Journal of Scientific Computing (2023) 97:49
https://doi.org/10.1007/s10915-023-02364-6

Low-Rank Tensor Data Reconstruction and Denoising via ADMM: Algorithm and Convergence Analysis

Jonathan Popa[1] · Yifei Lou[1] · Susan E. Minkoff[1]

Received: 21 February 2023 / Revised: 18 August 2023 / Accepted: 18 September 2023

Abstract

Seismic data is contaminated by noise due to a variety of factors including wind, ocean currents, vehicular traffic, and construction. Further receiver malfunction and limitations in acquisition geometry may lead to missing data. Seismic data naturally contains redundancies because surveys repeatedly sample the same region of the subsurface. If this data is organized into a high-dimensional data structure (a tensor), then the incomplete noisy data will have higher rank than the ground truth data. Direct rank reduction for tensors is an NP-hard problem. Thus one instead minimizes an approximation to rank such as the tensor nuclear norm (TNN), which sums the tensor's singular values. This TNN model together with the alternating direction method of multipliers (ADMM) optimization method is denoted TNN-ADMM. While convergence of ADMM has been shown previously for general minimization of functions, here we formulate tensor properties which allow us to prove a sufficient condition: by bounding the step size to be greater than one, the Lagrangian function is strictly decreasing and bounded. Therefore, ADMM applied to the TNN model converges. This condition reduces the computational burden of tuning the step size parameter. We demonstrate the performance of denoising and reconstruction on two synthetic seismic datasets and on a real dataset from offshore Western Australia. In the case of the linear synthetic data, TNN-ADMM out-performs another popular method by giving four times less error in half the computational time. We also illustrate that TNN-ADMM can effectively remove the majority of artificially added noise to real data.

Keywords ADMM convergence · Tensor reconstruction · Tensor denoising · Seismic data completion

Mathematics Subject Classification 65F55 · 15A83 · 15A72 · 15A69

✉ Susan E. Minkoff
sminkoff@utdallas.edu

Jonathan Popa
jonathan.popa1@utdallas.edu

Yifei Lou
yifei.lou@utdallas.edu

[1] Department of Mathematical Sciences, The University of Texas at Dallas, 800 W. Campbell Rd, Richardson, TX 75080, USA

Published online: 07 October 2023

1 Introduction

When recording seismic data, noise is usually present in the recording due to environmental factors and man-made sources. Examples of noise include weather, ocean currents, construction, and vehicular traffic. Additionally, data may be incomplete due to limitations in source and receiver placement and the range of frequencies possible for these devices. To mitigate these issues, simultaneous reconstruction and denoising methods have been developed to recover the desired data [10, 26].

We are interested in a class of reconstruction and denoising methods based on minimizing rank. The rank of a matrix is the number of linearly independent rows. Generalized definitions of rank exist for tensors, or multi-dimensional arrays, such as Tucker rank, tubal rank, and tensor train rank [10, 17, 21]. In the absence of noise, the complete data often has redundancies and hence has low rank. Noise and missing entries generally lead to observations of higher rank. These factors motivate low rank methods for recovering data [4]. As minimizing rank directly is NP-hard [4], two popular alternate approaches are minimizing rank approximations [4, 20, 22, 34] and factorizing a matrix into the product of two lower-rank matrices [8, 16, 41].

Methods that generalize rank minimization of tensors are of interest for a variety of applications that utilize high dimensional data including multiview clustering [38], remote sensing [13, 33], computerized tomography [24, 31], and seismic data completion [18, 19, 27, 28]. The multidimensional nature of seismic data makes tensors a natural choice for organizing this data (for example, x and y source and receiver coordinates and time). The tensor nuclear norm (TNN) is the sum of a tensor's singular values and is used in low rank tensor reconstruction methods [10, 40]. The alternating direction method of multipliers (ADMM) [2] can be used to minimize the TNN model. We refer to this combination of model and method as TNN-ADMM. Using ADMM allows for separation of the TNN and the data matching terms and is more efficient than other methods [27] such as multichannel singular spectrum analysis (MSSA) [26, 32] and projection onto convex sets (POCS) [1].

The convergence of ADMM has been studied previously for general functional minimization [2, 9, 35] where the function operates on vectors and matrices. However, these results cannot be directly applied to the tensor case. While Zhang et al. [39] mention convergence of the TNN-ADMM algorithm, they do not prove convergence. Instead they refer the reader to the work of Boyd et al. [2] in which they prove that if the function $f(x)$ is Lipschitz continuous and the function $g(x)$ is convex, for a vector $x \in \mathbb{R}^n$ (of dimension n), then ADMM converges to a stationary point of the function $f(x) + g(x)$. When minimizing a nonconvex logarithmic approximation of rank, Xue et al. [37] show that the sequence generated by ADMM has a convergent subsequence, but not that it converges to a stationary point. Here we prove tensor properties that then allow us to show that the sequence for the TNN minimization via ADMM converges to a stationary point. Our data completion problem involves tensor-tensor element-wise multiplication (a.k.a Hadamard product), not the standard vector–matrix multiplication as considered in [2]. We fully exploit the properties of tensors and include a lower bound for the internal parameter (the step size) introduced by ADMM, thereby reducing the computational burden of tuning this parameter.

The primary contributions of this paper include our proof of convergence for TNN-ADMM as well as numerical experiments that demonstrate denoising and reconstruction of both synthetic and real seismic data from offshore Western Australia. In the remainder of this paper we present background material on tensor linear algebra and develop several new properties in Sect. 2 in preparation for the convergence proof. In Sect. 3 we outline the data completion problem and ADMM algorithm, and we provide the proof of convergence for ADMM. Finally, in Sect. 4 we demonstrate TNN-ADMM for simultaneous reconstruction and denoising on two synthetic seismic examples (one with linear events, one with parabolic events). We compare results from this numerical method with a competing algorithm, MSSA, which has also been used for seismic data completion [26]. Lastly, we present a promising example that indicates TNN-ADMM can be successful at reconstructing and denoising real data.

2 Background Material

We consider $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_p}$ as an order-p tensor, $p \in \mathbb{Z}^+$. *Multi-index* notation is commonly used for referring to scalar elements of a tensor, using an index for each dimension, i.e. $\mathcal{X} = [\mathcal{X}_{i_1,i_2,\ldots,i_p}]$. We will instead use *linear indexing* to refer to scalar elements, e.g. $\mathcal{X} = [\mathcal{X}_i]$ for $i = 1, 2, \ldots, N$, with $N = \prod_{i=1}^{p} n_i$. Using linear indexing, the first element of the tensor (i.e. first column, row, and slice) is assigned an index value of one. Along the first column of the first slice the index of each scalar element is one greater than the previous row, resulting in the elements of this column being indexed from 1 to n_1. This indexing pattern continues for each column of the first slice, i.e. the second column is indexed from $n_1 + 1$ to $2n_1$ and the last column is indexed from $n_1 \times (n_2 - 1) + 1$ to $n_1 n_2$. The indexing then continues with the second slice, following the same pattern and so on. For completeness, the mapping from multi-indexing to linear indexing is given by,

$$\mathcal{X}_{i_1,i_2,\ldots,i_p} = \mathcal{X}_{1+\sum_{k=1}^{p}(i_k-1)\prod_{l=1}^{k-1} n_l} \tag{1}$$

We illustrate both conventional multi-indexing and linear indexing for a 3 × 3 × 2 example, with the two slices separated by a comma,

$$\mathcal{X} = \left(\begin{bmatrix} \mathcal{X}_{1,1,1} & \mathcal{X}_{1,2,1} & \mathcal{X}_{1,3,1} \\ \mathcal{X}_{2,1,1} & \mathcal{X}_{2,2,1} & \mathcal{X}_{2,3,1} \\ \mathcal{X}_{3,1,1} & \mathcal{X}_{3,2,1} & \mathcal{X}_{3,3,1} \end{bmatrix}, \begin{bmatrix} \mathcal{X}_{1,1,2} & \mathcal{X}_{1,2,2} & \mathcal{X}_{1,3,2} \\ \mathcal{X}_{2,1,2} & \mathcal{X}_{2,2,2} & \mathcal{X}_{2,3,2} \\ \mathcal{X}_{3,1,2} & \mathcal{X}_{3,2,2} & \mathcal{X}_{3,3,2} \end{bmatrix} \right) \tag{2}$$

$$= \left(\begin{bmatrix} \mathcal{X}_{1} & \mathcal{X}_{4} & \mathcal{X}_{7} \\ \mathcal{X}_{2} & \mathcal{X}_{5} & \mathcal{X}_{8} \\ \mathcal{X}_{3} & \mathcal{X}_{6} & \mathcal{X}_{9} \end{bmatrix}, \begin{bmatrix} \mathcal{X}_{10} & \mathcal{X}_{13} & \mathcal{X}_{16} \\ \mathcal{X}_{11} & \mathcal{X}_{14} & \mathcal{X}_{17} \\ \mathcal{X}_{12} & \mathcal{X}_{15} & \mathcal{X}_{18} \end{bmatrix} \right). \tag{3}$$

Linear indexing is very convenient when we wish to refer to elements of a tensor without concern for the number of dimensions. The set of tensors with the same dimension form a vector space, under which we can define the *Frobenius inner product* and *norm* [12] (see Appendix).

2.1 Tensor Algebra

For a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_p}$ we denote its multidimensional Fourier transform as $\hat{\mathcal{X}}$, which is implemented via the Fourier transform for each dimension from 3 to p. Specifically, we consider vectors oriented along the third dimension as *tubes*. For each tube we compute its Fourier transform treating the tube as a vector. Similarly we can compute the multidimensional inverse Fourier transform on $\hat{\mathcal{X}}$ that yields $\mathcal{X}$.

The *tensor product* (tProduct) [23] of two tensors $\mathcal{A} \in \mathbb{R}^{n_1 \times m \times n_3 \times \ldots \times n_p}$ and $\mathcal{B} \in \mathbb{R}^{m \times n_2 \times n_3 \times \ldots \times n_p}$ results in a tensor $\mathcal{C} = \mathcal{A}\mathcal{B} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times \ldots \times n_p}$. To compute $\mathcal{A}\mathcal{B}$, we multiply the frontal slices of $\hat{\mathcal{A}}$ and $\hat{\mathcal{B}}$, storing the result of each matrix product as a frontal slice in $\hat{\mathcal{C}}$, followed by the multidimensional inverse Fourier transform.

The *identity* tensor $\mathcal{I}_d \in \mathbb{R}^{n_1 \times n_1 \times n_3 \times \ldots \times n_p}$ has the identity matrix as its first frontal slice and zeros elsewhere. Transposing an order-3 tensor is computed by transposing each frontal slice and reversing the order of all but the first frontal slice. Transposing an order-p tensor ($p > 3$) is computed recursively by transposing each order-($p-1$) subtensor and reversing the order of all but the first order-($p-1$) subtensor. Just as with matrices, orthogonal tensors have the property that their transpose is their inverse, i.e., $\mathcal{U}\mathcal{U}^T = \mathcal{U}^T\mathcal{U} = \mathcal{I}_d$.

The *tensor singular value decomposition* (tSVD) decomposes a tensor into a product of three tensors [23],

$$\mathcal{X} = \mathcal{U}\mathcal{S}\mathcal{V}^T, \tag{4}$$

where the tensors $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3 \times \ldots \times n_p}$ and $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3 \times \ldots \times n_p}$ are orthogonal and $\mathcal{S}$ is a frontal diagonal tensor (meaning each frontal slice is a diagonal matrix).

The *tensor nuclear norm* (TNN) is defined as [10]:

$$||\mathcal{X}||_{\text{TNN}} = \sum_{i=1}^{N} ||\hat{\mathcal{X}}(:, :, i)||_*, \tag{5}$$

where $\hat{\mathcal{X}}(:, :, i)$ is the i^{th} slice of $\hat{\mathcal{X}}$, and $|| \cdot ||_*$ is the matrix nuclear norm. As TNN is not differentiable, we rely on the concept of *subgradient* [30] to define an optimality condition.

Definition 1 (Subgradient) For a convex function $f : \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_p} \to \mathbb{R}$ and a point $\mathcal{M} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_p}$, the subgradient of f at $\mathcal{M}$, denoted $\partial f(\mathcal{M})$, is the set of tensors,

$$\partial f(\mathcal{M}) = \left\{\mathcal{P} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_p} \mid \forall \mathcal{X}\ f(\mathcal{X}) - f(\mathcal{M}) \geq \langle \mathcal{P}, \mathcal{X} - \mathcal{M} \rangle_{\text{F}}\right\}. \tag{6}$$

Elements of the TNN's subgradient, $\mathcal{P} \in \partial\|\mathcal{M}\|_{\text{TNN}}$, can be expressed as [40]

$$\mathcal{P} = \mathcal{U}\mathcal{V}^T + \mathcal{W}, \tag{7}$$

such that $\mathcal{M} = \mathcal{U}\mathcal{S}\mathcal{V}^T$ is the tSVD of $\mathcal{M}$ and $\|\mathcal{W}\|_2 \leq 1$, where $\| \cdot \|_2$ is the tensor spectral norm (the largest singular value of the tensor).

