entropy

MDPI

Article

Rank-Adaptive Tensor Completion Based on Tucker Decomposition

Siqi Liu, Xiaoyu Shi and Qifeng Liao *

School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China
* Correspondence: liaoqf@shanghaitech.edu.cn

Abstract: Tensor completion is a fundamental tool to estimate unknown information from observed data, which is widely used in many areas, including image and video recovery, traffic data completion and the multi-input multi-output problems in information theory. Based on Tucker decomposition, this paper proposes a new algorithm to complete tensors with missing data. In decomposition-based tensor completion methods, underestimation or overestimation of tensor ranks can lead to inaccurate results. To tackle this problem, we design an alternative iterating method that breaks the original problem into several matrix completion subproblems and adaptively adjusts the multilinear rank of the model during optimization procedures. Through numerical experiments on synthetic data and authentic images, we show that the proposed method can effectively estimate the tensor ranks and predict the missing entries.

Keywords: tensor completion; Tucker decomposition; HOOI algorithm; rank-adaptive methods; SVT algorithm

check for updates

Citation: Liu, S.; Shi, X.; Liao, Q. Rank-Adaptive Tensor Completion Based on Tucker Decomposition. *Entropy* **2023**, *25*, 225. https://doi.org/10.3390/e25020225

Academic Editors: Yanlin Geng, Yinfei Xu and Min Zhu

Received: 24 December 2022
Revised: 21 January 2023
Accepted: 21 January 2023
Published: 24 January 2023

1. Introduction

Tensors, as a higher-order generalization of vectors and matrices, can preserve the original structure and the latent characteristics of multi-dimensional data, such as images and videos. To store and process these data, vectorization or matricization may break the adjacency relation and lead to the loss of crucial information in practical applications. Therefore, tensor analysis of data has gathered increasing attention due to its better performance in finding hidden structures and capturing potential features. Due to data missing from a transmission or insufficient collection, sometimes we have to deal with incomplete data, which increases the operation cost and the difficulty of multi-dimensional data analysis. Therefore, tensor completion, also referred to as tensor recovery, plays a key role in processing and analyzing incomplete multi-dimensional data. In tensor completion, the prediction of unknown data from a few observed entries is achieved based on the correlation between different parts of data or, simply, the hidden low-rank structure. Many real-world data lie in a latent low dimensional space since they may only relate to a small number of contributions; thus, we can use tensor completion to infer the missing data. It has already been widely applied in many scientific fields, including surrogate construction [1], image and video recovery [2–4], traffic data completion [5] and the MIMO (multi-input multi-output) problems in information theory [6].

The tensor completion methods can be roughly classified into model-based and non-model-based ones. Extending the idea from existing matrix completion algorithms [7,8], non-model-based methods directly formulate the tensor completion task as a rank minimization problem subject to linear constraints, and then relax it to a nuclear norm minimization problem, such as LRTC (low rank tensor completion) [9]. However, model-based methods achieve the goal differently. Based on a decomposition model with a given rank, these methods minimize the error on the known entries by adjusting the model parameters, such as CP-Wopt (CP-weighted optimization) [10], Tucker–Wopt (Tucker-weighted optimization) [5], M^{2}SA (multilinear subspace analysis with missing values) [11] and so on.

The computation cost of non-model-based methods is closely related to the size of the tensor, leaving little room for reducing the time complexity. On the contrary, this can be more flexibly controlled in model-based methods by altering the model size. Nonetheless, there still exists a big challenge when choosing the model's rank. Underestimation of the tensor rank makes it harder to reach our desired accuracy, whereas overestimation may lead to a poor generalization on the unobserved part of entries. Therefore, rank-adaptive methods are needed and worth studying. In [1], a greedy scheme is designed to find the fittest rank by gradually increasing the number of factors in the CP decomposition model. In [12], from an overestimated rank, a systematic approach to reducing the CP ranks to optimal ones is developed.

Focusing on tensor completion based on Tucker decomposition, we propose a novel rank-adaptive tensor completion method and verify its efficiency through experiments on synthetic and real practical data. The rest of the paper is organized as follows. Section 2 formulates the problem setting. Section 3 presents Tucker decomposition and the related works. We propose our rank-adaptive tensor completion based on the Tucker decomposition (RATC-TD) approach in Section 4 and verify it through numerical experiments in Section 5. Finally, we conclude our work in Section 6.

2. Notations and the Tensor Completion Problem

Before summarizing the related work and presenting our algorithm, this section first introduces some basic definitions that will be used in the following sections, and then states our problem settings.

2.1. Notations

We follow the notations in [13]. Let lowercase letters denote scalars, e.g., a; let lowercase bold letters denote vectors, e.g., $\mathbf{a}$; let bold capital letter denote matrices, e.g., $\boldsymbol{U}$; let capital fraktur letter denote high-order tensors, e.g., $\mathcal{T}$. The ith element of a vector $\mathbf{a}$ is denoted as a_i; the element of matrix $\boldsymbol{U}$ with index (i,j) is denoted by u_{ij}; the element of N-way tensor $\mathcal{T}$ with index $(i_1, i_2, \cdots, i_N)$ is denoted by $t_{i_1 i_2 \cdots i_N}$. A *tensor* can be viewed as a multi-dimensional array, and the number of dimensions is generally called the *order* of the tensor. For example, matrices are second-order tensors, while vectors are first-order tensors. A *mode-n fiber* is a vector extracted from a tensor by fixing all indices except the nth one. Consider a matrix as an example; mode-1 fibers refer to the columns, and mode-2 fibers refer to the rows. The *mode-n unfolding* of a tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is to rearrange the mode-n fibers as the columns of the resulting matrix, denoted as $\boldsymbol{T}_{(n)} \in \mathbb{R}^{I_n \times I_1 \cdots I_{n-1} I_{n+1} \cdots I_N}$. Naturally, the reverse of *unfolding*, the process of rearranging $\boldsymbol{T}_{(n)} \in \mathbb{R}^{I_n \times I_1 \cdots I_{n-1} I_{n+1} \cdots I_N}$ into $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, is called *folding*.

The *n-mode matrix product* is the multiplication of a tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with a matrix $\boldsymbol{U} \in \mathbb{R}^{J \times I_n}$, denoted as $\mathcal{X} = \mathcal{T} \times_n \boldsymbol{U}$, where $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N}$. It can be regarded as multiplying each mode-n fiber of $\mathcal{T}$ with $\boldsymbol{U}$. The matricized form is given by

$$\boldsymbol{X}_{(n)} = \boldsymbol{U}\boldsymbol{T}_{(n)}, \tag{1}$$

and the element-wise operations can be obtained by

$$x_{i_1 \dots i_{n-1} j i_{n+1} \dots i_N} = \sum_{i_n=1}^{I_n} t_{i_1 \dots i_{n-1} i_n i_{n+1} \dots i_N} u_{j i_n}. \tag{2}$$

The *norm* of a tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is a higher-order analog of the matrix Frobenius norm, defined by

$$\|\mathcal{T}\|_F = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} t^2_{i_1 i_2 \cdots i_N}}. \tag{3}$$

The *inner product* of two same sized tensors $\mathcal{T}, \mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is defined by

$$\langle \mathcal{T}, \mathcal{X} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} t_{i_1 i_2 \cdots i_N} x_{i_1 i_2 \cdots i_N}. \tag{4}$$

2.2. The Tensor Completion Problem

Tensor completion is the problem of predicting missing entries through a few sampled entries of a tensor based on the assumption that it has a low-rank structure. Supposing we have a partially observed tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, the set of indices where entries are observed is denoted as Ω. Let $\mathcal{P}_\Omega$ be an operator on $\mathbb{R}^{I_1 \times \cdots \times I_N}$, such that $\mathcal{P}_\Omega(\mathcal{X})$ is equal to $\mathcal{X}$ on Ω and is equal to 0 otherwise. The low-rank tensor completion problem can be formulated as

$$\min_{\mathcal{X}} \text{rank } (\mathcal{X}) \quad s.t.\ \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{T}). \tag{5}$$

Although the formulation seems easy, how to measure the rank of a tensor is still an open question. In previous research, the CP rank [1,12] and the Tucker rank [9,14,15] were mostly considered. After the tensor-train (TT) decomposition was proposed in [16], tensor completion based on the TT rank was studied in [3,17,18]. The low-tubal rank tensor completion problem was studied in [19–21]. In addition, the combination of the CP and the Tucker ranks was investigated in [22].

