In STISR, the core tensor $\mathcal{C}$ is further decomposed into a smaller core tensor $\mathcal{G}$ and three factor matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, where the rank of $\mathcal{G}$ is set to half of the rank R of $\mathcal{C}$, as shown in Table 1. When $R = (20, 20, 20)$, the core tensor size of Tucker-INR is $20 \times 20 \times 20 = 8000$, while the core tensor size of STISR is $10 \times 10 \times 10 + 3 \times 10 \times 20 = 1600$.

To compare performance across different ranks, we test ranks from 10 to 100 with a step size of 10, recording the best reconstruction quality. The results, shown in Table 2, indicate that STISR outperforms Tucker-INR in reconstruction quality, not only in the optimal reconstruction results but also in most rank conditions, where STISR achieves 1–2 dB higher than Tucker INR, showing more stable denoising performance. STISR slightly lags behind Tucker-INR in cases of smaller rank values; this may be because STISR's rank is further decomposed to half of Tucker-INR's, resulting in an insufficient rank to capture all the features of the data. Compared to Tucker-INR, the hierarchical re-decomposition strategy of STISR significantly reduces the number of parameters, thereby reducing model complexity and has rank stability. These advantages highlight the potential and practical value of STISR in 3D seismic data processing.

3.2. Synthetic Data Example

Fig. 5 presents the denoising results of DIP, MSSA, Tucker-INR, and STISR at noise levels $\sigma = 0.5$ and $\sigma = 2$. The noisy data in Figs. 4c and 4e exhibit

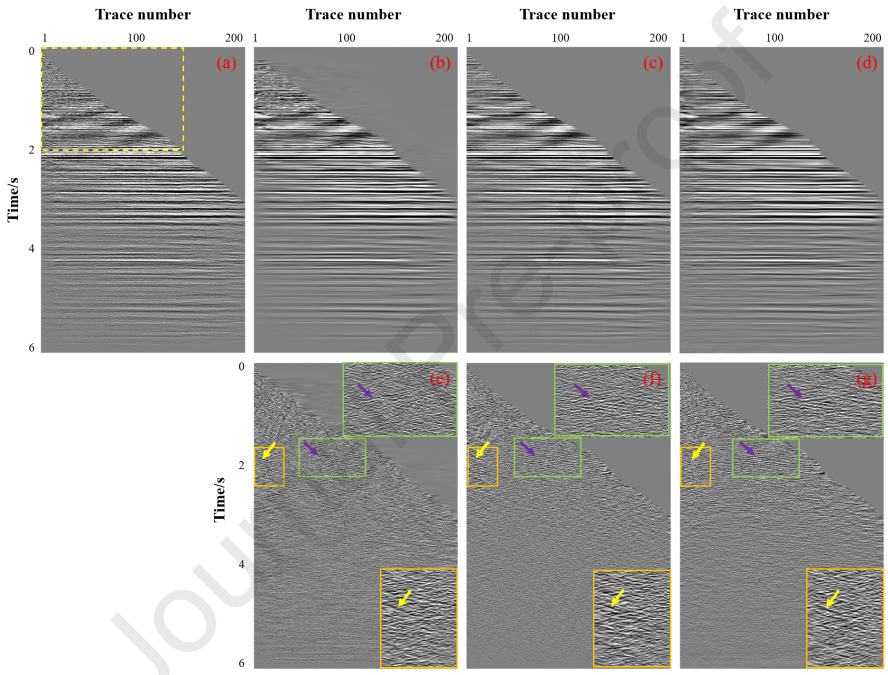

Figure 7: Denoising comparison on field seismic data. (a) Noisy dataset. Denoised using (b) DIP, (c) MSSA, and (d) the proposed method. The noise was removed using (e) DIP, (f) MSSA, and (g) the proposed method. The green and yellow boxes are magnified views of the corresponding regions. In the areas marked by yellow arrows, all three methods effectively suppress noise. However, at the locations indicated by the purple arrows, both MSSA and STISR methods attenuate stronger noise, demonstrating superior performance compared to the DIP method.

high-intensity noise that obscures the original signal, making visualization and structural interpretation challenging. The presence of erratic noise further complicates denoising. As shown in Figs. 5a, 5b, 5e, and 5f, DIP and MSSA reduce some random and erratic noise but leave noticeable residuals, as indicated by the green arrows.

In contrast, Tucker-INR and STISR show clear advantages in denoising. Tucker captures low-rank structures and removes random noise, while INR models nonlinear patterns to recover fault structures. STISR performs exceptionally well compared to Tucker-INR, with fault structures more clearly visible in Figs. 5d and 5h. This improvement is attributed to the enhanced expressiveness provided by the linear reparameterization, which better preserves the details of the wavefield. Furthermore, incorporating the l_1 norm in Tucker-INR and STISR significantly enhances noise suppression compared to DIP and MSSA, resulting in a denoising performance with no residual noise.