Next we introduce the *Hadamard product* and its related algebraic concepts [15, 29]. The Hadamard product is used to model the missing data problem by a *sampling operator*, which is a tensor with ones where data is sampled (receivers are located) and zeros where no data is recorded.

Definition 2 (Hadamard Product [15]) The *Hadamard product*, denoted $\mathcal{A} \circ \mathcal{B}$, between two tensors with matching dimensions is element-wise multiplication, i.e. $(\mathcal{A} \circ \mathcal{B})_i = \mathcal{A}_i\mathcal{B}_i$ for $i = 1, 2, \ldots, N$ where N is the number of elements of $\mathcal{A}$ and $\mathcal{B}$.

The Hadamard product has associated definitions for *identity*, *inverse*, and *idempotent* (see Appendix). Sampling operators in particular are idempotent since zero and one are idempotent under scalar multiplication.

2.2 Tensor Properties

This subsection lists useful properties on tensors and the Hadamard product which we will use in our convergence proof in Sect. 3. Additional Frobenius inner product properties on *symmetry*, *linearity*, the *polarization identity*, and *orthogonal invariance* are provided in the Appendix.

Property 1 For three real-valued tensors $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ of the same size, the composition of the Frobenius inner product and the Hadamard product has the following commutative property,

$$\langle \mathcal{A} \circ \mathcal{B}, \mathcal{C} \rangle_{\mathrm{F}} = \langle \mathcal{A} \circ \mathcal{C}, \mathcal{B} \rangle_{\mathrm{F}}. \tag{8}$$

Proof Using the definition of Frobenius inner product (Definition 6), the definition of Hadamard product (Definition 2), and the associativity and commutativity of scalar multiplication we obtain,

$$\begin{aligned} \langle \mathcal{A} \circ \mathcal{B}, \mathcal{C} \rangle_{\mathrm{F}} &= \sum_{i=1}^{N} (\mathcal{A} \circ \mathcal{B})_i \mathcal{C}_i = \sum_{i=1}^{N} (\mathcal{A}_i \mathcal{B}_i) \mathcal{C}_i & (9) \\ &= \sum_{i=1}^{N} (\mathcal{A}_i \mathcal{C}_i) \mathcal{B}_i = \sum_{i=1}^{N} (\mathcal{A} \circ \mathcal{C})_i \mathcal{B}_i = \langle \mathcal{A} \circ \mathcal{C}, \mathcal{B} \rangle_{\mathrm{F}}. & (10) \end{aligned}$$

□

Next we consider two *reshaping* operators that permute the elements of a tensor into another data structure, such as a vector or matrix, which are called *vectorization* and *diagonal matricization* [16], respectively. Then Properties 2 and 3 are introduced to relate vectorization, matricization, and a Hadamard product of tensors to a matrix–vector product. Both properties are useful when proving the Frobenius norm of a Hadamard product of tensors can be expressed in terms of the L_2 norm (Property 7).

Definition 3 (Vectorization [16]) Given a tensor $\mathcal{A}$, we use $\mathrm{vec}(\mathcal{A})$ to denote the *vectorization* of $\mathcal{A}$. This operation reshapes the scalar elements of the tensor into a vector as such,

$$\mathrm{vec}(\mathcal{A}) = \begin{bmatrix} \mathcal{A}_1 \\ \mathcal{A}_2 \\ \vdots \\ \mathcal{A}_N \end{bmatrix} \in \mathbb{R}^N, \tag{11}$$

where N is the number of element of $\mathcal{A}$.

Definition 4 (Diagonal Matricization [16]) We use $\mathrm{dimat}(\mathcal{A})$ to denote the *diagonal matricization* of tensor $\mathcal{A}$. This operation forms a diagonal matrix with the elements of $\mathcal{A}$,

$$\mathrm{dimat}(\mathcal{A}) = \begin{bmatrix} \mathcal{A}_1 & & & \\ & \mathcal{A}_2 & & \\ & & \ddots & \\ & & & \mathcal{A}_N \end{bmatrix} \in \mathbb{R}^{N \times N}. \tag{12}$$

Property 2 (Shape Invariance) The Frobenius norm is invariant under reshaping and is equivalent to the L_2 norm for vectors, i.e.

$$\|\mathcal{A}\|_{\mathrm{F}} = \|\mathrm{dimat}(\mathcal{A})\|_{\mathrm{F}} = \|\mathrm{vec}(\mathcal{A})\|_{\mathrm{F}} = \|\mathrm{vec}(\mathcal{A})\|_2.$$

Proof Reshaping operators do not affect the values of the scalar entries and do not result in additional non-zero entries, hence the Frobenius norm of any reshaped tensor has the same value as the Frobenius norm of the original tensor by the commutivity of addition. For a vector the Frobenius norm and L_2 norm have the same value by definition. □

Property 3 For tensors $\mathcal{A}$ and $\mathcal{B}$, the vectorization of their Hadamard product is equivalent to the matrix–vector product of the diagonal matricization of $\mathcal{A}$ and vectorization of $\mathcal{B}$, i.e.,

$$\text{vec}(\mathcal{A} \circ \mathcal{B}) = \text{dimat}(\mathcal{A})\text{vec}(\mathcal{B}). \tag{13}$$

Proof This property can be proved using matrix–vector multiplication,

$$\text{vec}(\mathcal{A} \circ \mathcal{B}) = \begin{bmatrix} \mathcal{A}_1\mathcal{B}_1 \\ \mathcal{A}_2\mathcal{B}_2 \\ \vdots \\ \mathcal{A}_N\mathcal{B}_N \end{bmatrix} = \begin{bmatrix} \mathcal{A}_1 & & & \\ & \mathcal{A}_2 & & \\ & & \ddots & \\ & & & \mathcal{A}_N \end{bmatrix} \begin{bmatrix} \mathcal{B}_1 \\ \mathcal{B}_2 \\ \vdots \\ \mathcal{B}_N \end{bmatrix} = \text{dimat}(\mathcal{A})\text{vec}(\mathcal{B}).$$

□

Below we define the *circulant* [16] of tensors, which will be used to show the block-diagonalization property (Property 4) of circulant matrices.

Definition 5 (Circulant [16]) The circulant of tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_p}$ is a matrix denoted $\text{circ}(\mathcal{A}) \in \mathbb{R}^{n_1 M \times n_2 M}$, where $M = \prod_{i=3}^{p} n_i$ is the number of frontal slices. Let $\mathcal{A}_{:,:,k}$ denote the k^{th} slice of $\mathcal{A}$ for $k = 1, 2, ..., M$. The circulant matrix is defined as,

$$\text{circ}(\mathcal{A}) = \begin{bmatrix} \mathcal{A}_{:,:,1} & \mathcal{A}_{:,:,M} & \mathcal{A}_{:,:,M-1} & \cdots & \mathcal{A}_{:,:,2} \\ \mathcal{A}_{:,:,2} & \mathcal{A}_{:,:,1} & \mathcal{A}_{:,:,M} & \cdots & \mathcal{A}_{:,:,3} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathcal{A}_{:,:,M} & \mathcal{A}_{:,:,M-1} & \cdots & \mathcal{A}_{:,:,2} & \mathcal{A}_{:,:,1} \end{bmatrix}. \tag{14}$$

Unlike the vectorization and diagonal matricization, the circulant operator is not a reshaping operator since elements of the tensor are repeated in the circulant. However, each block column and block row are reshaped forms of $\mathcal{A}$. Next we introduce a property of the circulant that diagonalizes the circulant matrix into a block diagonal matrix.

Property 4 (Block-Diagonalization [16]) The circulant matrix $\text{circ}(\mathcal{A})$ can be diagonalized into a block diagonal matrix,

$$(F_M \otimes I_{n_1})\text{circ}(\mathcal{A})(F_M^* \otimes I_{n_2}) = \begin{bmatrix} \hat{\mathcal{A}}_{:,:,1} & & & \\ & \hat{\mathcal{A}}_{:,:,2} & & \\ & & \ddots & \\ & & & \hat{\mathcal{A}}_{:,:,M} \end{bmatrix}, \tag{15}$$

where F_M is the discrete Fourier transform (DFT) matrix of size M, F_M^* is the conjugate transpose of F_M, I is the identity matrix (with size specified by the subscript), M is the total number of frontal slices of $\mathcal{A}$, and $\otimes$ is the *Kronecker product* [14] (see Definition 11 in the Appendix).

In the case that $\mathcal{A}$ is a diagonal tensor, the block diagonal matrix of the right-hand side of (15) is equivalent to the diagonal matriziation of $\hat{\mathcal{A}}$.

The circulant possesses a property similar to the Shape Invariance Property 2 under the Frobenius norm.

Property 5 The square of the Frobenius norm of tensor $\mathcal{A}$ is directly proportional to the square of the Frobenius norm of $\text{circ}(\mathcal{A})$ with the proportionality factor of $\frac{1}{M}$,

$$\|\mathcal{A}\|_{\text{F}}^2 = \frac{1}{M}\|\text{circ}(\mathcal{A})\|_{\text{F}}^2, \tag{16}$$

where M is the number of slices of $\mathcal{A}$.

Proof The matrix $\mathrm{circ}(\mathcal{A})$ repeats all the elements of $\mathcal{A}$ M-many times. Thus, $\|\mathrm{circ}(\mathcal{A})\|_{\mathrm{F}}^2 = M\sum_{i=1}^{N}|\mathcal{A}_i|^2 = M\|\mathcal{A}\|_{\mathrm{F}}^2$. □

Combining Properties 4, 5, and Orthogonal Invariance (see Property 12 in the Appendix) we relate the Frobenius norms of $\mathcal{A}$ and $\hat{\mathcal{A}}$.

Property 6 The square of the Frobenius norm of tensor $\mathcal{A}$ is directly proportional to the square of the Frobenius norm of $\hat{\mathcal{A}}$ (the multidimensional Fourier transform of $\mathcal{A}$) by a factor of $\frac{1}{M}$,

$$\|\mathcal{A}\|_{\mathrm{F}}^2 = \frac{1}{M}\|\hat{\mathcal{A}}\|_{\mathrm{F}}^2, \tag{17}$$

where M is the number of slices of $\mathcal{A}$.

Proof It follows from Property 4 (Block-Diagonalization) that we can express $\mathrm{circ}(\mathcal{A})$ into the product of three matrices. Further applying Property 5 we obtain,

$$\|\mathcal{A}\|_{\mathrm{F}}^2 = \frac{1}{M}\left\|(F_M\otimes I_{n_1})^{-1}\begin{bmatrix}\hat{\mathcal{A}}_{:,:,1} & & & \\ & \hat{\mathcal{A}}_{:,:,2} & & \\ & & \ddots & \\ & & & \hat{\mathcal{A}}_{:,:,M}\end{bmatrix}(F_M^*\otimes I_{n_2})^{-1}\right\|_{\mathrm{F}}^2. \tag{18}$$

Since the Kronecker product of orthogonal matrices is also orthogonal [14], we apply Property 12 to simplify the right-hand side to the Frobenius norm of the block-diagonal matrix,

$$\|\mathcal{A}\|_{\mathrm{F}}^2 = \frac{1}{M}\left\|\begin{bmatrix}\hat{\mathcal{A}}_{:,:,1} & & & \\ & \hat{\mathcal{A}}_{:,:,2} & & \\ & & \ddots & \\ & & & \hat{\mathcal{A}}_{:,:,M}\end{bmatrix}\right\|_{\mathrm{F}}^2. \tag{19}$$

Property 2 (Shape Invariance) guarantees that the Frobenius norm of the block diagonal matrix is equal to the Frobenius norm of $\hat{\mathcal{A}}$. □

Lastly we prove two properties to show that the Frobenius norm of a Hadamard product of tensors is bounded by the maximum value and Frobenius norm of the tensors.

