Adaptive Dynamic Convolution Module Based Low-Rank Tensor Factorization Framework for Multi-Dimensional Image Recovery

1st Dan Xu, 2nd Xiaoli Sun
School of Mathematical Sciences
Shenzhen University
Shenzhen, China
xxudan@outlook.com, xlsun@szu.edu.cn

3rd Zimeng Li, 4th Xiujun Zhang
School of Electronic and Communication Engineering
Shenzhen Polytechnic University
Shenzhen, China
Li_zimeng@szpu.edu.cn, zhangxiujun@szpu.edu.cn

***Abstract*—Due to sensor limitations, noise interference, and other factors, tensor data exhibit random or structured missing scenarios during acquisition and transmission. Thus, recovering the original high-dimensional tensor from incomplete observations has become a challenging problem. Although tensor recovery methods based on tube transform and low-rank decomposition have achieved certain progress, they remain insufficient in exploring local neighborhood structures and characterizing cross-channel interactions, making it difficult to effectively preserve spatial-spectral consistency. To address these issues, we propose an adaptive dynamic convolution module based low-rank tensor factorization framework for multi-dimensional image recovery, named LRADC. First, we propose a non-linear multilayer neural network guided by the adaptive dynamic convolution (ADC) module. The ADC module dynamically generates convolution kernels based on input content, boosting the network's capacity for local detail reconstruction and cross-channel feature interaction. Then, using this learned non-linear network, we define a low-rank tensor factorization framework. The newly proposed framework leverages low-rank factorization to model global structural priors and uses the ADC module to dynamically enhance local details and cross-channel feature interaction. Therefore, when addressing challenging tube missing scenarios, the model has obvious advantages in global structure mining and detailed information preservation. Extensive experiments on public datasets show that LRADC outperforms state-of-the-art methods in both visual and quantitative metrics.**

***Index Terms*—Low-Rank Tensor Factorization, Adaptive Dynamic Convolution, Multi-Dimensional Image Recovery.**

I. Introduction

In such fields as modern remote sensing observation [1], medical imaging [2], and video surveillance [3] and so on, the related data inherently exhibit a multi-dimensional structure. For example, hyperspectral/multispectral images integrate spatial and spectral information, while video sequences encompass spatial, temporal, and color modes. As a multi-dimensional generalization of matrices, tensors are capable of more comprehensively capturing the intrinsic correlations inherent in such data. However, during acquisition, transmission, and storage, multi-dimensional data often suffer from data loss or incomplete observation due to sensor limitations, occlusions, or noise [4]. This severely undermines subsequent processing and analysis. Therefore, recovering the original high-dimensional tensor from incomplete observations has become a challenging issue in computational imaging.

Suppose $\mathcal{O}$ is the observed tensor, Ω is the observation set, $\mathcal{X}$ is the recovered tensor, and a unified framework for the tensor completion problem can be formulated as [5] :

$$\min_{\mathcal{X}} \ \mathrm{rank}(\mathcal{X}) \quad \text{s.t.} \quad \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{O}), \tag{1}$$

Here, $\mathcal{P}_\Omega(\cdot)$ denotes the observation projection operator.

In the model, the definition of tensor rank depends on different tensor decompositions, such as the CP decomposition [6] and the Tucker decomposition [7]. Based on the theory of tensor singular value decomposition (t-SVD) [8], Kilmer et al. proposed the tensor tubal rank based on the discrete Fourier transform (DFT). This metric effectively characterizes the low-rank structure of the tensors along the tubal direction. With different transform domains, a series of variants have emerged, such as the discrete cosine transform (DCT) [9], [10] and the unitary transform [11], etc. To enhance the non-linear modeling capability for the low-rank structure of tensors, researchers introduce neural networks to obtain learnable transforms, such as S2NTNN [12] and HLRTF [13] methods. S2NTNN [12] method relies on a large number of t-SVD operations and therefore exhibits low computational efficiency. HLRTF [13] method proposes a framework based on hierarchical tensor factorization and effectively improves computational efficiency. LRTFR [14] method extends the tensor representation to the continuous function space, where the modeling is performed in the continuous function domain. However, since these approaches only consider the similarity along the tubal direction, they insufficiently exploit the local correlation in the spatial domain, which limits the recovery capacity of the model. When the tensor to be recovered exhibits structured missing, particularly in cases of the complete absence of spectral tubes, these methodologies lack the competence to capture local neighborhood information and cross-channel information fusion.

