SIAM J. IMAGING SCIENCES
Vol. 14, No. 4, pp. 1439–1471

Regularized Kaczmarz Algorithms for Tensor Recovery*

Xuemei Chen† and Jing Qin‡

Abstract. Tensor recovery has recently arisen in a lot of application fields, such as transportation, medical imaging, and remote sensing. Under the assumption that signals possess sparse and/or low-rank structures, many tensor recovery methods have been developed to apply various regularization techniques together with the operator-splitting type of algorithms. Due to the unprecedented growth of data, it becomes increasingly desirable to use streamlined algorithms to achieve real-time computation, such as stochastic optimization algorithms that have recently emerged as an efficient family of methods in machine learning. In this work, we propose a novel algorithmic framework based on the Kaczmarz algorithm for tensor recovery. We provide thorough convergence analysis and its applications from the vector case to the tensor one. Numerical results on a variety of tensor recovery applications, including sparse signal recovery, low-rank tensor recovery, image inpainting, and deconvolution, illustrate the enormous potential of the proposed methods.

Key words. Kaczmarz algorithm, tensor recovery, image inpainting, image deblurring, randomized algorithm

AMS subject classifications. 68Q25, 68R10, 68U05

DOI. 10.1137/21M1398562

1. Introduction.

Tensors are an important tool to represent, analyze, and process high-dimensional data. By generalizing vectors and matrices, tensors can be used to represent a variety of data sets in a versatile and efficient way. Besides the traditional low-dimensional signal processing problems such as image restoration, tensor modeling is able to improve complex data analysis and processing by exploiting the hidden relationships among data components. Recently, tensor recovery has arisen in many application areas, such as transportation systems [49], medical imaging [53], and hyperspectral image restoration [19]. The goal is to reconstruct a tensor-valued signal from its measurements with multiple channels which may be degraded by noise, blur, and so on. For example, many image restoration problems, e.g., image deblurring, can be cast as a tensor recovery problem by treating an image as a special type of tensor [25]. Moreover, tensor modeling typically imposes the assumption of sparsity and low-rank structures on the underlying tensor signal to be reconstructed, which brings the presence of sparsity-promoted regularizations in the objective function. In this work, we focus on third-order tensor recovery models given consistent linear measurements.

Tensor recovery problems usually involve massive data sets which result in large-scale systems of linear equations. As one of the most important iterative algorithms for solving

*Received by the editors February 12, 2021; accepted for publication (in revised form) July 13, 2021; published electronically October 18, 2021.
https://doi.org/10.1137/21M1398562
Funding: The work of the first author was supported by National Science Foundation grant DMS-2050028. The work of the second author was supported by National Science Foundation grant DMS-1941197.

†Department of Mathematics and Statistics, University of North Carolina Wilmington, Wilmington, NC 28409 USA (chenxuemei@uncw.edu).
‡Department of Mathematics, University of Kentucky, Lexington, KY 40506 USA (jing.qin@uky.edu).

linear systems, the Kaczmarz algorithm was first proposed by Stephen Kaczmarz [24] and then rediscovered as the algebraic reconstruction technique in computed tomography [21]. Due to its simplicity and efficiency, it has been used and developed in many applications, including ultrasound imaging [3], seismic imaging [40], positron emission tomography [22], electrical impedance tomography [30], and recently phase retrieval [23, 50].

There is a large amount of work on the interpretations, developments, and extensions of the Kaczmarz algorithm. For example, by treating each linear equation as a hyperplane of a high-dimensional space, it can be derived by applying the method of successive projections onto convex sets [10]. With random selection of projections, a randomized Kaczmarz algorithm for solving consistent overdetermined linear systems with a unique solution is proposed in [48], which can be considered as a special case of the stochastic gradient matching pursuit (StoGradMP) [37]. Convergence analysis of randomized Kaczmarz for noisy and random linear systems can be found in [36, 13]. In addition, application of coordinate descent to the dual formulation of a linear equality constrained least-norm problem leads to the Kaczmarz algorithm [51]. Some other Kaczmarz types of methods include accelerated randomized Kaczmarz [18], asynchronous parallel randomized Kaczmarz [31], block Kaczmarz algorithms [42, 39, 17], Kaczmarz method for fusion frame recovery [14], and greedy randomized Kaczmarz [4]. Similar to the standard version, randomized sparse block Kaczmarz can also be obtained from randomized dual block-coordinate descent [41]. Note that the randomized Kaczmarz algorithm can be considered as a special instance of stochastic gradient descent [37]. The Kaczmarz algorithm has also been incorporated into other iterative methods for solving ill-posed problems; a comprehensive review and convergence rate comparison can be found in [28].

Motivated by the advantages of Kaczmarz-type algorithms in solving linear systems, we intend to extend it from vectors to tensors. In this work, we integrate the Kaczmarz algorithm into sparse/low-rank tensor recovery to significantly reduce the computational cost while preserving high accuracy. To simplify the discussion, we focus on third-order tensors which can be further extended to other higher-order tensors. Specifically, we assume that the acquired tensor measurements $\mathcal{B} \in \mathbb{R}^{N_1 \times K \times N_3}$ are related with the sensing tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ and the underlying signal in a tensor form $\mathcal{X} \in \mathbb{R}^{N_2 \times K \times N_3}$ via the tensor equation $g(\mathcal{X}) = \mathcal{B}$. To recover $\mathcal{X}$, one can consider the following constrained minimization problem:

$$\hat{\mathcal{X}} = \underset{\mathcal{X} \in \mathbb{R}^{N_2 \times K \times N_3}}{\operatorname{argmin}} f(\mathcal{X}), \quad \text{s.t.} \quad g(\mathcal{X}) = \mathcal{B}. \tag{1.1}$$

Here the objective function f is convex on the real tensor space $\mathbb{R}^{N_2 \times K \times N_3}$, and g is a map from $\mathbb{R}^{N_2 \times K \times N_3}$ to $\mathbb{R}^{N_1 \times K \times N_3}$. In this paper, we assume $g(\mathcal{X}) = \mathcal{A} * \mathcal{X}$ where the symbol "$*$" denotes the t-product [26] (see section 2.2). To solve this linear constrained minimization problem, we propose a general regularized Kaczmarz tensor algorithm. The algorithm is "regularized" since the objective function contains regularization terms for preserving desirable characteristics of the underlying solution which is similar to regularization techniques for ill-posed inverse problems. Moreover, the proposed algorithm alternates the Kaczmarz algorithm and subgradient descent, which is thereby different from the classical Kaczmarz algorithm. We discuss convergence guarantees of the proposed algorithm with either a deterministic control sequence or a random sequence. The proposed framework is also adapted to solve the tensor

nuclear norm minimization problem. In addition, convergence guarantees of our algorithm in special cases when the third dimension of tensors is one have been provided for vector and matrix recovery problems. Furthermore, numerical experiments in various applications, including sparse vector recovery, image inpainting, low-rank tensor recovery, and single/multiple image deblurring, demonstrate the great potential of the proposed algorithms in terms of computational efficiency. There are three major contributions for this work detailed as follows.

1. We propose a novel regularized Kaczmarz algorithmic framework for tensor recovery problems with thorough convergence analysis; see Theorem 3.3. Due to the difference in data structures, extension of a Kaczmarz type of algorithm to tensors is not trivial. A linear convergence rate is proven for the randomized version; see Theorem 3.9. Moreover, we consider the noisy scenario with a slightly stronger assumption on the objective function; see Theorem 3.10.
2. We provide three important special cases of the proposed framework with detailed convergence discussions. The first case is for matrix recovery (see Corollary 4.1), which involves minimizing the sum of the (scaled) nuclear norm and the Frobenius norm. The second one is for vector recovery which has been studied extensively as mentioned above. Although Corollary 4.4 is comparable to results from the literature, Corollary 4.5 addresses the noisy case which is the first of its kind to the best of our knowledge. The last one is tensor nuclear minimization, which is particularly useful in a lot of high-dimensional signal processing problems. Convergence guarantees are also provided in Corollary 4.8 and Theorem 4.12.
3. Numerical experiments on various signal/image recovery problems have justified the proposed performance, which can be further extended to solve other related application problems in various areas. In addition, the proposed algorithms are friendly to parameter tuning, where the stepsize can be fixed as one. Batched versions have empirically shown the capability of further improvements.

When reduced to the vector or matrix cases, our work is relevant to [32], which has a more general form of constraints, and [45], which shows linear convergence for the randomized algorithm. Our results are compared with existing research when appropriate. The very recent work by Ma and Molitor [34] also uses the Kaczmarz algorithm for tensor recovery, but it focuses on solving the system $\mathcal{A} * \mathcal{X} = \mathcal{B}$ without minimizing a general objective function.

The rest of the paper is organized as follows. In section 2, we introduce basic concepts and results in convex optimization, and tensors. Our main results are in section 3, where we propose a tensor recovery method based on the Kaczmarz algorithm with the convergence guarantees. For the randomized version of this algorithm, we show linear convergence in expectation, even when in the presence of noise. As special cases of tensor recovery, section 4 discusses how the proposed framework is applied to solve the vector and matrix recovery problem, and the tensor nuclear norm regularized tensor recovery model. Section 5 lists various numerical experiments illustrating the efficiency of the proposed algorithms.

2. Preliminaries.

In this section, we provide clarification of notation and a brief review of fundamental concepts in convex optimization and tensor algebra.

Throughout the paper, we use boldface lowercase letters such as $\mathbf{x}$ for vectors, capital letters such as X for matrices, and calligraphic letters such as $\mathcal{X}$ for tensors. The sets of all

natural numbers and real numbers are denoted by $\mathbb{N}$ and $\mathbb{R}$, respectively. Given a real number $p \geq 1$, the ℓ_p-norm of a vector $\mathbf{x} \in \mathbb{R}^N$ is defined as $\|\mathbf{x}\|_p := (\sum_{i=1}^N |x_i|^p)^{1/p}$. Analogous to the vector ℓ_2-norm, the Frobenius norm of a matrix $X \in \mathbb{R}^{N_1 \times N_2}$ is defined as $\|X\|_F = \sqrt{\sum_{i=1}^{N_1} \sum_{j=1}^{N_2} x_{ij}^2}$. The trace of a square matrix X, denoted by $\mathrm{Tr}(X)$, is the sum of all diagonal entries of X. Furthermore, the nuclear norm of a matrix X, denoted by $\|X\|_*$, is defined as the sum of all the singular values of X. For a complex-valued matrix X, X^T is its transpose by interchanging the row and column index for each entry, and X^* is its complex conjugate transpose, i.e., performing both transpose and componentwise complex conjugate. For any positive integer k, the set $\{1, 2, \ldots, k\}$ is denoted by $[k]$. Given a finite set I, the cardinality of I is denoted by $|I|$.

For a convex set V, P_V denotes the orthogonal projection onto V. Given a matrix A, $R(A)$ is the row space of A, and $\sigma_{\min}(A)$, $\sigma_{\max}(A)$ are the smallest and largest nonzero singular values of A, respectively. One can show that

$$\sigma_{\min}(A)\|P_{R(A)}(\mathbf{x})\|_2 \leq \|A\mathbf{x}\|_2 \leq \sigma_{\max}(A)\|P_{R(A)}(\mathbf{x})\|_2. \tag{2.1}$$

The *soft thresholding* operator (also known as *shrinkage*) $S_\lambda(\cdot)$ is defined componentwise as

$$(S_\lambda(\mathbf{x}))_i = \mathrm{sgn}(x_i)\max\{|x_i| - \lambda, 0\}, \tag{2.2}$$

where $\mathbf{x} \in \mathbb{R}^N$ and $\mathrm{sgn}(\cdot)$ is the signum function which returns the sign of a nonzero number and zero otherwise. This operator can also be extended for matrices and tensors.

2.1. Convex optimization basics. To make the paper self-contained, we present basic definitions and properties about convex functions defined on real vector spaces. By reshaping matrices or tensors as vectors, these concepts and results can be naturally extended to functions defined on real matrix or tensor spaces.

For a continuous function $f : \mathbb{R}^N \to \mathbb{R}$, its *subdifferential* at $\mathbf{x} \in \mathbb{R}^N$ is defined as

$$\partial f(\mathbf{x}) = \left\{\mathbf{x}^* : f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \mathbf{x}^*, \mathbf{y} - \mathbf{x}\rangle \text{ for any } \mathbf{y} \in \mathbb{R}^N\right\}.$$

It can be shown that $\partial f(\mathbf{x})$ is closed and convex in $\mathbb{R}^N$. If, in addition, f is convex, then $\partial f(\mathbf{x})$ is nonempty for any $\mathbf{x} \in \mathbb{R}^N$. In addition, a convex function is called *proper* if its epigraph $\{(\mathbf{x}, \mu) : \mathbf{x} \in \mathbb{R}^N, \mu \geq f(\mathbf{x})\}$ is nonempty in $\mathbb{R}^{N+1}$.

Furthermore, f is α-*strongly convex* for some $\alpha > 0$ if for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$ and $\mathbf{x}^* \in \partial f(\mathbf{x})$ we have

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \mathbf{x}^*, \mathbf{y} - \mathbf{x}\rangle + \frac{\alpha}{2}\|\mathbf{y} - \mathbf{x}\|_2^2. \tag{2.3}$$

Example 2.1. Let $f_1(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2$, which is differentiable. In this case, the subdifferential becomes a singleton only consisting of the gradient, i.e., $\partial f_1(\mathbf{x}) = \{\nabla f_1(\mathbf{x})\} = \{\mathbf{x}\}$, and one can show that f_1 is 1-strongly convex.

Moreover, it is easy to show that $h(\mathbf{x}) + \frac{1}{2}\|\mathbf{x}\|_2^2$ is 1-strongly convex if h is convex. In what follows, we provide two such examples. For more details and examples, please refer to [5].

Example 2.2. Given a positive number λ, $f_\lambda(\mathbf{x}) = \lambda\|\mathbf{x}\|_1 + \frac{1}{2}\|\mathbf{x}\|_2^2$ is 1-strongly convex.

Example 2.3. Let U be a real matrix, $\lambda > 0$, then $f_U(\mathbf{x}) = \lambda\|U\mathbf{x}\|_1 + \frac{1}{2}\|\mathbf{x}\|_2^2$ is 1-strongly convex.

The *convex conjugate function* of f at $\mathbf{z} \in \mathbb{R}^N$ is defined as

$$f^*(\mathbf{z}) := \sup_{\mathbf{x}\in\mathbb{R}^N} \{\langle \mathbf{z}, \mathbf{x}\rangle - f(\mathbf{x})\}.$$

It can be shown that $\mathbf{z} \in \partial f(\mathbf{x})$ if and only if $\mathbf{x} = \nabla f^*(\mathbf{z})$ [43].

Following the ideas in [32], we can verify that if f is α-strongly convex, then the conjugate function f^* is differentiable and for any $\mathbf{z}, \mathbf{w} \in \mathbb{R}^N$, the following inequalities hold:

$$\|\nabla f^*(\mathbf{z}) - \nabla f^*(\mathbf{w})\|_2 \le \frac{1}{\alpha}\|\mathbf{z} - \mathbf{w}\|_2, \tag{2.4}$$

$$f^*(\mathbf{y}) \le f^*(\mathbf{x}) + \langle \nabla f^*(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle + \frac{1}{2\alpha}\|\mathbf{y} - \mathbf{x}\|_2^2. \tag{2.5}$$

For a convex function $f : \mathbb{R}^N \to \mathbb{R}$, the *Bregman distance* between $\mathbf{x}$ and $\mathbf{y}$ with respect to f and $\mathbf{x}^* \in \partial f(\mathbf{x})$ is defined as

$$D_{f,\mathbf{x}^*}(\mathbf{x}, \mathbf{y}) := f(\mathbf{y}) - f(\mathbf{x}) - \langle \mathbf{x}^*, \mathbf{y} - \mathbf{x}\rangle. \tag{2.6}$$

Since $\langle \mathbf{x}, \mathbf{x}^*\rangle = f(\mathbf{x}) + f^*(\mathbf{x}^*)$ if $x^* \in \partial f(\mathbf{x})$ [43], the Bregman distance can also be written as

$$D_{f,\mathbf{x}^*}(\mathbf{x}, \mathbf{y}) = f(\mathbf{y}) + f^*(\mathbf{x}^*) - \langle \mathbf{x}^*, \mathbf{y}\rangle. \tag{2.7}$$

Note that for $f_1(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2$, we have $D_{f_1,\mathbf{x}^*}(\mathbf{x}, \mathbf{y}) = \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2$. In general, if f is α_f-strongly convex, then the Bregman distance satisfies the property

$$\frac{\alpha_f}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 \le D_{f,\mathbf{x}^*}(\mathbf{x}, \mathbf{y}) \le \langle \mathbf{x}^* - \mathbf{y}^*, \mathbf{x} - \mathbf{y}\rangle. \tag{2.8}$$

The *proximal operator* of f is defined as

$$\mathrm{prox}_f(\mathbf{v}) = \operatorname*{argmin}_{\mathbf{x}} \left\{ f(\mathbf{x}) + \frac{1}{2}\|\mathbf{x} - \mathbf{v}\|_2^2 \right\}. \tag{2.9}$$

Due to the Fenchel's duality, we have

$$\mathrm{prox}_h(\mathbf{v}) = \nabla f^*(\mathbf{v}), \quad \text{where} \quad f(\mathbf{x}) = h(\mathbf{x}) + \frac{1}{2}\|\mathbf{x}\|_2^2.$$

It can be shown that the soft thresholding operator is in fact the proximal operator of the ℓ_1-norm, i.e., $S_\lambda(\mathbf{x}) = \mathrm{prox}_{\lambda\|\cdot\|_1}(\mathbf{x})$.

We provide an important lemma related to the Bregman distance, which will be used in proving Theorem 3.10.

Lemma 2.4. *If $f : \mathbb{R}^N \to \mathbb{R}$ is convex, continuous, and proper, then*

$$D_{f,\mathbf{x}^*}(\mathbf{x}, \mathbf{y}) - D_{f,\mathbf{w}^*}(\mathbf{w}, \mathbf{y}) \le \langle \mathbf{x}^* - \mathbf{w}^*, \mathbf{x} - \mathbf{y}\rangle$$

for any $\mathbf{x}^* \in \partial f(\mathbf{x})$ and $\mathbf{w}^* \in \partial f(\mathbf{w})$.