Property 7 The Frobenius norm of a Hadamard product of tensors $\mathcal{A}$ and $\mathcal{B}$ can be expressed as the L_2 norm of a product of a diagonal matricization and vectorization,

$$\|\mathcal{A}\circ\mathcal{B}\|_{\mathrm{F}} = \|\mathrm{dimat}(\mathcal{A})\mathrm{vec}(\mathcal{B})\|_2. \tag{20}$$

Proof The desired equality can be obtained by using Properties 2 and 3, i.e.,

$$\begin{aligned}\|\mathcal{A}\circ\mathcal{B}\|_{\mathrm{F}} &= \|\mathrm{vec}(\mathcal{A}\circ\mathcal{B})\|_{\mathrm{F}} \\ &= \|\mathrm{dimat}(\mathcal{A})\mathrm{vec}(\mathcal{B})\|_{\mathrm{F}} = \|\mathrm{dimat}(\mathcal{A})\mathrm{vec}(\mathcal{B})\|_2.\end{aligned}$$

□

Property 8 (Hadamard-Frobenius Inequality) The Frobenius norm of a Hadamard product of a non-negative tensor $\mathcal{A}$ and tensor $\mathcal{B}$ is bounded by the max value of $\mathcal{A}$ and the Frobenius norm of $\mathcal{B}$,

$$\|\mathcal{A}\circ\mathcal{B}\|_{\mathrm{F}} \le \max(\mathcal{A})\|\mathcal{B}\|_{\mathrm{F}}. \tag{21}$$

Proof Using Properties 2 and 7, we obtain the following inequality,

$$\|\mathcal{A} \circ \mathcal{B}\|_{\mathrm{F}} = \|\mathrm{dimat}(\mathcal{A})\mathrm{vec}(\mathcal{B})\|_2 \leq \|\mathrm{dimat}(\mathcal{A})\|_2 \|\mathrm{vec}(\mathcal{B})\|_2.$$

The L_2 norm of a matrix, also known as the spectral norm, is defined as the largest singular value. Since $\mathrm{dimat}(\mathcal{A})$ is a non-negative diagonal matrix, its largest singular value is its largest diagonal value. Hence,

$$\|\mathrm{dimat}(\mathcal{A})\|_2 = \max(\mathrm{diag}(\mathrm{dimat}(\mathcal{A}))) = \max(\mathcal{A}),$$

and thus,

$$\|\mathcal{A} \circ \mathcal{B}\|_{\mathrm{F}} \leq \max(\mathcal{A})\|\mathcal{B}\|_{\mathrm{F}}.$$

□

3 Convergence Analysis for TNN-ADMM Applied to Tensors

In Sect. 3.1 we describe the unconstrained TNN minimization problem for data completion and an iterative algorithm using ADMM with closed-form solutions of the subproblems. By analyzing the convergence in Sect. 3.2, we guarantee the sequence generated by ADMM has a subsequence convergent to a solution that minimizes the unconstrained TNN problem, provided the parameters of the problem satisfy strict inequalities derived in our analysis. These strict inequalities help guide the parameter selection in practice.

3.1 TNN-ADMM Problem Setting

Data acquired through experiments may be incomplete (i.e. it may contain gaps) and is likely be contaminated by noise. Gaps in seismic data can be caused by physical limitations on the placement of sources and receivers and receiver failure. Sources of noise include natural sources such as weather and man-made sources such as construction. We model this process by the equation,

$$\mathcal{Y} = \mathcal{A} \circ \mathcal{X} + \mathcal{N}, \tag{22}$$

where $\mathcal{Y}$ is the observation, $\mathcal{A}$ is the sampling operator, $\mathcal{X}$ is the underlying, complete data, and $\mathcal{N}$ is additive noise. Here the sampling operator is a tensor of zeros and ones, which acts on the data via the Hadamard product. It should be noted that we are only concerned with noise at sampled locations and thus consider the noise tensor $\mathcal{N}$ and the observation $\mathcal{Y}$ invariant to the sampling operator, i.e. $\mathcal{A} \circ \mathcal{N} = \mathcal{N}$ and $\mathcal{A} \circ \mathcal{Y} = \mathcal{Y}$. By simultaneously reconstructing and denoising the sampled noisy data $\mathcal{Y}$, our goal is to recover $\mathcal{X}$. We consider the unconstrained TNN minimization problem [10],

$$\min_{\mathcal{X}} \lambda \|\mathcal{X}\|_{\mathrm{TNN}} + \frac{1}{2}\|\mathcal{Y} - \mathcal{A} \circ \mathcal{X}\|_{\mathrm{F}}^2, \tag{23}$$

where $\lambda > 0$ is a weighting parameter. We introduce the auxiliary variable $\mathcal{Z}$ to decouple the TNN and data matching terms, rewriting (23) as

$$\min_{\mathcal{X}} \lambda \|\mathcal{X}\|_{\mathrm{TNN}} + \frac{1}{2}\|\mathcal{Y} - \mathcal{A} \circ \mathcal{Z}\|_{\mathrm{F}}^2 \quad \text{s.t. } \mathcal{X} = \mathcal{Z}. \tag{24}$$

With the TNN and data matching terms separated, we can apply ADMM to solve (24). The method proceeds as follows: first we form the augmented Lagrangian,

$$L_\rho(\mathcal{X}, \mathcal{Z}; \mathcal{B}) = \lambda \|\mathcal{X}\|_{\text{TNN}} + \frac{1}{2}\|\mathcal{Y} - \mathcal{A} \circ \mathcal{Z}\|_{\text{F}}^2 + \rho\langle \mathcal{B}, \mathcal{X} - \mathcal{Z}\rangle_{\text{F}} + \frac{\rho}{2}\|\mathcal{X} - \mathcal{Z}\|_{\text{F}}^2, \quad (25)$$

where $\mathcal{B}$ is the Lagrange multiplier or dual variable and the parameter $\rho > 0$ is the step size. Note that ρ appears in the Lagrange multiplier to make completing the square simpler. This form is referred to as a scaled augmented Lagrangian by Boyd et al. [2]. The dual variable serves to add the $\mathcal{X} = \mathcal{Z}$ equality constraint to the objective function. Additionally, the last term of the augmented Lagrangian is a quadratic penalty term for the equality constraint. The augmented Lagrangian includes both the Lagrangian and quadratic penalty terms to reduce the chance of ill-conditioning [5, 25]. When using only one penalty term, if the step size parameter tends to zero (or infinity), then the update steps could result in ill-conditioned tensors.

Using the augmented Lagrangian we can define the ADMM iterations as,

$$\mathcal{X}^{k+1} = \arg\min_{\mathcal{X}} L_\rho(\mathcal{X}, \mathcal{Z}^k; \mathcal{B}^k) \quad (26)$$

$$\mathcal{Z}^{k+1} = \arg\min_{\mathcal{Z}} L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}; \mathcal{B}^k) \quad (27)$$

$$\mathcal{B}^{k+1} = \mathcal{B}^k + \mathcal{X}^{k+1} - \mathcal{Z}^{k+1}, \quad (28)$$

where k denotes iteration number.

To solve the $\mathcal{X}$-problem (26),

$$\mathcal{X}^{k+1} = \arg\min_{\mathcal{X}} \lambda \|\mathcal{X}\|_{\text{TNN}} + \frac{1}{2}\|\mathcal{Y} - \mathcal{A} \circ \mathcal{Z}\|_{\text{F}}^2 + \rho\langle \mathcal{B}^k, \mathcal{X} - \mathcal{Z}^k\rangle_{\text{F}} + \frac{\rho}{2}\|\mathcal{X} - \mathcal{Z}^k\|_{\text{F}}^2, \quad (29)$$

we rewrite the last two terms using the polarization identity (see Property 11 in the Appendix),

$$\rho\langle \mathcal{B}^k, \mathcal{X} - \mathcal{Z}^k\rangle_{\text{F}} + \frac{\rho}{2}\|\mathcal{X} - \mathcal{Z}^k\|_{\text{F}}^2 = \frac{\rho}{2}\|\mathcal{X} - \mathcal{Z}^k + \mathcal{B}^k\|_{\text{F}}^2 - \frac{\rho}{2}\|\mathcal{B}^k\|_{\text{F}}^2. \quad (30)$$

Substituting this equality into the $\mathcal{X}$ update (29), we then simplify by considering minimization only over the terms that include $\mathcal{X}$,

$$\mathcal{X}^{k+1} = \arg\min_{\mathcal{X}} \lambda \|\mathcal{X}\|_{\text{TNN}} + \frac{\rho}{2}\|\mathcal{X} - \mathcal{Z}^k + \mathcal{B}^k\|_{\text{F}}^2. \quad (31)$$

Next we let $\mathcal{X} = \mathcal{U}\mathcal{S}\mathcal{V}^T$ and $\mathcal{Z}^k - \mathcal{B}^k = \mathcal{U}'\mathcal{S}'\mathcal{V}'^T$ be the tSVD of $\mathcal{X}$ and $\mathcal{Z}^k - \mathcal{B}^k$ respectively. Substituting these decompositions into (31) yields,

$$\mathcal{X}^{k+1} = \arg\min_{\mathcal{X} = \mathcal{U}\mathcal{S}\mathcal{V}^T} \lambda \|\mathcal{U}\mathcal{S}\mathcal{V}^T\|_{\text{TNN}} + \frac{\rho}{2}\|\mathcal{U}\mathcal{S}\mathcal{V}^T - \mathcal{U}'\mathcal{S}'\mathcal{V}'^T\|_{\text{F}}^2. \quad (32)$$

Next we factor $\mathcal{U}'$ and $\mathcal{V}'^T$ from the Frobenius norm term to obtain,

$$\mathcal{X}^{k+1} = \arg\min_{\mathcal{X} = \mathcal{U}\mathcal{S}\mathcal{V}^T} \lambda \|\mathcal{U}\mathcal{S}\mathcal{V}^T\|_{\text{TNN}} + \frac{\rho}{2}\|\mathcal{U}'(\mathcal{U}'^T\mathcal{U}\mathcal{S}\mathcal{V}^T\mathcal{V}' - \mathcal{S}')\mathcal{V}'^T\|_{\text{F}}^2. \quad (33)$$

We use the orthogonal invariance (Property 12) of the TNN and Frobenius norm to simplify (33),

$$\mathcal{X}^{k+1} = \arg\min_{\mathcal{X} = \mathcal{U}\mathcal{S}\mathcal{V}^T} \lambda \|\mathcal{S}\|_{\text{TNN}} + \frac{\rho}{2}\|\mathcal{U}'^T\mathcal{U}\mathcal{S}\mathcal{V}^T\mathcal{V}' - \mathcal{S}'\|_{\text{F}}^2. \quad (34)$$

Since $\mathcal{S}'$ is a diagonal tensor, the Frobenius norm is minimized when $\mathcal{U}'^T \mathcal{U}\mathcal{S}\mathcal{V}^T\mathcal{V}'$ is diagonal. By selecting $\mathcal{U} = \mathcal{U}'$ and $\mathcal{V} = \mathcal{V}'$ the product $\mathcal{U}'^T \mathcal{U}\mathcal{S}\mathcal{V}^T\mathcal{V}'$ simplifies to the diagonal tensor $\mathcal{S}$. We can now consider the $\mathcal{X}$ update as $\mathcal{X}^{k+1} = \mathcal{U}'\mathcal{S}^{k+1}\mathcal{V}'^T$, where $\mathcal{S}^{k+1}$ is the solution of the argmin problem,

$$\mathcal{S}^{k+1} = \arg\min_{\mathcal{S}} \lambda\|\mathcal{S}\|_{\text{TNN}} + \frac{\rho}{2}\|\mathcal{S} - \mathcal{S}'\|_{\text{F}}^2. \tag{35}$$

Next we use properties of TNN and the Frobenius norm to transform this problem into a series of independent 1-dimensional problems. Specifically, we denote $\bar{S}$ as the block diagonal matrix with the frontal slices of $\hat{\mathcal{S}}$ as the block-diagonal elements. Since the tensor $\mathcal{S}$ has the same singular values as the matrix $\bar{S}$ [27], $\|\mathcal{S}\|_{\text{TNN}} = \|\bar{S}\|_*$. The nuclear norm is the sum of the singular values, which is equivalent to the L_1 norm of a vector containing the singular values. We then apply Property 6 to express the Frobenius norm term of (35) in terms of $\hat{\mathcal{S}}$ and $\hat{\mathcal{S}}'$. We now consider the argmin problem as an update step for $\hat{\mathcal{S}}$,

$$\hat{\mathcal{S}}^{k+1} = \arg\min_{\hat{\mathcal{S}}} \lambda\|\bar{S}\|_* + \frac{\rho}{2M}\|\hat{\mathcal{S}} - \hat{\mathcal{S}}'\|_{\text{F}}^2, \tag{36}$$

with $\mathcal{S}^{k+1}$ as the inverse Fourier transform of $\hat{\mathcal{S}}^{k+1}$. By expressing the norms in terms of singular values the argmin problem can be written as,

$$\hat{\mathcal{S}}^{k+1} = \arg\min_{\hat{\mathcal{S}}} \lambda \sum_{i=1}^{L} \hat{\sigma}_i + \frac{\rho}{2M}\sum_{i=1}^{L}(\hat{\sigma}_i - \hat{\sigma}_i')^2, \tag{37}$$

where $\hat{\sigma}_i$ and $\hat{\sigma}_i'$ are the i^{th} diagonal values of the tensors $\hat{\mathcal{S}}$ and $\hat{\mathcal{S}}'$ respectively, and $L = \min(n_1, n_2)\prod_{k=3}^{p} n_k$ is the number of singular values. To find the minimum value, we calculate the partial derivative of (37) with respect to $\hat{\sigma}_i$ and set it equal to zero,

$$\lambda + \frac{\rho}{M}(\hat{\sigma}_i - \hat{\sigma}_i') = 0, \tag{38}$$

for $i = 1, 2, ..., L$. Solving for $\hat{\sigma}_i$ yields,

$$\hat{\sigma}_i = \hat{\sigma}_i' - \frac{\lambda M}{\rho}. \tag{39}$$

However, singular values must be positive, so we have the additional constraint that $\hat{\sigma}_i > 0$, and hence,

$$\hat{\sigma}_i = \max\left(\hat{\sigma}_i' - \frac{\lambda M}{\rho}, 0\right). \tag{40}$$

To summarize the closed-form solution of the $\mathcal{X}$-problem, let $\mathcal{X}^{k+1} = \mathcal{U}'\mathcal{S}^{k+1}\mathcal{V}'^T$, where $\mathcal{Z}^k - \mathcal{B}^k = \mathcal{U}'\mathcal{S}'\mathcal{V}'^T$ is the tSVD of $\mathcal{Z}^k - \mathcal{B}^k$. The tensor $\mathcal{S}^{k+1}$ is the inverse Fourier transform of $\hat{\mathcal{S}}^{k+1}$ and the diagonal values of $\hat{\mathcal{S}}^{k+1}$ are given by (40) in terms of the diagonal values of $\hat{\mathcal{S}}'$. This result is also known as singular value thresholding (SVT) [3] for matrices.