To better illustrate the performance of different methods in fault recovery, we performed fault recognition comparisons on the crossline = 100 profile for four methods, as shown in Fig. 6. The DIP method failed to effectively restore the fault during the denoising process, resulting in blurred fault boundaries that could not be accurately identified. In contrast, the MSSA method performed better in restoring fault structures, clearly displaying some fracture features, although noise interference still had an impact. The Tucker-INR and our proposed STISR methods showed significant advantages in preserving and recovering fault structural features. However, the blue box shows that STISR can better recover fault structures under high noise levels. This is due to the combination of linear and nonlinear reparameterization, which enhances the model's denoising capability under high-noise conditions while more effectively preserving the details of the wavefield.

The SNR comparison in Table 3 shows that STISR achieves the highest SNR at all noise levels, significantly outperforming MSSA, DIP, and Tucker-INR. Moreover, compared to MSSA and DIP, which require several hours of computation, Tucker INR and STISR complete the process in around 3 minutes. Although STISR's runtime is slightly slower than Tucker-INR, the difference is only 5 seconds. Combining Tucker decomposition and INR balances denoising performance with efficiency, and the re-decomposition strategy enhances stability. These results highlight STISR's exceptional performance in seismic data denoising and signal recovery, showcasing its potential in practical applications.

3.3. Field Data Example

To further validate the proposed method, we conduct experiments on the previously analyzed field CRP gathers of size $3001 \times 200 \times 570$ (Time, Inline, and Crossline). Fig. 2 shows a segment of the CRP gather collected at Crossline=50, which contains 200 seismic traces, each with 3001-time samples and a sampling interval of 2 ms. The data is contaminated by random noise, making weak signal components unclear, while strong reflection events exhibit approximately horizontal characteristics. The kurtosis of part of the random noise indicates sparse characteristics, making it suitable for the l_1 norm. Notably, due to the

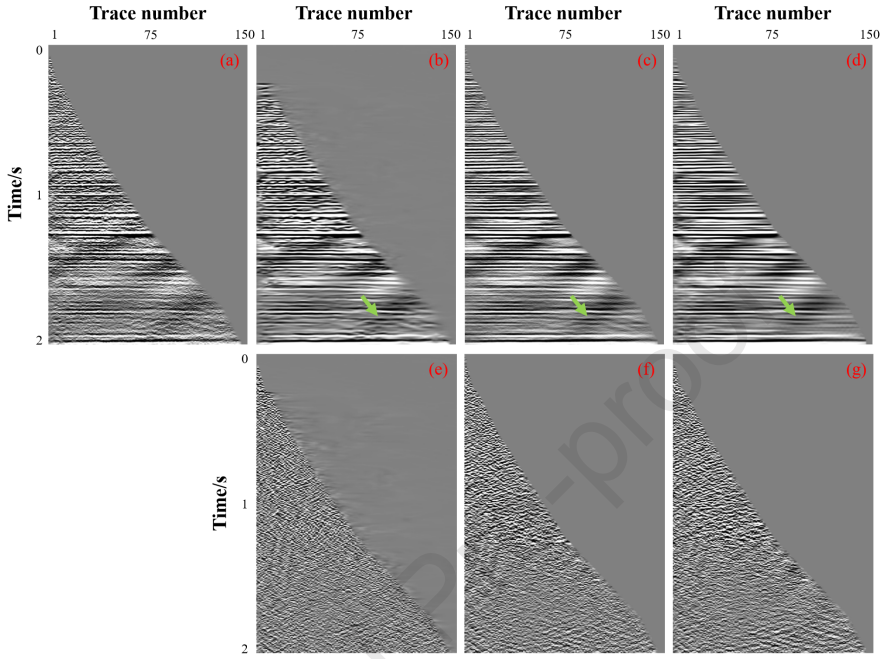

Figure 8: Denoising comparison is performed on the magnified region marked by the yellow-dotted box in Figure 7. (a) Noisy data. Denoised using (b) DIP, (c) MSSA, and (d) the proposed method. The noise was removed using (e) DIP, (f) MSSA, and (g) the proposed method. As highlighted by the green arrows, STISR effectively suppresses the residual interference exhibited in DIP and MSSA.

Table 3: Denoising performance evaluation of different methods at different noise levels σ (unit: dB)

	Noisy Data	**DIP**	**MSSA**	**Tucker-INR**	**STISR**
$\sigma = 0.1$	-11.80	-5.83	1.82	16.87	**18.18**
$\sigma = 0.5$	-7.79	2.14	7.99	19.01	**19.88**
$\sigma = 1$	-6.25	5.43	10.52	19.52	**20.19**
$\sigma = 2$	-5.67	8.68	12.46	19.55	**20.49**
$\sigma = 3$	-2.91	10.50	15.91	20.47	**21.91**
Time	–	5 h 36 m	1 h 27 m	**2 m 57 s**	3 m 2 s

Table 4: Comparison of running time on field CRP gathers

	Data size	**DIP**	**MSSA**	**STISR**
Time	$3001 \times 200 \times 570$	10 h 28 m	3 h 56 m	6 m 25 s