Proof. For any $\mathbf{x}^* \in \partial f(\mathbf{x})$ and $\mathbf{w}^* \in \partial f(\mathbf{w})$, we have

$$\begin{aligned} D_{f,\mathbf{x}^*}(\mathbf{x},\mathbf{y}) - D_{f,\mathbf{w}^*}(\mathbf{w},\mathbf{y}) &= f(\mathbf{y}) - f(\mathbf{x}) - \langle \mathbf{x}^*, \mathbf{y}-\mathbf{x}\rangle - f(\mathbf{y}) + f(\mathbf{w}) + \langle \mathbf{w}^*, \mathbf{y}-\mathbf{w}\rangle \\ &= f(\mathbf{w}) - f(\mathbf{x}) - \langle \mathbf{x}^*, \mathbf{y}-\mathbf{x}\rangle + \langle \mathbf{w}^*, \mathbf{y}-\mathbf{w}\rangle \\ &\le \langle \mathbf{w}^*, \mathbf{w}-\mathbf{x}\rangle - \langle \mathbf{x}^*, \mathbf{y}-\mathbf{x}\rangle + \langle \mathbf{w}^*, \mathbf{y}-\mathbf{w}\rangle \\ &= \langle \mathbf{w}^*, \mathbf{y}-\mathbf{x}\rangle - \langle \mathbf{x}^*, \mathbf{y}-\mathbf{x}\rangle = \langle \mathbf{x}^*-\mathbf{w}^*, \mathbf{x}-\mathbf{y}\rangle. \end{aligned}$$

■

Next we will introduce the concept of restricted strong convexity. If $f : \mathbb{R}^N \to \mathbb{R}$ is convex, differentiable, and proper, then (2.3) in the definition of an α-strongly convex function becomes

$$\langle \nabla f(\mathbf{y}) - \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle \ge \alpha \|\mathbf{y}-\mathbf{x}\|_2^2 \tag{2.10}$$

for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$. We define a weaker version as follows.

Definition 2.5. *Let $f : \mathbb{R}^N \to \mathbb{R}$ be convex differentiable with a nonempty minimizer set $X_f := \{\mathbf{x} : f(\mathbf{x}) \le f(\mathbf{y})$ for any $\mathbf{y} \in \mathbb{R}^N\}$. The function f is* restricted strongly convex *on a convex set $C \subseteq \mathbb{R}^N$ with $\alpha > 0$ if*

$$\langle \nabla f(\mathbf{y}) - \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle \ge \alpha \|\mathbf{y}-\mathbf{x}\|_2^2 \tag{2.11}$$

for any $\mathbf{x} \in C$ and $\mathbf{y} = P_{X_f \cap C}(\mathbf{x})$.

This definition can be found in [44]. The concept of restricted strong convexity first appeared in [29], where C is $\mathbb{R}^N$. We include below a useful lemma about the restricted strong convexity, which will be used for proving Lemma 3.6.

Lemma 2.6 ([44, Lemma 2.2]). *If f is restricted strongly convex on C with the constant α, then*

$$f(\mathbf{x}) - \min_{\mathbf{x}} f(\mathbf{x}) \le \frac{1}{\alpha}\|\nabla f(\mathbf{x})\|_2^2 \tag{2.12}$$

for any $\mathbf{x} \in C$.

2.2. Tensor basics. We follow the notation for tensor operators in [25]. If $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ with the kth frontal slice $A_k = \mathcal{A}(:,:,k)$, then we define the *block circulant operator* as follows:

$$\text{bcirc}(\mathcal{A}) := \begin{bmatrix} A_1 & A_{N_3} & \cdots & A_2 \\ A_2 & A_1 & \cdots & A_3 \\ \vdots & \vdots & \ddots & \vdots \\ A_{N_3} & A_{N_3-1} & \cdots & A_1 \end{bmatrix} \in \mathbb{R}^{N_1 N_3 \times N_2 N_3}. \tag{2.13}$$

Moreover, we define the operator unfold$(\cdot)$ and its inversion fold$(\cdot)$ for the conversion between tensors and matrices,

$$\text{unfold}(\mathcal{A}) = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_{N_3} \end{bmatrix} \in \mathbb{R}^{N_1 N_3 \times N_2}, \quad \text{fold}\left(\begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_{N_3} \end{bmatrix}\right) = \mathcal{A}. \tag{2.14}$$

For degenerate cases, we define the operator squeeze$(\cdot)$ that removes all the dimensions of length one from a tensor. For example, if $\mathcal{A} \in \mathbb{R}^{3\times 1\times 1}$, then squeeze$(\mathcal{A})$ returns a three-dimensional (3D) vector. The transpose of $\mathcal{A}$, denoted by $\mathcal{A}^T$, is the $N_2 \times N_1 \times N_3$ tensor obtained by transposing each of the frontal slices and then reversing the order of transposed frontal slices 2 through N_3. We have

$$\text{bcirc}(\mathcal{A}^T) = (\text{bcirc}(\mathcal{A}))^T .$$

For the two tensors $\mathcal{A}, \tilde{\mathcal{A}}$ of the same size, we define the inner product and the Frobenius norm for tensors as

$$\langle \mathcal{A}, \tilde{\mathcal{A}} \rangle := \sum_{i,j,k} \mathcal{A}(i,j,k)\tilde{\mathcal{A}}(i,j,k), \qquad \|\mathcal{A}\|_F^2 = \langle \mathcal{A}, \mathcal{A} \rangle.$$

One can see that $\langle \mathcal{A}, \tilde{\mathcal{A}} \rangle = \langle \text{unfold}(\mathcal{A}), \text{unfold}(\tilde{\mathcal{A}}) \rangle$ where the right-hand side is an inner product of two matrices. If $\mathcal{A} \in \mathbb{R}^{N_1\times N_2\times N_3}, \mathcal{C} \in \mathbb{R}^{N_2\times K\times N_3}$, then their *t-product* is the $N_1 \times K \times N_3$ tensor given by [27]

$$\mathcal{A} * \mathcal{C} := \text{fold}\left(\text{bcirc}(\mathcal{A})\,\text{unfold}(\mathcal{C})\right). \tag{2.15}$$

We list three important properties about the t-product as follows:

(a) Separability in the first dimension

$$(\mathcal{A} * \mathcal{C})(i,:,:) = \mathcal{A}(i,:,:) * \mathcal{C}. \tag{2.16}$$

(b) Sum separability in the second dimension

$$\mathcal{A} * \mathcal{C} = \sum_{j=1}^{N_2} \mathcal{A}(:,j,:) * \mathcal{C}(j,:,:). \tag{2.17}$$

(c) Circular convolution in the third dimension: if $\mathcal{A}, \mathcal{C} \in \mathbb{R}^{1\times 1\times N_3}$, then

$$\text{squeeze}(\mathcal{A} * \mathcal{C}) = \text{circ}(\mathbf{a})\mathbf{c}, \tag{2.18}$$

where both $\mathbf{a} = \text{squeeze}(\mathcal{A})$ and $\mathbf{c} = \text{squeeze}(\mathcal{C})$ are N_3-dimensional vectors. Here circ$(\mathbf{a})$ is the circular matrix generated by the vector $\mathbf{a}$, i.e., the reduced case of (2.13) when $\mathcal{A} \in \mathbb{R}^{1\times 1\times N_3}$.

Here $\mathcal{A}(i,:,:)$ is called the ith horizontal slice of $\mathcal{A}$. For notational convenience, $\mathcal{A}(i,:,:)$ will be denoted as $\mathcal{A}(i)$.

In Table 1, we summarize the t-products of tensors with reduced dimensions and their corresponding operators for vectors or matrices. In particular, based on (2.16) and (2.18), the t-product of $\mathcal{A} \in \mathbb{R}^{1\times 1\times N_3}$ and $\mathcal{C} \in \mathbb{R}^{1\times K\times N_3}$ can be implemented as

$$\text{squeeze}(\mathcal{A} * \mathcal{C}) = \text{squeeze}(\mathcal{C})\text{circ}(\text{squeeze}(\mathcal{A}))^T, \tag{2.19}$$

where squeeze$(\mathcal{A}) \in \mathbb{R}^{N_3}$ and squeeze$(\mathcal{C}) \in \mathbb{R}^{K\times N_3}$. If $\mathcal{A} \in \mathbb{R}^{N_1\times 1\times N_3}$ and $\mathcal{C} \in \mathbb{R}^{1\times 1\times N_3}$, then

$$\text{squeeze}(\mathcal{A} * \mathcal{C}) = \text{squeeze}(\mathcal{A})\text{circ}(\text{squeeze}(\mathcal{C}))^T, \tag{2.20}$$

Table 1
T-products of dimension-reduced tensors.

Condition	$\mathcal{A}$	$\mathcal{C}$	Vector/matrix operator
$N_2 = K = N_3 = 1$	$\mathbb{R}^{N_1\times 1\times 1}$	$\mathbb{R}^{1\times 1\times 1}$	scalar multiplication of a vector
$N_1 = N_2 = N_3 = 1$	$\mathbb{R}^{1\times 1\times 1}$	$\mathbb{R}^{1\times K\times 1}$	scalar multiplication of a vector
$N_1 = K = N_3 = 1$	$\mathbb{R}^{1\times N_2\times 1}$	$\mathbb{R}^{N_2\times 1\times 1}$	inner product of two vectors
$N_2 = N_3 = 1$	$\mathbb{R}^{N_1\times 1\times 1}$	$\mathbb{R}^{1\times K\times 1}$	outer product of two vectors
$N_1 = N_3 = 1$	$\mathbb{R}^{1\times N_2\times 1}$	$\mathbb{R}^{N_2\times K\times 1}$	vector-matrix multiplication
$K = N_3 = 1$	$\mathbb{R}^{N_1\times N_2\times 1}$	$\mathbb{R}^{N_2\times 1\times 1}$	matrix-vector multiplication (section 4.2)
$N_3 = 1$	$\mathbb{R}^{N_1\times N_2\times 1}$	$\mathbb{R}^{N_2\times K\times 1}$	matrix-matrix multiplication (section 4.1)
$N_1 = N_2 = K = 1$	$\mathbb{R}^{1\times 1\times N_3}$	$\mathbb{R}^{1\times 1\times N_3}$	vector circular convolution (2.18)
$N_1 = N_2 = 1$	$\mathbb{R}^{1\times 1\times N_3}$	$\mathbb{R}^{1\times K\times N_3}$	vector-matrix circular convolution (2.19)
$N_2 = K = 1$	$\mathbb{R}^{N_1\times 1\times N_3}$	$\mathbb{R}^{1\times 1\times N_3}$	matrix-vector circular convolution (2.20)
$N_1 = K = 1$	$\mathbb{R}^{1\times N_2\times N_3}$	$\mathbb{R}^{N_2\times 1\times N_3}$	sum of vector circular convolutions (2.21)

where $\text{squeeze}(\mathcal{A}) \in \mathbb{R}^{N_1\times N_3}$ and $\text{squeeze}(\mathcal{C}) \in \mathbb{R}^{N_3}$. Furthermore, if $\mathcal{A} \in \mathbb{R}^{1\times N_2\times N_3}$ and $\mathcal{C} \in \mathbb{R}^{N_2\times 1\times N_3}$, then the t-product becomes a sum of vector circular convolutions

$$\text{squeeze}(\mathcal{A} * \mathcal{C}) = \sum_{j=1}^{N_2} \text{circ}(\text{squeeze}(\mathcal{A})(j,:))\,\text{squeeze}(\mathcal{C})(j,:)^T, \tag{2.21}$$

where $\text{squeeze}(\mathcal{A})(j,:) \in \mathbb{R}^{N_3}$ is the jth row of the matrix $\text{squeeze}(\mathcal{A})$. These properties are particularly useful for representing the image blurring as a t-product; see section 5.4 for more details.

Based on the t-product, some concepts in the matrix case can be directly extended to the tensor case [26], e.g., tensor singular value decomposition (t-SVD) and tubal rank that is the number of nonsingular tubes in the t-SVD form.

Lemma 2.7. *If $\mathcal{A} \in \mathbb{R}^{N_1\times N_2\times N_3}, \mathcal{B} \in \mathbb{R}^{N_1\times K\times N_3}$, and $\mathcal{C} \in \mathbb{R}^{N_2\times K\times N_3}$, then*

$$\langle \mathcal{A} * \mathcal{C}, \mathcal{B}\rangle = \langle \mathcal{C}, \mathcal{A}^T * \mathcal{B}\rangle. \tag{2.22}$$

Proof.

$$\begin{aligned}\langle \mathcal{A} * \mathcal{C}, \mathcal{B}\rangle &= \langle \text{fold}(\text{bcirc}(\mathcal{A})\,\text{unfold}(\mathcal{C})), \mathcal{B}\rangle = \langle \text{bcirc}(\mathcal{A})\,\text{unfold}(\mathcal{C}), \text{unfold}(\mathcal{B})\rangle \\ &= \text{Tr}\left(\left(\text{bcirc}(\mathcal{A})\,\text{unfold}(\mathcal{C})\right)^T \text{unfold}(\mathcal{B})\right) = \text{Tr}\left(\left(\text{unfold}(\mathcal{C})\right)^T \text{bcirc}(\mathcal{A}^T)\,\text{unfold}(\mathcal{B})\right) \\ &= \langle \text{unfold}(\mathcal{C}), \text{bcirc}(\mathcal{A}^T)\,\text{unfold}(\mathcal{B})\rangle \\ &= \langle \mathcal{C}, \mathcal{A}^T * \mathcal{B}\rangle. \end{aligned}$$

Lemma 2.8. *If $\mathcal{A} \in \mathbb{R}^{N_1\times N_2\times N_3}$ and $\mathcal{X} \in \mathbb{R}^{N_2\times K\times N_3}$, then we have*

$$\|\mathcal{A} * \mathcal{X}\|_F \le \sqrt{N_3}\|\mathcal{A}\|_F\|\mathcal{X}\|_F. \tag{2.23}$$

Proof. $\|\mathcal{A} * \mathcal{X}\|_F^2 = \|\,\text{bcirc}(\mathcal{A})\,\text{unfold}(\mathcal{X})\|_F^2 \le \|\,\text{bcirc}(\mathcal{A})\|_F^2\|\mathcal{X}\|_F^2 = N_3\|\mathcal{A}\|_F^2\|\mathcal{X}\|_F^2$.

3. Tensor recovery. Let f be an α_f-strongly convex function defined on $\mathbb{R}^{N_2\times K\times N_3}$, $\mathcal{A}\in\mathbb{R}^{N_1\times N_2\times N_3}$, and $\mathcal{B}\in\mathbb{R}^{N_1\times K\times N_3}$. Consider a tensor recovery problem of the following form:

$$\hat{\mathcal{X}} = \underset{\mathcal{X}\in\mathbb{R}^{N_2\times K\times N_3}}{\operatorname{argmin}} f(\mathcal{X}) \quad \text{s.t.} \quad \mathcal{A}*\mathcal{X}=\mathcal{B}. \tag{3.1}$$

Using (2.15), the constraint $\mathcal{A}*\mathcal{X}=\mathcal{B}$ can be rewritten as the form

$$\operatorname{bcirc}(\mathcal{A})\operatorname{unfold}(\mathcal{X}) = \operatorname{unfold}(\mathcal{B}), \tag{3.2}$$

and we assume the linear system (3.2) is consistent and underdetermined. Recall that $\mathcal{A}(i)=\mathcal{A}(i,:,:)$. According to the separability of t-product in the first dimension (2.16), the linear constraint $\mathcal{A}*\mathcal{X}=\mathcal{B}$ can be split into N_1 reduced ones,

$$\mathcal{A}(i)*\mathcal{X}=\mathcal{B}(i), \quad i\in[N_1]. \tag{3.3}$$

One can see that

$$\sum_{i=1}^{N_1}\|\mathcal{A}(i)*\mathcal{X}\|_F^2 = \|\mathcal{A}*\mathcal{X}\|_F^2. \tag{3.4}$$

We further let $H_i:=\{\mathcal{X}:\mathcal{A}(i)*\mathcal{X}=\mathcal{B}(i)\}$, and let $H=\bigcap_{i=1}^{N_1}H_i$ be the feasible set of (3.1). The "row space" of $\mathcal{A}$ is defined as

$$R(\mathcal{A}) := \left\{\mathcal{A}^T*\mathcal{Y}:\mathcal{Y}\in\mathbb{R}^{N_1\times K\times N_3}\right\}.$$

If $N_3=1$, then $R(\mathcal{A})$ coincides with the row space of the $N_1\times N_2$ matrix $\mathcal{A}$.

The optimal solution $\hat{\mathcal{X}}$ of (3.1) satisfies the following optimality conditions:

$$\mathcal{A}*\hat{\mathcal{X}}=\mathcal{B}, \quad \partial f(\hat{\mathcal{X}})\cap R(\mathcal{A})\neq\emptyset. \tag{3.5}$$

We propose Algorithm 3.1, which only uses one of the horizontal slices of $\mathcal{A}$ at each iteration. In order to make sure all horizontal slices are used, we define a control sequence for slice selection at each iteration.

Definition 3.1. *A sequence $i:\mathbb{N}\to[M]$ is called a* control sequence *for $[M]$ if for any $m\in[M]$, there are infinitely many k's such that $i(k)=m$.*

In Algorithm 3.1, it can be shown that $\mathcal{Z}^{(k)}\in\partial f(\mathcal{X}^{(k)})\cap R(\mathcal{A})$ which will be used in our convergence analysis. Regarding the control sequence, one common choice is the cyclic sequence $\{1,2,\ldots,N_1,1,2,\ldots,N_1,\ldots\}$, which cycles sequentially through the constraints (3.3). Algorithm 3.1 can be viewed as a deterministic algorithm, whereas Algorithm 3.2 picks the slices in a random fashion and has gained much attention in recent years [48, 38, 54, 35]. At each iteration, the probability of picking the jth constraint/slice (3.3) is proportional to $\|\mathcal{A}(j)\|_F^2$, which is commonly used in the literature [48]. Nevertheless, other probability distributions are also allowed; see Corollary 4.1 and Remark 4.2. More discussion on picking appropriate probability distributions can be found in [16, 2, 14].