The $\mathcal{Z}$-problem (27) can be solved by setting its gradient evaluated at $\mathcal{Z}^{k+1}$ to zero and solving for $\mathcal{Z}^{k+1}$. This results in the $\mathcal{Z}$ update as follows,

$$\mathcal{Z}^{k+1} = (\mathcal{A} + \rho\mathcal{I})^{-1} \circ (\rho(\mathcal{X}^{k+1} + \mathcal{B}^k) + \mathcal{Y}), \tag{41}$$

where $(\mathcal{A} + \rho\mathcal{I})^{-1}$ is the Hadamard inverse and $\mathcal{I}$ is the Hadamard identity tensor. The dual variable $\mathcal{B}$ is updated in (28) using gradient ascent. Specifically, taking the gradient of the

Lagrangian function (25) with respect to the dual variable $\mathcal{B}$ yields $\rho(\mathcal{X}^{k+1} - \mathcal{Z}^{k+1})$. By taking the step size δ to be $1/\rho$ in the gradient ascent for $\mathcal{B}$, we obtain

$$\frac{\mathcal{B}^{k+1} - \mathcal{B}^k}{\delta} = \rho(\mathcal{X}^{k+1} - \mathcal{Z}^{k+1}), \tag{42}$$

which is equivalent to (28).

3.2 Convergence Analysis

We refer to the combination of applying ADMM to the TNN objective function as TNN-ADMM. To prove convergence of TNN-ADMM we first start by proving the following lemmas. Lemma 1 shows that the Lagrangian $L_\rho(\mathcal{X}^k, \mathcal{Z}^k; \mathcal{B}^k)$ is strictly decreasing with each iteration of ADMM and provides an upper bound on the difference between Lagrangians evaluated at consecutive iterations. Lemma 2 shows that both the sequence $\{\mathcal{X}^k, \mathcal{Z}^k, \mathcal{B}^k\}$ generated by ADMM and the Lagrangian $L_\rho(\mathcal{X}^k, \mathcal{Z}^k; \mathcal{B}^k)$ are bounded. Theorem 1 uses these two preceding lemmas to show that the limit point of the sequence generated by ADMM is a stationary point of the unconstrained minimization problem (23).

Lemma 1 *(Sufficient Decay) If $\rho > 1$, there exist two strictly positive parameters c_1, c_2 such that*

$$L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^{k+1}; \mathcal{B}^{k+1}) - L_\rho(\mathcal{X}^k, \mathcal{Z}^k; \mathcal{B}^k) \leq -c_1\|\mathcal{X}^{k+1} - \mathcal{X}^k\|_F^2 - c_2\|\mathcal{Z}^{k+1} - \mathcal{Z}^k\|_F^2. \tag{43}$$

Proof First we examine the difference in the Lagrangian when $\mathcal{Z}$ and $\mathcal{B}$ are constant,

$$\begin{aligned} L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^k; \mathcal{B}^k) - L_\rho(\mathcal{X}^k, \mathcal{Z}^k; \mathcal{B}^k) = \lambda\|\mathcal{X}^{k+1}\|_{\text{TNN}} + \rho\langle\mathcal{B}^k, \mathcal{X}^{k+1} - \mathcal{Z}^k\rangle_{\text{F}} \\ + \frac{\rho}{2}\|\mathcal{X}^{k+1} - \mathcal{Z}^k\|_{\text{F}}^2 - (\lambda(\|\mathcal{X}^k\|_{\text{TNN}} \\ + \rho\langle\mathcal{B}^k, \mathcal{X}^k - \mathcal{Z}^k\rangle_{\text{F}} + \frac{\rho}{2}\|\mathcal{X}^k - \mathcal{Z}^k\|_{\text{F}}^2). \end{aligned} \tag{44}$$

By applying the polarization identity (Property 11) the two Frobenius norm terms simplify as follows,

$$\begin{aligned} &\frac{\rho}{2}\|\mathcal{X}^{k+1} - \mathcal{Z}^k\|_{\text{F}}^2 - \frac{\rho}{2}\|\mathcal{X}^k - \mathcal{Z}^k\|_{\text{F}}^2 \\ &= \frac{\rho}{2}(\|\mathcal{X}^{k+1}\|_{\text{F}}^2 - \|\mathcal{X}^k\|_{\text{F}}^2 - 2\langle\mathcal{X}^{k+1}, \mathcal{Z}^k\rangle_{\text{F}} + 2\langle\mathcal{X}^k, \mathcal{Z}^k\rangle_{\text{F}}). \end{aligned} \tag{45}$$

By using the symmetry and linearity of the Frobenius inner product (see Properties 9 and 10 in the Appendix) the inner product terms in (44) and (45) can be simplified as follows,

$$\begin{aligned} \rho\langle\mathcal{B}^k - \mathcal{Z}^k, \mathcal{X}^{k+1} - \mathcal{X}^k\rangle_{\text{F}} = \rho\langle\mathcal{B}^k, \mathcal{X}^{k+1} - \mathcal{Z}^k\rangle_{\text{F}} - \rho\langle\mathcal{B}^k, \mathcal{X}^k - \mathcal{Z}^k\rangle_{\text{F}} \\ - \rho\langle\mathcal{X}^{k+1}, \mathcal{Z}^k\rangle_{\text{F}} + \rho\langle\mathcal{X}^k, \mathcal{Z}^k\rangle_{\text{F}}. \end{aligned} \tag{46}$$

Thus the difference (44) simplifies to,

$$\begin{aligned} &L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^k; \mathcal{B}^k) - L_\rho(\mathcal{X}^k, \mathcal{Z}^k; \mathcal{B}^k) \\ &= \lambda(\|\mathcal{X}^{k+1}\|_{\text{TNN}} - \|\mathcal{X}^k\|_{\text{TNN}}) + \rho\langle\mathcal{B}^k - \mathcal{Z}^k, \mathcal{X}^{k+1} - \mathcal{X}^k\rangle_{\text{F}} + \frac{\rho}{2}(\|\mathcal{X}^{k+1}\|_{\text{F}}^2 - \|\mathcal{X}^k\|_{\text{F}}^2). \end{aligned}$$

For $\mathcal{P}^{k+1} \in \partial \|\mathcal{X}^{k+1}\|_{\text{TNN}}$, we have $\|\mathcal{X}^{k+1}\|_{\text{TNN}} - \|\mathcal{X}^{k}\|_{\text{TNN}} \leq \langle \mathcal{P}^{k+1}, \mathcal{X}^{k+1} - \mathcal{X}^{k}\rangle_{\text{F}}$ by Definition 1, hence,

$$\begin{aligned} &L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^k; \mathcal{B}^k) - L_\rho(\mathcal{X}^{k}, \mathcal{Z}^k; \mathcal{B}^k) \\ &\leq \langle \lambda \mathcal{P}^{k+1} + \rho(\mathcal{B}^k - \mathcal{Z}^k), \mathcal{X}^{k+1} - \mathcal{X}^k\rangle_{\text{F}} + \frac{\rho}{2}(\|\mathcal{X}^{k+1}\|_{\text{F}}^2 - \|\mathcal{X}^{k}\|_{\text{F}}^2). \end{aligned} \tag{47}$$

Now we use the optimality condition of the $\mathcal{X}$-problem (26),

$$\lambda \mathcal{P}^{k+1} + \rho(\mathcal{B}^k + \mathcal{X}^{k+1} - \mathcal{Z}^k) = 0,$$

to simplify (47) as follows,

$$\begin{aligned} &L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^k; \mathcal{B}^k) - L_\rho(\mathcal{X}^{k}, \mathcal{Z}^k; \mathcal{B}^k) \\ &\leq \langle -\rho \mathcal{X}^{k+1}, \mathcal{X}^{k+1} - \mathcal{X}^k\rangle_{\text{F}} + \frac{\rho}{2}(\|\mathcal{X}^{k+1}\|_{\text{F}}^2 - \|\mathcal{X}^{k}\|_{\text{F}}^2) \\ &\leq -\frac{\rho}{2}\|\mathcal{X}^{k+1} - \mathcal{X}^k\|_{\text{F}}^2. \end{aligned} \tag{48}$$

Next we examine the difference in the Lagrangian when $\mathcal{X}$ and $\mathcal{B}$ are constant,

$$\begin{aligned} &L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^{k+1}; \mathcal{B}^k) - L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^k; \mathcal{B}^k) \\ &= \frac{1}{2}\|\mathcal{Y} - \mathcal{A} \circ \mathcal{Z}^{k+1}\|_{\text{F}}^2 + \rho\langle \mathcal{B}^k, \mathcal{X}^{k+1} - \mathcal{Z}^{k+1}\rangle_{\text{F}} + \frac{\rho}{2}\|\mathcal{X}^{k+1} - \mathcal{Z}^{k+1}\|_{\text{F}}^2 \\ &\quad - \left(\frac{1}{2}\|\mathcal{Y} - \mathcal{A} \circ \mathcal{Z}^{k}\|_{\text{F}}^2 + \rho\langle \mathcal{B}^k, \mathcal{X}^{k+1} - \mathcal{Z}^{k}\rangle_{\text{F}} + \frac{\rho}{2}\|\mathcal{X}^{k+1} - \mathcal{Z}^{k}\|_{\text{F}}^2\right). \end{aligned} \tag{49}$$

We apply Properties 10 and 11 to simplify the Frobenius norm terms,

$$\begin{aligned} \frac{1}{2}\|\mathcal{Y} - \mathcal{A} \circ \mathcal{Z}^{k+1}\|_{\text{F}}^2 - \frac{1}{2}\|\mathcal{Y} - \mathcal{A} \circ \mathcal{Z}^{k}\|_{\text{F}}^2 = &\frac{1}{2}(\|\mathcal{A} \circ \mathcal{Z}^{k+1}\|_{\text{F}}^2 - \|\mathcal{A} \circ \mathcal{Z}^{k}\|_{\text{F}}^2 \\ &- 2\langle \mathcal{Y}, \mathcal{A} \circ \mathcal{Z}^{k+1} - \mathcal{A} \circ \mathcal{Z}^{k}\rangle_{\text{F}}), \end{aligned} \tag{50}$$

$$\begin{aligned} \frac{\rho}{2}\|\mathcal{X}^{k+1} - \mathcal{Z}^{k+1}\|_{\text{F}}^2 - \frac{\rho}{2}\|\mathcal{X}^{k+1} - \mathcal{Z}^{k}\|_{\text{F}}^2 = &\frac{\rho}{2}(\|\mathcal{Z}^{k+1}\|_{\text{F}}^2 - \|\mathcal{Z}^{k}\|_{\text{F}}^2 \\ &- 2\langle \mathcal{X}^{k+1}, \mathcal{Z}^{k+1} - \mathcal{Z}^{k}\rangle_{\text{F}}). \end{aligned} \tag{51}$$

By applying Property 1 to the inner product term of (50) and utilizing the fact that the observation $\mathcal{Y}$ is invariant to the sampling operator $\mathcal{A}$, we can rewrite this term as,

$$2\langle \mathcal{Y}, \mathcal{A} \circ \mathcal{Z}^{k+1} - \mathcal{A} \circ \mathcal{Z}^{k}\rangle_{\text{F}} = 2\langle \mathcal{Y}, \mathcal{Z}^{k+1} - \mathcal{Z}^{k}\rangle_{\text{F}}. \tag{52}$$

We substitute the results of (50)-(52), into the Lagrangian difference (49), then simplify using Property 10 to obtain,

$$\begin{aligned} L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^{k+1}; \mathcal{B}^k) - L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^k; \mathcal{B}^k) = &\langle \mathcal{Y} + \rho(\mathcal{B}^k + \mathcal{X}^{k+1}), \mathcal{Z}^k - \mathcal{Z}^{k+1}\rangle_{\text{F}} \\ &+ \frac{1}{2}(\|\mathcal{A} \circ \mathcal{Z}^{k+1}\|_{\text{F}}^2 - \|\mathcal{A} \circ \mathcal{Z}^{k}\|_{\text{F}}^2) \\ &+ \frac{\rho}{2}(\|\mathcal{Z}^{k+1}\|_{\text{F}}^2 - \|\mathcal{Z}^{k}\|_{\text{F}}^2). \end{aligned} \tag{53}$$

Now we use the optimality condition of the $\mathcal{Z}$-problem (27),

$$(\mathcal{Y} - \mathcal{A} \circ \mathcal{Z}^{k+1}) \circ (-\mathcal{A}) - \rho(\mathcal{B}^k + \mathcal{X}^{k+1} - \mathcal{Z}^{k+1}) = 0. \tag{54}$$