To address the limitations of traditional models in handling structured missing, we propose a tensor low-rank factorization framework based on an adaptive dynamic convolution module induced non-linear multilayer neural network. First, we propose a non-linear multilayer neural network guided by the adaptive dynamic convolution (ADC) module. The learnable

This work is supported by the National Natural Science Foundation of China under grant 62272313 and 62501412. The corresponding author is Xiaoli Sun.

ADC module dynamically generates convolution kernels based on input content and effectively enhances the capabilities of the neural network to explore local detail structures and information of feature interactions among channels. Then, based on the learned multilayer neural network, we define the corresponding low-rank tensor factorization framework (LRADC) for tensor recovery. By adjusting the size of the factor tensors, we can effectively ensure the low-rankness of the recovered tensor. The overall LRADC framework is illustrated in Fig. 1.

We summarize the contributions of this work as follows:

- We propose a non-linear multilayer neural network guided by the adaptive dynamic convolution (ADC) module. The learnable ADC module combines global adaptive pooling, fully connected mapping, and per-sample convolution to dynamically generate convolution kernels. This effectively enhances the capabilities of the multilayer neural network to explore local detail reconstruction and cross-channel feature interaction.
- Based on the learned multilayer neural network, we design the corresponding low-rank tensor factorization framework for tensor recovery (LRADC). The framework leverages low-rank factorization to model global low-rank structural priors and uses the ADC module to dynamically enhance local details and cross-channel feature interactions. Therefore, when addressing complex tube missing problems, the model has obvious advantages in global structure mining and detailed information preservation.
- We develop an optimization algorithm based on the ADMM framework, with network parameters updated using Adam. The entire model is trained through unsupervised reconstruction loss, showing strong robustness and superior recovery performance. Extensive experiments are conducted on seven multispectral and hyperspectral datasets, including both random-missing and tube-missing scenarios. Experimental results demonstrate that the proposed method outperforms state-of-the-art methods in terms of PSNR and SSIM.

II. Our Method

A. Adaptive Dynamic Convolution Kernel

Traditional multilayer convolutional networks extract feature using fixed convolution kernels. Although standard convolutions can capture local patterns, their filters remain static and cannot adapt to changes in input content. To enhance the adaptability of network, we construct a learnable adaptive dynamic convolution kernel.

Let $\mathcal{X} \in \mathbb{R}^{C\times H\times W}$ denote a tensor. The dynamic convolution kernel tensor $\mathbf{K}(\cdot) : \mathbb{R}^{C\times H\times W} \to \mathbb{R}^{C\times s\times s}$ could be learned in the following two steps:

1) **Feature extraction.** For each channel c, we apply global average pooling to obtain a scalar feature

$$g_c \;=\; \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}\mathcal{X}_c^{(i,j)}, \tag{2}$$

which leads to the channel feature vector

$$\mathbf{g}(\mathcal{X}) \;=\; [g_1, g_2, \ldots, g_C]^\top \in \mathbb{R}^C. \tag{3}$$

2) **Kernel generation.** Each channel kernel depends only on its corresponding global feature g_c. The vectorized form of the kernel is given by

$$\mathbf{k}_c(\mathcal{X}) \;=\; \mathbf{a}_c\, g_c + \mathbf{b}_c, \tag{4}$$

where $\mathbf{a}_c \in \mathbb{R}^{s^2}$ and $\mathbf{b}_c \in \mathbb{R}^{s^2}$ are weight and bias vectors, respectively. After reshaping, we obtain a convolution kernel $\mathbf{k}_c(\mathcal{X}) \in \mathbb{R}^{s\times s}$. The dynamic convolution kernel tensor $\mathbf{K}(\mathcal{X}) \in \mathbb{R}^{C\times s\times s}$ is

$$\mathbf{K}(\mathcal{X}) = \big[\mathbf{k}_1(\mathcal{X}), \mathbf{k}_2(\mathcal{X}), \ldots, \mathbf{k}_C(\mathcal{X})\big], \tag{5}$$

Here, the bracket $[\cdot]$ indicates the stacking operation along the channel mode.