Since matrices are second-order tensors, matrix completion can be viewed as a special case of tensor completion. Using the matrix nuclear norm to approximate the rank of a matrix is studied for matrix completion in [23]. Liu et al. [9] generalized this to the tensor scheme by defining the nuclear norm of tensors and proposing the following low-rank tensor completion problem:

$$\min_{\mathcal{X}} \|\mathcal{X}\|_* \quad s.t.\ \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{T}). \tag{6}$$

The nuclear norm of a tensor is defined by $\|\mathcal{X}\|_* := \sum_{n=1}^N \|\boldsymbol{X}_{(n)}\|_*$, where $\|\boldsymbol{X}_{(n)}\|_*$ is the nuclear norm of the matrix $\boldsymbol{X}_{(n)}$. The nuclear norm can be computed by the sum of the singular values. However, the computation involves SVD (singular value decomposition) of the unfolding matrix $\boldsymbol{T}_{(n)}$ in each iteration, which is expensive when the size of $\mathcal{T}$ is large. To control the computation cost, this problem is reformulated based on tensor decomposition [5,11]. Particularly, the low-rank tensor completion problem based on Tucker decomposition can be written as

$$\begin{gathered} \min_{\mathcal{G}, \boldsymbol{A}_1 \cdots, \boldsymbol{A}_N} \|\mathcal{T} - \mathcal{X}\|_\Omega^2 \\ s.t.\ \mathcal{X} = \mathcal{G} \times_1 \boldsymbol{A}_1 \cdots \times_N \boldsymbol{A}_N, \quad \boldsymbol{A}_n^T \boldsymbol{A}_n = I^{R_n \times R_n}, \ n = 1, \cdots, N, \end{gathered} \tag{7}$$

where $\mathcal{X}$ is the completed goal tensor, $(R_1, \cdots, R_N)$ is a predefined multilinear rank, $\boldsymbol{A}_n \in \mathbb{R}^{I_n \times R_n}$ and $\| \cdot \|_\Omega$ denote the norm on the observation elements, i.e., $\|\mathcal{P}_\Omega(\cdot)\|_F$. Methods are actively developed to solve this problem, including M^{2}SA [11], gHOI (generalized higher-order orthogonal iteration) [24] and Tucker–Wopt [5]. However, these algorithms typically require the multilinear ranks given a priori, both underestimating and overestimating the ranks can result in inaccurate completion results. To handle this issue, we in this work propose a rank-adaptive tensor completion approach without requiring the multilinear ranks given a priori.

3. The Tucker Decomposition and Its Computation

We review Tucker decomposition and its related algorithms in this section.

3.1. Tucker Decomposition

Tucker decomposition was first introduced by Tucker [25]. For a Nth-order tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, its Tucker decomposition has the form

$$\mathcal{T} = \mathcal{G} \times_1 \boldsymbol{A}_1 \times_2 \boldsymbol{A}_2 \cdots \times_N \boldsymbol{A}_N, \tag{8}$$

where $\mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_N}$ is a core tensor and $\boldsymbol{A}_n \in \mathbb{R}^{I_n \times R_n}$, $n = 1, 2, \cdots, N$, are factor matrices. For simplicity, we assume all the factor matrices have orthogonal columns. R_n is the rank of the mode-n unfolding matrix $\boldsymbol{T}_{(n)}$. As described in [13], Tucker decomposition can be viewed as a higher-order generalization of PCA (principal component analysis) in some sense. In Tucker decomposition, the columns of the factor matrix represent the components in each mode, and each element of the core tensor characterizes the level of interaction between the components in different modes.

The tuple $(R_1, \cdots, R_N)$ is called Tucker rank or multilinear rank. However, practically, we prefer to use the truncated version of Tucker decomposition, where we set $R_n < \text{rank}(\boldsymbol{T}_{(n)})$ for one or more n. Given the truncation $(R_1, \cdots, R_N)$, the approximation of the truncated Tucker decomposition of a tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ can be described as

$$\begin{aligned} &\min_{\mathcal{G}, \boldsymbol{A}_1 \cdots, \boldsymbol{A}_N} \|\mathcal{T} - \mathcal{X}\|_F^2 \\ &s.t.\ \mathcal{X} = \mathcal{G} \times_1 \boldsymbol{A}_1 \cdots \times_N \boldsymbol{A}_N, \quad \boldsymbol{A}_n^T \boldsymbol{A}_n = I^{R_n \times R_n}, \ n = 1, \cdots, N, \end{aligned} \tag{9}$$

where $\mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_N}$, $\boldsymbol{A}_n \in \mathbb{R}^{I_n \times R_n}$ is the nth factor matrix, and $I^{R_n \times R_n}$ is an identity matrix of size $R_n \times R_n$.

3.2. The Higher-Order Orthogonal Iteration (HOOI) Algorithm

There are several approaches to computing the truncated Tucker decomposition of a tensor. One of the most popular methods is the higher-order orthogonal iteration (HOOI) algorithm [26], also referred to as the alternative least square (ALS) algorithm for Tucker decomposition. It is an alternative iterating method, where we fix all except one factor matrix each time, and then minimize the objective function in (9). Specifically, given initial guesses $\{\boldsymbol{A}_n^{(0)} : n = 1, \cdots, N\}$, at the kth iteration, for each $n = 1, \cdots, N$, we fix all the factor matrices except $\boldsymbol{A}_n$, and then find the optimal solution to the subproblem

$$\begin{aligned} &\min_{\mathcal{G}, \boldsymbol{A}_n^{(k)}} \|\mathcal{T} - \mathcal{X}\|_F^2 \\ &s.t.\ \mathcal{X} = \mathcal{G} \times_1 \boldsymbol{A}_1^{(k)} \cdots \times_n \boldsymbol{A}_n^{(k)} \times_{n+1} \boldsymbol{A}_{n+1}^{(k-1)} \cdots \times_N \boldsymbol{A}_N^{(k-1)}, \quad (\boldsymbol{A}_n^{(k)})^T \boldsymbol{A}_n^{(k)} = I^{R_n \times R_n}, \end{aligned} \tag{10}$$

where $\mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_N}$, $\boldsymbol{A}_n^{(k)} \in \mathbb{R}^{I_n \times R_n}$. Denoting $\mathcal{B} := \mathcal{G} \times_n \boldsymbol{A}_n^{(k)}$, this optimization problem can be simplified as

$$\begin{aligned} &\min_{\mathcal{B}} \|\mathcal{T} - \mathcal{X}\|_F^2 \\ &s.t.\ \mathcal{X} = \mathcal{B} \times_1 \boldsymbol{A}_1^{(k)} \cdots \times_{n-1} \boldsymbol{A}_{n-1}^{(k)} \times_{n+1} \boldsymbol{A}_{n+1}^{(k-1)} \cdots \times_N \boldsymbol{A}_N^{(k-1)}, \quad \text{rank}(\boldsymbol{B}_{(n)}) \leq R_n, \end{aligned} \tag{11}$$

where $\mathcal{B} \in \mathbb{R}^{R_1 \times \cdots R_{n-1} \times I_n \times R_{n+1} \cdots R_N}$ and $\boldsymbol{B}_{(n)}$ is the mode-n unfolding matrix form of $\mathcal{B}$. It can be considered as a constrained least squares problem [27], and thus the solution to (10) can be easily obtained by the following steps:

1. Compute $\mathcal{B}^* = \mathcal{T} \times_1 \left(\boldsymbol{A}_1^{(k)}\right)^T \cdots \times_{n-1} \left(\boldsymbol{A}_{n-1}^{(k)}\right)^T \times_{n+1} \left(\boldsymbol{A}_{n+1}^{(k-1)}\right)^T \cdots \times_N \left(\boldsymbol{A}_N^{(k-1)}\right)^T$;
2. Unfold $\mathcal{B}^*$ at the nth mode to obtain $\boldsymbol{B}_{(n)}^*$, then perform a truncated rank-R_n singular value decomposition $(\boldsymbol{B}_{(n)}^*)_{[R]} = \boldsymbol{U\Sigma V}^T = \boldsymbol{B}_{(n)}$;
3. Compute $\boldsymbol{A}_n^{(k)} = \boldsymbol{U}$ and $\boldsymbol{G}_{(n)} = \boldsymbol{\Sigma V}^T$.

We iteratively solve these subproblems until a given stopping criterion is reached, i.e., the difference between the solutions at two adjacent iterations is small enough, or the value of the objective function decreases very slightly. The complete procedure of HOOI is presented in Algorithm 1.