The $\mathcal{Z}$ terms of this optimality condition (54) can be moved to the right-hand side and then simplified using the idempotent property (see Definition 10 in the Appendix) of the sampling operator,

$$\mathcal{Y} + \rho(\mathcal{B}^k + \mathcal{X}^{k+1}) = \mathcal{A} \circ \mathcal{Z}^{k+1} + \rho \mathcal{Z}^{k+1}. \tag{55}$$

This expression (55) can be substituted into the inner product term of (53) and simplified using Properties 10 and 11 to obtain,

$$\begin{aligned} &L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^{k+1}; \mathcal{B}^k) - L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^k; \mathcal{B}^k) \\ &= -\frac{1}{2}\|\mathcal{A} \circ \mathcal{Z}^{k+1} - \mathcal{A} \circ \mathcal{Z}^k\|_{\mathrm{F}}^2 - \frac{\rho}{2}\|\mathcal{Z}^{k+1} - \mathcal{Z}^k\|_{\mathrm{F}}^2. \end{aligned} \tag{56}$$

By applying the Hadamard-Frobenius inequality (Property 8) to the first term of (56) and using $\max(\mathcal{A}) = 1$, we obtain,

$$L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^{k+1}; \mathcal{B}^k) - L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^k; \mathcal{B}^k) \le -\frac{(1+\rho)}{2}\|\mathcal{Z}^{k+1} - \mathcal{Z}^k\|_{\mathrm{F}}^2. \tag{57}$$

Next we examine the difference in the Lagrangian (25) when $\mathcal{X}$ and $\mathcal{Z}$ are constant,

$$L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^{k+1}; \mathcal{B}^{k+1}) - L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^{k+1}; \mathcal{B}^k) = \rho\langle \mathcal{B}^{k+1} - \mathcal{B}^k, \mathcal{X}^{k+1} - \mathcal{Z}^{k+1}\rangle_{\mathrm{F}},$$

which can be rewritten as

$$L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^{k+1}; \mathcal{B}^{k+1}) - L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^{k+1}; \mathcal{B}^k) = \rho\|\mathcal{B}^{k+1} - \mathcal{B}^k\|_{\mathrm{F}}^2. \tag{58}$$

by the dual variable update step (28). We return to the optimality condition of the $\mathcal{Z}$-problem (54) and use the idempotent property of the sampling operator $\mathcal{A}$ and the dual variable update (28) to obtain,

$$\rho\mathcal{B}^{k+1} = \mathcal{A} \circ \mathcal{Z}^{k+1} - \mathcal{Y}. \tag{59}$$

Substituting this relation (59) into the Lagrangian difference (58) for both $\mathcal{B}^{k+1}$ and $\mathcal{B}^k$ and simplifying yields,

$$L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^{k+1}; \mathcal{B}^{k+1}) - L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^{k+1}; \mathcal{B}^k) = \frac{1}{\rho}\|\mathcal{A} \circ (\mathcal{Z}^{k+1} - \mathcal{Z}^k)\|_{\mathrm{F}}^2. \tag{60}$$

Applying the Hadamard-Frobenius inequality (Property 8) to (60) and using $\max(\mathcal{A}) = 1$ produces the following inequality,

$$L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^{k+1}; \mathcal{B}^{k+1}) - L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^{k+1}; \mathcal{B}^k) \le \frac{1}{\rho}\|\mathcal{Z}^{k+1} - \mathcal{Z}^k\|_{\mathrm{F}}^2. \tag{61}$$

Summing the three Lagrangian differences (48), (57), and (61) and canceling the common terms (namely, $L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^k, \mathcal{B}^k)$ and $L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^{k+1}, \mathcal{B}^k)$) we obtain

$$\begin{aligned} &L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^{k+1}; \mathcal{B}^{k+1}) - L_\rho(\mathcal{X}^k, \mathcal{Z}^k; \mathcal{B}^k) \\ &\le -\frac{\rho}{2}\|\mathcal{X}^{k+1} - \mathcal{X}^k\|_{\mathrm{F}}^2 - \left(\frac{1+\rho}{2} - \frac{1}{\rho}\right)\|\mathcal{Z}^{k+1} - \mathcal{Z}^k\|_{\mathrm{F}}^2. \end{aligned} \tag{62}$$

When $\rho > 1$ the coefficients $c_1 = \frac{\rho}{2}$ and $c_2 = (\frac{1+\rho}{2} - \frac{1}{\rho})$ are strictly positive and we have the desired inequality. □

Lemma 2 *If $\rho > 1$, the sequence $\{\mathcal{X}^k, \mathcal{Z}^k, \mathcal{B}^k\}$ generated by (26)-(28) and its Lagrangian $L_\rho(\mathcal{X}^k, \mathcal{Z}^k; \mathcal{B}^k)$ are bounded.*

Proof By Lemma 1, $L_\rho(\mathcal{X}^k, \mathcal{Z}^k; \mathcal{B}^k) \le L_\rho(\mathcal{X}^0, \mathcal{Z}^0; \mathcal{B}^0)$, hence the Lagrangian has an upper bound. We consider the Lagrangian $L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^{k+1}; \mathcal{B}^{k+1})$ for the lower bound. By the optimality condition of the $\mathcal{Z}$ problem given by (59) we have

$$\begin{aligned} L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^{k+1}; \mathcal{B}^{k+1}) &= \lambda\|\mathcal{X}^{k+1}\|_{\mathrm{TNN}} + \frac{1}{2}\|\mathcal{Y} - \mathcal{A} \circ \mathcal{Z}^{k+1}\|_{\mathrm{F}}^2 \\ &\quad + \langle \mathcal{A} \circ \mathcal{Z}^{k+1} - \mathcal{Y}, \mathcal{X}^{k+1} - \mathcal{Z}^{k+1}\rangle_{\mathrm{F}} + \frac{\rho}{2}\|\mathcal{X}^{k+1} - \mathcal{Z}^{k+1}\|_{\mathrm{F}}^2. \end{aligned} \tag{63}$$

By applying Properties 1, 10, and 11, the second and third terms of (63) can be expressed as follows,

$$\begin{aligned} &\frac{1}{2}\|\mathcal{Y} - \mathcal{A} \circ \mathcal{Z}^{k+1}\|_{\mathrm{F}}^2 + \langle \mathcal{A} \circ \mathcal{Z}^{k+1} - \mathcal{Y}, \mathcal{X}^{k+1} - \mathcal{Z}^{k+1}\rangle_{\mathrm{F}} \\ &\quad = \frac{1}{2}\|\mathcal{Y} - \mathcal{A} \circ \mathcal{X}^{k+1}\|_{\mathrm{F}}^2 - \frac{1}{2}\|\mathcal{A} \circ \mathcal{Z}^{k+1} - \mathcal{A} \circ \mathcal{X}^{k+1}\|_{\mathrm{F}}^2. \end{aligned} \tag{64}$$

By substituting equality (64) into (63), applying Property 8 to the third term, and using $\max \mathcal{A} = 1$, we obtain the inequality,

$$\begin{aligned} &L_\rho(\mathcal{X}^{k+1}, \mathcal{Z}^{k+1}; \mathcal{B}^{k+1}) \\ &\ge \lambda\|\mathcal{X}^{k+1}\|_{\mathrm{TNN}} + \frac{1}{2}\|\mathcal{Y} - \mathcal{A} \circ \mathcal{Z}^{k+1}\|_{\mathrm{F}}^2 + \frac{\rho - 1}{2}\|\mathcal{X}^{k+1} - \mathcal{Z}^{k+1}\|_{\mathrm{F}}^2. \end{aligned} \tag{65}$$

The right hand side of (65) is positive if $\rho \ge 1$. Hence, the Lagrangian is bounded below by zero. Furthermore, due to the presence of $\|\mathcal{X}^k\|_{\mathrm{TNN}}$ and $\|\mathcal{X}^{k+1} - \mathcal{Z}^{k+1}\|_{\mathrm{F}}^2$ in the Lagrangian $L_\rho(\mathcal{X}^k, \mathcal{Z}^k; \mathcal{B}^k)$, the boundedness of the Lagrangian implies that the sequence $\{\mathcal{X}^k, \mathcal{Z}^k\}$ is bounded. By the relation between the dual variable $\mathcal{B}^k$ and the auxiliary variable $\mathcal{Z}^k$ given by (59), the boundedness of $\{\mathcal{Z}^k\}$ implies the boundedness of $\{\mathcal{B}^k\}$. □

Theorem 1 *If $\rho > 1$, every limit point $(\mathcal{X}^*, \mathcal{Z}^*, \mathcal{B}^*)$ of the sequence $\{\mathcal{X}^k, \mathcal{Z}^k, \mathcal{B}^k\}$ generated by the ADMM updates (26)-(28) is a stationary point of $L_\rho(\mathcal{X}, \mathcal{Z}; \mathcal{B})$ and $\mathcal{X}^*$ is a stationary point of the unconstrained minimization problem.*

Proof By Lemmas 1 and 2, $L_\rho(\mathcal{X}^k, \mathcal{Z}^k; \mathcal{B}^k)$ is decreasing and bounded below, hence there exists a subsequence $\{\mathcal{X}^{k_s}, \mathcal{Z}^{k_s}, \mathcal{B}^{k_s}\}$ that converges to a limit point $(\mathcal{X}^*, \mathcal{Z}^*, \mathcal{B}^*)$. By telescoping summation of (43) from $k = 0$ to K, we obtain

$$\begin{aligned} &c_1 \sum_{k=0}^{K} \|\mathcal{X}^{k+1} - \mathcal{X}^k\|_{\mathrm{F}}^2 + c_2 \sum_{k=0}^{K} \|\mathcal{Z}^{k+1} - \mathcal{Z}^k\|_{\mathrm{F}}^2 \\ &\quad \le L_\rho(\mathcal{X}^0, \mathcal{Z}^0; \mathcal{B}^0) - L_\rho(\mathcal{X}^{K+1}, \mathcal{Z}^{K+1}; \mathcal{B}^{K+1}). \end{aligned} \tag{66}$$

Since the Lagrangian L is bounded, we get

$$\sum_{k=0}^{\infty} \|\mathcal{X}^{k+1} - \mathcal{X}^k\|_2^2 \le \infty, \quad \sum_{k=0}^{\infty} \|\mathcal{Z}^{k+1} - \mathcal{Z}^k\|_2^2 \le \infty,$$

which implies that $\|\mathcal{X}^{k+1} - \mathcal{X}^k\|_2^2 \to 0$, $\|\mathcal{Z}^{k+1} - \mathcal{Z}^k\|_2^2 \to 0$. Furthermore, by using the dual variable update (28), the optimality condition of the $\mathcal{Z}$ problem (59), Property 8, and $\max(\mathcal{A}) = 1$, it can be shown that,

$$\|\mathcal{X}^{k+1} - \mathcal{Z}^{k+1}\|_{\mathrm{F}}^2 = \|\mathcal{B}^{k+1} - \mathcal{B}^k\|_{\mathrm{F}}^2 = \frac{1}{\rho}\|\mathcal{A} \circ (\mathcal{Z}^{k+1} - \mathcal{Z}^k)\|_{\mathrm{F}}^2 \le \frac{1}{\rho}\|\mathcal{Z}^{k+1} - \mathcal{Z}^k\|_{\mathrm{F}}^2 \to 0.$$

Therefore, we have $\mathcal{X}^* = \mathcal{Z}^*$ and $(\mathcal{X}^{k_s+1}, \mathcal{Z}^{k_s+1}, \mathcal{B}^{k_s+1}) \rightarrow (\mathcal{X}^*, \mathcal{Z}^*, \mathcal{B}^*)$.

Next we consider elements of the subgradient, $\mathcal{P}^{k_s} \in \partial\|\mathcal{X}^{k_s}\|_{\text{TNN}}$, for the subsequence $\{\mathcal{X}^{k_s}\}$ that converges to $\mathcal{X}^*$. It follows from the formula of the TNN's subgradient (7) that

$$\|\mathcal{P}^{k_s}\|_2 \leq \|\mathcal{U}\mathcal{V}^T\|_2 + \|\mathcal{W}\|_2 \leq 2. \tag{67}$$

Thus $\{\mathcal{P}^{k_s}\}$ is a bounded sequence, so there exists a convergent subsequence. Let $\mathcal{P}^*$ denote the limit point of this subsequence. Without loss of generality, let us use $\{\mathcal{P}^{k_s}\}$ to denote this convergent subsequence. Since $\mathcal{P}^{k_s} \in \partial\|\mathcal{X}^{k_s}\|_{\text{TNN}}$, we have $\|\mathcal{X}\|_{\text{TNN}} - \|\mathcal{X}^{k_s}\|_{\text{TNN}} \geq \langle\mathcal{P}^{k_s}, \mathcal{X} - \mathcal{X}^{k_s}\rangle_{\text{F}}, \forall\mathcal{X}$. By the continuity of the TNN norm, $\lim\limits_{k_s\rightarrow\infty} \|\mathcal{X}^{k_s}\|_{\text{TNN}} = \|\mathcal{X}^*\|_{\text{TNN}}$. Similarly, by the continuity of the Frobenius inner product, $\lim\limits_{k_s\rightarrow\infty} \langle\mathcal{P}^{k_s}, \mathcal{X}-\mathcal{X}^{k_s}\rangle_{\text{F}} = \langle\mathcal{P}^*, \mathcal{X}-\mathcal{X}^*\rangle_{\text{F}}$. Thus, $\|\mathcal{X}\|_{\text{TNN}} - \|\mathcal{X}^*\|_{\text{TNN}} \geq \langle\mathcal{P}^*, \mathcal{X} - \mathcal{X}^*\rangle_{\text{F}}$, and hence $\mathcal{P}^* \in \partial\|\mathcal{X}^*\|_{\text{TNN}}$.