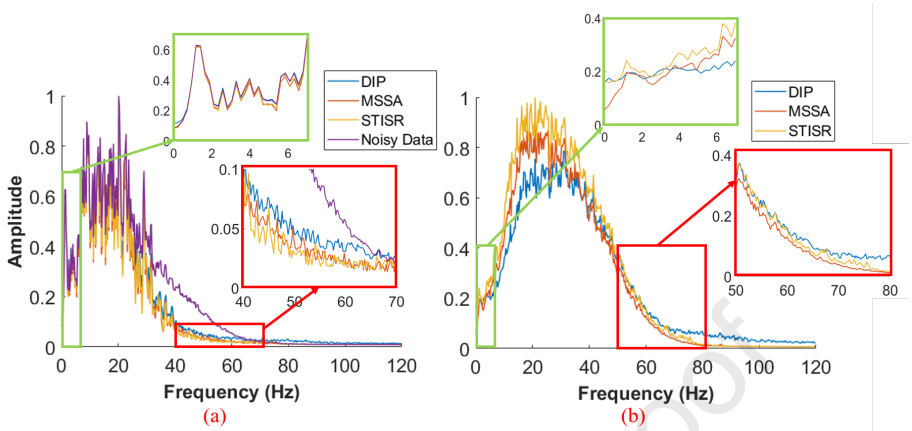

Figure 9: Normalized multichannel amplitude spectrum. (a) Denoised data. (b) Removed noise. The red and green boxes represent magnified views of the corresponding areas.

relatively horizontal alignment of reflection events, we adopt a different TV regularization compared to the model data, as shown in Eq. 12.

The denoising results of DIP, MSSA, and STISR are presented in Fig. 7. Specifically, STISR methods use the Adam optimizer with a weight decay of 0.1, $\omega_0 = 2$, a maximum iteration count of 5001, $\gamma_1 = 0.07$, and $\gamma_2 = 5 \times 10^{-6}$. Although all three methods effectively suppress noise, STISR and MSSA outperform DIP in noise attenuation. As indicated by the yellow arrows, all three methods reduce noise, while the purple arrows highlight the superior suppression achieved by MSSA and STISR compared to DIP. Tables 3 and 4 demonstrate that, in contrast to the several-hour runtime required by MSSA and DIP, STISR strikes a balance between performance and efficiency, completing denoising within six minutes even for large datasets.

In the shallow region (Fig. 8), the STISR method demonstrates superior performance by effectively removing stronger noise energy compared to the other two methods, while preserving useful signals. Additionally, it removes more intense interference, ensuring the lateral continuity of reflection events, as highlighted by the green arrows. By integrating both linear and nonlinear parameterization, STISR enhances the model's expressive capacity, enabling more precise differentiation between useful signals and noise. This approach significantly improves the suppression of both sparse noise (as indicated by the green arrows) and gaussian noise while maintaining the smoothness and continuity of the signals. Compared to DIP and MSSA, STISR achieves more effective noise removal, demonstrating the robustness of the l_1 norm in suppressing noise.

We further analyze noise suppression in the normalized multichannel amplitude spectrum in Fig. 9. In the low-frequency range, all three methods exhibit similar performance, indicating their effectiveness in preserving low-frequency components. However, as shown in Fig. 9b, STISR achieves stronger noise suppression within the primary noise frequency band compared to the other two