Algorithm 1 The high-order orthogonal iteration (HOOI) algorithm [26]

```
Input: Tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and truncation $(R_1, R_2, \cdots, R_N)$.
Output: Core tensor $\mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_N}$, and factor matrices $\boldsymbol{A}_n \in \mathbb{R}^{I_n \times R_n}$ for $n = 1, 2, \cdots, N$.
Initialize $\boldsymbol{A}_n^{(0)} \in \mathbb{R}^{I_n \times R_n}$ for $n = 1, 2, \cdots, N$ using HOSVD.
$k \leftarrow 0$.
while not converge do
    $k \leftarrow k + 1$.
    for $n = 1, 2, \cdots, N$ do
        $\mathcal{B} \leftarrow \mathcal{T} \times_1 \left(\boldsymbol{A}_1^{(k)}\right)^T \cdots \times_{n-1} \left(\boldsymbol{A}_{n-1}^{(k)}\right)^T \times_{n+1} \left(\boldsymbol{A}_{n+1}^{(k-1)}\right)^T \cdots \times_N \left(\boldsymbol{A}_N^{(k-1)}\right)^T$.
        $\boldsymbol{B}_{(n)} \leftarrow$ mode-$n$ unfolding matrix of $\mathcal{B}$.
        $\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}^T \leftarrow$ truncated rank-$R_n$ SVD of $\boldsymbol{B}_{(n)}$.
        $\boldsymbol{A}_n^{(k)} \leftarrow \boldsymbol{U}$.
    end for
end while
$\mathcal{G} \leftarrow \boldsymbol{\Sigma} \boldsymbol{V}^T$ folding at mode-$n$.
```

3.3. The Rank-Adaptive HOOI Algorithm

HOOI requires the multilinear rank $(R_1, \cdots, R_N)$ given a priori, which is hard to be determined in practice. Instead of (9), Xiao and Yang [27] consider the following form of the low multilinear rank approximation problem:

$$\begin{aligned} &\min_{\mathcal{G}, \boldsymbol{A}_1 \cdots, \boldsymbol{A}_N} \ (R_1, R_2, \cdots, R_N) \quad s.t. \ \|\mathcal{X} - \mathcal{T}\|_F^2 < \epsilon \|\mathcal{T}\|_F^2, \\ &\mathcal{X} = \mathcal{G} \times_1 \boldsymbol{A}_1 \cdots \times_N \boldsymbol{A}_N, \quad \boldsymbol{A}_n^T \boldsymbol{A}_n = I^{R_n \times R_n}, \ n = 1, \cdots, N, \end{aligned} \tag{12}$$

where ϵ is a given tolerance. A rank-adaptive HOOI algorithm is proposed in [27], which adjusts the truncation R_n for dimension n in the HOOI iterations by

$$R_n^{(k)} = \underset{R}{\operatorname{argmin}} \, \|\boldsymbol{B}_{(n)} - (\boldsymbol{B}_{(n)})_{[R]}\|_F^2 \leq \|\mathcal{B}\|_F^2 - (1 - \epsilon)\|\mathcal{T}\|_F^2, \tag{13}$$

where $(\boldsymbol{B}_{(n)})_{[R]}$ is the best rank-R approximation of $\boldsymbol{B}_{(n)}$. For the full SVD of $\boldsymbol{B}_{(n)} = \boldsymbol{U\Sigma V}^T$, it can be calculated by $(\boldsymbol{B}_{(n)})_{[R]} = \boldsymbol{U}_{:,1:R} \boldsymbol{\Sigma}_{1:R,1:R} \boldsymbol{V}_{:,1:R}^T$. In [27], it is proven that (13) is a local optimal strategy for updating R_n, i.e., the optimal solution of the following problem:

$$\begin{aligned} &\min_{\mathcal{B}} \operatorname{rank}(\boldsymbol{B}_{(n)}) \quad s.t. \ \|\mathcal{X} - \mathcal{T}\|_F^2 < \epsilon \|\mathcal{T}\|_F^2, \\ &\mathcal{X} = \mathcal{B} \times_1 \boldsymbol{A}_1^{(k)} \cdots \times_{n-1} \boldsymbol{A}_{n-1}^{(k)} \times_{n+1} \boldsymbol{A}_{n+1}^{(k-1)} \cdots \times_N \boldsymbol{A}_N^{(k-1)}. \end{aligned} \tag{14}$$

Details of the above procedure are summarized in Algorithm 2.

Algorithm 2 The rank-adaptive HOOI algorithm [27]

```
**Input:** Tensor $\mathcal{T} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$, error tolerance $\epsilon$, initial guess of factor matrices $\boldsymbol{A}_n^{(0)} \in \mathbb{R}^{I_n\times R_n^{(0)}}$ for $n=1,2,\cdots,N$ and initial truncation $(R_1^{(0)},R_2^{(0)},\cdots,R_N^{(0)})$.
**Output:** Truncation $(R_1^{(k)},R_2^{(k)},\cdots,R_N^{(k)})$, core tensor $\mathcal{G} \in \mathbb{R}^{R_1^{(k)}\times\cdots\times R_N^{(k)}}$ and factor matrices $\boldsymbol{A}_n^{(k)} \in \mathbb{R}^{I_n\times R_n^{(k)}}$ for $n=1,2,\cdots,N$.
$\mathcal{G}^{(0)} \leftarrow \mathcal{T}\times_1 (\boldsymbol{A}_1^{(0)})^T\cdots\times_N(\boldsymbol{A}_N^{(0)})^T$.
$k \leftarrow 0$.
**while** not converge **do**
    $k \leftarrow k+1$.
    **for all** $n \in \{1,2,\cdots,N\}$ **do**
        $\mathcal{B} \leftarrow \mathcal{T}\times_1\left(\boldsymbol{A}_1^{(k)}\right)^T\cdots\times_{n-1}\left(\boldsymbol{A}_{n-1}^{(k)}\right)^T\times_{n+1}\left(\boldsymbol{A}_{n+1}^{(k-1)}\right)^T\cdots\times_N\left(\boldsymbol{A}_N^{(k-1)}\right)^T$.
        $\boldsymbol{B}_{(n)} \leftarrow$ mode-$n$ unfolding matrix of $\mathcal{B}$.
        $\boldsymbol{U},\boldsymbol{\Sigma},\boldsymbol{V}^T \leftarrow$ full SVD of $\boldsymbol{B}_{(n)}$.
        $R_n^{(k)} \leftarrow$ minimum $R$ such that $\sum_{r>R}\boldsymbol{\Sigma}_{r,r}^2 < \|\mathcal{B}\|_F^2-(1-\epsilon)\|\mathcal{T}\|_F^2$.
        $\boldsymbol{A}_n^{(k)} \leftarrow \boldsymbol{U}_{:,1:R_n^{(k)}}$.
    **end for**
    $\mathcal{G} \leftarrow \boldsymbol{\Sigma}_{1:R_n^{(k)},1:R_n^{(k)}}\boldsymbol{V}^T_{:,1:R_n^{(k)}}$ folding at mode-$n$.
**end while**
```

4. A Rank-Adaptive Tensor Recovery Scheme

This section proposes a rank-adaptive tensor completion scheme based on truncated Tucker decomposition (RATC-TD).

Analogous to the rank-adaptive HOOI algorithm, instead of (7), we consider a different form of the low-rank tensor completion problem:

$$\begin{gathered}\min_{\mathcal{G},\boldsymbol{A}_1\cdots,\boldsymbol{A}_N} \ (R_1,R_2,\cdots,R_N) \quad s.t.\ \mathcal{P}_\Omega(\mathcal{X})=\mathcal{P}_\Omega(\mathcal{T}),\\ \mathcal{X}=\mathcal{G}\times_1\boldsymbol{A}_1\cdots\times_N\boldsymbol{A}_N,\quad \boldsymbol{A}_n^T\boldsymbol{A}_n=I^{R_n\times R_n},\ n=1,\cdots,N.\end{gathered} \tag{15}$$

Note that if the data are noisy, the constraint must be relaxed to $\|\mathcal{X}-\mathcal{T}\|_\Omega^2<\epsilon\|\mathcal{T}\|_\Omega^2$, where ϵ is a given tolerance of the relative error on the observation part between the original tensor and the completed one.

Using an alternative optimization technique, problem (15) can be divided into N subproblems, which will be detailedly represented in Section 4.1. To address those subproblems, the singular value thresholding (SVT) algorithm [7] is introduced in Section 4.2. The entire algorithm is summarized in Section 4.3.

4.1. Alternative Optimization in Tensor Completion

We solve (15) by an alternative iteration method. For each $n=1,\cdots,N$, initialize $R_n^{(0)}=I_n$, and the initial guess $\boldsymbol{A}_n^{(0)}$ can be either randomly set or given by HOSVD (higher-order SVD) [13] of $\mathcal{P}_\Omega(\mathcal{T})$. At the kth iteration, for each $n=1,\cdots,N$, by fixing all the factor matrices except $\boldsymbol{A}_n$, we solve the subproblem

$$\begin{gathered}\min_{\mathcal{G},\boldsymbol{A}_n^{(k)}} \ (R_n^{(k)}) \quad s.t.\ \mathcal{P}_\Omega(\mathcal{X})=\mathcal{P}_\Omega(\mathcal{T}),\\ \mathcal{X}=\mathcal{G}\times_1\boldsymbol{A}_1^{(k)}\cdots\times_n\boldsymbol{A}_n^{(k)}\times_{n+1}\boldsymbol{A}_{n+1}^{(k-1)}\cdots\times_N\boldsymbol{A}_N^{(k-1)},\quad (\boldsymbol{A}_n^{(k)})^T\boldsymbol{A}_n^{(k)}=I^{R_n\times R_n}.\end{gathered} \tag{16}$$