Algorithm 3.1 Regularized Kaczmarz algorithm for tensor recovery.

Input: $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, $\mathcal{B} \in \mathbb{R}^{N_1 \times K \times N_3}$, control sequence $\{i(k)\}_{k=0}^{\infty} \subseteq [N_1]$, stepsize t, maximum number of iterations T, and the tolerance tol.
Output: an approximation of $\hat{\mathcal{X}}$
Initialize: $\mathcal{Z}^{(0)} \in R(\mathcal{A}) \subset \mathbb{R}^{N_2 \times K \times N_3}$, $\mathcal{X}^{(0)} = \nabla f^*(\mathcal{Z}^{(0)})$.
for $k = 0, 1, \ldots, T-1$ **do**
$\mathcal{Z}^{(k+1)} = \mathcal{Z}^{(k)} + t\mathcal{A}(i(k))^T * \frac{\mathcal{B}(i(k)) - \mathcal{A}(i(k)) * \mathcal{X}^{(k)}}{\|\mathcal{A}(i(k))\|_F^2}$
$\mathcal{X}^{(k+1)} = \nabla f^*(\mathcal{Z}^{(k+1)})$
Terminate if $\|\mathcal{X}^{(k+1)} - \mathcal{X}^{(k)}\|_F / \|\mathcal{X}^{(k)}\|_F < tol$.
end for

Algorithm 3.2 Randomized regularized Kaczmarz algorithm for tensor recovery.

Input: $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, $\mathcal{B} \in \mathbb{R}^{N_1 \times K \times N_3}$, stepsize t, maximum number of iterations T, and the tolerance tol.
Output: an approximation of $\hat{\mathcal{X}}$
Initialize: $\mathcal{Z}^{(0)} \in R(\mathcal{A}) \subset \mathbb{R}^{N_2 \times K \times N_3}$, $\mathcal{X}^{(0)} = \nabla f^*(\mathcal{Z}^{(0)})$.
for $k = 0, 1, \ldots, T-1$ **do**
pick $i(k)$ randomly from $[N_1]$ with $\Pr(i(k) = j) = \|\mathcal{A}(j)\|_F^2 / \|\mathcal{A}\|_F^2$,
$\mathcal{Z}^{(k+1)} = \mathcal{Z}^{(k)} + t\mathcal{A}(i(k))^T * \frac{\mathcal{B}(i(k)) - \mathcal{A}(i(k)) * \mathcal{X}^{(k)}}{\|\mathcal{A}(i(k))\|_F^2}$
$\mathcal{X}^{(k+1)} = \nabla f^*(\mathcal{Z}^{(k+1)})$
Terminate if $\|\mathcal{X}^{(k+1)} - \mathcal{X}^{(k)}\|_F / \|\mathcal{X}^{(k)}\|_F < tol$.
end for

3.1. Convergence analysis with a control sequence.

To analyze the convergence of the proposed Algorithms 3.1 and 3.2, we state the following proposition involving a dimension-reduced t-product in Table 1 which plays an important role in the convergence analysis.

Proposition 3.2. *Suppose $\mathcal{A} \in \mathbb{R}^{1 \times N_2 \times N_3}$, $\mathcal{B} \in \mathbb{R}^{1 \times K \times N_3}$, and f is an α_f-strongly convex function defined on $\mathbb{R}^{N_2 \times K \times N_3}$. Given an arbitrary $\bar{\mathcal{Z}} \in \mathbb{R}^{N_2 \times K \times N_3}$ and $\bar{\mathcal{X}} = \nabla f^*(\bar{\mathcal{Z}})$, let $\mathcal{Z} = \bar{\mathcal{Z}} + t\frac{\mathcal{A}^T}{\|\mathcal{A}\|_F^2} * (\mathcal{B} - \mathcal{A} * \bar{\mathcal{X}})$ and $\mathcal{X} = \nabla f^*(\mathcal{Z})$; we have*

$$D_{f,\mathcal{Z}}(\mathcal{X}, \mathcal{H}) \le D_{f,\bar{\mathcal{Z}}}(\bar{\mathcal{X}}, \mathcal{H}) - \frac{t}{\|\mathcal{A}\|_F^2}\left(1 - \frac{tN_3}{2\alpha_f}\right)\|\mathcal{B} - \mathcal{A} * \bar{\mathcal{X}}\|_F^2$$

for any $\mathcal{H}$ that satisfies $\mathcal{A} * \mathcal{H} = \mathcal{B}$.

Proof. For simplicity of notation, let $\mathcal{Z} = \bar{\mathcal{Z}} + s\mathcal{W}$, where $\mathcal{W} = \mathcal{A}^T * (\mathcal{B} - \mathcal{A} * \bar{\mathcal{X}})$, and $s = \frac{t}{\|\mathcal{A}\|_F^2}$. By Lemma 2.8, we have $\|\mathcal{W}\|_F \le \sqrt{N_3}\|\mathcal{A}\|_F\|\mathcal{B} - \mathcal{A} * \bar{\mathcal{X}}\|_F$. By Lemma 2.7, we have

$$\langle \mathcal{W}, \mathcal{H} - \bar{\mathcal{X}} \rangle = \langle \mathcal{A}^T * (\mathcal{A} * \mathcal{H} - \mathcal{A} * \bar{\mathcal{X}}), \mathcal{H} - \bar{\mathcal{X}} \rangle = \|\mathcal{A} * (\mathcal{H} - \bar{\mathcal{X}})\|_F^2 = \|\mathcal{B} - \mathcal{A} * \bar{\mathcal{X}}\|_F^2.$$

By (2.7), the Bregman distance between $\mathcal{X}$ and $\mathcal{H}$ with respect to f and $\mathcal{Z}$ satisfies

$$
\begin{aligned}
D_{f,\mathcal{Z}}(\mathcal{X},\mathcal{H}) &= f^*(\mathcal{Z}) - \langle \mathcal{Z},\mathcal{H}\rangle + f(\mathcal{H}) \\
&= f^*(\bar{\mathcal{Z}} + s\mathcal{W}) - \langle \bar{\mathcal{Z}} + s\mathcal{W},\mathcal{H}\rangle + f(\mathcal{H}) \\
&\le f^*(\bar{\mathcal{Z}}) + \langle \nabla f^*(\bar{\mathcal{Z}}), s\mathcal{W}\rangle + \frac{1}{2\alpha_f}\|s\mathcal{W}\|_F^2 - \langle \bar{\mathcal{Z}} + s\mathcal{W},\mathcal{H}\rangle + f(\mathcal{H}) \\
&= D_{f,\bar{\mathcal{Z}}}(\bar{\mathcal{X}},\mathcal{H}) - \langle s\mathcal{W},\mathcal{H}\rangle + \langle \bar{\mathcal{X}}, s\mathcal{W}\rangle + \frac{1}{2\alpha_f}\|s\mathcal{W}\|_F^2 \\
&= D_{f,\bar{\mathcal{Z}}}(\bar{\mathcal{X}},\mathcal{H}) - \langle s\mathcal{W},\mathcal{H} - \bar{\mathcal{X}}\rangle + \frac{1}{2\alpha_f}\|s\mathcal{W}\|_F^2 \\
&\le D_{f,\bar{\mathcal{Z}}}(\bar{\mathcal{X}},\mathcal{H}) - s\|\mathcal{B} - \mathcal{A} * \bar{\mathcal{X}}\|_F^2 + \frac{s^2}{2\alpha_f} N_3 \|\mathcal{A}\|_F^2 \|\mathcal{B} - \mathcal{A} * \bar{\mathcal{X}}\|_F^2 \\
&= D_{f,\bar{\mathcal{Z}}}(\bar{\mathcal{X}},\mathcal{H}) - s\|\mathcal{B} - \mathcal{A} * \bar{\mathcal{X}}\|_F^2 \left(1 - \frac{sN_3\|\mathcal{A}\|_F^2}{2\alpha_f}\right).
\end{aligned}
$$

The desired result is obtained since $s = \frac{t}{\|\mathcal{A}\|_F^2}$. ∎

This proposition essentially shows how much the Bregman distance decreases after one iteration. We are now ready to state our first main result.

Theorem 3.3. *Let f be α_f-strongly convex. If $t < 2\alpha_f/N_3$, then the sequence $\{\mathcal{X}^{(k)}\}$ generated by Algorithm 3.1 satisfies*

$$
D_{f,\mathcal{Z}^{(k+1)}}(\mathcal{X}^{(k+1)},\mathcal{X}) \le D_{f,\mathcal{Z}^{(k)}}(\mathcal{X}^{(k)},\mathcal{X}) - t\left(1 - \frac{tN_3}{2\alpha_f}\right)\frac{\|\mathcal{A}(i(k)) * (\mathcal{X}^{(k)} - \mathcal{X})\|_F^2}{\|\mathcal{A}(i(k))\|_F^2} \tag{3.6}
$$

for all $\mathcal{X} \in H_{i(k)}$. Moreover, the sequence $\{\mathcal{X}^{(k)}\}$ converges to the solution of (3.1).

Proof. We apply Proposition 3.2 where $\mathcal{A}(i(k))$, $\mathcal{B}(i(k))$, $\mathcal{Z}^{(k)}$, $\mathcal{Z}^{(k+1)}$ are replaced by $\mathcal{A}$, $\mathcal{B}$, $\bar{\mathcal{Z}}$, $\mathcal{Z}$, respectively, and then obtain (3.6). The rest of the proof is for the convergence of the sequence $\{\mathcal{X}^{(k)}\}$.

It is known that $\lim_{\|\mathcal{Z}\|_F\to\infty} \frac{f^*(\mathcal{Z})}{\|\mathcal{Z}\|_F} = \infty$ (see [32, equation (5)]). Then for any $\mathcal{Z} \in \partial f(\mathcal{X})$ and arbitrary $\mathcal{Y}$, we have

$$
\lim_{\|\mathcal{Z}\|_F\to\infty} \frac{D_{f,\mathcal{Z}}(\mathcal{X},\mathcal{Y})}{\|\mathcal{Z}\|_F} = \lim_{\|\mathcal{Z}\|_F\to\infty} \frac{f^*(\mathcal{Z}) - \langle \mathcal{Z},\mathcal{Y}\rangle + f(\mathcal{Y})}{\|\mathcal{Z}\|_F} \ge \lim_{\|\mathcal{Z}\|_F\to\infty} \frac{f^*(\mathcal{Z}) - \|\mathcal{Z}\|_F\|\mathcal{Y}\|_F}{\|\mathcal{Z}\|_F} = \infty.
$$

By (3.6), the sequence $\{D_{f,\mathcal{Z}^{(k)}}(\mathcal{X}^{(k)},\hat{\mathcal{X}})\}$ is decreasing, hence bounded and convergent. The above limit implies that $\|\mathcal{Z}^{(k)}\|_F$ must be bounded. So for a subsequence $\{k_l\}$, we have $\lim_{l\to\infty} \mathcal{Z}^{(k_l)} = \tilde{\mathcal{Z}}$. Thus, we get

$$
\lim_{l\to\infty} \mathcal{X}^{(k_l)} = \lim_{l\to\infty} \nabla f^*\left(\mathcal{Z}^{(k_l)}\right) = \nabla f^*(\tilde{\mathcal{Z}}) := \tilde{\mathcal{X}}.
$$

We denote the constant $t(1 - \frac{tN_3}{2\alpha_f})$ by r. Since $\{i(k)\}$ is a control sequence for $[N_1]$, there exists $j_0 \in [N_1]$ such that $\{i(k_l)\}$ has infinitely many terms of j_0. Without loss of generality, the subsequence can be picked such that

$$
\forall l, i(k_l) \equiv j_0, \text{ and } \{i(k_l), i(k_l+1), \ldots, i(k_{l+1}-1)\} = [N_1].
$$

Therefore, we have

$$\begin{aligned} D_{f,\mathcal{Z}^{(k_{l+1})}}(\mathcal{X}^{(k_{l+1})},\hat{\mathcal{X}}) &\le D_{f,\mathcal{Z}^{(k_l+1)}}(\mathcal{X}^{(k_l+1)},\hat{\mathcal{X}}) \\ &\le D_{f,\mathcal{Z}^{(k_l)}}(\mathcal{X}^{(k_l)},\hat{\mathcal{X}}) - r\frac{\|\mathcal{A}(i(k_l))*(\mathcal{X}^{(k_l)}-\hat{\mathcal{X}})\|_F^2}{\|\mathcal{A}(i(k_l))\|_F^2}. \end{aligned} \tag{3.7}$$

By letting $l\to\infty$, we can get

$$D_{f,\tilde{\mathcal{Z}}}(\tilde{\mathcal{X}},\hat{\mathcal{X}}) \le D_{f,\tilde{\mathcal{Z}}}(\tilde{\mathcal{X}},\hat{\mathcal{X}}) - r\frac{\|\mathcal{A}(j_0)*(\tilde{\mathcal{X}}-\hat{\mathcal{X}})\|_F^2}{\|\mathcal{A}(j_0)\|_F^2},$$

which implies that $\mathcal{A}(j_0)*(\tilde{\mathcal{X}}-\hat{\mathcal{X}})=0$ and thereby $\tilde{\mathcal{X}}\in H_{j_0}$.

Let $J_{in}:=\{j:\tilde{\mathcal{X}}\in H_j\}$. Obviously, $j_0\in J_{in}$ by the previous analysis. Next we use the proof by contradiction to further show $J_{in}=[N_1]$. Suppose that $J_{out}=[N_1]\backslash J_{in}\neq\emptyset$. For each l, we define n_l to be the index in $\{k_l,k_l+1,\dots,k_{l+1}-1\}$ such that $\{i(k_l),i(k_l+1),\dots,i(n_l-1)\}\subset J_{in}$ and $i(n_l)\in J_{out}$.

Since $\tilde{\mathcal{X}}\in H_j$ for any $j\in\{i(k_l),i(k_l+1),\cdots,i(n_l-1)\}$, we have, by (3.6),

$$D_{f,\mathcal{Z}^{(n_l)}}(\mathcal{X}^{(n_l)},\tilde{\mathcal{X}}) \le D_{f,\mathcal{Z}^{(n_l-1)}}(\mathcal{X}^{(n_l-1)},\tilde{\mathcal{X}}) \le \cdots \le D_{f,\mathcal{Z}^{(k_l)}}(\mathcal{X}^{(k_l)},\tilde{\mathcal{X}}).$$

By letting $l\to\infty$, we have $\lim_{l\to\infty}\mathcal{X}^{(n_l)}=\tilde{\mathcal{X}}$.

By choosing a subsequence, we can assume $i(n_l)\equiv j_1\in J_{out}$. By the same analysis as in (3.7), we can get $\tilde{\mathcal{X}}\in H_{j_1}$ which is a contradiction. Therefore, we must have $\tilde{\mathcal{X}}\in H=\bigcap_{i=1}^{N_1}H_i$.

By (3.6), $D_{f,\mathcal{Z}^{(k)}}(\mathcal{X}^{(k)},\tilde{\mathcal{X}})$ is decreasing. Since its subsequence $D_{f,\mathcal{Z}^{(k_l)}}(\mathcal{X}^{(k_l)},\tilde{\mathcal{X}})\to 0$, the entire sequence $D_{f,\mathcal{Z}^{(k)}}(\mathcal{X}^{(k)},\tilde{\mathcal{X}})$ converges to zero as well. This shows $\mathcal{X}^{(k)}\to\tilde{\mathcal{X}}$ due to (2.8).

Now we have $\tilde{\mathcal{Z}}=\lim_{k\to\infty}\mathcal{Z}^{(k)}\in R(\mathcal{A})$. Since $\mathcal{Z}^{(k)}\in\partial f(\mathcal{X}^{(k)})$, we have $\tilde{\mathcal{Z}}\in\partial f(\tilde{\mathcal{X}})$. So $\mathcal{A}*\tilde{\mathcal{X}}=\mathcal{B}$ and $\tilde{\mathcal{Z}}\in\partial f(\tilde{\mathcal{X}})\cup R(\mathcal{A})$ fulfill the optimality condition (3.5). Since the solution of (3.1) is unique, we have $\hat{\mathcal{X}}=\tilde{\mathcal{X}}$. ∎

Remark 3.4. We would like to compare Theorem 3.3 with [32, Theorem 2.7]. Note that [32, Theorem 2.7] considers the split feasibility problem, that is, finding vectors belonging to the set $\{\mathbf{x}:A_i\mathbf{x}\in Q_i\}$, where A_i are matrices and Q_i are convex sets. Our theorem focuses on a specific minimization problem that recovers tensors.

3.2. Convergence analysis with a random sequence. For the randomized Algorithm 3.2, Theorem 3.3 no longer applies. We will prove a linear convergence rate in expectation if the objective function f satisfies certain conditions.

The dual problem of (3.1) is the unconstrained problem

$$\min_{\mathcal{Y}\in\mathbb{R}^{N_1\times K\times N_3}} g_f(\mathcal{Y}),$$

where

$$g_f(\mathcal{Y}) = f^*(\mathcal{A}^T*\mathcal{Y}) - \langle\mathcal{Y},\mathcal{B}\rangle. \tag{3.8}$$

Definition 3.5. *Let $m := \min_{\mathcal{Y}} g_f(\mathcal{Y})$. We will call a function f* admissible *if g_f, as defined in (3.8), is restricted strongly convex on any of g_f's level set $L_\delta^{g_f} := \{\mathcal{Y} : g_f(\mathcal{Y}) \le m+\delta\}$. We call f* strongly admissible *if g_f is restricted strongly convex on $\mathbb{R}^{N_1\times K\times N_3}$.*

The following lemma is a consequence of Definition 2.5 and Lemma 2.6.