B. Adaptive Dynamic Convolution Module Based Nonlinear Multilayer Neural Network

Firstly, based on the proposed dynamic convolution kernel tensor, we propose the definition of the face-wise adaptive convolution operator as follows.

*Definition 1 (**Tensor Face-wise Convolution Operator**): For two tensors $\mathcal{X} \in \mathbb{R}^{C\times H\times W}$ and $\mathbf{K}(\mathcal{X}) \in \mathbb{R}^{C\times s\times s}$, the tensor face-wise convolution is defined as*

$$\mathcal{Y} = \mathcal{X} \circledast_t \mathbf{K}(\mathcal{X}) \tag{6}$$

where $\circledast_t$ denotes the tensor face-wise convolution operator.

For each slice of $\mathcal{Y}$, that is $\mathcal{Y}_c$, we have:

$$\mathcal{Y}_c = \mathcal{X}_c \circledast \mathbf{k}_c(\mathcal{X}), \qquad c = 1, \ldots, C. \tag{7}$$

where the symbol $\circledast$ denotes the traditional convolution operator. Its element-wise form is given by:

$$\mathcal{Y}_c^{(i,j)} = \sum_{u=-\tau}^{\tau}\sum_{v=-\tau}^{\tau}\mathbf{k}_c^{(u,v)}(\mathcal{X}) \cdot \mathcal{X}_c^{(i+u,\,j+v)}. \tag{8}$$

where $s = 2\tau+1$ and the summation is taken only over indices satisfying $(i+u, j+v) \in \{1, \ldots, H\} \times \{1, \ldots, W\}$.

Based on the learned adaptive dynamic convolution kernel tensor and the tensor face-wise convolution in Definition 1, we stack per-channel convolution output along the channel mode and define the ADC module as:

$$f(\mathcal{X}; \mathbf{K}(\mathcal{X})) \;=\; \mathcal{X} \circledast_t \mathbf{K}(\mathcal{X}) \;\in\; \mathbb{R}^{C\times H\times W} \tag{9}$$

where $k_c(\mathcal{X}) \in \mathbb{R}^{s\times s}$ denotes the convolution kernel of the c-th channel, which is adaptively generated from the input $\mathcal{X}$ with size $s = 2\tau + 1$. The specific operation of Eq. 9 is illustrated in Fig. 2.

To introduce cross-channel interaction and non-linear expressiveness, we introduce a channel mixing nonlinearity module $\phi_\ell(\cdot)$, which is composed of a non-linear channel transform and the stacked face-wise convolution, as follows:

$$\phi_\ell(\mathcal{X}) \;:=\; \sigma\big(f(\mathcal{X}; \mathbf{K}_\ell(\mathcal{X})) \times_1 \mathbf{W}_\ell\big), \qquad \ell = 1, \ldots, L \tag{10}$$

Finally, we recursively stack multiple modules $\phi_\ell(\cdot)$, and the final L-layer neural network $\Phi(\cdot)$ is expressed as

$$\Phi(\mathcal{X}) \;=\; \phi_L \circ \phi_{L-1} \circ \cdots \circ \phi_1(\mathcal{X}), \qquad \ell = 1, \ldots, L \tag{11}$$