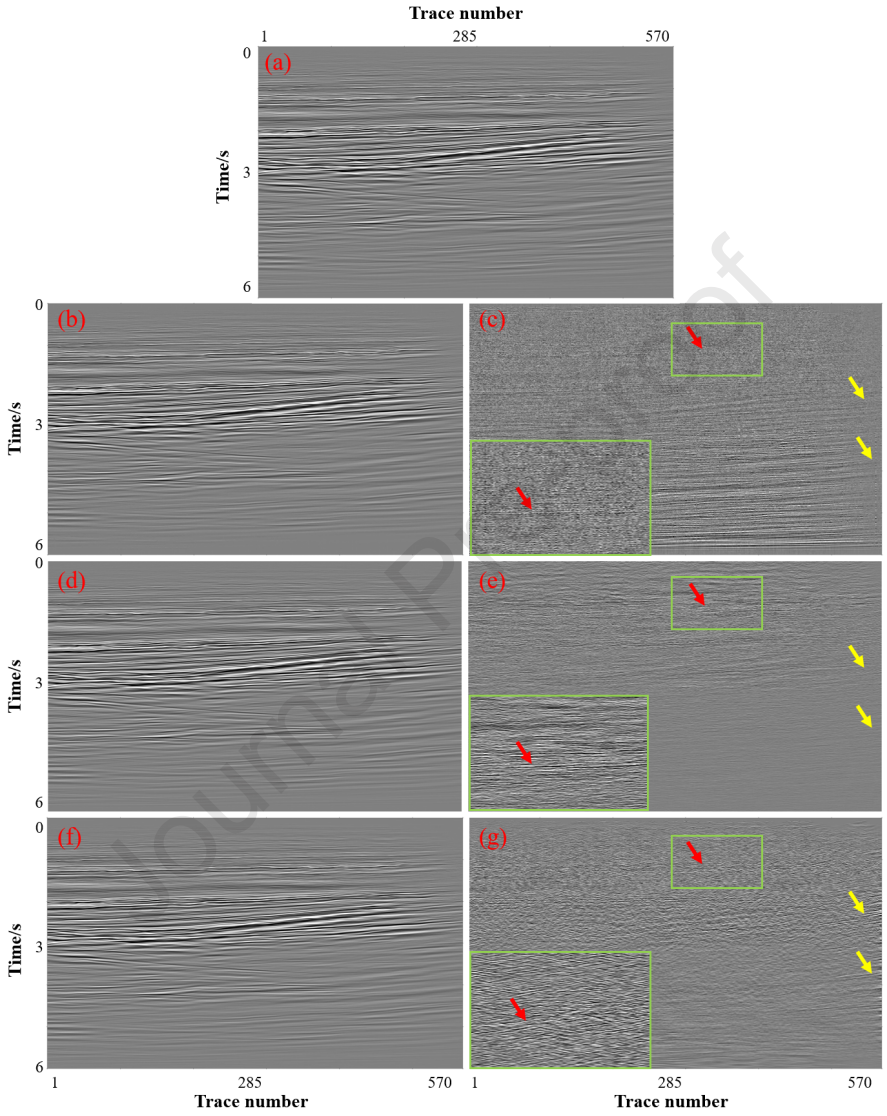

Figure 10: Denoising comparison on stacked data. (a) Stacked result of the original data. Stacked data results after pre-stack denoising using (b) DIP, (d) MSSA, and (f) the proposed method. Stacked noise results after pre-stack denoising using (c) DIP, (e) MSSA, and (g) the proposed method. The green boxes represent magnified views of the corresponding regions. As indicated by the red arrows, signal leakage is present in the DIP and MSSA methods, whereas STISR effectively suppresses this leakage. Additionally, the random noise highlighted by the yellow arrows is not effectively eliminated in the DIP and MSSA results, whereas STISR achieves effective attenuation.

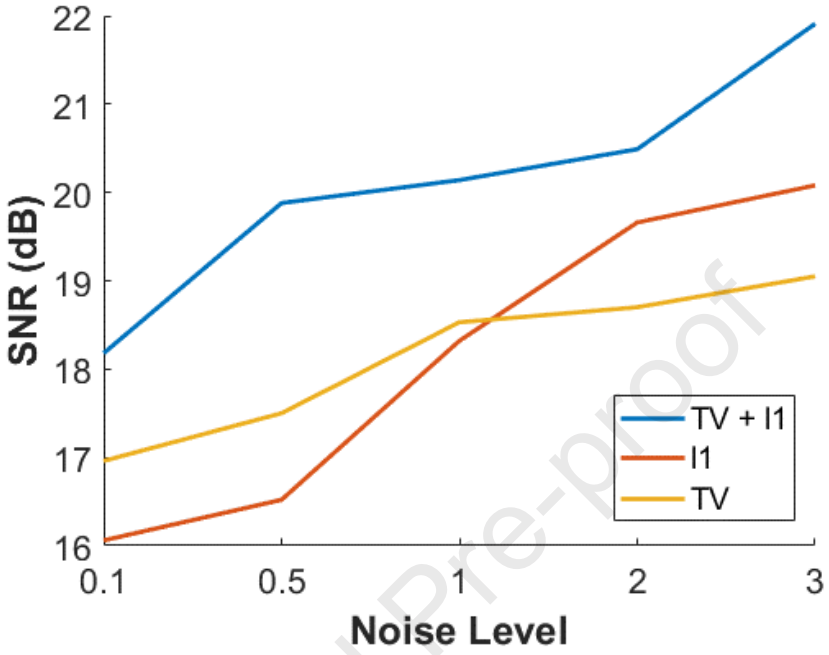

Figure 11: Quantitative performance of regularization at different noise levels σ.

methods. Overall, STISR demonstrates superior noise attenuation within the noise frequency range, highlighting its exceptional denoising capability.

To further validate fidelity and denoise performance, we present stacked data (Fig. 10) for each method. For each of the 570 Crossline coordinates in the processed three-dimensional data, we perform horizontal stacking multiple times to generate corresponding stacked data. Although all three methods effectively attenuate noise, signal leakage is observed in the DIP and MSSA methods, whereas STISR achieves the least signal leakage. Moreover, STISR not only eliminates stronger noise energy but also significantly suppresses noise, particularly in the regions highlighted by the yellow arrows, demonstrating superior performance.

3.4. Ablation Experiment

3.4.1. Impact of Regularization

To better illustrate the role of TV regularization and the l_1 norm in the proposed method, we conducted ablation experiments on synthetic data, with results shown in Fig. 11. The experimental results validate the effectiveness of TV regularization and the l_1 norm in noise suppression. When $\sigma = 0.1$ and $\sigma = 0.5$, random noise in the data is significantly more prevalent than erratic noise. However, when $\sigma = 1$, the ratio between the two types of noise is more balanced. As σ increases to 2 and 3, erratic noise becomes more prominent in the synthetic data. By comparing the experimental results, we find that

Table 5: Comparison of running time with or without INR on different data

	Data size	**Tucker**	**STISR**
Synthetic data	$300 \times 300 \times 300$	228 s	182 s
Field data	$3001 \times 200 \times 570$	960 s	325 s

experiments using only the l_1 norm outperform the TV regularization method at $\sigma = 2$ and $\sigma = 3$. This may be due to the stronger denoising ability of the l_1 norm when dealing with erratic noise. TV regularization excels in smoothing and removing random noise, while the l_1 norm is more effective in handling erratic noise.