Lemma 3.6. *Let $\hat{\mathcal{X}}$ be the solution of (3.1).*

(a) *Assume f is admissible. For $\tilde{\mathcal{X}}$ and $\tilde{\mathcal{Z}} \in \partial f(\tilde{\mathcal{X}}) \cap R(\mathcal{A})$, there exists $\nu > 0$ such that for all $\mathcal{X}$ and $\mathcal{Z} \in \partial f(\mathcal{X}) \cap R(\mathcal{A})$ with $D_{f,\mathcal{Z}}(\mathcal{X}, \hat{\mathcal{X}}) \le D_{f,\tilde{\mathcal{Z}}}(\tilde{\mathcal{X}}, \hat{\mathcal{X}})$ it holds that*

$$D_{f,\mathcal{Z}}(\mathcal{X}, \hat{\mathcal{X}}) \le \frac{1}{\nu}\|\mathcal{A} * (\mathcal{X} - \hat{\mathcal{X}})\|_F^2. \tag{3.9}$$

(b) *If f is strongly admissible, there exists $\nu > 0$ such that (3.9) holds for all $\mathcal{X}$ and $\mathcal{Z} \in \partial f(\mathcal{X}) \cap R(\mathcal{A})$.*

Proof. (a) Due to the strong duality, we have that $f(\hat{\mathcal{X}}) = -m = -\min_{\mathcal{Y}} g_f(\mathcal{Y})$. Since $\mathcal{Z} \in R(\mathcal{A})$, we let $\mathcal{Z} = \mathcal{A}^T * \mathcal{Y}$ for some $\mathcal{Y}$. Then

$$\begin{aligned} D_{f,\mathcal{Z}}(\mathcal{X}, \hat{\mathcal{X}}) &= f^*(\mathcal{Z}) - \langle \mathcal{Z}, \hat{\mathcal{X}}\rangle + f(\hat{\mathcal{X}}) = f^*(\mathcal{A}^T * \mathcal{Y}) - \langle \mathcal{A}^T * \mathcal{Y}, \hat{\mathcal{X}}\rangle + f(\hat{\mathcal{X}}) \\ &= f^*(\mathcal{A}^T * \mathcal{Y}) - \langle \mathcal{Y}, \mathcal{A} * \hat{\mathcal{X}}\rangle + f(\hat{\mathcal{X}}) = f^*(\mathcal{A}^T * \mathcal{Y}) - \langle \mathcal{Y}, \mathcal{B}\rangle - m \\ &= g_f(\mathcal{Y}) - m. \end{aligned}$$

Similarly, for $\tilde{\mathcal{X}}$, there exists $\tilde{\mathcal{Y}}$ such that $D_{f,\mathcal{Z}}(\tilde{\mathcal{X}}, \hat{\mathcal{X}}) = g_f(\tilde{\mathcal{Y}}) - m$.

The assumption $D_{f,\mathcal{Z}}(\mathcal{X}, \hat{\mathcal{X}}) \le D_{f,\tilde{\mathcal{Z}}}(\tilde{\mathcal{X}}, \hat{\mathcal{X}})$ implies that $\mathcal{Y} \in \{\mathcal{W} : g_f(\mathcal{W}) \le g_f(\tilde{\mathcal{Y}})\}$, a level set of g_f. By Lemma 2.6, the restricted strong convexity of g_f on its level set implies the existence of $\nu > 0$ such that

$$g_f(\mathcal{Y}) - m \le \frac{1}{\nu}\|\nabla g_f(\mathcal{Y})\|_F^2 \quad \text{for all } \mathcal{Y} \in \{\mathcal{W} : g_f(\mathcal{W}) \le g_f(\tilde{\mathcal{Y}})\}.$$

Furthermore, the gradient of g_f is computed as

$$\nabla g_f(\mathcal{Y}) = \mathcal{A} * \nabla f^*(\mathcal{A}^T * \mathcal{Y}) - \mathcal{B} = \mathcal{A} * \nabla f^*(\mathcal{Z}) - \mathcal{B} = \mathcal{A} * \mathcal{X} - \mathcal{B} = \mathcal{A} * (\mathcal{X} - \hat{\mathcal{X}}).$$

This proves part (a).

(b) The proof is similar to (a) and simpler. Under the assumption that g_f is restricted strong convex on $\mathbb{R}^{N_1\times K\times N_3}$, we no longer need to require the dual variable $\mathcal{Y}$ to be in a level set. The inequality (3.9) is thus true for any $\mathcal{X}$ and $\mathcal{Z} \in \partial f(\mathcal{X}) \cap R(\mathcal{A})$. ∎

Lemma 3.6 is a key lemma in proving the linear convergence rate for the randomized version. However, its assumptions are not easy to check. Below we provide some important examples.

Example 3.7. *Admissible functions.* See [44].

(a) If f is strongly convex and piecewise quadratic and bcirc($\mathcal{A}$) has full row rank, then f is strongly admissible. There are many functions that fall under this category, e.g., $f_\lambda(\mathcal{X}) = \frac{1}{2}\|\mathcal{X}\|_F^2 + \lambda\|\mathcal{X}\|_1$.

(b) Let f be defined on matrices. The objective function of the regularized nuclear norm problem $f(X) = \frac{1}{2}\|X\|_F^2 + \lambda\|X\|_*$ is admissible.

(c) When f is defined on $\mathbb{R}^N$, important examples include $f_\lambda(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1$, or more generally $\frac{1}{2}\|\mathbf{x}\|_2^2 + \lambda\|U\mathbf{x}\|_1$, where U is a linear operator.

Remark 3.8. The constant ν in Lemma 3.6 depends on the tensor $\mathcal{A}$, the function f, and the corresponding level set (if f is only admissible). For a simple example, we let $K = N_3 = 1$, so $\mathcal{A} * \mathcal{X}$ degenerates to the regular matrix-vector multiplication $A\mathbf{x}$. We let the objective function be $f_1(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2$, we have $\mathbf{x} = \mathbf{z} \in R(A)$. Furthermore, the minimizer $\hat{\mathbf{x}}$ is the projection of the any feasible vector onto $R(A)$, so $\mathbf{x} - \hat{\mathbf{x}} \in R(A)$. Then $D_{f_1,\mathbf{z}}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{2}\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 = \frac{1}{2}\|P_{R(A)}(\mathbf{x} - \hat{\mathbf{x}})\|_2^2 \overset{(2.1)}{\leq} \frac{1}{2\sigma_{\min}^2(A)}\|A(\mathbf{x} - \hat{\mathbf{x}})\|_2^2$, which means that $\nu = 2\sigma_{\min}^2(A)$. For a less trivial example, we refer readers to [29, Lemma 7] for an explicit computation of ν for the function $f_\lambda(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1$ (A does not need to be full row rank). In general, it is hard to quantify ν.

Theorem 3.9. *Let f be α_f-strongly convex and admissible, and ν is the constant from (3.9). Let $\mathcal{X}^{(k)}, \mathcal{Z}^{(k)}$ be generated by the Algorithm 3.2. If $0 < \frac{\nu t}{\|\mathcal{A}\|_F^2}(1 - \frac{tN_3}{2\alpha_f}) < 1$, then $\mathcal{X}^{(k)}$ converges linearly to $\hat{\mathcal{X}}$ in expectation, i.e.,*

$$\mathbb{E}\left[D_{f,\mathcal{Z}^{(k+1)}}(\mathcal{X}^{(k+1)}, \hat{\mathcal{X}})\right] \leq \left(1 - \frac{\nu t}{\|\mathcal{A}\|_F^2}\left(1 - \frac{tN_3}{2\alpha_f}\right)\right)\mathbb{E}\left[D_{f,\mathcal{Z}^{(k)}}(\mathcal{X}^{(k)}, \hat{\mathcal{X}})\right], \tag{3.10}$$

and thereby

$$\mathbb{E}\|\mathcal{X}^{(k)} - \hat{\mathcal{X}}\|_F^2 \leq \left[\frac{2}{\alpha_f} D_{f,\mathcal{Z}^{(0)}}(\mathcal{X}^{(0)}, \hat{\mathcal{X}})\right]\left(1 - \frac{\nu t}{\|\mathcal{A}\|_F^2}\left(1 - \frac{tN_3}{2\alpha_f}\right)\right)^k. \tag{3.11}$$

Proof. Let $\mathcal{G}^{(k)} := \mathcal{X}^{(k)} - \hat{\mathcal{X}}$. By Theorem 3.3,

$$D_{f,\mathcal{Z}^{(k+1)}}(\mathcal{X}^{(k+1)}, \hat{\mathcal{X}}) \leq D_{f,\mathcal{Z}^{(k)}}(\mathcal{X}^{(k)}, \hat{\mathcal{X}}) - r\frac{\|\mathcal{A}(i(k)) * \mathcal{G}^{(k)}\|_F^2}{\|\mathcal{A}(i(k))\|_F^2}, \tag{3.12}$$

where $r = t(1 - \frac{tN_3}{2\alpha_f})$. We need to analyze the expectation of $\frac{\|\mathcal{A}(i(k)) * \mathcal{G}^{(k)}\|_F^2}{\|\mathcal{A}(i(k))\|_F^2}$.

Since $D_{f,\mathcal{Z}^{(k)}}(\mathcal{X}^{(k)}, \hat{\mathcal{X}}) \leq D_{f,\mathcal{Z}^{(0)}}(\mathcal{X}^{(0)}, \hat{\mathcal{X}})$ for any $k \geq 1$, we apply Lemma 3.6(a) and get

$$D_{f,\mathcal{Z}^{(k)}}(\mathcal{X}^{(k)}, \hat{\mathcal{X}}) \leq \frac{1}{\nu}\|\mathcal{A} * \mathcal{G}^{(k)}\|_F^2. \tag{3.13}$$

For the sake of convenience, we let $\mathbb{E}_c$ be the expectation conditioned on $i(0), \ldots, i(k-1)$. At the kth iteration, the probability that $i(k) = j$ is proportional to $\|\mathcal{A}(j)\|_F^2$. Therefore, we have

$$\mathbb{E}_c\left[\frac{\|\mathcal{A}(i(k)) * \mathcal{G}^{(k)}\|_F^2}{\|\mathcal{A}(i(k))\|_F^2}\right] = \sum_{j=1}^{N_1}\frac{\|\mathcal{A}(j)\|_F^2}{\|\mathcal{A}\|_F^2}\frac{\|\mathcal{A}(j) * \mathcal{G}^{(k)}\|_F^2}{\|\mathcal{A}(j)\|_F^2} \tag{3.14}$$

$$\overset{(3.4)}{=} \frac{1}{\|\mathcal{A}\|_F^2}\|\mathcal{A} * \mathcal{G}^{(k)}\|_F^2 \overset{(3.13)}{\geq} \frac{\nu}{\|\mathcal{A}\|_F^2} D_{f,\mathcal{Z}^{(k)}}(\mathcal{X}^{(k)}, \hat{\mathcal{X}}). \tag{3.15}$$

Finally, we take the expectation of (3.12) and get

$$\begin{aligned}
&\mathbb{E}\left[D_{f,\mathcal{Z}^{(k+1)}}(\mathcal{X}^{(k+1)},\hat{\mathcal{X}})\right]\\
&\le \mathbb{E}\left[D_{f,\mathcal{Z}^{(k)}}(\mathcal{X}^{(k)},\hat{\mathcal{X}}) - r\frac{\|\mathcal{A}(i(k))*\mathcal{G}^{(k)}\|_F^2}{\|\mathcal{A}(i(k))\|_F^2}\right]\\
&= \mathbb{E}_{i(0),\dots,i(k-1)}\mathbb{E}_c\left[D_{f,\mathcal{Z}^{(k)}}(\mathcal{X}^{(k)},\hat{\mathcal{X}}) - r\frac{\|\mathcal{A}(i(k))*\mathcal{G}^{(k)}\|_F^2}{\|\mathcal{A}(i(k))\|_F^2}\right]\\
&\le \mathbb{E}_{i(0),\dots,i(k-1)}\left[D_{f,\mathcal{Z}^{(k)}}(\mathcal{X}^{(k)},\hat{\mathcal{X}}) - \frac{r\nu}{\|\mathcal{A}\|_F^2}D_{f,\mathcal{Z}^{(k)}}(\mathcal{X}^{(k)},\hat{\mathcal{X}})\right]\\
&= \left(1-\frac{r\nu}{\|\mathcal{A}\|_F^2}\right)\mathbb{E}\left[D_{f,\mathcal{Z}^{(k)}}(\mathcal{X}^{(k)},\hat{\mathcal{X}})\right].
\end{aligned}$$

To prove (3.11), we note that $\mathbb{E}[D_{f,\mathcal{Z}^{(k)}}(\mathcal{X}^{(k)},\hat{\mathcal{X}})] \le (1-\frac{r\nu}{\|\mathcal{A}\|_F^2})^k D_{f,\mathcal{Z}^{(0)}}(\mathcal{X}^{(0)},\hat{\mathcal{X}})$. Then by (2.8), we have

$$\mathbb{E}\|\mathcal{X}^{(k)}-\hat{\mathcal{X}}\|_F^2 \le \frac{2}{\alpha_f}\mathbb{E}\left[D_{f,\mathcal{Z}^{(k)}}(\mathcal{X}^{(k)},\hat{\mathcal{X}})\right] \le \left[\frac{2}{\alpha_f}D_{f,\mathcal{Z}^{(0)}}(\mathcal{X}^{(0)},\hat{\mathcal{X}})\right]\left(1-\frac{r\nu}{\|\mathcal{A}\|_F^2}\right)^k. \quad \blacksquare$$

So far, we have taken care of consistent and noise-free constraints of the form $\mathcal{A}*\mathcal{X}=\mathcal{B}$. In what follows, we analyze the sensitivity of Algorithm 3.2 by considering the perturbed constraint $\mathcal{A}*\mathcal{X}=\tilde{\mathcal{B}}$ where $\tilde{\mathcal{B}}=\mathcal{B}+\mathcal{E}$. Here $\mathcal{E}$ is typically assumed to be Gaussian noise.

Theorem 3.10. *Let $\mathcal{A}*\mathcal{X}=\mathcal{B}$ be the original consistent linear constraint. For the perturbed measurements $\tilde{\mathcal{B}}=\mathcal{B}+\mathcal{E}$, Algorithm 3.2 performs the following updating scheme:*

$$\bar{\mathcal{Z}}^{(k+1)} = \bar{\mathcal{Z}}^{(k)} + t\mathcal{A}(i(k))^T * \frac{\mathcal{B}(i(k))+\mathcal{E}(i(k))-\mathcal{A}(i(k))*\bar{\mathcal{X}}^{(k)}}{\|\mathcal{A}(i(k))\|_F^2}, \tag{3.16}$$

$$\bar{\mathcal{X}}^{(k+1)} = \nabla f^*(\bar{\mathcal{Z}}^{(k+1)}). \tag{3.17}$$

Let f be α_f-strongly convex and strongly admissible, and ν is the constant from (3.9). If $0<\frac{\nu t}{\|\mathcal{A}\|_F^2}(1-\frac{tN_3}{2\alpha_f})<1$ and $\epsilon=\max_{i\in[N_1]}\frac{\|\mathcal{E}(i)\|_F}{\|\mathcal{A}(i)\|_F}$, then we have

(3.18)

$$\mathbb{E}\sqrt{D_{f,\bar{\mathcal{Z}}^{(k)}}(\bar{\mathcal{X}}^{(k)},\hat{\mathcal{X}})} \le \left(\sqrt{1-\frac{\nu t}{\|\mathcal{A}\|_F^2}\left(1-\frac{tN_3}{2\alpha_f}\right)}\right)^k D_{f,\bar{\mathcal{Z}}^{(0)}}(\bar{\mathcal{X}}^{(0)},\hat{\mathcal{X}}) + \frac{\sqrt{2\alpha_f N_3}\|\mathcal{A}\|_F^2\epsilon}{\nu\left(1-\frac{tN_3}{2\alpha_f}\right)},$$

where $\hat{\mathcal{X}}$ is the solution of (3.1).

Proof. Given $\bar{\mathcal{Z}}^{(k)}$ and $\bar{\mathcal{X}}^{(k)}$, we define

$$\mathcal{Z}^{(k+1)} = \bar{\mathcal{Z}}^{(k)} + t\mathcal{A}(i(k))^T * \frac{\mathcal{B}(i(k))-\mathcal{A}(i(k))*\bar{\mathcal{X}}^{(k)}}{\|\mathcal{A}(i(k))\|_F^2}, \tag{3.19}$$

$$\mathcal{X}^{(k+1)} = \nabla f^*(\mathcal{Z}^{(k+1)}). \tag{3.20}$$

Note that $\mathcal{Z}^{(k+1)}, \mathcal{X}^{(k+1)}$ are not those defined in Algorithm 3.2. By (3.16) and (3.19), we have that $\bar{\mathcal{Z}}^{(k+1)} = \mathcal{Z}^{(k+1)} + \frac{t}{\|\mathcal{A}(i)\|_F^2}\mathcal{A}(i)^T * \mathcal{E}(i)$.