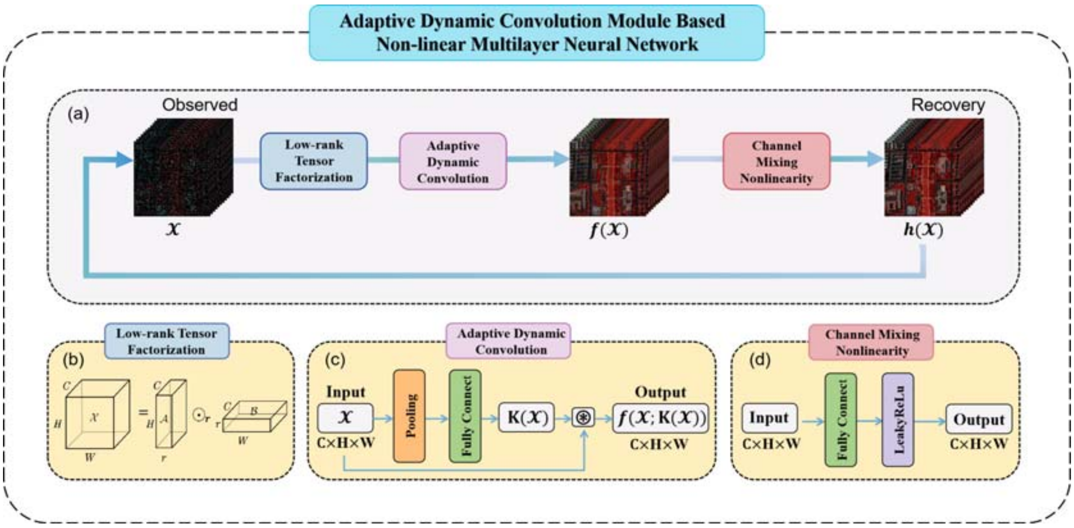


Fig. 1. Illustration of the proposed LRADC framework. (a) Overall network pipeline; (b) Low-rank tensor factorization; (c) Structure of the adaptive dynamic convolution (ADC) module, which enhances local features and promotes cross-channel interactions; (d) Structure of the channel mixing nonlinearity module, which performs nonlinear feature fusion.

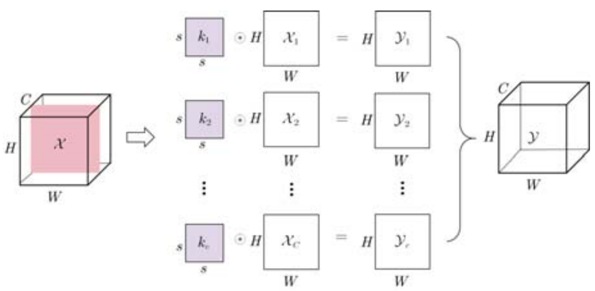


Fig. 2. Illustration of the tensor face-wise convolution operator.

C. Low-Rank Tensor Factorization

In multi-dimensional image recovery, where multispectral/hyperspectral images can be viewed as three-dimensional data (1D spectral domain and 2D spatial domain), the target tensor $\mathcal{X} \in \mathbb{R}^{C\times H\times W}$ often exhibits high redundancy along both the spectral and spatial dimensions. Therefore, the third-order tensor has the low-rank property. Recovering low-rank tensors from observed data inevitably involves a large number of SVD operations, which incurs substantial computational costs. Moreover, its stability may degrade when the data is severely missing.

According to Theorem 2 in [13], we know that under the definition of tensor product (T-product) induced by linear or non-linear transforms, tensor factorization can not only preserve the low-rank property of tensors, but also effectively reduce computational complexity. Specifically, if a third-order tensor $\mathcal{X} \in \mathbb{R}^{C\times H\times W}$ has rank r, it can be decomposed into two smaller tensors $\mathcal{A} \in \mathbb{R}^{C\times H\times r}$ and $\mathcal{B} \in \mathbb{R}^{C\times r\times W}$. Consequently, a low-rank tensor factorization based on the multilayer neural network $\Phi(\cdot)$ proposed in the above section is defined as:

$$\mathcal{X} = \Phi(\mathcal{A} \odot_r \mathcal{B}), \tag{12}$$

$$(\mathcal{A} \odot_r \mathcal{B})_{c,h,w} = \sum_{k=1}^{r} \mathcal{A}_{c,h,k}\, \mathcal{B}_{c,k,w}. \tag{13}$$

where $\odot_r$ denotes contraction along the rank dimension r. Using the tensor factorization based on $\Phi(\cdot)$ defined in Eq. 12, we can ensure that $\mathrm{rank}(\mathcal{X}) \leq r$. By changing r (which can be done efficiently by changing the size of $\mathcal{A}$ and $\mathcal{B}$), we can control the rank of the recovered tensor $\mathcal{X}$.

This factorization not only reduces the number of parameters and computational cost, but also preserves the key spatial-spectral correlations, thereby improving the efficiency of tensor recovery.