3.4.2. Impact of INR

Fig. 12 presents the denoising results under noise levels of $\sigma = 0.5$ and $\sigma = 2$, with and without using INR. While Tucker decomposition, constrained by low-rank assumptions, removes part of the noise, it exhibits limited capability in preserving intricate structures and fine details, as indicated by the green and blue arrows. In contrast, introducing INR in the STISR framework enhances its ability to capture fine details while maintaining the low-rank structure, leading to cleaner denoised results and reduced computational time. On large-scale field data, STISR achieves denoising in just one-third of the time Tucker requires, as shown in the table 5. This demonstrates that the combination of Tucker decomposition and INR not only improves the network's ability to fit nonlinear features but also enhances the robustness of denoising at different noise levels.

4. Discussion

4.1. Noise-level analysis

Table 6: Denoising performance with fixed 10% erratic noise (unit: dB)

Input SNR (dB)	-30.23	-23.09	-20.47	-13.18
Output SNR (dB)	-0.58	6.52	9.53	14.98

Table 7: Denoising performance with fixed noise level of 0.1 (unit: dB)

Erratic noise ratio	10%	20%	30%	40%	50%
Input SNR	-11.80	-11.95	-12.14	-12.21	-12.37
Output SNR	18.18	16.40	14.19	11.95	9.78

To illustrate the applicability of the proposed method, its stability under different noise intensities is further investigated. As shown in Table 6, when the erratic noise ratio is fixed at 10%, the proposed method is able to improve the SNR even when the input SNR is as low as -30 dB. No abrupt performance collapse is observed, indicating a certain degree of robustness to strong

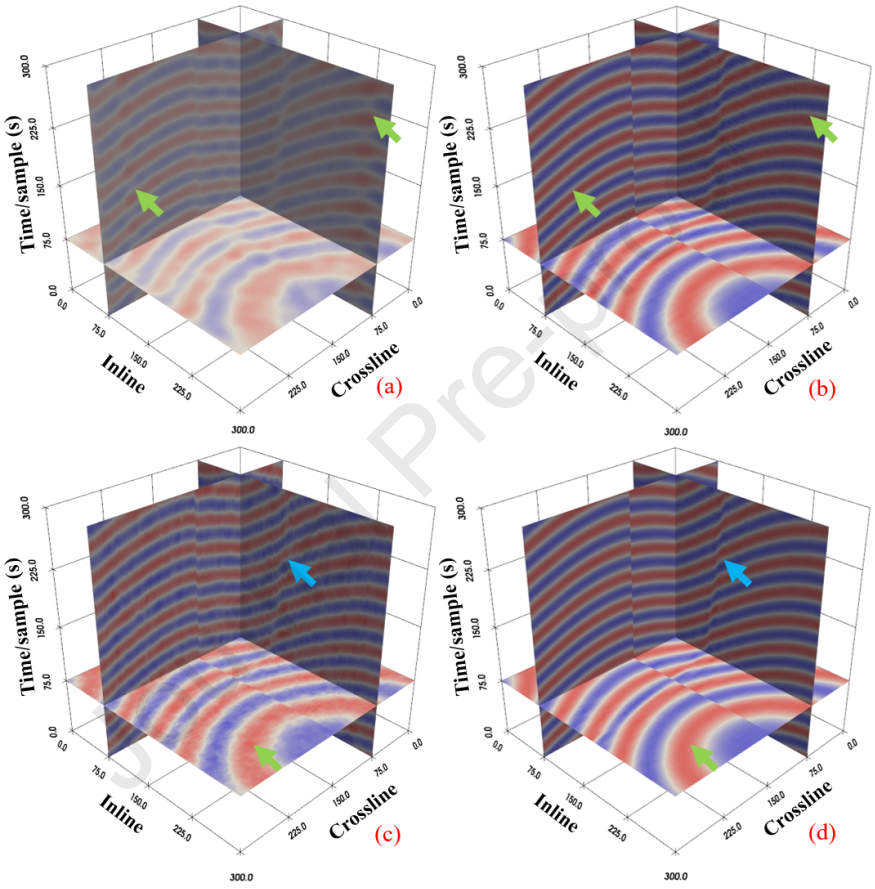

Figure 12: Denoising results with and without INR. (a)-(b) show the results for $\sigma = 0.5$: (a) Tucker decomposition, (b) STISR. (c)-(d) show the results for $\sigma = 2$: (c) Tucker decomposition, (d) STISR. As indicated by the green arrows, Tucker decomposition retains residual noise, whereas STISR effectively eliminates it. As highlighted by the blue arrows, STISR outperforms Tucker decomposition in recovering fault structures, demonstrating superior capability in detail preservation.

additive random noise. Table 7 presents the denoising results under different erratic noise ratios with a fixed noise level of 0.1. As the proportion of erratic noise increases, the output SNR gradually decreases but remains stable without sudden degradation, suggesting that the proposed method can maintain reliable denoising performance under relatively high levels of nonstationary interference.