With the update (3.19), we apply Proposition 3.2:

$$D_{f,\mathcal{Z}^{(k+1)}}(\mathcal{X}^{(k+1)}, \hat{\mathcal{X}}) \le D_{f,\bar{\mathcal{Z}}^{(k)}}(\bar{\mathcal{X}}^{(k)}, \hat{\mathcal{X}}) - t\left(1 - \frac{tN_3}{2\alpha_f}\right)\frac{\left\|\mathcal{A}(i) * (\hat{\mathcal{X}} - \bar{\mathcal{X}}^{(k)})\right\|_F^2}{\|\mathcal{A}(i)\|_F^2}. \tag{3.21}$$

Due to the presence of noise, it is not clear if the assumption of Lemma 3.6(a) holds. This is why we require strong admissibility, in which case Lemma 3.6(b) applies and we have

$$D_{f,\bar{\mathcal{Z}}^{(k)}}(\bar{\mathcal{X}}^{(k)}, \hat{\mathcal{X}}) \le \frac{1}{\nu}\|\mathcal{A} * (\bar{\mathcal{X}}^{(k)} - \hat{\mathcal{X}})\|_F^2. \tag{3.22}$$

Similar to the analysis in (3.14)–(3.15), we take the conditional expectation of (3.21) and get

$$\mathbb{E}\left[D_{f,\mathcal{Z}^{(k+1)}}(\mathcal{X}^{(k+1)}, \hat{\mathcal{X}})\middle| i(1), \ldots, i(k-1)\right] \le \left(1 - \frac{\nu t}{\|\mathcal{A}\|_F^2}\left(1 - \frac{tN_3}{2\alpha_f}\right)\right) D_{f,\bar{\mathcal{Z}}^{(k)}}(\bar{\mathcal{X}}^{(k)}, \hat{\mathcal{X}}). \tag{3.23}$$

By Lemma 2.4, we get

$$\begin{aligned}
&D_{f,\bar{\mathcal{Z}}^{(k+1)}}(\bar{\mathcal{X}}^{(k+1)}, \hat{\mathcal{X}}) - D_{f,\mathcal{Z}^{(k+1)}}(\mathcal{X}^{(k+1)}, \hat{\mathcal{X}}) \\
&\le \langle \bar{\mathcal{Z}}^{(k+1)} - \mathcal{Z}^{(k+1)}, \bar{\mathcal{X}}^{(k+1)} - \hat{\mathcal{X}}\rangle \le \|\bar{\mathcal{Z}}^{(k+1)} - \mathcal{Z}^{(k+1)}\|_F \|\bar{\mathcal{X}}^{(k+1)} - \hat{\mathcal{X}}\|_F \\
&\overset{(2.8)}{\le} \frac{t}{\|\mathcal{A}(i)\|_F^2}\|\mathcal{A}(i)^T * \mathcal{E}(i)\|_F \sqrt{\frac{\alpha_f}{2}}\sqrt{D_{f,\bar{\mathcal{Z}}^{(k+1)}}(\bar{\mathcal{X}}^{(k+1)}, \hat{\mathcal{X}})} \\
&\overset{(2.23)}{\le} \frac{t\sqrt{N_3}\|\mathcal{E}(i)\|_F}{\|\mathcal{A}(i)\|_F}\sqrt{\frac{\alpha_f}{2}}\sqrt{D_{f,\bar{\mathcal{Z}}^{(k+1)}}(\bar{\mathcal{X}}^{(k+1)}, \hat{\mathcal{X}})} \le t\epsilon\sqrt{\frac{\alpha_f N_3}{2}}\sqrt{D_{f,\bar{\mathcal{Z}}^{(k+1)}}(\bar{\mathcal{X}}^{(k+1)}, \hat{\mathcal{X}})}.
\end{aligned}$$

By denoting $a = t\epsilon\sqrt{\frac{\alpha_f N_3}{2}}$, $\alpha^2 = D_{f,\bar{\mathcal{Z}}^{(k+1)}}(\bar{\mathcal{X}}^{(k+1)}, \hat{\mathcal{X}})$, and $\beta^2 = D_{f,\mathcal{Z}^{(k+1)}}(\bar{\mathcal{X}}^{(k+1)}, \hat{\mathcal{X}})$, we can rewrite the above inequality as $\alpha^2 - \beta^2 \le a\alpha$, which implies

$$\alpha \le \frac{1}{2}\left(a + \sqrt{a^2 + 4\beta^2}\right)$$

and thereby $\alpha \le a + \beta$. That is,

$$\sqrt{D_{f,\bar{\mathcal{Z}}^{(k+1)}}(\bar{\mathcal{X}}^{(k+1)}, \hat{\mathcal{X}})} \le a + \sqrt{D_{f,\mathcal{Z}^{(k+1)}}(\mathcal{X}^{(k+1)}, \hat{\mathcal{X}})}.$$

We again let $\mathbb{E}_c$ be the expectation conditioned on $i(0), \ldots, i(k-1)$. Then we have

$$\begin{aligned}
&\mathbb{E}_c\sqrt{D_{f,\bar{\mathcal{Z}}^{(k+1)}}(\bar{\mathcal{X}}^{(k+1)}, \hat{\mathcal{X}})} \le a + \mathbb{E}_c\sqrt{D_{f,\mathcal{Z}^{(k+1)}}(\mathcal{X}^{(k+1)}, \hat{\mathcal{X}})} \\
&\le a + \sqrt{\mathbb{E}_c\left[D_{f,\mathcal{Z}^{(k+1)}}(\mathcal{X}^{(k+1)}, \hat{\mathcal{X}})\right]} \overset{(3.23)}{\le} a + \sqrt{\left(1 - \frac{\nu t}{\|\mathcal{A}\|_F^2}\left(1 - \frac{tN_3}{2\alpha_f}\right)\right) D_{f,\bar{\mathcal{Z}}^{(k)}}(\bar{\mathcal{X}}^{(k)}, \hat{\mathcal{X}})},
\end{aligned}$$

which implies

$$d_{k+1} \le a + \sqrt{1 - \frac{\nu t}{\|\mathcal{A}\|_F^2}\left(1 - \frac{tN_3}{2\alpha_f}\right)}\, d_k := a + \sqrt{1-b}\, d_k, \tag{3.24}$$

where $d_k := \mathbb{E}\sqrt{D_{f,\bar{Z}^{(k)}}(\bar{\mathcal{X}}^{(k)}, \hat{\mathcal{X}})}$. After applying (3.24) iteratively, we get

$$\begin{aligned} d_k &\le (\sqrt{1-b})^k d_0 + a\left(1 + \sqrt{1-b} + (\sqrt{1-b})^2 + \cdots + (\sqrt{1-b})^{k-1}\right) \\ &= (\sqrt{1-b})^k d_0 + a\frac{1-(\sqrt{1-b})^k}{1-\sqrt{1-b}} \le (\sqrt{1-b})^k d_0 + a\frac{1}{1-\sqrt{1-b}} \\ &= (\sqrt{1-b})^k d_0 + a\frac{1+\sqrt{1-b}}{b} \le (\sqrt{1-b})^k d_0 + \frac{2a}{b}, \end{aligned}$$

which reduces to (3.18). ∎

4. Special cases of the proposed algorithm.

4.1. Matrix recovery. This section discusses the special cases when $N_3 = 1$. The tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times 1}$ degenerates to the $N_1 \times N_2$ matrix A. Although this has been brought up in Example 3.7, we would like to include further discussions here. Specifically, $\mathcal{X}$ degenerates to the $N_2 \times K$ matrix X and $\mathcal{B}$ degenerates to the $N_1 \times K$ matrix B. The minimization problem (3.1) becomes a matrix recovery problem

$$\min_X f(X) \quad \text{s.t.} \quad AX = B. \tag{4.1}$$

If $f(X) = \lambda\|X\|_* + \frac{1}{2}\|X\|_F^2$, which is an admissible function, we have

$$\nabla f^*(Z) = \text{prox}_{\lambda\|\cdot\|_*}(Z) = D_\lambda(Z).$$

The singular value thresholding operator $D_\lambda(Z)$ is defined as $U \max\{S - \lambda, 0\}V^T$, provided that $Z = USV^T$ is the singular value decomposition.

The following corollary is a consequence of Theorems 3.3 and 3.9. Note this function is 1-strongly convex. Similar to the tensor notation, we will let $A(i)$ be the ith row of A, and $R_M(A) = \{A^T Y : Y \in \mathbb{R}^{N_1 \times K}\}$.

Corollary 4.1. *Let $\hat{X}$ be the solution of (4.1) where $f(X) = \lambda\|X\|_* + \frac{1}{2}\|X\|_F^2$. This function is admissible with the constant ν (see (3.9)). Initializing with $Z^{(0)} \in R_M(A) \subset \mathbb{R}^{N_2 \times K}$ and $X^{(0)} = D_\lambda(Z^{(0)})$, we perform the updating scheme*

$$Z^{(k+1)} = Z^{(k)} + tA(i(k))^T \frac{B(i(k)) - A(i(k))X^{(k)}}{\|A(i(k))\|_F^2}, \tag{4.2}$$

$$X^{(k+1)} = D_\lambda(Z^{(k+1)}), \tag{4.3}$$

where $\{i(k)\}$ is a slice selection sequence and $t < 2$.

(a) *If* $\{i(k)\}$ *is a control sequence for* $[N_1]$*, then the sequence* $X^{(k)}$ *converges to* $\hat{X}$.
(b) *If* $\{i(k)\}$ *is a random sequence such that* $\Pr(i(k) = j) = p_j$*, then*

(4.4)
$$\mathbb{E}\left[D_{f,Z^{(k+1)}}(X^{(k+1)}, \hat{X})\right] \leq \left(1 - \nu t\left(1 - \frac{t}{2}\right)\min_j\left\{\frac{p_j}{\|A(j)\|_F^2}\right\}\right)\mathbb{E}\left[D_{f,Z^{(k)}}(X^{(k)}, \hat{X})\right].$$

Remark 4.2. Corollary 4.1(b) allows a more general probability distribution instead of the specific distribution in Algorithm 3.2. The proof can be easily modified from that in Theorem 3.9.

Remark 4.3. It is worth noting that the linear constraint $AX = B$ is in a different form from $\{X : \langle A_i, X\rangle = b_i, i \in [m]\}$, where each A_i is a matrix. However, the proof of Theorem 3.9 can be easily adapted to these types of constraints. If we focus on the regularized nuclear norm optimization problem

$$\min_{X\in\mathbb{R}^{N_2\times K}} \lambda\|X\|_* + \frac{1}{2}\|X\|_F^2 \quad \text{s.t.} \quad \langle A_i, X\rangle = b_i, i \in [m], \tag{4.5}$$

it is stated in the introduction of [45] that the algorithm

$$\begin{aligned} Z^{(k+1)} &= Z^{(k)} + tA_{i(k)}\frac{b(i(k)) - \langle A_{i(k)}, X^{(k)}\rangle}{\|A_{i(k)}\|_F^2}, \\ X^{(k+1)} &= D_\lambda(Z^{(k+1)}) \end{aligned} \tag{4.6}$$

with a random sequence $\{i(k)\}$ has a linear convergence rate in expectation. Related numerical experiments can be found in section 5.2.

4.2. Vector recovery. As mentioned in Remark 3.8, when $\mathcal{A} \in \mathbb{R}^{N_1\times N_2\times 1}$ and $\mathcal{X} \in \mathbb{R}^{N_2\times 1\times 1}$, $\mathcal{A}$ degenerates to the $N_1 \times N_2$ matrix A and $\mathcal{X}$ degenerates to the vector $\mathbf{x}$. In this section, we further discuss this special case in more detail. Specifically, we consider the following vector version of the minimization problem (3.1):

$$\hat{\mathbf{x}} = \operatorname*{argmin}_x f(\mathbf{x}) \quad \text{s.t.} \quad A\mathbf{x} = \mathbf{b}, \tag{4.7}$$

where $R(A)$ in this context is the row space of A. As a consequence, Algorithm 3.1 or Algorithm 3.2 becomes Algorithm 4.1.

In this setting, Theorems 3.3 and 3.9 reduce to the following results.

Corollary 4.4. *Let* f *be* α_f*-strongly convex.*
(a) *If* $\{i(k)\}$ *is a control sequence for* $[N_1]$*, then* $\mathbf{x}^{(k)}$ *from Algorithm 4.1 converges to* $\hat{\mathbf{x}}$.
(b) *If* f *is admissible and* $\{i(k)\}$ *is a random sequence such that* $\Pr(i(k) = j) = p_j$*, then the iterates from Algorithm 4.1 satisfy*

(4.8)
$$\mathbb{E}\left[D_{f,\mathbf{z}^{(k+1)}}\left(\mathbf{x}^{(k+1)}, \hat{\mathbf{x}}\right)\right] \leq \left(1 - \nu t\left(1 - \frac{t}{2\alpha_f}\right)\min_j\left\{\frac{p_j}{\|A(j)\|_2^2}\right\}\right)\mathbb{E}\left[D_{f,\mathbf{z}^{(k)}}(\mathbf{x}^{(k)}, \hat{\mathbf{x}})\right].$$

Note that Corollary 4.4 aligns with the previous results in literature. Part (a) can be found in [32, Theorem 2.7] in a more general setting, and part (b) can be found in [45, Theorem 4.5]. The following noisy setting is a new contribution, which is a result of Theorem 3.10.

Algorithm 4.1 Regularized Kaczmarz algorithm for vector recovery.

Input: $A \in \mathbb{R}^{N_1 \times N_2}$, $b \in \mathbb{R}^{N_1}$, row selection sequence $\{i(k)\}_{k=0}^{\infty} \subseteq [N_1]$, stepsize t, maximal number of iterations T, and tolerance tol.
Output: an approximation of $\hat{\mathbf{x}}$
Initialize: $\mathbf{z}^{(0)} \in R(A) \subset \mathbb{R}^{N_2}, \mathbf{x}^{(0)} = \nabla f^*(\mathbf{z}^{(0)})$.
for $k = 0, 1, \ldots, T-1$ **do**
$\mathbf{z}^{(k+1)} = \mathbf{z}^{(k)} + t\frac{\mathbf{b}_{i(k)} - A(i(k))\mathbf{x}^{(k)}}{\|A(i(k))\|_2^2} A(i(k))^T$
$\mathbf{x}^{(k+1)} = \nabla f^*(\mathbf{z}^{(k+1)})$
Terminate if $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|_2 / \|\mathbf{x}^{(k)}\|_2 < tol$.
end for

Corollary 4.5. *Let f be α_f-strongly convex and strongly admissible. Let $\bar{\mathbf{z}}^{(k)}, \bar{\mathbf{x}}^{(k)}$ be generated from Algorithm 4.1 ($t < 2\alpha_f$) with the noisy constraint $A\mathbf{x} = \mathbf{b} + \mathbf{e}$, i.e.,*

$$\bar{\mathbf{z}}^{(k+1)} = \bar{\mathbf{z}}^{(k)} + t\frac{\mathbf{b}_{i(k)} + \mathbf{e}_{i(k)} - A(i(k))\bar{\mathbf{x}}^{(k)}}{\|A(i(k))\|_2^2} A(i(k))^T, \tag{4.9}$$

$$\bar{\mathbf{x}}^{(k+1)} = \nabla f^*(\bar{\mathbf{z}}^{(k+1)}), \tag{4.10}$$

Moreover, $\{i(k)\}$ is a random sequence such that $\Pr(i(k) = j) = \frac{\|A(j)\|_2^2}{\|A\|_F^2}$. Let $\epsilon = \max_{i \in [N_1]} \frac{|\mathbf{e}_i|}{\|A(i)\|_2}$. We have

$$\mathbb{E}\sqrt{D_{f,\bar{\mathbf{z}}^{(k)}}(\bar{\mathbf{x}}^{(k)}, \hat{\mathbf{x}})} \le \left(\sqrt{1 - \frac{\nu t\left(1 - \frac{t}{2\alpha_f}\right)}{\|A\|_F^2}}\right)^k D_{f,\bar{\mathbf{z}}^{(0)}}(\bar{\mathbf{x}}^{(0)}, \hat{\mathbf{x}}) + \frac{\sqrt{2\alpha_f}\|A\|_F^2 \epsilon}{\nu\left(1 - \frac{t}{2\alpha_f}\right)}, \tag{4.11}$$

where $\hat{\mathbf{x}}$ is still the solution of (4.7).

In the work [32], some noisy models in practice have been briefly mentioned but without theoretical analysis. Corollary 4.5 states that with perturbed measurements, the expected square root of Bregman distance still enjoys an exponential decay and the iterates are within certain radius (proportional to the noise level) of the true solution.

4.2.1. Kaczmarz type of algorithms. Important examples that satisfy the assumptions of Corollaries 4.4 and 4.5 are when f is strongly convex and piecewise quadratic and A is full row rank (see Example 3.7).

For $f_1(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2$, we have $\mathbf{x}^{(k)} = \mathbf{z}^{(k)}$, and Algorithm 4.1 becomes the well-known Kaczmarz algorithm [24]

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t\frac{\mathbf{b}_{i(k)} - A(i(k))\mathbf{x}^{(k)}}{\|A(i(k))\|_2^2} A(i(k))^T,$$

which is known to converge to the minimum norm solution of $A\mathbf{x} = \mathbf{b}$ if the initial $\mathbf{x}^{(0)}$ is in the row space of A. Linear convergence for the randomized Kaczmarz algorithm can be

found in [48, 13]. In fact, if we apply Corollary 4.4(b) and use the fact that $\nu = 2\sigma_{\min}^2(A)$ (see Remark 3.8), and $\alpha_f = 1$, then (4.8) reduces to

$$\mathbb{E}\left[\left\|\mathbf{x}^{(k+1)} - \hat{\mathbf{x}}\right\|_2^2\right] \leq \left(1 - 2\sigma_{\min}^2(A)t\left(1 - \frac{t}{2}\right)\min_j\left\{\frac{p_j}{\|A(j)\|_2^2}\right\}\right)\mathbb{E}\left[\left\|\mathbf{x}^{(k)} - \hat{\mathbf{x}}\right\|_2^2\right],$$

which is a more general version of the result in [48].

For $f_\lambda(\mathbf{x}) = \lambda\|\mathbf{x}\|_1 + \frac{1}{2}\|\mathbf{x}\|_2^2$, (4.7) becomes a regularized version of the basis pursuit [52, 9],

$$\hat{\mathbf{x}} = \operatorname*{argmin}_x \lambda\|\mathbf{x}\|_1 + \frac{1}{2}\|\mathbf{x}\|_2^2 \quad \text{s.t.} \quad A\mathbf{x} = \mathbf{b}, \tag{4.12}$$

and Algorithm 4.1 becomes the *sparse Kaczmarz method*

$$\begin{aligned} \mathbf{z}^{(k+1)} &= \mathbf{z}^{(k)} + t\frac{\mathbf{b}_{i(k)} - A(i(k))\mathbf{x}^{(k)}}{\|A(i(k))\|_2^2}A(i(k))^T, \\ \mathbf{x}^{(k+1)} &= D_\lambda(\mathbf{z}^{(k+1)}) \end{aligned} \tag{4.13}$$

that was proposed in [32].