Denoting $\mathcal{B}:=\mathcal{G}\times_n\boldsymbol{A}_n^{(k)}$, (16) is equivalent to

$$\begin{aligned} &\min_{\mathcal{B}} \operatorname{rank}(\boldsymbol{B}_{(n)}) \quad s.t.\ \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{T}), \\ \mathcal{X} = \mathcal{B} \times_1 \boldsymbol{A}_1^{(k)} &\cdots \times_{n-1} \boldsymbol{A}_{n-1}^{(k)} \times_{n+1} \boldsymbol{A}_{n+1}^{(k-1)} \cdots \times_N \boldsymbol{A}_N^{(k-1)}, \end{aligned} \tag{17}$$

where $\boldsymbol{B}_{(n)}$ is the mode-n unfolding matrix form of $\mathcal{B}$. However, this rank minimization problem is NP-hard. Its tightest convex relaxation is

$$\begin{aligned} &\min_{\mathcal{B}} \|\boldsymbol{B}_{(n)}\|_* \quad s.t.\ \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{T}), \\ \mathcal{X} = \mathcal{B} \times_1 \boldsymbol{A}_1^{(k)} &\cdots \times_{n-1} \boldsymbol{A}_{n-1}^{(k)} \times_{n+1} \boldsymbol{A}_{n+1}^{(k-1)} \cdots \times_N \boldsymbol{A}_N^{(k-1)}, \end{aligned} \tag{18}$$

where $\|\boldsymbol{B}_{(n)}\|_*$ is the nuclear norm of the matrix $\boldsymbol{B}_{(n)}$, which can be computed by the sum of the singular values. The matrix form of this problem can be written as [27]

$$\begin{aligned} &\min_{\boldsymbol{B}} \|\boldsymbol{B}\|_* \quad s.t.\ \boldsymbol{P}_\Omega(\boldsymbol{B}\boldsymbol{M}^T) = \boldsymbol{P}_\Omega(\boldsymbol{T}_{(n)}), \\ &\boldsymbol{M} = \boldsymbol{A}_N \cdots \otimes \boldsymbol{A}_{n+1} \otimes \boldsymbol{A}_{n-1} \cdots \otimes \boldsymbol{A}_1, \end{aligned} \tag{19}$$

where $\boldsymbol{T}_{(n)}$ is the mode-n unfolding matrix form of $\mathcal{T}$, and $\otimes$ is the Kronecker product. In this work, we use the singular value thresholding (SVT) algorithm [7] to obtain a proximity solution to (19).

4.2. Solving the Subproblems Using SVT

Following the notation used in [7], we first introduce the singular value thresholding operator. Consider the skinny singular value decomposition of a matrix $\boldsymbol{X} \in \mathbb{R}^{n_1 \times n_2}$ of rank r,

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T, \quad \boldsymbol{\Sigma} = \operatorname{diag}(\{\sigma_i\}_{1\le i\le r}), \tag{20}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are, respectively, $n_1 \times r$ and $n_2 \times r$ matrices with orthonormal columns, and the singular values $\sigma_i > 0$ for $i = 1, \cdots, r$. Given the shrinkage threshold $\tau > 0$, the singular value thresholding operator $\mathcal{D}_\tau$ is defined by

$$\mathcal{D}_\tau(\boldsymbol{X}) := \boldsymbol{U}\mathcal{D}_\tau(\boldsymbol{\Sigma})\boldsymbol{V}^T, \quad \mathcal{D}_\tau(\boldsymbol{\Sigma}) = \operatorname{diag}(\{\max(\sigma_i - \tau, 0)\}_{1\le i\le r}). \tag{21}$$

According to the deduction of [7], for each $\tau > 0$ and $\boldsymbol{Y} \in \mathbb{R}^{n_1 \times n_2}$, the singular value shrinkage operator (21) obeys

$$\mathcal{D}_\tau(\boldsymbol{Y}) = \operatorname*{argmin}_{\boldsymbol{X}} \left\{ \tau\|\boldsymbol{X}\|_* + \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{Y}\|_F^2 \right\}. \tag{22}$$

The SVT algorithm utilizes the singular value thresholding operator $\mathcal{D}_\tau$ and its property (22) to handle the subproblem (19). Consider the proximal problem for $\tau > 0$,

$$\begin{aligned} &\min_{\boldsymbol{B}} \tau\|\boldsymbol{B}\|_* + \frac{1}{2}\|\boldsymbol{B}\|_F^2 \\ &s.t.\ \boldsymbol{P}_\Omega(\boldsymbol{B}\boldsymbol{M}^T) = \boldsymbol{P}_\Omega(\boldsymbol{T}_{(n)}), \end{aligned} \tag{23}$$

the solution of which converges to that of (19) as $\tau \to \infty$. This optimization problem can be solved by a Lagrangian multiplier method known as the Uzawa's algorithm [28].

Given the Lagrangian of (23)

$$\mathcal{L}(\boldsymbol{B}, \boldsymbol{Y}) = \tau\|\boldsymbol{B}\|_* + \frac{1}{2}\|\boldsymbol{B}\|_F^2 + \langle \boldsymbol{Y}, \boldsymbol{P}_\Omega(\boldsymbol{T}_{(n)} - \boldsymbol{B}\boldsymbol{M}^T)\rangle, \tag{24}$$

where $\boldsymbol{Y}$ has the same size as $\boldsymbol{T}_{(n)}$, the optimal $\boldsymbol{B}^*$ and $\boldsymbol{Y}^*$ should satisfy

$$\mathcal{L}(\boldsymbol{B}^*, \boldsymbol{Y}^*) = \inf_{\boldsymbol{B}} \sup_{\boldsymbol{Y}} \mathcal{L}(\boldsymbol{B}, \boldsymbol{Y}) = \sup_{\boldsymbol{Y}} \inf_{\boldsymbol{B}} \mathcal{L}(\boldsymbol{B}, \boldsymbol{Y}). \tag{25}$$

Starting with $\boldsymbol{Y}^{(0)} = \boldsymbol{0}$, Uzawa's algorithm finds the saddle point $(\boldsymbol{B}^*, \boldsymbol{Y}^*)$ through an iterative procedure given by

$$\begin{cases} \boldsymbol{B}^{(k)} = \operatorname{argmin}_{\boldsymbol{B}} \mathcal{L}(\boldsymbol{B}, \boldsymbol{Y}^{(k-1)}) \\ \boldsymbol{Y}^{(k)} = \boldsymbol{Y}^{(k-1)} + \delta_k \boldsymbol{P}_{\Omega}(\boldsymbol{T}_{(n)} - \boldsymbol{B}^{(k)} \boldsymbol{M}^T), \end{cases} \tag{26}$$

where $\{\delta_k\}_{k \geq 1} > 0$ are scalar step sizes. The sequence $\{\boldsymbol{B}^{(k)}\}$ converges to the unique solution to (23). The update of $\boldsymbol{Y}$ is actually based on the gradient descent method if we note that

$$\partial_{\boldsymbol{Y}} \mathcal{L}(\boldsymbol{B}, \boldsymbol{Y}) = \boldsymbol{P}_{\Omega}(\boldsymbol{T}_{(n)} - \boldsymbol{B}\boldsymbol{M}^T). \tag{27}$$

Now, we have to compute the minimizer of (26). Observe that the factor matrices $\{\boldsymbol{A}_k : k = 1, \cdots, N\}$ have orthogonal columns. Based on the orthogonal invariance property of the Frobenius norm, we have

$$\|\mathcal{B}\|_F^2 = \|\mathcal{B} \times_1 \boldsymbol{A}_1^{(k)} \cdots \times_{n-1} \boldsymbol{A}_{n-1}^{(k)} \times_{n+1} \boldsymbol{A}_{n+1}^{(k-1)} \cdots \times_N \boldsymbol{A}_N^{(k-1)}\|_F^2, \tag{28}$$

which in the matrix form gives that

$$\|\boldsymbol{B}\|_F^2 = \|\boldsymbol{B}\boldsymbol{M}^T\|_F^2. \tag{29}$$

Utilizing this property,

$$\begin{aligned} &\operatorname{argmin}_{\boldsymbol{B}} \tau\|\boldsymbol{B}\|_* + \frac{1}{2}\|\boldsymbol{B}\|_F^2 + \langle \boldsymbol{Y}, \boldsymbol{P}_{\Omega}(\boldsymbol{T}_{(n)} - \boldsymbol{B}\boldsymbol{M}^T) \rangle \\ &= \operatorname{argmin}_{\boldsymbol{B}} \tau\|\boldsymbol{B}\|_* + \frac{1}{2}\|\boldsymbol{B}\boldsymbol{M}^T\|_F^2 + \langle \boldsymbol{Y}, \boldsymbol{P}_{\Omega}(\boldsymbol{T}_{(n)} - \boldsymbol{B}\boldsymbol{M}^T) \rangle \\ &= \operatorname{argmin}_{\boldsymbol{B}} \tau\|\boldsymbol{B}\|_* + \frac{1}{2}\|\boldsymbol{B}\boldsymbol{M}^T - \boldsymbol{P}_{\Omega}(\boldsymbol{Y})\|_F^2 \\ &= \operatorname{argmin}_{\boldsymbol{B}} \tau\|\boldsymbol{B}\|_* + \frac{1}{2}\|\boldsymbol{B} - \boldsymbol{P}_{\Omega}(\boldsymbol{Y})\boldsymbol{M}\|_F^2. \end{aligned} \tag{30}$$

According to (22), the optimal $\boldsymbol{B}^*$ is given by $\mathcal{D}_\tau(\boldsymbol{P}_{\Omega}(\boldsymbol{Y})\boldsymbol{M}) = \mathcal{D}_\tau(\boldsymbol{Y}\boldsymbol{M})$ since $\boldsymbol{P}_{\Omega}(\boldsymbol{Y}) = \boldsymbol{Y}$ for all $k \geq 0$. Therefore, Uzawa's algorithm finally takes the form

$$\begin{cases} \boldsymbol{B}^{(k)} = \mathcal{D}_\tau(\boldsymbol{Y}^{(k-1)}\boldsymbol{M}) \\ \boldsymbol{Y}^{(k)} = \boldsymbol{Y}^{(k-1)} + \delta_k \boldsymbol{P}_{\Omega}(\boldsymbol{T}_{(n)} - \boldsymbol{B}^{(k)} \boldsymbol{M}^T), \end{cases} \tag{31}$$

also referred to as the shrinkage iterations in SVT.