4.2. Limitations of STISR method

Although the proposed STISR method demonstrates effective compact representation and noise attenuation for 3D seismic data, its applicability is subject to several inherent limitations. As an extension of the low-rank regularization framework, STISR implicitly assumes that seismic data are low-rank or compressible along certain dimensions. This assumption generally holds for smooth and slowly varying seismic events, enabling effective redundancy exploitation. However, in regions with steep dips, rapid local variations, or complex interference patterns, the data tend to exhibit higher-rank characteristics, which weakens the low-rank modeling capability. In practice, preprocessing techniques such as normal moveout (NMO) correction on CMP gathers can be employed to make the data more consistent with low-rank assumptions. Moreover, the global nature of Tucker decomposition relies on strong inter-dimensional correlations, which may degrade in structurally complex scenarios, leading to reduced reconstruction accuracy. Overall, STISR is more suitable for continuous and smoothly varying seismic reflections, while data containing complex geological structures generally require appropriate preprocessing to enhance their low-rank characteristics.

5. Conclusion

The proposed approach integrates Tucker decomposition with INR, where the Tucker factor matrices are modeled using INR, and a progressive hierarchical re-decomposition strategy is employed to optimize the core tensor. This design enhances the representation of fine-scale features while preserving low-rank stability. The incorporation of l_1 regularization further improves the model's adaptability to heterogeneous noise. Experiments on synthetic data demonstrate that the proposed method can effectively recover clean signals and reconstruct fault structures under low signal-to-noise ratio conditions. Compared with DIP and MSSA methods, which typically require several hours, the proposed method achieves comparable or superior denoising performance within approximately 3 minutes. Field seismic data experiments further verify the robustness of the method in suppressing both sparse and Gaussian random noise under vertically nonstationary signal conditions. Poststack results indicate that the proposed method removes noise with minimal signal leakage. Future work will explore integration with alternative tensor decomposition schemes (such as TT or TR decomposition), extension to higher-dimensional seismic data, the suppression of more complex coherent noise (e.g., linear noise and multiples), and the development of adaptive rank estimation and online processing strategies.

References

[1] S. Mandelli, V. Lipari, P. Bestagini, S. Tubaro, Interpolation and denoising of seismic data using convolutional neural networks, arXiv preprint arXiv:1901.07927 (2019).

[2] Y. Wang, Random noise attenuation using forward-backward linear prediction, Journal of Seismic Exploration 8 (2) (1999) 133–142.

[3] N. Iqbal, Deepseg: Deep segmental denoising neural network for seismic data, IEEE Transactions on Neural Networks and Learning Systems 34 (7) (2022) 3397–3404.

[4] F. Wang, B. Yang, Y. Wang, M. Wang, Learning from noisy data: An unsupervised random denoising method for seismic data using model-based deep learning, IEEE Transactions on Geoscience and Remote Sensing 60 (2022) 1–14.

[5] F. Meng, Q. Fan, Y. Li, Self-supervised learning for seismic data reconstruction and denoising, IEEE Geoscience and Remote Sensing Letters 19 (2021) 1–5.

[6] L. Zhuang, M. K. Ng, Fasthymix: Fast and parameter-free hyperspectral image mixed noise removal, IEEE Transactions on Neural Networks and Learning Systems 34 (8) (2021) 4702–4716.

[7] M. A. Davenport, J. Romberg, An overview of low-rank matrix recovery from incomplete observations, IEEE Journal of Selected Topics in Signal Processing 10 (4) (2016) 608–622.

[8] E. Kim, M. Lee, C.-H. Choi, N. Kwak, S. Oh, Efficient l_1 -norm-based low-rank matrix approximations for large-scale problems using alternating rectified gradient method, IEEE Transactions on Neural Networks and Learning Systems 26 (2) (2015) 237–251. doi:10.1109/TNNLS.2014.2312535.

[9] T. G. Kolda, B. W. Bader, Tensor decompositions and applications, SIAM review 51 (3) (2009) 455–500.

[10] J. Xue, Y. Zhao, W. Liao, J. C.-W. Chan, S. G. Kong, Enhanced sparsity prior model for low-rank tensor completion, IEEE Transactions on Neural Networks and Learning Systems 31 (11) (2020) 4567–4581. doi:10.1109/TNNLS.2019.2956153.

[11] Y.-D. Kim, S. Choi, Nonnegative tucker decomposition, in: 2007 IEEE conference on computer vision and pattern recognition, IEEE, 2007, pp. 1–8.

[12] A. Karami, M. Yazdi, A. Z. Asli, Noise reduction of hyperspectral images using kernel non-negative tucker decomposition, IEEE Journal of Selected Topics in Signal Processing 5 (3) (2011) 487–493.