4.3. Tensor nuclear norm regularized minimization. In this section, we consider one special case of tensor recovery involving the nuclear norm regularization and linear measurements. Specifically, we adapt the proposed algorithms for solving the *tensor nuclear norm regularized minimization* problem

$$\hat{\mathcal{X}} = \operatorname*{argmin}_{\mathcal{X}\in\mathbb{R}^{N_2\times K\times N_3}} \frac{1}{2}\|\mathcal{X}\|_F^2 + \lambda\|\mathcal{X}\|_{\text{tnn}} \quad \text{s.t.} \quad \mathcal{A} * \mathcal{X} = \mathcal{B}, \tag{4.14}$$

where $\mathcal{A} \in \mathbb{R}^{N_1\times N_2\times N_3}$ and $\mathcal{B} \in \mathbb{R}^{N_1\times K\times N_3}$. Here $\|\mathcal{X}\|_{tnn}$ is the *tensor nuclear norm* of $\mathcal{X}$, which is defined through the Fourier transform.

Definition 4.6. *Given a tensor $\mathcal{X} \in \mathbb{R}^{N_2\times K\times N_3}$, $F(\mathcal{X})$ is the $N_2 \times K \times N_3$ tensor obtained by taking the 1D Fourier transform along each tube of $\mathcal{X}$, i.e.,*

$$F(\mathcal{X})(i, j, :) = \text{fft}(\mathcal{X}(i, j, :)) \quad \textit{for } i \in [N_2], j \in [K].$$

In what follows, we use $F(\mathcal{X})_i$ to denote the ith frontal slice of the tensor $F(\mathcal{X})$.

Finally, we define

$$\|\mathcal{X}\|_{\text{tnn}} := \sum_{k=1}^{N_3} \|F(\mathcal{X})_k\|_*.$$

In addition, the tensor nuclear norm can be computed through t-SVD [26], which involves the SVD of each frontal slice $F(\mathcal{X})_k$:

$$F(\mathcal{X})_k = \tilde{U}_k\tilde{S}_k\tilde{V}_k^T.$$

Let $\tilde{\mathcal{U}}$ be the tensor generated by concatenating $\tilde{U}_k$'s along the third dimension such that $\tilde{U}_k$ is the kth frontal slice of $\tilde{\mathcal{U}}$. Likewise, we obtain $\tilde{\mathcal{S}}$ and $\tilde{\mathcal{V}}$. Denote $\mathcal{U} = F^{-1}(\tilde{\mathcal{U}})$ and $\mathcal{V} = F^{-1}(\tilde{\mathcal{V}})$. The *singular tube thresholding operator* [46] is defined as

$$\mathscr{S}_\tau(\mathcal{X}) := \mathcal{U} * F^{-1}(\max(\tilde{\mathcal{S}} - \tau, 0)) * \mathcal{V}^T.$$

As a straightforward application of this definition, we obtain the following lemma.

Lemma 4.7. *For any $\mathcal{X} \in \mathbb{R}^{N_2 \times K \times N_3}$, the equality $\mathcal{Y} = \mathscr{S}_\lambda(\mathcal{X})$ holds if and only if $F(\mathcal{Y})_k = D_\lambda(F(\mathcal{X}_k))$ for $k = 1, \ldots, N_3$.*

Next we develop regularized Kaczmarz algorithms for solving (4.14). It can be shown that $\nabla f^*(\mathcal{Z}) = \operatorname{prox}_{\lambda\|\cdot\|_{\text{tnn}}}(\mathcal{Z}) = \mathscr{S}_\lambda(\mathcal{Z})$; see [33, Theorem 4.2] for more examples. In this case, Algorithm 3.1 with a given control sequence reduces to Algorithm 4.2, which alternates the Kaczmarz step and singular tube thresholding.

Algorithm 4.2 Regularized Kaczmarz algorithm for solving (4.14).

Input: $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, $\mathcal{B} \in \mathbb{R}^{N_1 \times K \times N_3}$, control sequence $\{i(k)\}_{k=0}^{\infty} \subseteq [N_1]$, stepsize t, maximum number of iterations T, and tolerance tol.
Output: an approximation of $\hat{\mathcal{X}}$
Initialize: $\mathcal{Z}^{(0)} \in R(\mathcal{A}) \subset \mathbb{R}^{N_2 \times K \times N_3}$, $\mathcal{X}^{(0)} = \mathscr{S}_\lambda(\mathcal{Z}^{(0)})$.
for $k = 0, 1, \ldots, T-1$ **do**
  $\mathcal{Z}^{(k+1)} = \mathcal{Z}^{(k)} + t\mathcal{A}(i(k))^T * \frac{\mathcal{B}(i(k)) - \mathcal{A}(i(k)) * \mathcal{X}^{(k)}}{\|\mathcal{A}(i(k))\|_F^2}$
  $\mathcal{X}^{(k+1)} = \mathscr{S}_\lambda(\mathcal{Z}^{(k+1)})$
  Terminate if $\|\mathcal{X}^{(k+1)} - \mathcal{X}^{(k)}\|_F / \|\mathcal{X}^{(k)}\|_F < tol$.
end for

The following corollary is a direct consequence of Theorem 3.3 as the objective function is 1-strongly convex.

Corollary 4.8. *The sequence generated by Algorithm 4.2 with $t < 2/N_3$ satisfies*

$$D_{f,\mathcal{Z}^{(k+1)}}(\mathcal{X}^{(k+1)}, \mathcal{X}) \le D_{f,\mathcal{Z}^{(k)}}(\mathcal{X}^{(k)}, \mathcal{X}) - t\left(1 - \frac{tN_3}{2}\right) \frac{\|\mathcal{A}(i(k)) * (\mathcal{X}^{(k)} - \mathcal{X})\|_F^2}{\|\mathcal{A}(i(k))\|_F^2} \tag{4.15}$$

for all $\mathcal{X} \in H_{i(k)}$. Moreover, the sequence $\{\mathcal{X}^{(k)}\}$ converges to the solution of (4.14).

Similarly, we can adapt Algorithm 3.2 with a random selection of row index to get a reduced version, i.e., Algorithm 4.3, for solving (4.14).

Algorithm 4.3 Randomized regularized Kaczmarz algorithm for solving (4.14).

Input: $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, $\mathcal{B} \in \mathbb{R}^{N_1 \times K \times N_3}$, stepsize t.
Output: an approximation of $\hat{\mathcal{X}}$
Initialize: $\mathcal{Z}^{(0)} \in R(\mathcal{A}) \subset \mathbb{R}^{N_2 \times K \times N_3}$, $\mathcal{X}^{(0)} = \mathscr{S}_\lambda(\mathcal{Z}^{(0)})$.
while termination criteria not satisfied **do**
  pick $i(k)$ randomly from $[N_1]$ with $\Pr(i(k) = j) = \|\mathcal{A}(j)\|_F^2 / \|\mathcal{A}\|_F^2$,
  $\mathcal{Z}^{(k+1)} = \mathcal{Z}^{(k)} + t\mathcal{A}(i(k))^T * \frac{\mathcal{B}(i(k)) - \mathcal{A}(i(k)) * \mathcal{X}^{(k)}}{\|\mathcal{A}(i(k))\|_F^2}$
  $\mathcal{X}^{(k+1)} = \mathscr{S}_\lambda(\mathcal{Z}^{(k+1)})$
end while

Regarding the convergence analysis of Algorithm 4.3, Theorem 3.9 cannot be applied directly as it is not immediately clear if this function is admissible. To address this issue, we propose the following lemma, which can be shown using the definition of Bregman distance and Lemma 4.7.

Lemma 4.9. *Let $f(\mathcal{X}) = \lambda\|\mathcal{X}\|_{\text{tnn}} + \frac{1}{2}\|\mathcal{X}\|_F^2$ be defined on $\mathbb{R}^{N_2\times K\times N_3}$ and its reduced version $f_M(X) = \lambda\|X\|_* + \frac{1}{2}\|X\|_F^2$ be defined on $\mathbb{R}^{N_2\times K}$. Then given $z \in \partial f(\mathcal{X})$, we have*

$$D_{f,Z}(\mathcal{X},\mathcal{W}) = \sum_{j=1}^{N_3} D_{f_M,F(\mathcal{Z})_j}(F(\mathcal{X})_j, F(\mathcal{W})_j).$$

Next we provide two lemmas showing that the Fourier transform of a t-product of two tensors can be efficiently implemented in the matrix setting. Their derivations are based on the fact that block circulant matrices can be block diagonalized by the Fourier transform; see [26]. For a third-order tensor $\mathcal{X}$, we let $\text{diag}(\mathcal{X})$ denote the block diagonal matrix whose ith diagonal block is the ith frontal slice of $\mathcal{X}$.

Lemma 4.10. *If $\mathcal{A} \in \mathbb{R}^{N_1\times N_2\times N_3}$ and $\mathcal{X} \in \mathbb{R}^{N_2\times K\times N_3}$, then*

$$\begin{aligned}\text{unfold}\left(F(\mathcal{A} * \mathcal{X})\right) &= \begin{bmatrix} F(\mathcal{A})_1 & & & \\ & F(\mathcal{A})_2 & & \\ & & \ddots & \\ & & & F(\mathcal{A})_{N_3}\end{bmatrix}\begin{bmatrix} F(\mathcal{X})_1 \\ F(\mathcal{X})_2 \\ \vdots \\ F(\mathcal{X})_{N_3}\end{bmatrix} \\ &= \text{diag}(F(\mathcal{A}))\,\text{unfold}(F(\mathcal{X})).\end{aligned} \tag{4.16}$$

Lemma 4.11. *If $\mathcal{A} \in \mathbb{R}^{N_1\times N_2\times N_3}$ and $\mathcal{Y} \in \mathbb{R}^{N_1\times K\times N_3}$, then*

$$\begin{aligned}\text{unfold}(F(\mathcal{A}^T * \mathcal{Y})) &= \begin{bmatrix} [F(\mathcal{A})_1]^* & & & \\ & [F(\mathcal{A})_2]^* & & \\ & & \ddots & \\ & & & [F(\mathcal{A})_{N_3}]^*\end{bmatrix}\begin{bmatrix} F(\mathcal{Y})_1 \\ F(\mathcal{Y})_2 \\ \vdots \\ F(\mathcal{Y})_{N_3}\end{bmatrix} \\ &= (\text{diag}(F(\mathcal{A})))^*\,\text{unfold}(F(\mathcal{Y})).\end{aligned} \tag{4.17}$$

Theorem 4.12. *If $t < 2/N_3$, then the sequence generated by Algorithm 4.3 converges in expectation with*

$$\mathbb{E}\left[D_{f,\mathcal{Z}^{(k+1)}}(\mathcal{X}^{(k+1)},\hat{\mathcal{X}})\right] \le \left(1 - \frac{\beta\nu}{\|\mathcal{A}\|_F^2}\right)\mathbb{E}\left[D_{f,\mathcal{Z}^{(k)}}(\mathcal{X}^{(k)},\hat{\mathcal{X}})\right], \tag{4.18}$$

where ν is the admissible constant for the function $\lambda\|X\|_* + \frac{1}{2}\|X\|_F^2$ (see Corollary 4.1), and β is given by

$$\beta = \min_{i\in[N_1],j\in[N_3]} t\frac{\|F(\mathcal{A}(i))_j\|_F^2}{\|F(\mathcal{A}(i))\|_F^2}\left(1 - \frac{t\|F(\mathcal{A}(i))_j\|_F^2}{\|F(\mathcal{A}(i))\|_F^2}\right). \tag{4.19}$$

Proof. The objective function can be written as

$$f(\mathcal{X}) = \lambda\|\mathcal{X}\|_{\text{tnn}} + \frac{1}{2}\|\mathcal{X}\|_F^2 = \sum_{j=1}^{N_3}\left(\lambda\|F(\mathcal{X})_j\|_* + \frac{1}{2}\|F(\mathcal{X})_j\|_F^2\right).$$

In light of Lemma 4.10, the constraint can be expressed in the Fourier domain as

$$F(\mathcal{A})_j F(\mathcal{X})_j = F(\mathcal{B})_j, \quad j \in [N_3].$$

Let $\mathcal{V} \in \mathbb{R}^{N_2 \times K \times N_3}$ and $\mathcal{V}_k$ be the kth frontal slice of $\mathcal{V}$. Therefore, if

$$\hat{\mathcal{V}} = \arg\min_{\mathcal{V}} \sum_{j=1}^{N_3} \left(\lambda \|\mathcal{V}_j\|_* + \frac{1}{2}\|\mathcal{V}_j\|_F^2 \right) \quad \text{s.t.} \quad F(\mathcal{A})_j \mathcal{V}_j = F(\mathcal{B})_j, \quad j \in [N_3], \tag{4.20}$$

then $F^{-1}(\hat{\mathcal{V}})$ is the minimizer of (4.14). Note that (4.20) can be split into N_3 subproblems.

We apply $F(\cdot)$ to the first step of Algorithm 4.3:

$$F(\mathcal{Z}^{(k+1)}) = F(\mathcal{Z}^{(k)}) + \frac{t}{\|\mathcal{A}(i)\|_F^2} F\left(\mathcal{A}(i)^T * (\mathcal{B}(i) - \mathcal{A}(i) * \mathcal{X}^{(k)}) \right). \tag{4.21}$$

By Lemmas 4.10 and 4.11,

$$\begin{aligned}
&\text{unfold}\left(F\left(\mathcal{A}(i)^T * (\mathcal{B}(i) - \mathcal{A}(i) * \mathcal{X}^{(k)}) \right) \right) \\
&= (\text{diag}[F(\mathcal{A}(i))])^* \left(\text{unfold}\left(F(\mathcal{B}(i))\right) - \text{unfold}\left(F(\mathcal{A}(i) * \mathcal{X}^{(k)}) \right) \right) \\
&= (\text{diag}[F(\mathcal{A}(i))])^* \left(\text{unfold}\left(F(\mathcal{B}(i))\right) - \text{diag}[F(\mathcal{A}(i))]\, \text{unfold}(F(\mathcal{X}^{(k)})) \right).
\end{aligned}$$

So (4.21) becomes

$$\begin{aligned}
\text{unfold}(F(\mathcal{Z}^{(k+1)})) = \text{unfold}(F(\mathcal{Z}^{(k)})) \\
+ \frac{t}{\|\mathcal{A}(i)\|_F^2} (\text{diag}[F(\mathcal{A}(i))])^* \left(\text{unfold}\left(F(\mathcal{B}(i))\right) - \text{diag}[F(\mathcal{A}(i))]\, \text{unfold}(F(\mathcal{X}^{(k)})) \right).
\end{aligned}$$

By separating the above updating equation into N_3 pieces, we obtain

$$F(\mathcal{Z}^{(k+1)})_j = F(\mathcal{Z}^{(k)})_j + \frac{t}{\|\mathcal{A}(i)\|_F^2} F(\mathcal{A}(i))_j^* \left(F(\mathcal{B}(i))_j - F(\mathcal{A}(i))_j F(\mathcal{X}^{(k)})_j \right), \quad j \in [N_3]. \tag{4.22}$$

By Lemma 4.7, the second step of Algorithm 4.3 becomes

$$F(\mathcal{X}^{(k+1)})_j = D_\lambda(F(\mathcal{Z}^{(k+1)})_j), \quad j \in [N_3]. \tag{4.23}$$

Now we make a change of variables as $F(\mathcal{X}^{(k)})_j = V_j^{(k)}, F(\mathcal{Z}^{(k)})_j = W_j^{(k)}$. Then (4.22)–(4.23) become

$$W_j^{(k+1)} = W_j^{(k)} + \frac{t}{\|\mathcal{A}(i)\|_F^2} F(\mathcal{A}(i))_j^* \left(F(\mathcal{B}(i))_j - F(\mathcal{A}(i))_j V_j^{(k)} \right), \quad j \in [N_3], \tag{4.24}$$

$$V_j^{(k+1)} = D_\lambda(W_j^{(k+1)}), \quad j \in [N_3]. \tag{4.25}$$

Note that (4.24)–(4.25) are the two major iteration steps for solving the matrix recovery problem

$$\hat{V}_j = \arg\min_{V_j} \lambda\|V_j\|_* + \frac{1}{2}\|V_j\|_F^2 \quad \text{s.t.} \quad F(\mathcal{A})_j V_j = F(\mathcal{B})_j, j \in [N_3]. \tag{4.26}$$

This is in fact the jth subproblem of (4.20), so we wish to apply Corollary 4.1(b). To fit into the format of (4.2), we rewrite

$$\frac{t}{\|\mathcal{A}(i)\|_F^2} = \frac{t}{\|F(\mathcal{A}(i))\|_F^2} = \frac{t\|F(\mathcal{A}(i))_j\|_F^2}{\|F(\mathcal{A}(i))\|_F^2}\frac{1}{\|F(\mathcal{A}(i))_j\|_F^2}.$$

The stepsize given by

$$s_j = \frac{t\|F(\mathcal{A}(i))_j\|_F^2}{\|F(\mathcal{A}(i))\|_F^2} \le t \le 2/N_3 \le 2$$

fits the assumption of Corollary 4.1. The probability distribution is $p_i = \frac{\|F(\mathcal{A}(i))\|_F^2}{\|F(\mathcal{A})\|_F^2}$. We compute

$$\frac{p_i}{\|F(\mathcal{A}(i))_j\|_F^2} = \frac{\|F(\mathcal{A}(i))\|_F^2}{\|F(\mathcal{A})\|_F^2}\frac{1}{\|F(\mathcal{A}(i))_j\|_F^2} \ge \frac{1}{\|F(\mathcal{A})\|_F^2} = \frac{1}{\|\mathcal{A}\|_F^2}.$$

So applying Corollary 4.1(b) yields

$$\mathbb{E}\left[D_{f_M,W_j^{(k+1)}}(V_j^{(k+1)}, \hat{V}_j)\right] \le \left(1 - \frac{\nu s_j\left(1-\frac{s_j}{2}\right)}{\|\mathcal{A}\|_F^2}\right)\mathbb{E}\left[D_{f_M,W_j^{(k)}}\left(V_j^{(k)}, \hat{V}_j\right)\right]. \tag{4.27}$$

Finally, using Lemma 4.9, we have

$$\begin{aligned}
\mathbb{E}\left[D_{f,\mathcal{Z}^{(k+1)}}\left(\mathcal{X}^{(k+1)}, \hat{\mathcal{X}}\right)\right] &= \sum_{j=1}^{N_3}\mathbb{E}\left[D_{f_M,W_j^{(k+1)}}\left(V_j^{(k+1)}, \hat{V}_j\right)\right] \\
&\le \sum_{j=1}^{N_3}\left(1 - \frac{\nu s_j\left(1-\frac{s_j}{2}\right)}{\|\mathcal{A}\|_F^2}\right)\mathbb{E}\left[D_{f_M,W_j^{(k)}}\left(V_j^{(k)}, \hat{V}_j\right)\right] \\
&\le \left(1 - \frac{\nu\beta}{\|\mathcal{A}\|_F^2}\right)\mathbb{E}\left[D_{f,\mathcal{Z}^{(k)}}(\mathcal{X}^{(k)}, \hat{\mathcal{X}})\right],
\end{aligned}$$

where

$$\beta = \min_{i\in[N_1],j\in[N_3]} s_j(1-s_j/2) = \min_{i\in[N_1],j\in[N_3]} t\frac{\|F(\mathcal{A}(i))_j\|_F^2}{\|F(\mathcal{A}(i))\|_F^2}\left(1 - \frac{t\|F(\mathcal{A}(i))_j\|_F^2}{\|F(\mathcal{A}(i))\|_F^2}\right) > 0.$$

■

5. Numerical experiments. In this section, we illustrate the performance of the proposed algorithms in several application problems, including 1D sparse signal recovery, low-rank image inpainting, low-rank tensor recovery, and image deblurring. There are two special cases of our proposed algorithms when the constraint selection sequence is either cyclic or random. For the random version, we take the average of all the results obtained by running 50 trials. To save computational time, we only execute the deterministic version with a cyclic control sequence for image inpainting and deblurring tests. For image deblurring, we also consider a

batched version of the proposed algorithm to improve the performance. However, the batching technique has shown very limited performance enhancement in our other experiments, so we skip those experiments.