D. Multi-dimensional Image Recovery Model

We propose a new self-supervised framework for multi-dimensional image recovery, named LRADC, which integrates low-rank tensor factorization with adaptive dynamic convolution module. The optimization model is formulated as follows.

$$\begin{aligned} \min_{\mathcal{A},\mathcal{B},\{\mathbf{K}_\ell\},\{\mathbf{W}_\ell\}} & \left\|\mathcal{P}_\Omega(\Phi(\mathcal{A} \odot_r \mathcal{B})) - \mathcal{P}_\Omega(\mathcal{O})\right\|_F^2 \\ & + \lambda \sum_{d=1}^{3} \|\nabla_d(\Phi(\mathcal{A} \odot_r \mathcal{B}))\|_1, \end{aligned} \tag{14}$$

where $\mathcal{O} \in \mathbb{R}^{C\times H\times W}$ is the observed incomplete tensor; $\Phi(\mathcal{A} \odot_r \mathcal{B})$ denotes the recovered tensor; $\mathcal{A} \in \mathbb{R}^{C\times H\times r}$ and $\mathcal{B} \in \mathbb{R}^{C\times r\times W}$. $\mathcal{P}_\Omega$ is the projection operator, Ω is the set of observed indices. The second term in the model is the total variation (TV) regularization [15], which promotes smoothness of the recovered results in both spatial and spectral domains. ∇_d $(d = 1, 2, 3)$ denotes the gradient operator along the spectral, row, and column dimensions of the tensor, respectively, capturing local variations of the data across different dimensions. λ is the regularization parameter. Through this optimization, LRADC is able to model the global low-rank structural prior while enhancing local details and cross-channel interactions via adaptive dynamic convolution, thereby achieving complementary integration of global and local features.

Define the set of parameters as $\Theta = \{\mathcal{A}, \mathcal{B}, \{\mathbf{K}_\ell\}, \{\mathbf{W}_\ell\}\}$, and update Θ using Adam optimizer [16]. To handle the problem in Eq. 14, we develop an ADMM-based algorithm. Introducing auxiliary variables $\mathcal{U}_d$ $(d = 1, 2, 3)$, we reformulate the model as follows.

$$\begin{aligned} \min_{\Theta,\{\mathcal{U}_d\}} \ & \|P_\Omega(\Phi(\mathcal{A} \odot_r \mathcal{B})) - P_\Omega(\mathcal{O})\|_F^2 + \lambda \sum_{d=1}^{3} \|\mathcal{U}_d\|_1 \\ \text{s.t.} \ & \mathcal{U}_d = \nabla_d(\Phi(\mathcal{A} \odot_r \mathcal{B})), \ d = 1, 2, 3. \end{aligned} \tag{15}$$

In this case, the augmented Lagrangian function is as follows:

$$\begin{aligned} \mathcal{L}_\alpha(\Theta, \mathcal{U}_d, \mathcal{V}_d) = & \sum_{d=1}^{3} \Big(\langle \mathcal{V}_d, \nabla_d(\Phi(\mathcal{A} \odot_r \mathcal{B})) - \mathcal{U}_d \rangle \\ & + \frac{\alpha}{2} \|\nabla_d(\Phi(\mathcal{A} \odot_r \mathcal{B})) - \mathcal{U}_d\|_F^2 \Big) + \lambda \sum_{d=1}^{3} \|\mathcal{U}_d\|_1, \end{aligned} \tag{16}$$

where α is the penalty parameter and $\mathcal{V}_d$ denotes the Lagrangian multipliers. Under the ADMM framework [17], the optimization is decomposed into three subproblems.

1) Update Θ: We update Θ using Adam [16] by solving

$$\begin{aligned} \min_{\Theta} \sum_{d=1}^{3} \Big(& \langle \mathcal{V}_d, \nabla_d(\Phi(\mathcal{A} \odot_r \mathcal{B})) - \mathcal{U}_d \rangle \\ & + \tfrac{\alpha}{2} \|\nabla_d(\Phi(\mathcal{A} \odot_r \mathcal{B})) - \mathcal{U}_d\|_F^2 \Big). \end{aligned} \tag{17}$$