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

[14] R. Tibi, P. Hammond, R. Brogan, C. J. Young, K. Koper, Deep learning denoising applied to regional distance seismic data in utah, Bulletin of the Seismological Society of America 111 (2) (2021).

[15] A. Apicella, F. Donnarumma, F. Isgrò, R. Prevete, A survey on modern trainable activation functions, Neural Networks 138 (2021) 14–32.

[16] P. Ramachandran, B. Zoph, Q. V. Le, Searching for activation functions, arXiv preprint arXiv:1710.05941 (2017).

[17] W. Wang, J. Yang, N. Qin, Z. Li, J. Huang, T. Shan, An adaptive high-dimensional progressive denoising method for seismic weak signal enhancement, IEEE Transactions on Geoscience and Remote Sensing 62 (2024) 1–10. doi:10.1109/TGRS.2024.3493135.

[18] Y. Hu, S. Yu, Seismic data registration based on a physically constrained unsupervised framework, IEEE Transactions on Geoscience and Remote Sensing 63 (2025) 1–10. doi:10.1109/TGRS.2025.3625936.

[19] Z. Li, T. Wang, J. D. Lee, S. Arora, Implicit bias of gradient descent on reparametrized models: On equivalence to mirror descent, Advances in Neural Information Processing Systems 35 (2022) 34626–34640.

[20] J. Li, D. Liu, M. D. Sacchi, Unsupervised ground-roll attenuation via implicit neural representations, Geophysics 90 (2) (2025) V101–V111.

[21] F. Wang, Z. Liu, F. Li, Q. Hu, J. Ma, Seismic data interpolation by learning a denoiser with deep implicit prior, IEEE Geoscience and Remote Sensing Letters 20 (2023) 1–5.

[22] W. Gao, D. Liu, W. Chen, M. D. Sacchi, X. Wang, Nersi: Neural implicit representations for 5d seismic data interpolation, Geophysics 90 (1) (2025) V29–V42.

[23] S. Hoyer, J. Sohl-Dickstein, S. Greydanus, Neural reparameterization improves structural optimization, arXiv preprint arXiv:1909.04240 (2019).

[24] W. S. Harlan, Regularization by model reparameterization, This website: papers/regularization. pdf (1995).

[25] T. Salimans, D. P. Kingma, Weight normalization: A simple reparameterization to accelerate training of deep neural networks, Advances in neural information processing systems 29 (2016).

[26] M. Figurnov, S. Mohamed, A. Mnih, Implicit reparameterization gradients, Advances in neural information processing systems 31 (2018).

[27] J. Xue, Y. Zhao, S. Huang, W. Liao, J. C.-W. Chan, S. G. Kong, Multilayer sparsity-based tensor decomposition for low-rank tensor completion, IEEE Transactions on Neural Networks and Learning Systems 33 (11) (2021) 6916–6930.

[28] L. Li, W. Li, Q. Du, R. Tao, Low-rank and sparse decomposition with mixture of gaussian for hyperspectral anomaly detection, IEEE Transactions on Cybernetics 51 (9) (2020) 4363–4372.

[29] D. Vidaurre, C. Bielza, P. Larranaga, A survey of l1 regression, International Statistical Review 81 (3) (2013) 361–387.

[30] E. J. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis?, Journal of the ACM (JACM) 58 (3) (2011) 1–37.

[31] Y.-Q. Zhao, J. Yang, Hyperspectral image denoising via sparse representation and low-rank constraint, IEEE Transactions on Geoscience and Remote Sensing 53 (1) (2014) 296–308.

[32] D. Yu, H. Zhang, W. Chen, J. Yin, T.-Y. Liu, Large scale private learning via low-rank reparametrization, in: International Conference on Machine Learning, PMLR, 2021, pp. 12208–12218.

[33] K. W. Jorgensen, L. K. Hansen, Model selection for gaussian kernel pca denoising, IEEE transactions on neural networks and learning systems 23 (1) (2011) 163–168.

[34] G. H. Golub, C. F. Van Loan, Matrix computations, JHU press, 2013.

[35] V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, V. Lempitsky, Speeding-up convolutional neural networks using fine-tuned cp-decomposition, arXiv preprint arXiv:1412.6553 (2014).

[36] A. Karami, M. Yazdi, G. Mercier, Compression of hyperspectral images using discerete wavelet transform and tucker decomposition, IEEE journal of selected topics in applied earth observations and remote sensing 5 (2) (2012) 444–450.

[37] S. Li, L. Cheng, T. Zhang, H. Zhao, J. Li, Striking the right balance: Three-dimensional ocean sound speed field reconstruction using tensor neural networks, The Journal of the Acoustical Society of America 154 (2) (2023) 1106–1123.

[38] D. Yu, L. Deng, F. Seide, The deep tensor neural network with applications to large vocabulary speech recognition, IEEE Transactions on audio, speech, and language processing 21 (2) (2012) 388–396.

[39] E. Newman, L. Horesh, H. Avron, M. Kilmer, Stable tensor neural networks for rapid deep learning, arXiv preprint arXiv:1811.06569 (2018).