Evaluating the optimality condition of the $\mathcal{X}$-problem (26) at k_s and taking the limit of the sequence we obtain,

$$\lambda\mathcal{P}^* + \rho(\mathcal{B}^* + \mathcal{X}^* - \mathcal{Z}^*) = 0, \tag{68}$$

Taking the same limit for the optimality condition of the $\mathcal{Z}$-problem (59) we obtain,

$$\rho\mathcal{B}^* = \mathcal{A} \circ \mathcal{Z}^* - \mathcal{Y}, \tag{69}$$

hence $(\mathcal{X}^*, \mathcal{Z}^*, \mathcal{B}^*)$ is a stationary point of the Lagrangian $L_\rho(\mathcal{X}, \mathcal{Z}; \mathcal{B})$. Furthermore, by substituting the limit of the $\mathcal{Z}$-problem (69) into the limit of the $\mathcal{X}$-problem (68) and using the fact $\mathcal{X}^* = \mathcal{Z}^*$, we obtain,

$$\lambda\mathcal{P}^* + \mathcal{A} \circ \mathcal{Z}^* - \mathcal{Y} = 0, \tag{70}$$

which implies that $\mathcal{X}^*$ is a stationary point of the original unconstrained minimization problem (23). □

Remark 1 We consider additive Gaussian noise in Equation (22) because it is straight-forward to handle. Specifically in the unconstrained problem (23) the data fitting term for Gaussian noise is modeled by a least-squares misfit whose gradient is linear. Hence the corresponding optimization problem is efficient to solve. For other types of noise we can still use TNN as a low-rank regularization term, but we would need to use a different data fidelity term. As one example, with Poisson noise the data discrepancy is modeled by a log-likelihood term. The convergence analysis conducted in this paper is applicable to non-additive noise, so long as the corresponding data fidelity term is convex and Lipschitz continuous.

With the proof of convergence, we are now ready to apply TNN-ADMM to numerical experiments for data completion and denoising. Furthermore, to ensure the success of our experiments we restrict the parameter selection to $\rho > 1$ based on the convergence analysis.

4 Numerical Experiments

In this section we illustrate simultaneous denoising and completion of seismic data arranged into a tensor using the TNN-ADMM algorithm. We perform experiments on two synthetic and one real dataset. The real data was recorded offshore Western Australia. (While we chose not to do so here, one could also test the method using benchmark datasets, such as those contained in the OpenFWI suite [7].) For both synthetic examples, we compare the performance of TNN-ADMM to a competing method, multi-channel singular spectrum analysis (MSSA). MSSA

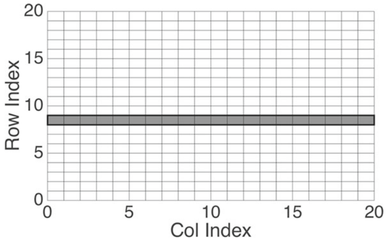


Fig. 1 A 20 × 20 $x - y$ grid slice representing part of the computational domain with the ninth row highlighted. Each grid cell is associated with a seismic trace recorded at that location

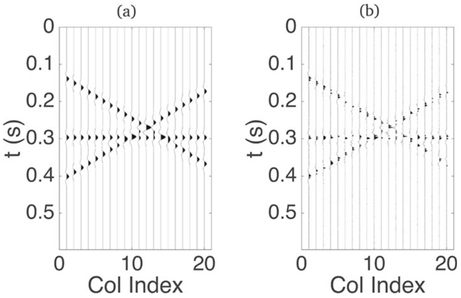


Fig. 2 **a** The slice of the ground truth synthetic data corresponding to the ninth row of the grid. **b** The noisy observation resulting from adding Gaussian noise with a SNR of 10 to 1 and subsequently decimating 60% of the data

is a reconstruction and denoising algorithm that has been successfully applied previously to seismic data [26]. MSSA has also been applied to astrophysical data (see [36]) where the authors state that singular spectrum analysis or "SSA is a non-parametric analysis tool for time-series analysis.... It makes no strong prior assumptions about the spectrum. An oscillation does not have to be periodic to be accurately represented." The paper by Ghil et al. [11] provides a review. Our goal with this comparison is to quantify how TNN-ADMM performs relative to a reasonable established method in terms of accuracy and performance.

4.1 Linear Synthetic Example

We use a synthetic dataset consisting of three linear events created using a code made publicly available by Chen et al. [6]. The events are recorded on a grid of dimension 20 × 20 with 2 ms time sampling and a recording duration of 0.6 s. This data arranged as an order-3 tensor, with dimensions of (x, y, t), represents the underlying ground truth. We add Gaussian noise to this data with a signal-to-noise ratio (SNR) of 10 to 1 and then apply a sampling operator that randomly decimates 60% of the data. For the purposes of our experiments, we regard the data *before* adding noise and decimating as the ground truth. In Fig. 1 we plot an $x - y$ grid slice of the domain. Each grid cell is associated with a seismic trace recorded at that location, hence each row is associated with a seismic gather. In Fig. 2a, b we plot a slice of the ground truth synthetic data and the noisy decimated observation respectively along a fixed row of the grid.

The orientation of a tensor is the order of its dimensions, and reordering or permuting its dimensions produces a different orientation. We arrange the data into a tensor using the most-square orientation for optimal results [27]. (The most-square orientation is the permutation of dimensions that produces the most-square frontal faces out of all possible orientations.) We then apply TNN-ADMM with parameter values of $\rho = 1.1$ and $\lambda = 0.1$.

These parameters were chosen based on a grid search in which we vary the magnitude of the parameters and select the pair that produced the least relative error. (In the case of real

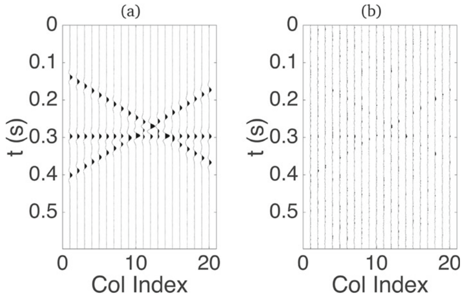


Fig. 3 **a** The recovered result after applying TNN-ADMM to the noisy observation shown in Fig. 2b. **b** The difference between the recovered result and the true data, amplified by a factor of five. The error in reconstruction is 9%

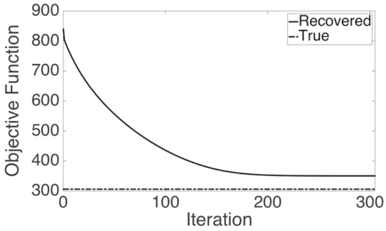


Fig. 4 The TNN-ADMM objective function versus iteration for reconstruction of the synthetic data. The dashed line shows the value of the objective function for the true data

data we use an approximation to the error.) We note that in our prior work (Popa et al. [27]) we discuss tuning parameters for both MSSA and TNN-ADMM in some detail. (Specifically, see p. V225 of [27].) We do not repeat that discussion here but instead, refer the reader to that paper for details. The relative error is computed as the Frobenius norm of the difference between the ground truth and the reconstructed data produced by the algorithm, divided by the Frobenius norm of the ground truth.

Additionally, the grid search was restricted to values of $\rho > 1$ to satisfy the theory presented in the convergence analysis. A grid search suggests that our algorithm is not particularly sensitive to ρ, in the sense that for many values of ρ the algorithm will still converge but at different rates. In general λ should be chosen proportional to the noise. We define convergence of the algorithm to be when the relative error between consecutive iterations reaches a tolerance of 10^{-4}. The TNN-ADMM algorithm converges in 305 iterations with a runtime of 8.21 s. The error in reconstruction is error around 9%. We plot the recovered result and difference plots in Fig. 3a, b respectively. From these plots we observe accurate reconstruction and successful denoising. We plot the unconstrained TNN-ADMM objective function (23) as a function of iteration in Fig. 4. The objective function decreases with each iteration, and for this number of iterations it converges to a value of 350.33, which is slightly greater than the true objective value of 306.11.

Next we compare our results to a competing algorithm, MSSA. The MSSA method is based on reducing the rank of block Hankel matrices formed from frequency slices of the data [6, 26]. The MSSA algorithm uses a parameter N to represent the number of non-zero singular values preserved when reducing the rank of the block Hankel matrix for each frequency slice. Each block Hankel matrix of the underlying data has the same number of non-zero singular values as the number of linear events, so we select $N = 3$ for the synthetic experiment. We use a version of the MSSA algorithm that weights and sums the observation and the rank-reduced data to produce the reconstructed data. This algorithm can be expressed as,

$$\mathcal{X}^{k+1} = a_k\mathcal{Y} + (\mathcal{I} - a_k\mathcal{A}) \circ F(\mathcal{X}^k), \quad k = 1, 2, ..., k_{max}, \tag{71}$$

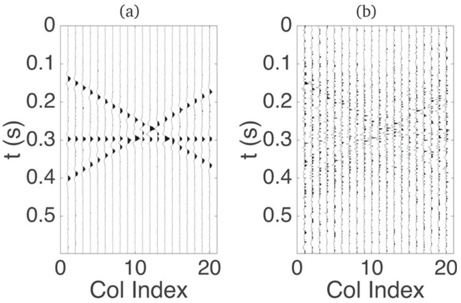


Fig. 5 **a** The recovered result from applying MSSA to the noisy observation shown in Fig. 2b. **b** The difference between the recovered result and the true data, amplified by a factor of five. The error in reconstruction is 36%

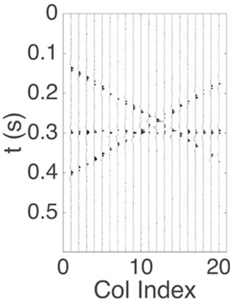


Fig. 6 A slice of the noisy, decimated observation with a SNR of 10–3. This slice corresponds with the ninth row of the grid as shown in Fig. 1

where a_k is the weighting parameter at iteration k, and $F(\mathcal{X}^k)$ represents the result of MSSA reducing the rank of the block Hankel matrices of the frequency slices of $\mathcal{X}^k$. We select the initial and final values of the weighting parameter as $a_1 = 1$ and $a_{k_{max}} = 0$. This parameter decreases linearly over a fixed number of iterations. Experimenting over a range of values for the fixed number of iterations, we found 10 iterations to be optimal for this synthetic example. The runtime for 10 iterations of MSSA is 20.78 s for this experiment. While MSSA converges in fewer iterations than TNN-ADMM, the computational cost per iteration is significantly greater for MSSA due to the rank reduction of the block Hankel matrix for each frequency slice. So the overall time to convergence for MSSA for this example is more than twice the time for TNN-ADMM convergence.

We plot the recovered result and the error in Fig. 5. We observe the reconstruction result has error around 36%, significantly greater than the error from TNN-ADMM. In the recovered and difference plots we see noise is still present in the reconstructed result – MSSA was only able to partially denoise this example.

We now repeat this same experiment with data that has 3 times the added noise of the previous experiment (a SNR of 10–3). We apply the same sampling operator which decimates 60% of the data and plot this observation in Fig. 6. On the scale of the data it is difficult to observe the increase in noise compared to Fig. 2b. For TNN-ADMM this data uses parameters of $\rho = 1.1$, $\lambda = 1.5$ and a convergence tolerance of 10^{-4}. The algorithm converges in 43 iterations and 1.19 s, with error around 19%. We plot the reconstructed result and the difference in Fig. 7a, b respectively. We observe that the error is concentrated primarily where the events occur (error in the event amplitude).