2) Update $\mathcal{U}_d$ $(d = 1, 2, 3)$: The subproblem is

$$\min_{\mathcal{U}_d} \ \lambda \|\mathcal{U}_d\|_1 + \frac{\alpha}{2} \Big\| (\nabla_d(\Phi(\mathcal{A} \odot_r \mathcal{B})) + \frac{\mathcal{V}_d}{\alpha}) - \mathcal{U}_d \Big\|_F^2. \tag{18}$$

which can be solved by

$$\mathcal{U}_d = \text{Soft}_{\lambda/\alpha} \left(\nabla_d(\Phi(\mathcal{A} \odot_r \mathcal{B})) + \frac{\mathcal{V}_d}{\alpha} \right), \tag{19}$$

where $\text{Soft}_\beta(\cdot)$ denotes the soft-thresholding operator with threshold β, defined as

$$\text{Soft}_\beta(x) = \max(|x| - \beta, 0) \cdot \text{sgn}(x). \tag{20}$$

3) Update $\mathcal{V}_d$ $(d = 1, 2, 3)$: The update rule is

$$\mathcal{V}_d = \mathcal{V}_d + \alpha \big(\nabla_d(\Phi(\mathcal{A} \odot_r \mathcal{B})) - \mathcal{U}_d \big). \tag{21}$$

We set the maximum number of iterations as $I_{\max}$, and the overall procedure is summarized in Algorithm 1.

Algorithm 1 The whole algorithm of our method

1: **Input:** The observed tensor $\mathcal{O}$; parameter λ; Lagrange parameter α; maximum iteration $I_{\max}$
2: **Initialization:** $\mathcal{U}_d = \nabla_d(\Phi(\mathcal{A} \odot_r \mathcal{B}))$, $\mathcal{V}_d = \mathbf{0}$, $I \leftarrow 0$
3: **while** $I < I_{\max}$ **do**
4: Update Θ using Adam optimizer [16];
5: $\mathcal{U}_d \leftarrow \text{Soft}_{\lambda/\alpha} \left(\nabla_d(\Phi(\mathcal{A} \odot_r \mathcal{B})) + \frac{\mathcal{V}_d}{\alpha} \right)$ via Eq. 19;
6: $\mathcal{V}_d \leftarrow \mathcal{V}_d + \alpha \left(\nabla_d(\Phi(\mathcal{A} \odot_r \mathcal{B})) - \mathcal{U}_d \right)$ via Eq. 21;
7: $I \leftarrow I + 1$;
8: **end while**
9: **Output:** The recovered tensor $\Phi(\mathcal{A} \odot_r \mathcal{B})$

III. Experiments

All experiments are carried out on a computer equipped with an Intel(R) Core(TM) i7-8700 CPU, 16GB RAM, and an NVIDIA Quadro P620 GPU. The proposed method is implemented using Python with the PyTorch framework.

A. Experimental Settings

1) Datasets: In experiments, we adopt a variety of hyperspectral and multispectral datasets to evaluate the performance of our method. The HSI datasets[1] include *Pavia* and *WDC*. The MSI datasets[2] include *Flowers*, *Balloons*, *Feathers*, *Cd*, *Cloth*, and *Beans*. For all experiments, the missing rates (MR) are set to 0.9, 0.8 and 0.7, respectively.

2) Compared Methods: To more comprehensively evaluate the effectiveness of the proposed LRADC method, we compared it with several representative tensor recovery approaches, including TNN [18], UTNN [11], FTNN [19], NTTNN [20], HLRTF [13], S2NTNN [12] and LRTFR [14].

3) Evaluation Metrics: In our experiments, we adopt the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) as the main evaluation metrics. A higher PSNR value and an SSIM value closer to 1 indicate that the reconstruction is closer to the original image.