To make fair performance comparisons, we adopt the widely used quantitative metrics. For tensor recovery, we use the relative error (RelErr) defined as

$$\text{RelErr} = \frac{\|\mathcal{X} - \widehat{\mathcal{X}}\|_F}{\|\mathcal{X}\|_F},$$

where $\widehat{\mathcal{X}}$ is an estimate of the ground truth tensor $\mathcal{X}$. As one of the most important image quality metrics, the peak signal-to-noise ratio (PSNR) is defined as

$$\text{PSNR} = 20 \log \left(\frac{I_{\max}}{\|\widehat{X} - X\|_F} \right),$$

where $\widehat{X}$ is an estimate of the noise-free image X and $I_{\max}$ is the maximum possible image intensity. In addition, the structural similarity index (SSIM) between two images X and Y is defined as

$$\text{SSIM} = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)},$$

where μ_x, σ_x are the mean and standard deviation of the image X, σ_{xy} is the cross-covariance between X and Y, and C_1 and C_2 are the luminance and contrast constants. Both PSNR and SSIM values can be obtained efficiently in MATLAB via `PSNR` and `SSIM`, respectively.

All the numerical experiments are implemented using MATLAB R2019a for Windows 10 on a desktop PC with 64GB RAM and a 3.10GHz Intel Core i9-9960X CPU.

5.1. One-dimensional sparse signal recovery. We will compare the performance of the following three methods for solving (4.12): (1) linearized Bregman iteration [52, 9] (denoted by LinBreg); (2) alternating direction method of multipliers (ADMM) [6]; (3) our proposed regularized Kaczmarz method (4.13) with random or deterministic cyclic sequence (denoted by RK-rand and RK-cyc). In particular, LinBreg has the following iterations:

$$\begin{aligned} \mathbf{z}^{(k+1)} &= \mathbf{z}^{(k)} + tA^T(\mathbf{b} - A\mathbf{x}^{(k)}), \\ \mathbf{x}^{(k+1)} &= S_\lambda(\mathbf{z}^{(k+1)}). \end{aligned} \tag{5.1}$$

For ADMM, we rewrite (4.12) as

$$\min_{\mathbf{x},\mathbf{w}} \frac{1}{2}\|\mathbf{x}\|_2^2 + \lambda\|\mathbf{w}\|_1 \quad \text{s.t.} \quad A\mathbf{x} = \mathbf{b},\ \mathbf{w} = \mathbf{x},$$

and the corresponding augmented Lagrangian reads as

$$L(\mathbf{x}, \mathbf{w}, \mathbf{u}_1, \mathbf{u}_2) = \frac{1}{2}\|\mathbf{x}\|_2^2 + \lambda\|\mathbf{w}\|_1 + \frac{\rho_1}{2}\|A\mathbf{x} - \mathbf{b} + \mathbf{u}_1\|_2^2 + \frac{\rho_2}{2}\|\mathbf{x} - \mathbf{w} + \mathbf{u}_2\|_2^2.$$

In our experiment, A is a 200×1000 Gaussian matrix. The ground truth vector $\mathbf{x}$ is a 10-sparse vector whose support is randomly generated and the entries on each support index

are independent and follow the normal distribution with mean 1 and standard deviation 1. The parameters are all tuned to achieve optimal performance. For the regularized Kaczmarz method, the stepsize is $t = 40$, and we show the results when the indices are chosen cyclically or randomly. For the linearized Bregman iteration, the stepsize $t = 20$. For ADMM, we pick $\rho_1 = 10$ and $\rho_2 = 100$.

Figure 1 shows the relative error of all four methods versus the running time. Both versions of the regularized Kaczmarz algorithms are outperforming LinBreg and ADMM. Moreover, the cyclic version of the regularized Kaczmarz method performs slightly better than the randomized one.

5.2. Image inpainting. In the second experiment, we consider a low-rank image inpainting problem. The test image is a checkerboard image of size 128×128 with a large missing rectangular area; see the first image of Figure 2. This image can be described by a rank-two matrix, so it is appropriate to be recovered via the model (4.5). Let I be the image to be

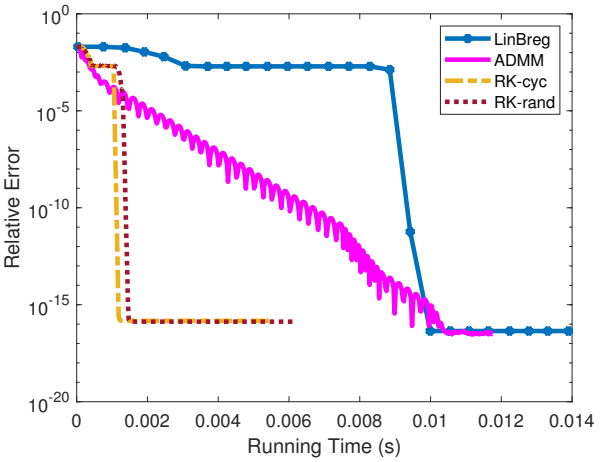

Figure 1. *One-dimensional sparse signal recovery.*

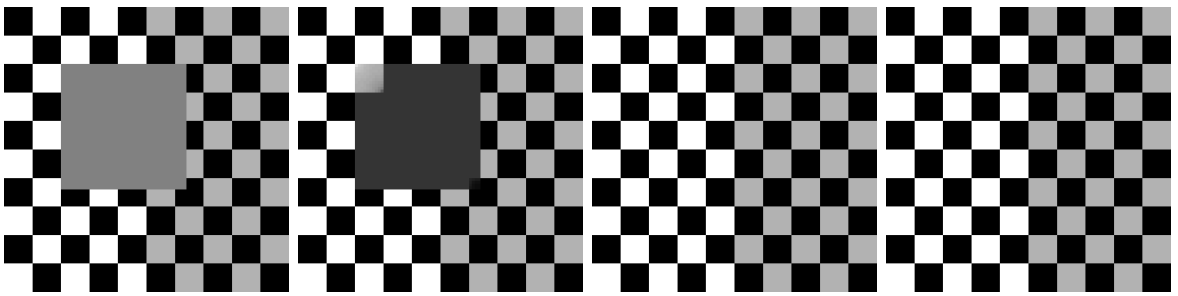

Figure 2. *Image inpainting without noise. The original checkerboard image has a missing box. The linearized Bregman result uses (5.2) with $t = 1$. Our result uses (5.3) with $t = 9$ and batch size 2000. The running times are 1.24s (TV), 0.95s (LinBreg), 0.60s (proposed).*

recovered, and let Ω be the set of indices whose pixel values are known. Then the linear constraints are $\{\langle P_{ij}, X\rangle = I_{ij}, (i,j) \in \Omega\}$, where P_{ij} is the matrix whose entries are all zeros except its ijth entry is one. We use X_Ω for keeping the pixel intensities in Ω and setting other pixel intensities to be zero.

We consider three image inpainting methods: (1) total variation (TV) based image inpainting [47, 20], (2) linearized Bregman iteration [8] (denoted by LinBreg as before), and (3) the proposed regularized Kaczmarz method (4.6). For LinBreg, we adopt the following updating scheme in analogy to (5.1):

$$\begin{aligned} Z^{(k+1)} &= Z^{(k)} + t(I - X^{(k)})_\Omega, \\ X^{(k+1)} &= D_\lambda(Z^{(k+1)}). \end{aligned} \tag{5.2}$$

The second is the regularized Kaczmarz method (4.6). In this application, the first step of (4.6) becomes $Z^{(k+1)} = Z^{(k)} + t(I - X^{(k)})_{(i,j)}$, which means only one pixel from Ω is updated. Again, the choice of indexing can be cyclic or random. In our numerical experiment, we will use a more general version $Z^{(k+1)} = Z^{(k)} + \sum_{(i,j)\in T(k)} t(I - X^{(k)})_{ij} = Z^{(k)} + t(I - X^{(k)})_{T(k)}$, where $T(k) \subset \Omega$, so that $|T(k)|$ many pixels are updated in one iteration. A different index set $T(k)$ is chosen at each iteration k, but we do require that they have the same *batch size*, i.e., the cardinality $|T(k)| = b$. Therefore, our algorithm for this specific case reads as

$$\begin{aligned} Z^{(k+1)} &= Z^{(k)} + t(I - X^{(k)})_{T(k)}, \\ X^{(k+1)} &= D_\lambda(Z^{(k+1)}). \end{aligned} \tag{5.3}$$

In the methods involving singular value thresholding [8], we choose $\lambda = 1500$.

Figure 2 compares all the listed methods quantitatively and qualitatively. For the TV result, the image was recovered by minimizing the functional $\lambda\|\nabla X_x\|_1 + \|\nabla X_y\|_1 + \frac{1}{2}\|(X - I)_\Omega\|_2^2$. The TV regularization only considers the piecewise constant type of smoothness and thus it may not handle texture-like images with a low-rank structure very well. Since the missing area is relatively large in the test image, TV fails to recover the image as expected. As shown in the last two subfigures of Figure 2, either the linearized Bregman iteration or the regularized Kaczmarz iteration is able to achieve almost perfect reconstruction. In our regularized Kaczmarz iteration, we choose batch size to be 2000 with a cyclic indexing. The regularized Kaczmarz enjoys faster convergence.

The batch size b plays an important role in a lot of optimization algorithms, e.g., stochastic gradient descent. In this application, b can be viewed as the number of pixels updated in step 1 of (5.3). If $b = |\Omega|$, then our algorithm coincides with (5.2). This is related to StoGradMP [37], especially with a random choice of the constraints. Some other relevant works include the block Kaczmarz algorithm [38, 14]. The proof presented in this paper can also be adapted to show that the algorithm

$$\begin{cases} \text{Pick an index set } T(k) \text{ whose cardinality is } b, \\ Z^{(k+1)} = Z^{(k)} + t\left(P_{span(A_i, i\in T(k))}\hat{X} - P_{span(A_i, i\in T(k))}X^{(k)}\right), \\ X^{(k+1)} = D_\lambda(Z^{(k+1)}) \end{cases} \tag{5.4}$$

produces a sequence $\{X^{(k)}\}$ that converges to the solution of (4.5). Note that in the image inpainting problem, (5.3) is exactly this block version (5.4) due to the orthogonality of the

indicator matrices P_{ij}. The batch size does influence the convergence of (5.3). For this particular example, we have $|\Omega| = 8064$ and we found a batch size around 2000 to be optimal.

5.3. Low-rank tensor recovery. Assume that the ground truth tensor $\mathcal{X} \in \mathbb{R}^{N_2 \times K \times N_3}$ with a small tubal rank satisfies the tensor system

$$\mathcal{A} * \mathcal{X} = \mathcal{B}, \quad \mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}.$$

Then we consider the tensor recovery model (4.14) with the tensor nuclear norm regularization. It can be empirically shown that Algorithm 3.1 can achieve good performance in terms of accuracy and convergence speed when N_1 is larger than the other dimensions N_2, N_3, K but may fail to converge in other scenarios. As an illustration, we show the performance of our algorithm with cyclic and random control sequences in Figure 3, where $N_1 = 200$, $N_2 = N_3 = K = 100$, and the maximal number of iterations as 2000. Both the coefficient tensor $\mathcal{A}$ and the ground truth tensor $\mathcal{X}$ are normally distributed, and the tubal rank of $\mathcal{X}$ is two by taking the hard thresholding of singular tubes after t-SVD. For the random case, we take the average of 50 trials. One can see that the cyclic control sequence achieves faster convergence with much smaller error but with slightly more running time.

5.4. Tensor-based image deconvolution. Consider an image $\widehat{X} \in \mathbb{R}^{m_1 \times n_1}$ that is degraded by taking the convolution with a point spread function $\widehat{H} \in \mathbb{R}^{m_2 \times n_2}$. Let $m = m_1 + m_2 - 1$ and $n = n_1 + n_2 - 1$. Using the zero padding, both $\widehat{X}$ and $\widehat{H}$ can be extended to two respective matrices X, H of size $m \times n$. By the construction, we have

$$H \circledast X = Y,$$

where $Y \in \mathbb{R}^{m \times n}$ and $\circledast$ is the 2D convolution. Next we establish the equivalence between 2D convolution and t-product by creating a doubly block circulant matrix [1, Appendix]. Let h_i be the ith row of H, and let $A_i := \text{circ}(h_i) \in \mathbb{R}^{n \times n}, i \in [m]$ be the circulant matrix

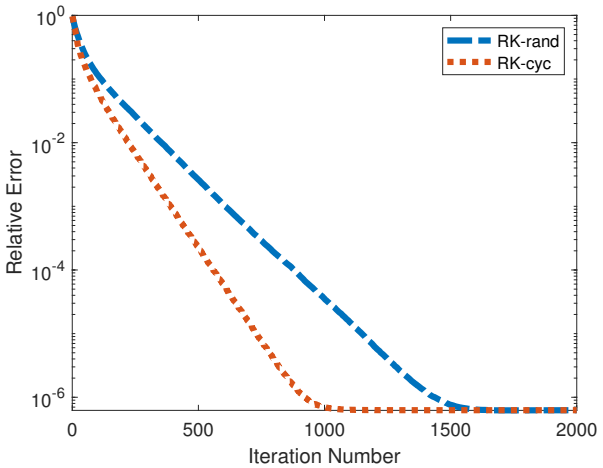

Figure 3. *Low-rank tensor recovery. Running times for the random and cyclic control sequences are* 0.8721*s and* 0.8612*s, respectively. Both run* 2000 *iterations.*

generated by h_i. Let A_i be the ith frontal slice of $\mathcal{A} \in \mathbb{R}^{n\times n\times m}$. Then we create a doubly block circulant matrix $\text{bcirc}(\mathcal{A})$ based on $\mathcal{A}$. Let $\mathcal{X} \in \mathbb{R}^{n\times 1\times m}$ be the tensor version of X by setting $\mathcal{X}(j,1,i) = X(i,j)$ for $i \in [m]$ and $j \in [n]$. Based on (2.14) and (2.15), the 2D convolution can be represented as

$$\text{vec}(H \circledast X) = \text{bcirc}(\mathcal{A})\,\text{vec}(X) = \text{unfold}(\mathcal{A} * \mathcal{X}).$$

Here $\text{vec}(\cdot)$ is a vectorization operator by rowwise stacking such that $\text{vec}(X) = \text{unfold}(\mathcal{X})$. The form of $\mathcal{A} * \mathcal{X}$ in this setting can also be derived from (2.16) and (2.21). To recover $\mathcal{X}$, we consider the low-rank tensor recovery model

$$\min_{\mathcal{X}\in\mathbb{R}^{n\times 1\times m}} \lambda\|\mathcal{X}\|_* \quad \text{s.t.} \quad \mathcal{A} * \mathcal{X} = \mathcal{Y},$$

where $\mathcal{Y} \in \mathbb{R}^{n\times 1\times m}$ is the tensor version of the observed blurry image Y. Note that the tubal rank of the ground truth $\mathcal{X}$ is at most one since the second dimension of $\mathcal{X}$ is one, which implies that $\mathcal{X}$ is low-rank. The conversion between Y and $\mathcal{Y}$ is the same as that between X and $\mathcal{X}$. Next we apply Algorithm 4.3 to recover $\mathcal{X}$ and thereby the image X.

We test an image "house" of size 256×256, which is degraded by a Gaussian convolution kernel of size 9×9 with standard deviation 2. Then we compare various image deblurring methods, including TV image deblurring [11], nonlocal means (NLM) [7], BM3D [15], and our algorithm with batch sizes $b = 20, 40, 60, 80$ and $\alpha = 1$, $\lambda = 0.1$. Here TV, NLM, and BM3D are performed in the plug-and-play ADMM image recovery framework [12]. The MATLAB sources codes can be found in https://www.mathworks.com/matlabcentral/fileexchange/60641-plug-and-play-admm-for-image-restoration. To avoid ringing artifacts, we extend the image symmetrically along the boundary 14 pixels, apply the algorithm, and cut the results back to the normal size. Figure 4 shows the images recovered by TV, NLM, BM3D, and our algorithm with $b = 80$. All results are compared quantitatively in Table 2 in terms of PSNR, SSIM, and running time.