To obtain an approximation to the solution of (19), we choose a large enough τ and perform the iterations (31) until the stopping criteria $\|\boldsymbol{T} - \boldsymbol{B}^{(k)}\boldsymbol{M}^T\|_\Omega^2 < \epsilon\|\boldsymbol{T}\|_\Omega^2$ are reached, starting with $\boldsymbol{Y}^{(0)} = \boldsymbol{0} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$.

The overall process of solving the subproblem (18) is shown in Algorithm 3.

Algorithm 3 The SVT algorithm for solving (18)

```
Input: Set of observed indices $\Omega$, tensor with observed entries $\mathcal{P}_{\Omega}(\mathcal{T}) \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, fixed factor matrices $\boldsymbol{A}_m \in \mathbb{R}^{I_m \times R_m}$ for $m = \{1, 2, \cdots, N\}/n$, error tolerance $\epsilon$, shrinkage threshold $\tau$ and scalar step sizes $\{\delta_k\}_{k \geq 1}$.
Output: Optimized $\mathcal{B}$.
$\boldsymbol{P}_{\Omega}(\boldsymbol{T}) \leftarrow$ mode-$n$ unfolding matrix of $\mathcal{P}_{\Omega}(\mathcal{T})$.
$\boldsymbol{M} \leftarrow \boldsymbol{A}_N \cdots \otimes \boldsymbol{A}_{n+1} \otimes \boldsymbol{A}_{n-1} \cdots \otimes \boldsymbol{A}_1$.
Initialize $\boldsymbol{B}^{(0)} \leftarrow (\boldsymbol{P}_{\Omega}(\boldsymbol{T}))\boldsymbol{M}$,   $\boldsymbol{Y}^{(0)} \leftarrow \boldsymbol{0}$.
$k \leftarrow 0$.
while $\|\boldsymbol{T} - \boldsymbol{B}^{(k)}\boldsymbol{M}^T\|_\Omega^2 \geq \epsilon\|\boldsymbol{T}\|_\Omega^2$ do
    $k \leftarrow k + 1$.
    Update $\boldsymbol{B}^{(k)} \leftarrow \mathcal{D}_\tau(\boldsymbol{Y}^{(k-1)}\boldsymbol{M})$, where $\mathcal{D}_\tau$ is defined in (21).
    Update $\boldsymbol{Y}^{(k)} \leftarrow \boldsymbol{Y}^{(k-1)} + \delta_k \boldsymbol{P}_{\Omega}(\boldsymbol{T}_{(n)} - \boldsymbol{B}^{(k)}\boldsymbol{M}^T)$.
end while
$\mathcal{B} \leftarrow \boldsymbol{B}^{(k)}$ folding at mode-$n$.
```

4.3. The Rank-Adaptive Tensor Completion Algorithm

This section summarizes our proposed method, namely the rank-adaptive tensor completion algorithm. Based on Tucker decomposition, this algorithm can complete a tensor with only a few observed entries and adaptively estimate its multilinear rank.

Given a tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, only entries with indices in the set Ω are observed. Our goal is to predict the unobserved entries based on the premise that $\mathcal{T}$ has a low-rank structure. In this work, we consider the multilinear rank of the tensor. We solve this problem by finding a low multilinear rank tensor $\mathcal{X}$ whose entries on the observation part satisfy $\|\mathcal{X} - \mathcal{T}\|_\Omega^2 < \epsilon \|\mathcal{T}\|_\Omega^2$, where ϵ is a given tolerance. Here, we restate our problem setting (15)

$$\min_{\mathcal{G}, \boldsymbol{A}_1 \cdots, \boldsymbol{A}_N} \ (R_1, R_2, \cdots, R_N) \quad s.t.\ \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{T}),$$

$$\mathcal{X} = \mathcal{G} \times_1 \boldsymbol{A}_1 \cdots \times_N \boldsymbol{A}_N, \quad \boldsymbol{A}_n^T \boldsymbol{A}_n = I^{R_n \times R_n}\ , \ n = 1, \cdots, N.$$

Similar to HOOI, we solve (15) by alternative iterations. By fixing all the factor matrices except one, we obtain the rank minimization subproblem (17). Since it is NP-hard, we instead consider its convex relaxation form (18), i.e., minimizing the nuclear norm, which can be handled by the SVT algorithm. The solution can be computed using Algorithm 3. After optimized $\mathcal{B}$ in (18) is obtained, we update $R_n^{(k)} = \min(R_n^{(k-1)}, \ \mathrm{rank}(\boldsymbol{B}_{(n)}))$ to ensure that $R_n^{(k)}$ is monotone decreasing. We perform the iterations until some stopping criteria are satisfied, e.g., the estimated rank $(R_1, R_2, \cdots, R_N)$ remains unchanged, and the factor matrices have slight improvement, or the maximum number of iterations is reached. We present the detailed procedure of our proposed method in Algorithm 4.

Algorithm 4 The rank-adaptive tensor completion based on Tucker decomposition (RATC-TD) algorithm

Input: Set of observed indices Ω, tensor with observed entries $\mathcal{P}_\Omega(\mathcal{T}) \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and initial guess of factor matrices $\boldsymbol{A}_n^{(0)} \in \mathbb{R}^{I_n \times R_n^{(0)}}$ for $n = 1, 2, \cdots, N$.

Output: Estimated rank $(R_1^{(k)}, R_2^{(k)}, \cdots, R_N^{(k)})$, core tensor $\mathcal{G} \in \mathbb{R}^{R_1^{(k)} \times \cdots \times R_N^{(k)}}$ and factor matrices $\boldsymbol{A}_n^{(k)} \in \mathbb{R}^{I_n \times R_n^{(k)}}$ for $n = 1, 2, \cdots, N$.

```
$k \leftarrow 0$.
Initialize $(R_1^{(0)}, R_2^{(0)}, \cdots, R_N^{(0)}) \leftarrow (I_1, I_2, \cdots, I_N)$.
while not converge do
    for all $n = 1, 2, \cdots, N$ do
        $k \leftarrow k + 1$.
        $\mathcal{B} \leftarrow$ solution to (18) using Algorithm 3.
        $\boldsymbol{B}_{(n)} \leftarrow$ mode-$n$ unfolding matrix of $\mathcal{B}$.
        $\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}^T \leftarrow$ full-SVD of $\boldsymbol{B}_{(n)}$.
        $R_n^{(k)} \leftarrow \min(R_n^{(k-1)}, \ \mathrm{rank}(\boldsymbol{B}_{(n)}))$ .
        $\boldsymbol{A}_n^{(k)} \leftarrow \boldsymbol{U}_{:,1:R_n^{(k)}}$ .
    end for
    $\mathcal{G} \leftarrow \boldsymbol{\Sigma}_{1:R_n^{(k)},1:R_n^{(k)}} \boldsymbol{V}^T_{:,1:R_n^{(k)}}$ folding at mode-$n$.
end while
```

5. Numerical Experiments

In this section, numerical experiments are conducted to the effectiveness of our proposed rank-adaptive tensor completion based on Tucker decomposition (RATC-TD) algorithm.

5.1. Test Problem 1: The Recovery of Third-Order Tensors

Many tensor recovery algorithms in the Tucker scheme require a priori given the rank of the tensor, but our method does not require this. In this test problem, synthetic data are considered, and we set the random composition tensor in the same way as [9]. We consider the tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ and obtain tensors by multiplying a randomly generated kernel $\mathcal{G}$ of size $r_1 \times r_2 \times r_3$ with randomly generated factor matrices $\boldsymbol{A}_i$, where $\boldsymbol{A}_i \in \mathbb{R}^{I_i \times r_i}, i = 1, 2, 3$. So, the tensor $\mathcal{T}$ can be represented by $\mathcal{G} \times_1 \boldsymbol{A}_1 \times_2 \boldsymbol{A}_2 \times_3 \boldsymbol{A}_3$, and the number of entries of $\mathcal{T}$ is denoted by $|\mathcal{T}|$. By setting tensors in this way, we can obtain random tensors and ensure that the obtained tensors have a low-rank structure, which can be handled well by the classical low-rank tensor recovery methods.