[40] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, Advances in neural information processing systems 33 (2020) 6840–6851.

[41] A. Miller, N. Foti, A. D'Amour, R. P. Adams, Reducing reparameterization gradient variance, Advances in Neural Information Processing Systems 30 (2017).

[42] G. Tucker, D. Lawson, S. Gu, C. J. Maddison, Doubly reparameterized gradient estimators for monte carlo objectives, arXiv preprint arXiv:1810.04152 (2018).

[43] Y. Luo, X. Zhao, Z. Li, M. K. Ng, D. Meng, Low-rank tensor function representation for multi-dimensional data recovery, IEEE Transactions on Pattern Analysis and Machine Intelligence (2023).

[44] Y. Luo, X. Zhao, D. Meng, Revisiting nonlocal self-similarity from continuous representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 47 (1) (2025) 450–468. doi:10.1109/TPAMI.2024.3464875.

[45] L. R. Tucker, Some mathematical notes on three-mode factor analysis, Psychometrika 31 (3) (1966) 279–311.

[46] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, R. Ng, Nerf: Representing scenes as neural radiance fields for view synthesis, Communications of the ACM 65 (1) (2021) 99–106.

[47] D. Liu, W. Gao, W. Xu, J. Li, X. Wang, W. Chen, 5d seismic data interpolation by continuous representation, IEEE Transactions on Geoscience and Remote Sensing (2024).

[48] W. Gao, D. Liu, W. Chen, M. D. Sacchi, X. Wang, Nersi: Neural implicit representations for 5d seismic data interpolation, Geophysics 90 (1) (2024) 1–58.

[49] A. Bakulin, D. Neklyudov, I. Silvestrov, Multiplicative random seismic noise caused by small-scale near-surface scattering and its transformation during stacking, Geophysics 87 (5) (2022) V419–V435.

[50] H. Fu, M. K. Ng, M. Nikolova, J. L. Barlow, Efficient minimization methods of mixed l2-l1 and l1-l1 norms for image restoration, SIAM Journal on Scientific computing 27 (6) (2006) 1881–1902.

[51] J. Portilla, A. Tristan-Vega, I. W. Selesnick, Efficient and robust image restoration using multiple-feature l2-relaxed sparse analysis priors, IEEE Transactions on Image Processing 24 (12) (2015) 5046–5059.

[52] E. Candes, B. Recht, Exact matrix completion via convex optimization, Communications of the ACM 55 (6) (2012) 111–119.

[53] N. Vervliet, O. Debals, L. De Lathauwer, Tensorlab 3.0—numerical optimization strategies for large-scale constrained and coupled matrix/tensor factorization, in: 2016 50th Asilomar Conference on Signals, Systems and Computers, IEEE, 2016, pp. 1733–1738.

[54] D. Liu, Q. Wang, N. You, M. D. Sacchi, W. Chen, Filling the gap: Enhancing borehole imaging with a tensor neural network, Geophysics 90 (3) (2025) D71–D83.

[55] L. I. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D: nonlinear phenomena 60 (1-4) (1992) 259–268.

[56] S. Holtz, T. Rohwedder, R. Schneider, The alternating linear scheme for tensor optimization in the tensor train format, SIAM Journal on Scientific Computing 34 (2) (2012) A683–A713.

[57] P. K. Diederik, Adam: A method for stochastic optimization, (No Title) (2014).

[58] T. T. Wu, K. Lange, Coordinate descent algorithms for lasso penalized regression (2008).

[59] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM journal on imaging sciences 2 (1) (2009) 183–202.

[60] A. Jacot, F. Gabriel, C. Hongler, Neural tangent kernel: Convergence and generalization in neural networks, Advances in neural information processing systems 31 (2018).

[61] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, A. Courville, On the spectral bias of neural networks, in: International conference on machine learning, PMLR, 2019, pp. 5301–5310.

[62] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, R. Ng, Fourier features let networks learn high frequency functions in low dimensional domains, Advances in neural information processing systems 33 (2020) 7537–7547.

[63] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, G. Wetzstein, Implicit neural representations with periodic activation functions, Advances in neural information processing systems 33 (2020) 7462–7473.

[64] D. Ulyanov, A. Vedaldi, V. Lempitsky, Deep image prior, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 9446–9454.

[65] V. E. Oropeza, M. D. Sacchi, A randomized svd for multichannel singular spectrum analysis (mssa) noise attenuation, in: SEG technical program expanded abstracts 2010, Society of Exploration Geophysicists, 2010, pp. 3539–3544.

This work was supported in part by the National Key Research and Development Program of China under grant 2021YFA0716903, in part by the National Natural Science Foundation of China under grant 42374135, in part by the Fundamental Research Funds for the Central Universities under grant xzy012023073, and in part by the Innovation Capability Support Program of Shaanxi under grant No. 2021TD-08. The contribution of the second author was supported by the industrial sponsors of the Signal Analysis and Imaging Group (SAIG) consortium at the University of Alberta.
The authors declare that they have no conflicts of interest.