We then apply MSSA to this example and plot a recovered and difference slice in Fig. 8. We observe these results appear very similar to the previous example with less noise, with a runtime of 20.72 s and error in recovery around 37%. These results indicate the performance

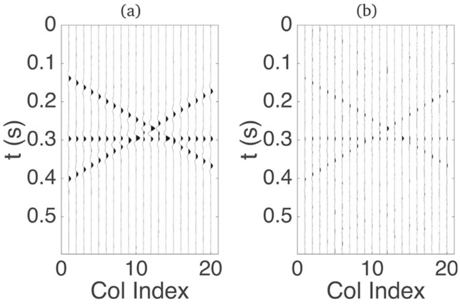


Fig. 7 **a** The recovered result from applying TNN-ADMM to the observation shown in Fig. 6. **b** The difference between the recovered result and the true data, amplified by a factor of two. The error in reconstruction is 19%

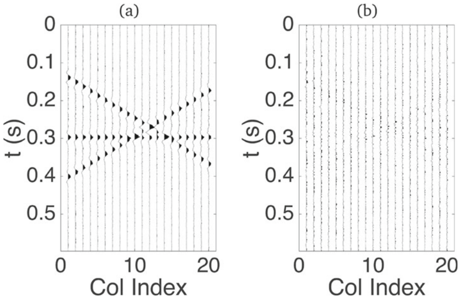


Fig. 8 **a** The recovered result from applying MSSA to the observation shown in Fig. 6. **b** The difference between the recovered result and the true data, amplified by a factor of two. The error in reconstruction is 37%

Table 1 The initial and final reconstruction errors for TNN-ADMM and MSSA

	SNR 10–1 (%)	SNR 10–3 (%)
Initial error	77	79
TNN-ADMM	9	19
MSSA	36	37

The initial error is the error between the initial observation (with noise added and 60% of traces decimated) and the ground truth

of MSSA is less sensitive to the SNR than TNN-ADMM. We summarize the reconstruction error for each synthetic experiment in Table 1.

4.2 Parabolic Synthetic Example

We construct a similar synthetic dataset as in the previous example but with curved, parabolic events rather than linear events. One of the events is linear in x and parabolic in y, one is linear in y and parabolic in x, and one is parabolic in both x and y. The events are recorded on a grid of dimension 30×50 with 2 ms time sampling and a recording duration of 1 s. Gaussian noise with SNR 10–3 is added to the data. The resulting noisy data is then decimated, removing 60% selected randomly. We show the synthetic data and noisy observation in Fig. 9. We apply TNN-ADMM and MSSA to denoise and complete the noisy observation.

For the TNN-ADMM method we tune the parameters ρ and λ by determining the pair that produces the least error while restricting the grid search to $\rho > 1$. For this example, we find the parameters $\rho = 1.1$ and $\lambda = 3$ are optimal. We use a convergence tolerance of 10^{-4}. The TNN-ADMM method converges in 41 iterations, with a runtime of 6.37 s. We plot the recovered and difference plots in Fig. 10. The final relative error is 16%.

Fig. 9 **a** A slice of the parabolic synthetic data. This data is considered the ground truth. **b** The noisy observation resulting from adding noise and decimating the ground truth

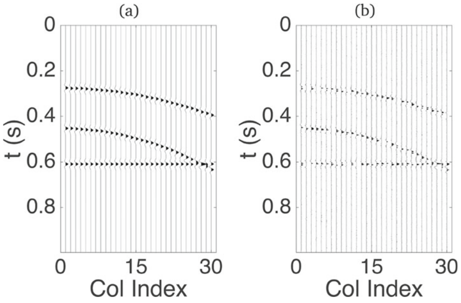


Fig. 10 **a** The recovered result produced by applying TNN-ADMM to the noisy observation for the parabolic synthetic data shown in Fig. 9. **b** The difference between the recovered result and the ground truth. The relative error in reconstruction is approximately 16%

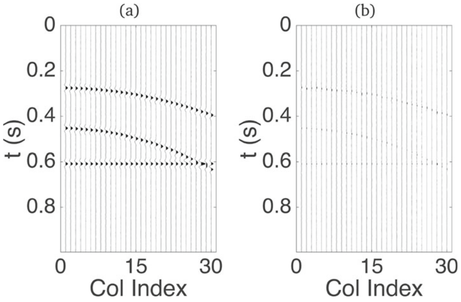
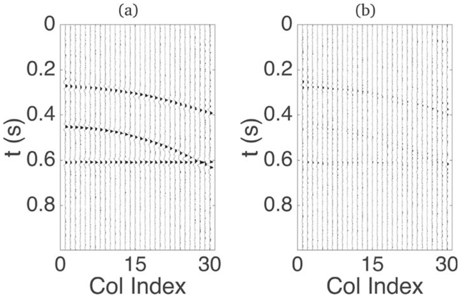


Fig. 11 **a** The recovered result from applying MSSA to the noisy observation for the parabolic synthetic data shown in Fig. 9. **b** The difference between the recovered result and the ground truth. The relative error is 76%



For MSSA we run 10 iterations with a linearly decreasing weighting parameter as in the previous example. We determine that the parameter value $N = 25$ produces the least error for the example. We plot the recovered and difference plots in Fig. 11. The MSSA method has a runtime of 228.75 s and results in a relative error of 76%.

4.3 Real Data Example

In a marine seismic experiment, a disturbance (wave) is generated when a source of energy such as an airgun emits a pressure pulse in the ocean. The sound waves generated by the source are transmitted into the subsurface and reflect off of rock and fluid interfaces back up to the ocean surface. A seismic survey vessel is equipped with the airgun source and an array of streamers containing receivers which record the returning waves.

For our real data experiments we use a single shot from seismic survey data recorded offshore Western Australia. We plot the acquisition geometry for this shot location in Fig. 12.

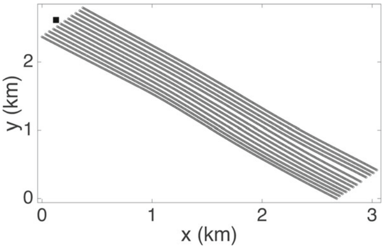


Fig. 12 The acquisition geometry for a single shot location of a survey conducted offshore Western Australia. The circles represent receivers and the square in the top left represents the source location

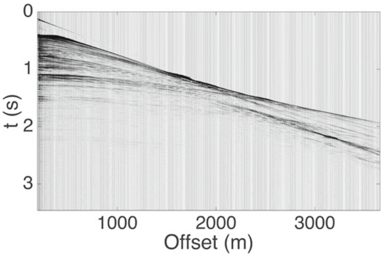


Fig. 13 The data gather of one streamer shown in Fig. 12 from offshore Western Australia

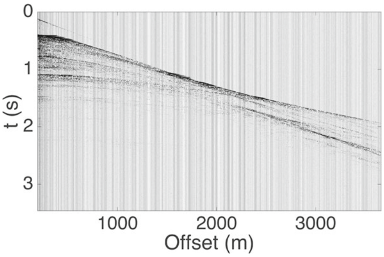


Fig. 14 A gather produced by adding noise and randomly decimating 60% of the real data shown in Fig. 13

The acquisition geometry shows the relative locations of the source and receivers. The data is recorded using 12 streamers, each with 288 receivers and a 2 ms sampling rate.

Real data contains noise from many sources such as ocean currents and wind. The addition of noise makes seismic data more difficult to interpret, increasing uncertainty in identifying subsurface resources. Filling in missing data and removing noise improves the precision of interpretation and hydrocarbon detection.

Quantifying the success of a denoising method on real data is difficult without considerable processing by experts. Instead here we again add 10% artificial random Gaussian noise to the recorded data and measure removing this added noise and reconstructing the data. We compute the relative reconstruction error by taking the Frobenius norm of the difference between the recorded and reconstructed data and normalizing by the recorded data.

The data for all receivers along a streamer constitutes a gather. Each gather may be stored as a matrix, and these 12 gathers become slices of a three-dimensional tensor. In Fig. 13 we plot a single streamer gather of the recorded data. We then add random Gaussian noise and apply a sampling operator to decimate 60% of the data. The indices where the sampling operator decimates the data are chosen at random. Figure 14 shows this noisy, decimated gather. Applying this sampling operator to the entire tensor of data, the resulting decimated data is the "observation," from which we will reconstruct and denoise to recover the recorded data. The error between the recorded data and the noisy, decimated observation is 77.89%.

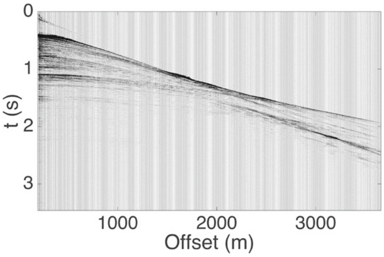


Fig. 15 The result from applying TNN-ADMM to the decimated data shown in Fig. 14 to simultaneously reconstruct and denoise

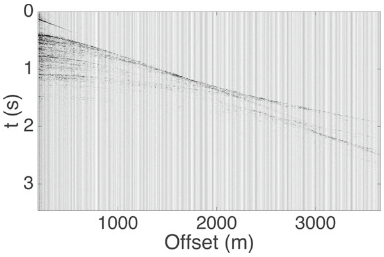


Fig. 16 The difference between the denoised and reconstructed gather and the recorded data shown in Fig. 14. The error in this reconstruction is about 29%

We tune the ρ and λ parameters using the same procedure as described in the synthetic data example. Note that λ determines the recovery performance that typically depends on the noise level, while ρ controls the convergence rate. We showed a sufficient condition of $\rho > 1$ for guaranteed convergence, but in practice, ADMM converges faster for smaller values of ρ. In the case of real data, we can determine a reasonable value for ρ by considering a subsampled portion of the data (treated as an observation) while the recorded data, before subsampling, is treated as the ground truth. This subsampling results in a completion problem where the solution is known and the parameter ρ can be tuned. The λ parameter must normally be manually tuned without a priori knowledge of the noise present in the data by an optimization over values which give the best denoising result. For our example, which includes added synthetic noise, we are able to tune the lambda parameter to remove as much of this noise as possible. We find that $\rho = 1.1$ and $\lambda = 400$ are reasonable parameter choices for our problem. We use a convergence tolerance of 10^{-3}. The TNN-ADMM method converges in 83 iterations for this example. The reconstructed result is shown in Fig. 15, and the difference between the reconstructed and recorded data is plotted in Fig. 16. We observe successful reconstruction with an error of approximately 29.29%.

To better understand how effective the denoising process is, we perform an additional experiment using the recorded data without adding any artificial noise. We apply the same sampling operator to decimate 60% of the data. we then use the constrained version of TNN-ADMM solely for data completion without denoising to recover the missing data. The error for this example is approximately 25.53%. The fact that the two errors are similar between data completion alone and simultaneous denoising and completion indicates that TNN-ADMM is able to remove the majority of the artificial noise in this experiment.

5 Conclusions

The TNN-ADMM method is useful for simultaneous reconstruction and denoising. To prove convergence of this method we first show that the Lagrangian evaluated at each iteration and the sequence generated by ADMM for each variable (the reconstructed data, the auxiliary variable, and the dual variable) is sufficiently decreasing and bounded. We are then able to show that the limit point of each sequence is a stationary point of the Lagrangian and of the unconstrained minimization problem.

We compare TNN-ADMM with MSSA for reconstruction and denoising for two synthetic datasets. For the linear synthetic experiment MSSA has similar performance for two different SNRs, but TNN-ADMM performs better in each case both in terms of final error and computational runtime. For the parabolic synthetic experiment TNN-ADMM strongly outperforms MSSA. Finally we apply the TNN-ADMM denoising and reconstruction algorithm to real data from offshore Western Australia. The algorithm appears to reconstruct the data and remove the majority of the added noise. Properly denoising real data requires significant processing to estimate the noise level in the data and the sources of the noise which is beyond the scope of this work. However, in future incorporating geoscience expertise could further illucidate the promise of using TNN-ADMM to denoise and reconstruct real data.

Acknowledgements We are grateful to two anonymous reviewers for their insightful comments which helped to improve this manuscript. This research is partially supported by the sponsors of the UT Dallas "3D+4D Seismic FWI" research consortium and by NSF CAREER Award no. 1846690. We thank BHP, and its partners, Santos and INPEX, for providing the offshore Western Australia data and James Cai for his expertise with this dataset and time-lapse data processing in general. We also thank David Lumley, Kai Xiong, and Wei Zhou for their help in preparing the real data used in our experiments.

Data Availibility The synthetic data used in this study is available upon request. The real data in this study is strictly confidential and not available for public use.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Appendix

We present here some additional definitions and properties which form part of the background material but which may be known to those familiar with tensor algebra.

Definition 6 (Frobenius inner product [12]) The Frobenius inner product between two real-valued tensors of the same size, $\mathcal{A}$ and $\mathcal{B}$, is defined as the sum of the entry-wise product of elements,

$$\langle \mathcal{A}, \mathcal{B} \rangle_{\mathrm{F}} = \sum_{i=1}^{N} \mathcal{A}_i \mathcal{B}_i . \tag{72}$$

The Frobenius inner product induces the *Frobenius norm*.

Definition 7 (Frobenius norm [12]) The Frobenius norm of a tensor $\mathcal{A}$ is given by the sum of the squares of the magnitude of the elements,

$$\|\mathcal{A}\|_{\mathrm{F}}^2 = \langle \mathcal{A}, \mathcal{A} \rangle_{\mathrm{F}} = \sum_{i=1}^{N} |\mathcal{A}_i|^2 . \tag{73}$$

Definition 8 (Hadamard Identity [15]) The Hadamard identity tensor, denoted $\mathcal{I}$, is a tensor comprised entirely of ones which has the property,

$$\mathcal{A} \circ \mathcal{I} = \mathcal{I} \circ \mathcal{A} = \mathcal{A}, \tag{74}$$

for any tensor $\mathcal{A}$.