In this experiment, we set the tensor size as $50 \times 50 \times 50$, the kernel size as $5 \times 5 \times 5$, and the kernel data are randomly generated from the interval $[0, 1]$. The factor matrices are of size 50×5, and the data of factor matrices are randomly generated from $[-0.5, 0.5]$. In order to show the robustness and reliability of our proposed algorithm, we add a small perturbation to the synthetic data. We add Gaussian noise with zero mean and standard deviation as 0.1 times the element mean and take the tensor with Gaussian noise as the ground truth data. As described above, our algorithm does not require the rank in advance, and the algorithm takes the size of the tensor as the initial tensor rank. With the continuous operation of the algorithm, the tucker rank is constantly reduced, and the exact rank can be approximated.

We compare the proposed algorithm with other Tucker-decomposition-based tensor completion algorithms (that require appropriate ranks given a priori), including gHOI [24], M^{2}SA [11] and Tucker–Wopt [5], and test the sample estimation error $error_{obs}$ and out-of-sample estimation error $error_{val}$ with different initial ranks at sampling rates r (the proportion of known data in the total data) of 0.05, 0.1 and 0.2, respectively. The size of the observed index set Ω is $r|\mathcal{T}|$, and each index in Ω is randomly generated through the uniform distribution. $\overline{\Omega}$ denotes the complementary set of Ω. The specific definitions of two kinds of errors are given by the following formulas:

$$error_{obs} = \frac{\|\boldsymbol{P}_{\Omega}(\mathcal{X} - \mathcal{T})\|_F^2}{\|\boldsymbol{P}_{\Omega}(\mathcal{T})\|_F^2}, \tag{32}$$

$$error_{val} = \frac{\|\boldsymbol{P}_{\overline{\Omega}}(\mathcal{X} - \mathcal{T})\|_F^2}{\|\boldsymbol{P}_{\overline{\Omega}}(\mathcal{T})\|_F^2}, \tag{33}$$

where $\mathcal{X}$ is the result obtained by completion algorithms.

To ensure the generalization of the proposed algorithms, we set the error tolerance ϵ on the observed data to 0.0025, which means we stop optimizing $\mathcal{B}$ with SVT when the relative error is less than ϵ or the maximum number of iterations is reached. When using SVT to optimize $\mathcal{B}$, we set the relative error $error_{SVT}$ in the following format:

$$error_{SVT} = \frac{\|\boldsymbol{P}_{\Omega}(\boldsymbol{T} - \boldsymbol{B}^{(k)}\boldsymbol{M}^T)\|_F^2}{\|\boldsymbol{P}_{\Omega}(\boldsymbol{T})\|_F^2}. \tag{34}$$

For the estimated tensor $\mathcal{X}$ obtained at each iteration in Algorithm 4, the relative error is assessed as (32) and (33), for other methods considered, the relative errors are also assessed using the same form. In addition to the relative error and the maximum number of iterations, Algorithm 4 also terminates if the difference of the relative errors obtained from the algorithm for two consecutive steps is less than a certain threshold η, i.e.,

$$\left| \frac{error_{iter}^{(k+1)} - error_{iter}^{(k)}}{error_{iter}^{(k)}} \right| < \eta, \tag{35}$$

where the $error_{iter}^{(k)}$ represents the relative error obtained at the kth iteration.

The initial rank is set to $R^{(0)} = [r, r, r],\ r = 5, 10, 15$. Tables 1 and 2 respectively show recovery error (32) and (33) of test problem 1. It can be seen that when the given initial rank does not match the ground truth data, gHOI, M^{2}SA and Tucker–Wopt have large errors, while our RATC-TD has small errors. We emphasize that our proposed method does not require a given tensor rank; the initial rank of our proposed algorithm is set to the size of the tensor we aim to complete, i.e., $R^{(0)} = \text{size}(\mathcal{T})$.

Table 1. Relative error (32) comparison with different initial rank on observed data, test problem 1.

Sampling Ratio	Initial Rank	M^{2}SA	gHOI	Tucker–Wopt	RATC-TD
0.05	5, 5, 5	0.0354	0.0431	0.0354	0.0354
0.05	10, 10, 10	0.0322	0.0404	0.0284	0.0354
0.05	15, 15, 15	0.0270	0.0362	0.1128	0.0354
0.1	5, 5, 5	0.0264	0.0278	0.0264	0.0264
0.1	10, 10, 10	0.0250	0.0267	0.0233	0.0264
0.1	15, 15, 15	0.0229	0.0229	0.0199	0.0264
0.2	5, 5, 5	0.0235	0.0236	0.0235	0.0235
0.2	10, 10, 10	0.0230	0.0229	0.0225	0.0235
0.2	15, 15, 15	0.0223	0.0215	0.0208	0.0235

Table 2. Relative error (33) comparison with different initial rank on unknown data, test problem 1.

Sampling Ratio	Initial Rank	M^{2}SA	gHOI	Tucker–Wopt	RATC-TD
0.05	5, 5, 5	0.0187	0.0518	0.0187	0.0186
0.05	10, 10, 10	0.3438	0.4192	0.0429	0.0186
0.05	15, 15, 15	0.6262	0.6143	1.4444	0.0186
0.1	5, 5, 5	0.0153	0.0231	0.0153	0.0153
0.1	10, 10, 10	0.2300	0.3102	0.0241	0.0153
0.1	15, 15, 15	0.4935	0.4521	0.0610	0.0153
0.2	5, 5, 5	0.0145	0.0162	0.0145	0.0145
0.2	10, 10, 10	0.2480	0.2390	0.0181	0.0145
0.2	15, 15, 15	0.4872	0.2800	0.0222	0.0145

In this experiment, we use our RATC-TD algorithm to estimate the tensor ranks. We next use the M^{2}SA method to improve the estimated results. Our algorithm can be considered an initial step for other algorithms, and the effects of tensor completion are also sound when only our method is used for optimization. The recovery results without using M^{2}SA are shown in Table 3. As seen from Table 2, since our method was used for optimization in advance, compared with the results obtained by simply using the M^{2}SA method with a given rank, it gives a better recovery effect. Notably, our proposed algorithm can also stably complete the tensor under a small sampling rate and does not need to give an appropriate rank in advance.

Table 3. Numerical results only using our proposed algorithm RATC-TD, test problem 1.

Sampling Ratio	$error_{obs}$	$error_{val}$
0.05	0.0500	0.0410
0.1	0.0489	0.0523
0.2	0.0495	0.0443

5.2. Test Problem 2: The Recovery of Fourth-Order Tensors

Similarly to test problem 1, following the procedures in [9], we in this test problem generate the object tensor $\mathcal{T} \in \mathbb{R}^{30\times30\times30\times30}$ by multiplying a randomly generated kernel $\mathcal{G} \in \mathbb{R}^{5\times5\times5\times5}$ with randomly generated factor matrices of size $\mathbb{R}^{30\times5}$. The ground truth data is defined by $\mathcal{T}$ with Gaussian noise added (the same setting as in test problem 1). For gHOI, M^{2}SA, Tucker–Wopt and our proposed RATC-TD, different initial ranks are tested, and the sampling rate r is set to 0.1. The corresponding number of entries in the observed data set Ω is $r|\mathcal{T}|$, and the index of each entry in Ω is randomly generated through the uniform distribution. The errors for this test problem are shown in Table 4, where $error_{obs}$ is the error on the observed data set (32) and $error_{val}$ is the error on the validation set (33). It can be seen that our RATC-TD method performs well for this test problem.

Table 4. The results of tensor completion quality of methods in the case of sampling rate $r = 0.1$, test problem 2.

Types of Error	Initial Rank	M^{2}SA	gHOI	Tucker–Wopt	RATC-TD
$error_{obs}$	5, 5, 5, 5	0.0300	0.0342	0.0300	0.0300
$error_{val}$	5, 5, 5, 5	0.0147	0.0251	0.0147	0.0147
$error_{obs}$	15, 15, 15, 15	0.0258	0.0255	0.1552	0.0300
$error_{val}$	15, 15, 15, 15	0.7130	0.4807	6.6025	0.0147

5.3. Test Problem 3: The Recovery of Real Missing Pictures

In this test problem, real practical data are considered, and the performance of our RATC-TD is compared with that of gHOI and M^{2}SA. The initial complete images considered are shown in Figure 1. The data format of each image can be regarded as a third-order tensor. Each image here is stored as a tensor with size $256 \times 256 \times 3$, where the third dimension is 3, representing three color channels of red, green, and blue. The ground truth data $\mathcal{T}$ for this test problem are the images in Figure 1 (details are as follows).