B. Experimental Results

1) Random Missing Scenario: The numerical results for tensor completion on the HSI and MSI datasets are shown in Table I. LRADC achieves the best PSNR/SSIM on most datasets, demonstrating that the proposed method exploits spatial and spectral information more fully. For better visual comparison, we also show the corresponding zoom-in regions of different methods in Fig. 3. It can be seen that, compared to traditional methods, LRADC more effectively recovers fine structures, texture details, and edge information. In Table I, though S2NTNN marginally outperforms our method on the MSI *Balloons* dataset at a missing rate of 0.7, it still suffers from loss of details in regions with complex structures. In contrast, LRADC is able to reconstruct clear, natural, and spectrally consistent images even with highly missing observations. This is because the low-rank decomposition in our model captures global structural information, while the adaptive

[1] http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes
[2] https://cave.cs.columbia.edu/repository/Multispectral

TABLE I
AVERAGE QUANTITATIVE COMPARISON OF DIFFERENT METHODS ON HSI AND MSI DATASETS (RANDOM MISSING)

Method		TNN		UTNN		FTNN		NTTNN		HLRTF		S2NTNN		LRTFR		Ours	
Data	MR	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
HSI	0.9	32.64	0.9197	37.88	0.9728	37.82	0.9734	37.90	0.9746	41.94	0.9913	35.49	0.9837	41.26	0.9901	**44.68**	**0.9947**
Pavia	0.8	37.80	0.9656	47.08	0.9951	45.21	0.9919	43.34	0.9915	48.50	0.9972	43.05	0.9968	48.08	0.9971	**49.12**	**0.9982**
(200×200×80)	0.7	41.72	0.9813	45.48	0.9978	50.94	0.9964	46.96	0.9961	49.96	0.9976	50.95	0.9982	49.50	0.9987	**51.53**	**0.9988**
HSI	0.9	32.96	0.9376	39.28	0.9834	37.08	0.9728	40.31	0.9874	39.55	0.9878	45.78	0.9934	39.09	0.9871	**48.22**	**0.9965**
WDC	0.8	37.75	0.9750	46.11	0.9953	43.32	0.9910	44.87	0.9948	42.45	0.9930	47.87	0.9964	43.55	0.9943	**50.66**	**0.9977**
(256×256×191)	0.7	41.10	0.9867	49.87	0.9975	47.12	0.9952	47.55	0.9969	44.71	0.9957	49.86	0.9974	45.46	0.9962	**51.92**	**0.9982**
MSI	0.9	33.78	0.8858	36.01	0.9317	36.86	0.9487	36.68	0.9336	38.98	0.9593	36.99	0.9781	39.31	0.9846	**44.62**	**0.9878**
Flowers	0.8	38.65	0.9506	41.64	0.9738	41.93	0.9784	41.44	0.9701	43.73	0.9810	43.27	0.9899	45.39	0.9931	**49.51**	**0.9937**
(512×512×31)	0.7	42.04	0.9730	45.20	0.9853	45.22	0.9873	44.39	0.9824	47.13	0.9859	46.03	0.9881	47.68	0.9931	**51.62**	**0.9952**
MSI	0.9	36.30	0.9441	39.24	0.9738	39.74	0.9783	36.95	0.9588	41.10	0.9857	42.16	0.9825	42.35	0.9870	**45.85**	**0.9945**
Balloons	0.8	42.00	0.9815	46.05	0.9931	45.60	0.9930	41.81	0.9827	45.98	0.9938	47.31	0.9942	44.87	0.9924	**49.80**	**0.9969**
(256×256×31)	0.7	46.08	0.9920	49.85	0.9972	49.35	0.9965	44.85	0.9901	49.01	0.9963	**50.74**	0.9974	46.21	0.9940	50.33	**0.9978**
MSI	0.9	33.44	0.9015	35.36	0.9386	36.77	0.9604	35.89	0.9473	39.04	0.9748	32.84	0.8715	39.13	0.9701	**40.56**	**0.9834**
Feathers	0.8	38.15	0.9578	40.87	0.9775	41.67	0.9833	40.68	0.9767	43.78	0.9874	38.22	0.9607	41.68	0.9837	**44.38**	**0.9919**
(512×512×31)	0.7	41.55	0.9779	44.72	0.9889	45.11	0.9908	43.71	0.9865	47.81	0.9913	41.91	0.9799	43.28	0.9871	**47.93**	**0.9944**
MSI	0.9	31.85	0.9050	32.56	0.9251	33.89	0.9443	31.27	0.9233	32.14	0.9444	31.15	0.9332	36.10	0.9620	**38.20**	**0.9654**
Cd	0.8	35.98	0.9507	36.31	0.9597	38.28	0.9613	35.64	0.9565	36.15	0.9691	35.71	0.9589	38.93	0.9813	**40.96**	**0.9845**
(512×512×31)	0.7	38.99	0.9712	39.23	0.9756	41.18	0.9767	38.59	0.9729	38.47	0.9784	37.07	0.9737	42.37	0.9888	**42.54**	**0.9889**
MSI	0.9	26.76	0.7685	27.12	0.7887	29.51	0.8703	29.61	0.8821	31.21	0.9037	30.61	0.8721	33.63	0.9751	**33.88**	**0.9794**
Cloth	0.8	31.07	0.8888	32.80	0.9237	34.34	0.9441	35.09	0.9510	37.65	0.9818	35.58	0.9538	35.56	0.9788	**39.91**	**0.9895**
(512×512×31)	0.7	34.39	0.9383	37.26	0.9641	37.84	0.9693	38.83	0.9727	41.55	0.9915	38.45	0.9733	39.02	0.9713	**44.51**	**0.9926**
MSI	0.9	25.48	0.6834	26.66	0.7531	27.66	0.8366	28.51	0.8527	31.24	0.8874	32.85	0.9325	31.84	0.9094	**35.14**	**0.9639**
Beans	0.8	29.90	0.8429	31.83	0.9015	32.87	0.9372	33.64	0.9411	36.98	0.9607	37.16	0.9768	36.70	0.9689	**38.64**	**0.9798**
(512×512×31)	0.7	33.22	0.9156	35.64	0.9533	36.27	0.9678	37.02	0.9696	40.02	0.9804	40.43	0.9825	40.98	0.9892	**41.15**	**0.9901**