More generally, we consider an image sequence $\mathcal{X} \in \mathbb{R}^{n\times p\times m}$ with p frames (a video). Assume that all frames are convolved with the same spatial blurring kernel in its extended tensor form $\mathcal{A} \in \mathbb{R}^{n\times n\times m}$. As an illustrative example, we test the 3D MRI image data set `mri` in MATLAB, which consists of 12 slices of size 128×128 from an MRI data scan of a human cranium. When $p \ll \min\{m,n\}$, the ground truth $\mathcal{X}$ can be considered as low-rank. Each blurry image is generated by convolving the ground truth with a Gaussian convolution kernel of size 5×5 with standard deviation 2. The parameters are $\alpha = 1$, $\lambda = 10^{-2}$, the maximum iteration number is 1000, and the batch size is 60. Figure 5 shows the first four frames of blurry observations and their respective recovered image. To suppress ringing artifacts, projection of all intensities onto the positive values is set as a postprocessing step.

6. Conclusions and future work. In many tensor recovery problems, the underlying tensor is either sparse or low-rank, which can be exploited in the design of efficient algorithms. In this paper, we propose a regularized Kaczmarz algorithm framework for tensor recovery. Precisely, we adopt the t-product for third-order tensors with rapid implementation through the fast Fourier transform and establish a linear convergence rate in expectation for the proposed algorithm with random sequence. In addition, we provide extensive discussions

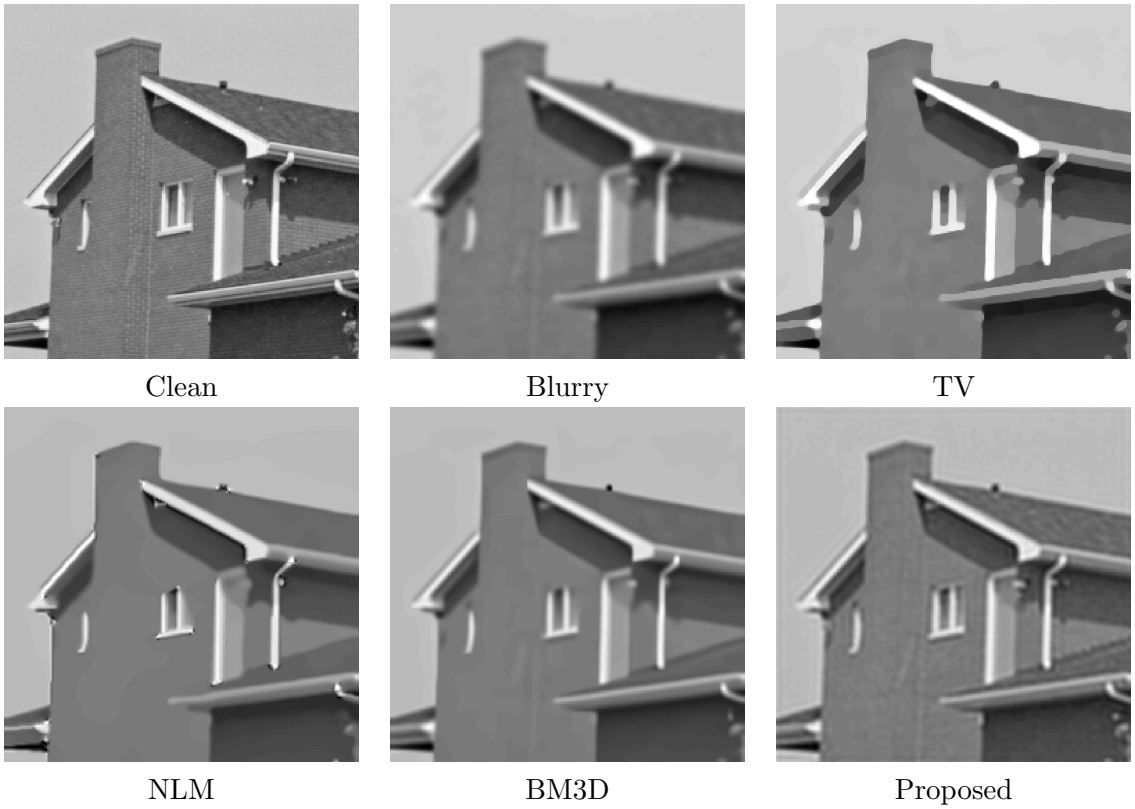

Figure 4. *Visual comparison of various image deblurring methods, including TV, NLM, BM3D, and our proposed tensor-based deblurring algorithm with a cyclic control sequence and blocksize* $b = 80$.

Table 2

Quantitative comparison of various image deblurring methods. Rows 5–8*: The proposed tensor-based image deblurring algorithm with a cyclic control sequence and batch size* $b = 20, 40, 60, 80$.

Method	PSNR	SSIM	Running time (s)
TV	29.48	0.8180	33.34
NLM	29.16	0.8113	29.27
BM3D	30.69	0.8381	318.58
$b = 20$	30.90	0.8257	53.34
$b = 40$	31.10	0.8451	72.88
$b = 60$	31.11	0.8457	97.96
$b = 80$	31.12	0.8461	122.13

on the matrix and vector recovery together with tensor nuclear norm minimization as special cases. A showcase of numerical experiments demonstrates its considerable potential in various applications, including sparse signal recovery, low-rank tensor recovery, image inpainting, and deblurring. In the future, we intend to explore the convergence rate for the deterministic method and discuss theoretical guarantees for its superior performance over the randomized version. Moreover, it would be extremely intriguing to make a thorough discussion on the acceleration effects by choosing an appropriate batch size or stepsize. Furthermore, the current

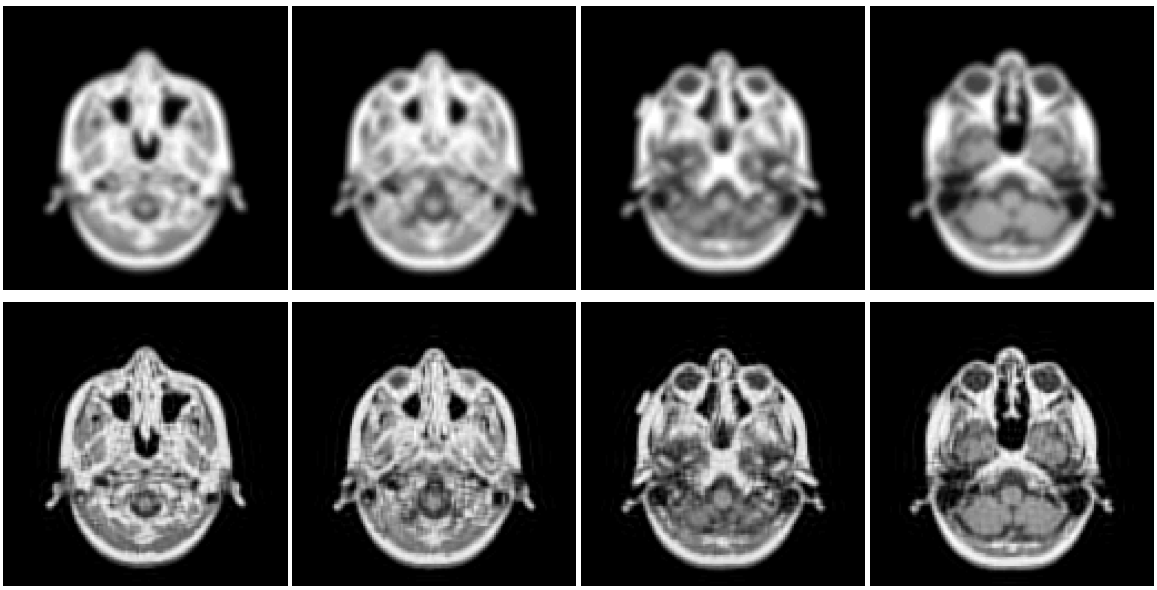

Figure 5. *Image sequence deblurring. Row* 1*: blurry images; row* 2*: recovered images. Overall runtime is* 72.90 *seconds and the entire relative error between the ground truth and recovered sequences is* 13.90%*.*

framework can be adapted to other types of tensor products or tensors of order higher than three.

Acknowledgment. The authors would like to thank the anonymous reviewers for improving the clarity and presentation of the paper.

REFERENCES

[1] M. A. Abidi, A. V. Gribok, and J. Paik, *Optimization Techniques in Computer Vision*, Springer, New York, 2016.

[2] A. Agaskar, C. Wang, and Y. M. Lu, *Randomized Kaczmarz algorithms: Exact MSE analysis and optimal sampling probabilities*, in Proceedings of the 2014 IEEE Global Conference on Signal and Information Processing, IEEE, 2014, pp. 389–393.

[3] A. H. Andersen and A. C. Kak, *Simultaneous algebraic reconstruction technique (SART): A superior implementation of the ART algorithm*, Ultrasonic Imaging, 6 (1984), pp. 81–94.

[4] Z.-Z. Bai and W.-T. Wu, *On greedy randomized Kaczmarz method for solving large sparse linear systems*, SIAM J. Sci. Comput., 40 (2018), pp. A592–A606.

[5] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, CMS Books Math. 408, Springer, New York, 2011.

[6] S. Boyd, N. Parikh, and E. Chu, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*, Now Publishers, Boston, 2011.

[7] A. Buades, B. Coll, and J.-M. Morel, *Non-local means denoising*, IPOL J. Image Process. Online, 1 (2011), pp. 208–212.

[8] J.-F. Cai, E. J. Candès, and Z. Shen, *A singular value thresholding algorithm for matrix completion*, SIAM J. Optim., 20 (2010), pp. 1956–1982.

[9] J.-F. Cai, S. Osher, and Z. Shen, *Convergence of the linearized Bregman iteration for ℓ_1-norm minimization*, Math. Comp., 78 (2009), pp. 2127–2136.

[10] Y. Censor and S. A. Zenios, *Parallel Optimization: Theory, Algorithms, and Applications*, Oxford University Press, New York, 1997.
[11] S. H. Chan, R. Khoshabeh, K. B. Gibson, P. E. Gill, and T. Q. Nguyen, *An augmented Lagrangian method for total variation video restoration*, IEEE Trans. Image Processing, 20 (2011), pp. 3097–3111.
[12] S. H. Chan, X. Wang, and O. A. Elgendy, *Plug-and-play ADMM for image restoration: Fixed-point convergence and applications*, IEEE Trans. Comput. Imaging, 3 (2016), pp. 84–98.
[13] X. Chen and A. M. Powell, *Almost sure convergence of the Kaczmarz algorithm with random measurements*, J. Fourier Anal. Appl., 18 (2012), pp. 1195–1214.
[14] X. Chen and A. M. Powell, *Randomized subspace actions and fusion frames*, Constr. Approx., 43 (2016), pp. 103–134.
[15] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, *Image denoising with block-matching and 3D filtering*, in Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning, Proc. SPIE, 6064, International Society for Optics and Photonics, 2006, 606414.
[16] L. Dai, M. Soltanalian, and K. Pelckmans, *On the randomized Kaczmarz algorithm*, IEEE Signal Process. Lett., 21 (2013), pp. 330–333.
[17] N. Durgin, R. Grotheer, C. Huang, S. Li, A. Ma, D. Needell, and J. Qin, *Sparse Randomized Kaczmarz for Support Recovery of Jointly Sparse Corrupted Multiple Measurement Vectors*, in Research in Data Science, Springer, New York, 2019, pp. 1–14.
[18] Y. C. Eldar and D. Needell, *Acceleration of randomized Kaczmarz method via the Johnson–Lindenstrauss lemma*, Numer. Algorithms, 58 (2011), pp. 163–177.
[19] H. Fan, Y. Chen, Y. Guo, H. Zhang, and G. Kuang, *Hyperspectral image restoration using low-rank tensor recovery*, IEEE J. Selected Topics Applied Earth Observations Remote Sensing, 10 (2017), pp. 4589–4604.
[20] P. Getreuer, *Total variation inpainting using split Bregman*, IPOL J. Image Process. Online, 2 (2012), pp. 147–157.
[21] R. Gordon, R. Bender, and G. T. Herman, *Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography*, J. Theoret. Biol., 29 (1970), pp. 471–481.
[22] G. T. Herman and L. B. Meyer, *Algebraic reconstruction techniques can be made computationally efficient (positron emission tomography application)*, IEEE Trans. Medical Imaging, 12 (1993), pp. 600–609.
[23] H. Jeong and C. S. Güntürk, *Convergence of the Randomized Kaczmarz Method for Phase Retrieval*, preprint, arXiv:1706.10291, 2017.
[24] S. Kaczmarz, *Angenaherte auflosung von systemen linearer glei-chungen*, Bull. Int. Acad. Pol. Sic. Let., Cl. Sci. Math. Nat., 35 (1937), pp. 355–357.
[25] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover, *Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 148–172.
[26] M. E. Kilmer and C. D. Martin, *Factorization strategies for third-order tensors*, Linear Algebra Appl., 435 (2011), pp. 641–658.
[27] M. E. Kilmer, C. D. Martin, and L. Perrone, *A Third-Order Generalization of the Matrix SVD as a Product of Third-Order Tensors*, Tech. rep. TR-2008-4, Department of Computer Science, Tufts University, 2008.
[28] S. Kindermann and A. Leitao, *Convergence rates for Kaczmarz-type regularization methods*, Inverse Problems Imaging, 8 (2014).
[29] M.-J. Lai and W. Yin, *Augmented ℓ_1 and nuclear-norm models with a globally linearly convergent algorithm*, SIAM J. Imaging Sci., 6 (2013), pp. 1059–1091.
[30] T. Li, T.-J. Kao, D. Isaacson, J. C. Newell, and G. J. Saulnier, *Adaptive Kaczmarz method for image reconstruction in electrical impedance tomography*, Physiological Measurement, 34 (2013), p. 595.
[31] J. Liu, S. J. Wright, and S. Sridhar, *An Asynchronous Parallel Randomized Kaczmarz Algorithm*, preprint, arXiv:1401.4780, 2014.
[32] D. A. Lorenz, F. Schöpfer, and S. Wenger, *The linearized Bregman method via split feasibility problems: Analysis and generalizations*, SIAM J. Imaging Sci., 7 (2014), pp. 1237–1262.

[33] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, *Tensor robust principal component analysis with a new tensor nuclear norm*, IEEE Trans. Pattern Anal. Machine Intelligence, 42 (2019), pp. 925–938.
[34] A. Ma and D. Molitor, *Randomized Kaczmarz for Tensor Linear Systems*, preprint, arXiv:2006.01246, 2020.
[35] A. Ma, D. Needell, and A. Ramdas, *Convergence properties of the randomized extended Gauss–Seidel and Kaczmarz methods*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 1590–1604.
[36] D. Needell, *Randomized Kaczmarz solver for noisy linear systems*, BIT, 50 (2010), pp. 395–403.
[37] D. Needell, N. Srebro, and R. Ward, *Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm*, Math. Program., 155 (2016), pp. 549–573.
[38] D. Needell and J. A. Tropp, *Paved with good intentions: Analysis of a randomized block Kaczmarz method*, Linear Algebra Appl., 441 (2014), pp. 199–221.
[39] D. Needell, R. Zhao, and A. Zouzias, *Randomized block Kaczmarz method with projection for solving least squares*, Linear Algebra Appl., 484 (2015), pp. 322–343.
[40] J. E. Peterson, B. N. Paulsson, and T. V. McEvilly, *Applications of algebraic reconstruction techniques to crosshole seismic data*, Geophysics, 50 (1985), pp. 1566–1580.
[41] S. Petra, *Randomized sparse block Kaczmarz as randomized dual block-coordinate descent*, An. Ştiinţ. Univ. "Ovidius" Constanţa Ser. Mat., 23 (2015), pp. 129–149.
[42] C. Popa and R. Zdunek, *Kaczmarz extended algorithm for tomographic image reconstruction from limited-data*, Math. Comput. Simul., 65 (2004), pp. 579–598.
[43] R. T. Rockafellar, *Convex Analysis*, Princeton Math. Ser. 28, Princeton University Press, Princeton, NJ, 1970.
[44] F. Schöpfer, *Linear convergence of descent methods for the unconstrained minimization of restricted strongly convex functions*, SIAM J. Optim., 26 (2016), pp. 1883–1911.
[45] F. Schöpfer and D. A. Lorenz, *Linear convergence of the randomized sparse Kaczmarz method*, Math. Program., 173 (2019), pp. 509–536.
[46] O. Semerci, N. Hao, M. E. Kilmer, and E. L. Miller, *Tensor-based formulation and nuclear norm regularization for multienergy computed tomography*, IEEE Trans. Image Process., 23 (2014), pp. 1678–1693.
[47] J. Shen and T. F. Chan, *Mathematical models for local nontexture inpaintings*, SIAM J. Appl. Math., 62 (2002), pp. 1019–1043.
[48] T. Strohmer and R. Vershynin, *A randomized Kaczmarz algorithm with exponential convergence*, J. Fourier Anal. Appl., 15 (2009), 262.
[49] H. Tan, G. Feng, J. Feng, W. Wang, Y.-J. Zhang, and F. Li, *A tensor-based method for missing traffic data completion*, Transportation Research Part C Emerging Technologies, 28 (2013), pp. 15–27.
[50] Y. S. Tan and R. Vershynin, *Phase retrieval via randomized Kaczmarz: Theoretical guarantees*, Inform. Inference, 8 (2018), pp. 97–123.
[51] S. J. Wright, *Coordinate descent algorithms*, Math. Program., 151 (2015), pp. 3–34.
[52] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, *Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing*, SIAM J. Imaging Sci., 1 (2008), pp. 143–168.
[53] H. Zhou, L. Li, and H. Zhu, *Tensor regression with applications in neuroimaging data analysis*, J. Amer. Statist. Assoc., 108 (2013), pp. 540–552.
[54] A. Zouzias and N. M. Freris, *Randomized extended Kaczmarz for solving least squares*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 773–793.