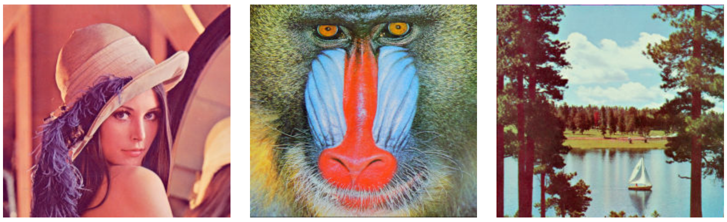

(**a**) girl (**b**) baboon (**c**) boat

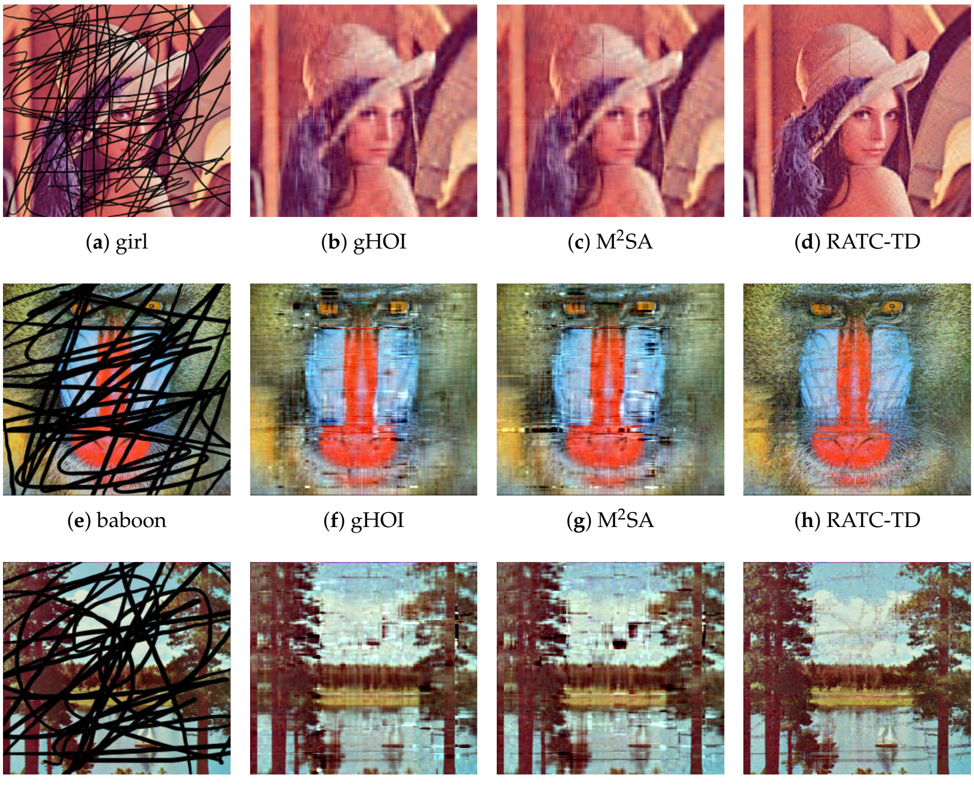
Figure 1. The original images, which are referred to as ‘girl’, ‘baboon’ and ‘boat’, test problem 3.

Two ways to construct partial images are considered, and our method is tested to recover the original images using the partial images. First, some black lines are added to the images, which can be considered a kind of structural missing, and the corrupted images are shown in the first column of Figure 2. The black line parts of images correspond to $\overline{\Omega}$, and the rest correspond to Ω. Second, after the initial image is converted into the data format of a third-order tensor $\mathcal{T}$, the observed index set Ω is constructed with each index randomly generated through the uniform distribution, and the size of Ω is $r|\mathcal{T}|$ (where the sampling rate is $r = 0.1$). The images associated with Ω are shown in the first column of Figure 3.

From Figure 2, it is clear that the images obtained by our RATC-TD are closer to the ground truth images than the ones obtained by gHOI and M^{2}SA. The numerical results representing the qualities of recovery are shown in Table 5. In addition to the errors $error_{obs}$ (32) on the observation set and the errors $error_{val}$ (33) on the unobserved data set, we also use SSIM (structure similarity index measure) [29] and PSNR (peak signal-to-noise ratio) parameters to evaluate the effect of our image restoration, which are often used to assess the quality of image restoration. SSIM ranges from 0 to 1; the larger the SSIM is, the smaller the image distortion is. Similarly, the larger the PSNR is, the less distortion there is. PSNR is used to measure the difference between two images. For the restored image $\mathcal{X}$ and the original image $\mathcal{T}$, the PSNR between them is given by the following formula:

$$\mathrm{PSNR} = 10 \times \lg\left(\frac{\mathrm{Maxpixel}^2}{\mathrm{MSE}}\right), \tag{36}$$

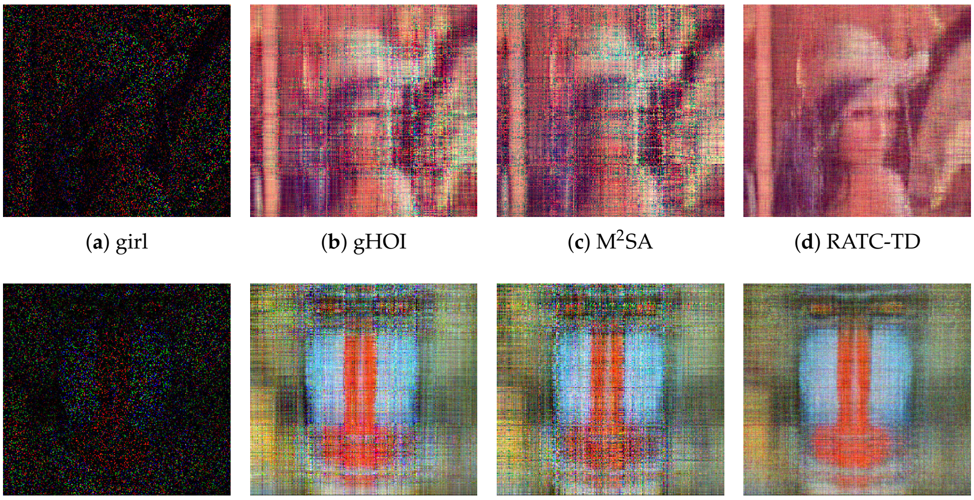
where Maxpixel is the maximum value of the image pixel. In our test problem, Maxpixel $= 255$, and MSE is the mean square error between $\mathcal{X}$ and $\mathcal{T}$. For the case with sampling rate $r = 0.1$, the corresponding restoration results are shown in Figure 3, and the numerical results representing the qualities of recovery are shown in Table 6. In this case, we only use 10 percent of the ground truth data, but it can be seen that our RATC-TD gives effective recovery results for this test problem.


(**a**) girl (**b**) gHOI (**c**) M^{2}SA (**d**) RATC-TD

(**e**) baboon (**f**) gHOI (**g**) M^{2}SA (**h**) RATC-TD

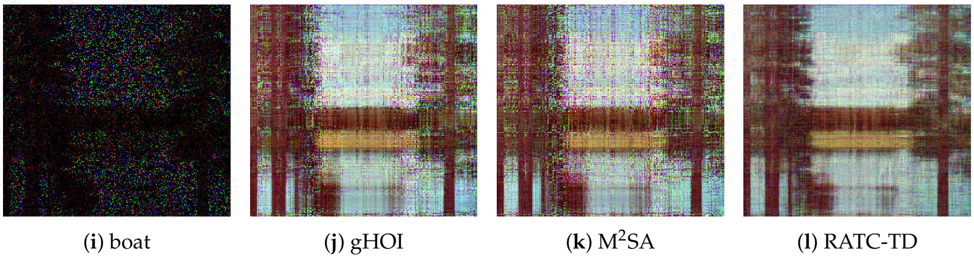
(**i**) boat (**j**) gHOI (**k**) M^{2}SA (**l**) RATC-TD

Figure 2. The restoration results of the images with black lines, test problem 3.


(**a**) girl (**b**) gHOI (**c**) M^{2}SA (**d**) RATC-TD

(**e**) baboon (**f**) gHOI (**g**) M^{2}SA (**h**) RATC-TD

Figure 3. *Cont.*



(**i**) boat (**j**) gHOI (**k**) M^{2}SA (**l**) RATC-TD

Figure 3. The restoration results with sampling rate $r = 0.1$, test problem 3.

Table 5. Numerical characterization of the recovery quality of images with black lines, test problem 3.

Girl	**PSNR**	**SSIM**	$error_{obs}$	$error_{val}$
missing figure	11.4347	0.5044	/	/
gHOI	24.8558	0.9536	0.0236	0.0985
M^{2}SA	24.8196	0.9532	0.0235	0.1080
RATC-TD	28.9981	0.9833	0.0099	0.0751
Baboon	**PSNR**	**SSIM**	$error_{obs}$	$error_{val}$
missing figure	8.6372	0.1878	/	/
gHOI	18.3661	0.7477	0.0268	0.1670
M^{2}SA	17.2923	0.6948	0.0263	0.2341
RATC-TD	20.9857	0.8612	0.0098	0.1004
Boat	**PSNR**	**SSIM**	$error_{obs}$	$error_{val}$
missing figure	8.5502	0.3229	/	/
gHOI	17.4365	0.8717	0.0282	0.2419
M^{2}SA	16.9671	0.8556	0.0276	0.2635
RATC-TD	22.0162	0.9524	0.0099	0.0998

Table 6. Numerical characterization of the recovery quality of images obtained by random sampling at sampling rate of 0.1, test problem 3.