Definition 9 (Hadamard Inverse [29]) The *Hadamard inverse* of a tensor $\mathcal{A}$ with only non-zero elements is denoted $\mathcal{A}^{-1}$ and has the property,

$$\mathcal{A} \circ \mathcal{A}^{-1} = \mathcal{A}^{-1} \circ \mathcal{A} = \mathcal{I}. \tag{75}$$

Note that the Hadamard inverse is the entry-wise inverse, such that $(\mathcal{A}^{-1})_i = \frac{1}{\mathcal{A}_i}$. Since the Hadamard inverse is only defined for tensors with non-zero entries, sampling operators do not have a Hadamard inverse, except for the Hadamard identity tensor.

Definition 10 (Idempotent [15]) A tensor $\mathcal{A}$ is *idempotent* under the Hadamard product if its product with itself produces the same tensor, i.e.,

$$\mathcal{A} \circ \mathcal{A} = \mathcal{A}. \tag{76}$$

It can be observed that any tensor comprised exclusively of zeros and ones is idempotent under the Hadamard product since zero and one are idempotent under scalar multiplication. Hence, every sampling operator is idempotent.

Property 9 (Symmetry [12]) The Frobenius inner product is *symmetric* for real tensors $\mathcal{A}$ and $\mathcal{B}$,

$$\langle \mathcal{A}, \mathcal{B} \rangle_{\mathrm{F}} = \langle \mathcal{B}, \mathcal{A} \rangle_{\mathrm{F}}. \tag{77}$$

Property 10 (Linearity [12]) The Frobenius inner product is *linear*, for tensors $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, and real-valued scalar s,

$$\langle \mathcal{A} + \mathcal{B}, \mathcal{C} \rangle_{\mathrm{F}} = \langle \mathcal{A}, \mathcal{C} \rangle_{\mathrm{F}} + \langle \mathcal{B}, \mathcal{C} \rangle_{\mathrm{F}} \quad \text{(additivity)} \tag{78}$$

$$\langle s\mathcal{A}, \mathcal{B} \rangle_{\mathrm{F}} = s \langle \mathcal{A}, \mathcal{B} \rangle_{\mathrm{F}} \qquad \text{(homogeneity).} \tag{79}$$

These fundamental inner product properties imply the *polarization identity* (Property 11) and *orthogonal invariance* (Property 12) for the Frobenius norm [12]. In addition, we state a new property (Properties 1) regarding the Frobenius inner product.

Property 11 (Polarization Identity [12]) For real-valued tensors $\mathcal{A}$ and $\mathcal{B}$,

$$\|\mathcal{A} + \mathcal{B}\|_{\mathrm{F}}^2 = \|\mathcal{A}\|_{\mathrm{F}}^2 + 2\langle \mathcal{A}, \mathcal{B} \rangle_{\mathrm{F}} + \|\mathcal{B}\|_{\mathrm{F}}^2. \tag{80}$$

Property 12 (Orthogonal Invariance [12]) The Frobenius norm is invariant to orthogonal tensors, such that for any tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_p}$ and any orthogonal tensors $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times \ldots \times n_p}$ and $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times \ldots \times n_p}$,

$$\|\mathcal{U}\mathcal{A}\|_{\mathrm{F}} = \|\mathcal{A}\|_{\mathrm{F}}, \quad \|\mathcal{A}\mathcal{V}^T\|_{\mathrm{F}} = \|\mathcal{A}\|_{\mathrm{F}}. \tag{81}$$

It is worth noting that the TNN also has the orthogonal invariance property [31].

Definition 11 (Kronecker Product [14]) The *Kronecker product* of matrices $A \in \mathbb{R}^{n_1 \times n_2}$ and $B \in \mathbb{R}^{m_1 \times m_2}$ is a matrix denoted $A \otimes B \in \mathbb{R}^{n_1 m_1 \times n_2 m_2}$, defined as follows,

$$A \otimes B = \begin{bmatrix} A_{1,1}B & \ldots & A_{1,n_2}B \\ \vdots & \ddots & \vdots \\ A_{n_1,1}B & \ldots & A_{n_1,n_2}B \end{bmatrix}, \tag{82}$$

where $A_{i,j}$ is the scalar entry of A in row i and column j.

References

1. Abma, R., Kabir, N.: 3D interpolation of irregular data with a POCS algorithm. Geophysics **71**(6), E91–E97 (2006). https://doi.org/10.1190/1.2356088
2. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. **3**, 1–122 (2011). https://doi.org/10.1561/2200000016
3. Cai, J.F., Candès, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion. SIAM J. Optim. **20**(4), 1956–1982 (2010). https://doi.org/10.1137/080738970
4. Candès, E., Recht, B.: Exact matrix completion via convex optimization. Found. Comput. Math. **9**(6), 717–772 (2009). https://doi.org/10.1007/s10208-009-9045-5
5. Charalambous, C.: A method to overcome the ill-conditioning problem of differentiable penalty functions. Oper. Res. **28**(3), 650–667 (1980)
6. Chen, Y., Huang, W., Zhang, D., Chen, W.: An open-source Matlab code package for improved rank-reduction 3D seismic data denoising and reconstruction. Comput. Geosci. **95**, 59–66 (2016). https://doi.org/10.1016/j.cageo.2016.06.017
7. Deng, C., Feng, S., Wang, H., Zhang, X., Jin, P., Feng, Y., Zeng, Q., Chen, Y., Lin, Y.: OpenFWI: large-scale multi-structural benchmark datasets for seismic full waveform inversion. arXiv preprint arXiv:2111.02926 (2021)
8. Du, S., Xiao, Q., Shi, Y., Cucchiara, R., Ma, Y.: Unifying tensor factorization and tensor nuclear norm approaches for low-rank tensor completion. Neurocomputing **458**, 204–218 (2021). https://doi.org/10.1016/j.neucom.2021.06.020
9. Eckstein, J., Bertseka, D.: On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. Math. Program. **55**, 293–318 (1992). https://doi.org/10.1007/BF01581204
10. Ely, G., Aeron, S., Hao, N., Kilmer, M.E.: 5D seismic data completion and denoising using a novel class of tensor decompositions. Geophysics **80**, V83–V95 (2015). https://doi.org/10.1190/geo2014-0467.1
11. Ghil, M., Allen, M.R., Dettinger, M.D., Ide, K., Kondrashov, D., Mann, M.E., Robertson, A.W., Saunders, A., Tian, Y., Varadi, F., Yiou, P.: Advanced spectral methods for climatic time series. Rev. Geophys. (2002). https://doi.org/10.1029/2000RG000092
12. Golub, G.H., Van Loan, C.F.: Matrix computations. In: Johns Hopkins Studies in the Mathematical Sciences, 4th edn. Johns Hopkins University Press, Baltimore (2013)
13. He, W., Yokoya, N., Yuan, L., Zhao, Q.: Remote sensing image reconstruction using tensor ring completion and total variation. IEEE Trans. Geosci. Remote Sens. **57**(11), 8998–9009 (2019). https://doi.org/10.1109/TGRS.2019.2924017
14. Horn, R.A., Johnson, C.R.: Topics in Matrix Analysis. Cambridge University Press, Cambridge (1991)
15. Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press, Cambridge (2013)
16. Kilmer, M.E., Martin, C.D.: Factorization strategies for third-order tensors. Linear Algebra Appl. **435**(3), 641–658 (2011). https://doi.org/10.1016/j.laa.2010.09.020
17. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Rev. **51**(3), 455–500 (2009). https://doi.org/10.1137/07070111X
18. Kreimer, N., Stanton, A., Sacchi, M.D.: Tensor completion based on nuclear norm minimization for 5D seismic data reconstruction. Geophysics **78**(6), 273–284 (2013). https://doi.org/10.1190/geo2013-0022.1
19. Kumar, R., Da Silva, C., Akalin, O., Aravkin, A.Y., Mansour, H., Recht, B., Herrmann, F.J.: Efficient matrix completion for seismic data reconstruction. Geophysics **80**, V97–V113 (2015). https://doi.org/10.1190/geo2014-0369.1
20. Li, T., Ma, J.: T-SVD based non-convex tensor completion and robust principal component analysis. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 6980–6987. IEEE (2021). https://doi.org/10.1109/ICPR48806.2021.9412248
21. Long, Z., Liu, Y., Chen, L., Zhu, C.: Low rank tensor completion for multiway visual data. Signal Process. **155**, 301–316 (2019). https://doi.org/10.1016/j.sigpro.2018.09.039
22. Lu, C., Feng, J., Chen, Y., Liu, W., Lin, Z., Yan, S.: Tensor robust principal component analysis with a new tensor nuclear norm. IEEE Trans. Pattern Anal. Mach. Intell. **42**(4), 925–938 (2019). https://doi.org/10.1109/TPAMI.2019.2891760
23. Martin, C.D., Shafer, R., Larue, B.: An order-p tensor factorization with applications in imaging. SIAM J. Sci. Comput. **35**(1), A474–A490 (2013)
24. Mohd Sagheer, S.V., George, S.N.: Denoising of low-dose CT images via low-rank tensor modeling and total variation regularization. Artif. Intell. Med. **94**, 1–17 (2019). https://doi.org/10.1016/j.artmed.2018.12.006

25. Nocedal, J., Wright, S.J.: Penalty and Augmented Lagrangian Methods, pp. 497–528. Springer, New York (2006)
26. Oropeza, V., Sacchi, M.: Simultaneous seismic data denoising and reconstruction via multichannel singular spectrum analysis. Geophysics **76**, V25–V32 (2011). https://doi.org/10.1190/1.3552706
27. Popa, J., Minkoff, S.E., Lou, Y.: An improved seismic data completion algorithm using low-rank tensor optimization: cost reduction and optimal data orientation. Geophysics **86**(3), V219–V232 (2021). https://doi.org/10.1190/geo2020-0539.1
28. Popa, J., Minkoff, S.E., Lou, Y.: Tensor-based reconstruction applied to regularized time-lapse data. Geophys. J. Int. **231**(1), 638–649 (2022). https://doi.org/10.1093/gji/ggac211
29. Reams, R.: Hadamard inverses, square roots and products of almost semidefinite matrices. Linear Algebra Appl. **288**, 35–43 (1999). https://doi.org/10.1016/S0024-3795(98)10162-3
30. Rockafellar, R.T., Wets, R.J.B.: Subderivatives and Subgradients, pp. 298–348. Springer, Berlin Heidelberg (1998)
31. Semerci, O., Hao, N., Kilmer, M.E., Miller, E.L.: Tensor-based formulation and nuclear norm regularization for multienergy computed tomography. IEEE Trans. Image Process. **23**(4), 1678–1693 (2014). https://doi.org/10.1109/TIP.2014.2305840
32. Trickett, S., Burroughs, L., Milton, A., Walton, L., Dack, R.: Rank-reduction-based trace interpolation. SEG Tech. Program Expand. Abstr. **2010**, 3829–3833 (2010). https://doi.org/10.1190/1.3513645
33. Wang, A., Zhou, G., Zhao, Q.: Guaranteed robust tensor completion via $*_L$-SVD with applications to remote sensing data. Remote Sens. **13**(18), 1–45 (2021). https://doi.org/10.3390/rs13183671
34. Wang, H., Zhang, F., Wang, J., Huang, T., Huang, J., Liu, X.: Generalized nonconvex approach for low-tubal-rank tensor recovery. IEEE Trans. Neural Netw. Learn. Syst. (2021). https://doi.org/10.1109/TNNLS.2021.3051650
35. Wang, Y., Yin, W., Zeng, J.: Global convergence of admm in nonconvex nonsmooth optimization. J. Sci. Comput. **78**(1), 29–63 (2019)
36. Weinberg, M., Petersen, M.: Using multichannel singular spectrum analysis to study galaxy dynamics. Mon. Not. R. Astron. Soc. **501**(4), 5408–5423 (2021). https://doi.org/10.1093/mnras/staa3997
37. Xue, J., Zhao, Y., Liao, W., Cheung-Wai Chan, J.: Nonconvex tensor rank minimization and its applications to tensor recovery. Inf. Sci. **503**, 109–128 (2019). https://doi.org/10.1016/j.ins.2019.06.061
38. Yang, M., Luo, Q., Li, W., Xiao, M.: Multiview clustering of images with tensor rank minimization via nonconvex approach. SIAM J. Imag. Sci. **13**(4), 2361–2392 (2020). https://doi.org/10.1137/20M1318006
39. Zhang, F., Wang, J., Wang, W., Xu, C.: Low-tubal-rank plus sparse tensor recovery with prior subspace information. IEEE Trans. Pattern Anal. Mach. Intell. **43**(10), 3492–3507 (2020). https://doi.org/10.1109/TPAMI.2020.2986773
40. Zhang, Z., Aeron, S.: Exact tensor completion using T-SVD. IEEE Trans. Signal Process. **65**(6), 1511–1526 (2017). https://doi.org/10.1109/TSP.2016.2639466
41. Zhou, P., Lu, C., Lin, Z., Zhang, C.: Tensor factorization for low-rank tensor completion. IEEE Trans. Image Process. **27**(3), 1152–1163 (2017). https://doi.org/10.1109/TIP.2017.2762595

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.