Girl	**PSNR**	**SSIM**	$error_{obs}$	$error_{val}$
missing figure	5.5747	0.0329	/	/
gHOI	13.2045	0.6585	0.0309	0.1336
M^{2}SA	11.9943	0.5977	0.0305	0.1663
RATC-TD	18.9566	0.8049	0.0100	0.0764
Baboon	**PSNR**	**SSIM**	$error_{obs}$	$error_{val}$
missing figure	5.8165	0.0306	/	/
gHOI	13.6954	0.5862	0.0319	0.1121
M^{2}SA	12.8840	0.5404	0.0351	0.1441
RATC-TD	17.4101	0.7008	0.0098	0.0968
Boat	**PSNR**	**SSIM**	$error_{obs}$	$error_{val}$
missing figure	5.6110	0.0346	/	/
gHOI	12.5033	0.7168	0.0312	0.1336
M^{2}SA	11.6136	0.6549	0.0343	0.1615
RATC-TD	17.4527	0.8735	0.0096	0.0869

For the gHOI method and the M^{2}SA method, which need to be given the rank of the initial tensor in advance, we choose the initial rank as $R^{(0)} = [20, 20, 3]$. It is worth noting that for real image data, it is difficult for us to know the exact Tucker rank in advance. When there is no way to know the rank in advance, in order to use gHOI and M^{2}SA to achieve data completion, the initial rank can only be constantly adjusted through experiments, or the rank is given based on prior experience.

6. Conclusions

In this paper, we propose a novel rank-adaptive tensor completion based on the Tucker decomposition (RATC-TD) method. As the existing tensor completion methods based on Tucker decomposition, such as gHOI, M^{2}SA and Tucker–Wopt, typically require the tensor rank given a priori, overestimating or underestimating the tensor ranks can lead to poor results. Inspired by the RaHOOI algorithm, we propose our algorithm based on the HOOI structure. Our proposed algorithm can adaptively estimate the multilinear rank of data in the process of tensor completion. We show the algorithm's effectiveness through experiments on completing synthetic data and genuine pictures. The results of our algorithm can also provide effective initial data for other tensor completion methods, such as M^{2}SA. The further work we expect is to extend the algorithm to higher-dimensional problems, which requires us to optimize the algorithm further to reduce the algorithm's computational time.

Author Contributions: Conceptualization, S.L., X.S. and Q.L.; methodology, S.L., X.S. and Q.L.; software, S.L. and X.S.; validation, S.L. and X.S.; investigation, S.L. and X.S.; writing—original draft preparation, S.L. and X.S.; writing—review and editing, S.L., X.S. and Q.L.; visualization, S.L. and X.S.; project administration, Q.L.; funding acquisition, Q.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 12071291), the Science and Technology Commission of Shanghai Municipality (No. 20JC1414300) and the Natural Science Foundation of Shanghai (No. 20ZR1436200).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tang, K.; Liao, Q. Rank adaptive tensor recovery based model reduction for partial differential equations with high-dimensional random inputs. *J. Comput. Phys.* **2020**, *409*, 109326.
2. Ahmadi-Asl, S.; Asante-Mensah, M.G.; Cichocki, A.; Phan, A.H.; Oseledets, I.; Wang, J. Cross Tensor Approximation for Image and Video Completion. *arXiv* **2022**, arXiv:2207.06072
3. Bengua, J.A.; Phien, H.N.; Tuan, H.D.; Do, M.N. Efficient tensor completion for color image and video recovery: Low-rank tensor train. *IEEE Trans. Image Process.* **2017**, *26*, 2466–2479. [CrossRef]
4. Long, Z.; Liu, Y.; Chen, L.; Zhu, C. Low rank tensor completion for multiway visual data. *Signal Process.* **2019**, *155*, 301–316. [CrossRef]
5. Tan, H.; Feng, J.; Chen, Z.; Yang, F.; Wang, W. Low multilinear rank approximation of tensors and application in missing traffic data. *Adv. Mech. Eng.* **2014**, *6*, 157597.
6. Araújo, D.C.; De Almeida, A.L.; Da Costa, J.P.; de Sousa, R.T. Tensor-based channel estimation for massive MIMO-OFDM systems. *IEEE Access* **2019**, *7*, 42133–42147.
7. Cai, J.F.; Candès, E.J.; Shen, Z. A singular value thresholding algorithm for matrix completion. *SIAM J. Optim.* **2010**, *20*, 1956–1982. [CrossRef]
8. Ma, S.; Goldfarb, D.; Chen, L. Fixed point and Bregman iterative methods for matrix rank minimization. *Math. Program.* **2011**, *128*, 321–353.
9. Liu, J.; Musialski, P.; Wonka, P.; Ye, J. Tensor completion for estimating missing values in visual data. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 208–220. [CrossRef]
10. Acar, E.; Dunlavy, D.M.; Kolda, T.G.; Mørup, M. Scalable tensor factorizations for incomplete data. *Chemom. Intell. Lab. Syst.* **2011**, *106*, 41–56.
11. Geng, X.; Smith-Miles, K.; Zhou, Z.H.; Wang, L. Face image modeling by multilinear subspace analysis with missing values. In Proceedings of the 17th ACM international conference on Multimedia, Beijing, China, 19–24 October 2009.
12. He, Z.; Zhang, Z. High-dimensional uncertainty quantification via tensor regression with rank determination and adaptive sampling. *IEEE Trans. Compon. Packag. Manuf. Technol.* **2021**, *11*, 1317–1328. [CrossRef]
13. Kolda, T.G.; Bader, B.W. Tensor decompositions and applications. *SIAM Rev.* **2009**, *51*, 455–500. [CrossRef]
14. Gandy, S.; Recht, B.; Yamada, I. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Probl.* **2011**, *27*, 025010. [CrossRef]
15. Kressner, D.; Steinlechner, M.; Vandereycken, B. Low-rank tensor completion by Riemannian optimization. *BIT Numer. Math.* **2014**, *54*, 447–468.
16. Oseledets, I.V. Tensor-train decomposition. *SIAM J. Sci. Comput.* **2011**, *33*, 2295–2317. [CrossRef]

17. Ding, M.; Huang, T.Z.; Ji, T.Y.; Zhao, X.L.; Yang, J.H. Low-rank tensor completion using matrix factorization based on tensor train rank and total variation. *J. Sci. Comput.* **2019**, *81*, 941–964. [CrossRef]
18. Yang, J.; Zhu, Y.; Li, K.; Yang, J.; Hou, C. Tensor completion from structurally-missing entries by low-TT-rankness and fiber-wise sparsity. *IEEE J. Sel. Top. Signal Process.* **2018**, *12*, 1420–1434.
19. Jiang, Q.; Ng, M. Robust Low-Tubal-Rank Tensor Completion via Convex Optimization. In Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI), Macao, China, 10–16 August 2019.
20. Liu, X.Y.; Aeron, S.; Aggarwal, V.; Wang, X. Low-tubal-rank tensor completion using alternating minimization. *IEEE Trans. Inf. Theory* **2020**, *66*, 1714–1737. [CrossRef]
21. Wang, A.; Lai, Z.; Jin, Z. Noisy low-tubal-rank tensor completion. *Neurocomputing* **2019**, *330*, 267–279. [CrossRef]
22. Liu, Y.; Long, Z.; Huang, H.; Zhu, C. Low CP rank and tucker rank tensor completion for estimating missing components in image data. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 944–954.
23. Hu, Y.; Zhang, D.; Ye, J.; Li, X.; He, X. Fast and accurate matrix completion via truncated nuclear norm regularization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 2117–2130.
24. Liu, Y.; Shang, F.; Fan, W.; Cheng, J.; Cheng, H. Generalized higher-order orthogonal iteration for tensor decomposition and completion. In Proceedings of the Advances in Neural Information Processing Systems 27 (NIPS 2014), Montreal, QC, Canada, 8–13 December 2014.
25. Tucker, L.R. Some mathematical notes on three-mode factor analysis. *Psychometrika* **1966**, *31*, 279–311. [CrossRef]
26. De Lathauwer, L.; De Moor, B.; Vandewalle, J. On the best rank-1 and rank-$(R_1, R_2, \ldots, R_N)$ approximation of higher-order tensors. *SIAM J. Matrix Anal. Appl.* **2000**, *21*, 1324–1342.
27. Xiao, C.; Yang, C. A rank-adaptive higher-order orthogonal iteration algorithm for truncated Tucker decomposition. *arXiv* **2021**, arXiv:2110.12564
28. Uzawa, H. Iterative methods for concave programming. *Stud. Linear Nonlinear Program.* **1958**, *6*, 154–165.
29. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.