The **best** and second-best values are highlighted.

dynamic convolution enhances local feature representation by using information across channels.

2) Tube Missing Scenario: In the more challenging scenario of tube missing, the performance gap among different methods becomes more evident. Table II reports the quantitative results on different datasets. It can be observed that traditional low-rank methods almost fail to recover the original structures under missing large-area tubes. HLRTF [13] and LRTFR [14] improve the ability to capture non-linear information, thus performing better than traditional methods in some datasets, but their performance still degrades significantly under high missing rates. In contrast, LRADC consistently demonstrates remarkable superiority across all datasets, achieving the best results and maintaining strong recovery capability even under extreme conditions.

This superiority is more clearly reflected in the visual comparisons, as shown in Fig. 4. In tube missing scenarios, other methods often suffer from severe structural loss, and sometimes fail to recover the overall spatial structure. Although HLRTF and LRTFR partially alleviate these issues, they still lack sufficient detail recovery due to limited cross-channel information fusion. In contrast, LRADC effectively completes the missing regions with structures and textures, while preserving smoothness and consistency along the spectral dimension, producing results that are very close to the groundtruth. This indicates that our adaptive dynamic convolution fully captures spatial-spectral information and enhances cross-channel feature interaction, enabling LRADC to achieve superior recovery performance in tube missing scenarios.

C. Ablation Study

1) Contributions of Different Components:: To evaluate the contribution of each component, we conducted an ablation study on the MSI *Cloth* dataset under tube missing scenario with MR = 0.7. The variants are defined as follows: **Ours** is the complete model; **Only ADC** denotes retaining only the ADC module; **LRADC*** denotes retaining only the low-rank factorization; **t-SVD + DFT** denotes replacing both proposed modules with tensor singular value decomposition and discrete Fourier transform.

The results are shown in Fig. 5. In terms of PSNR/SSIM, t-SVD + DFT performs the worst, indicating that fixed transforms are ineffective in recovering the structural missing scenario. Only ADC shows strong feature extraction ability, but its performance is limited by the lack of global low-rank constraint. LRADC* suffers a significant performance drop, demonstrating that while low-rank factorization captures global structure, it fails to model local details and cross-channel information without the ADC module. In contrast, the complete model LRADC achieves the best results on both metrics, showing that low-rank factorization and the ADC module are complementary, jointly enhancing the recovery of global structural information and local details modeling.

2) Parameters Analysis: To analyze the influence of rank r in low-rank decomposition on the recovery performance, we conduct a quantitative comparison of LRADC with different r values on the MSI *Cd* dataset under tube missing patterns, and

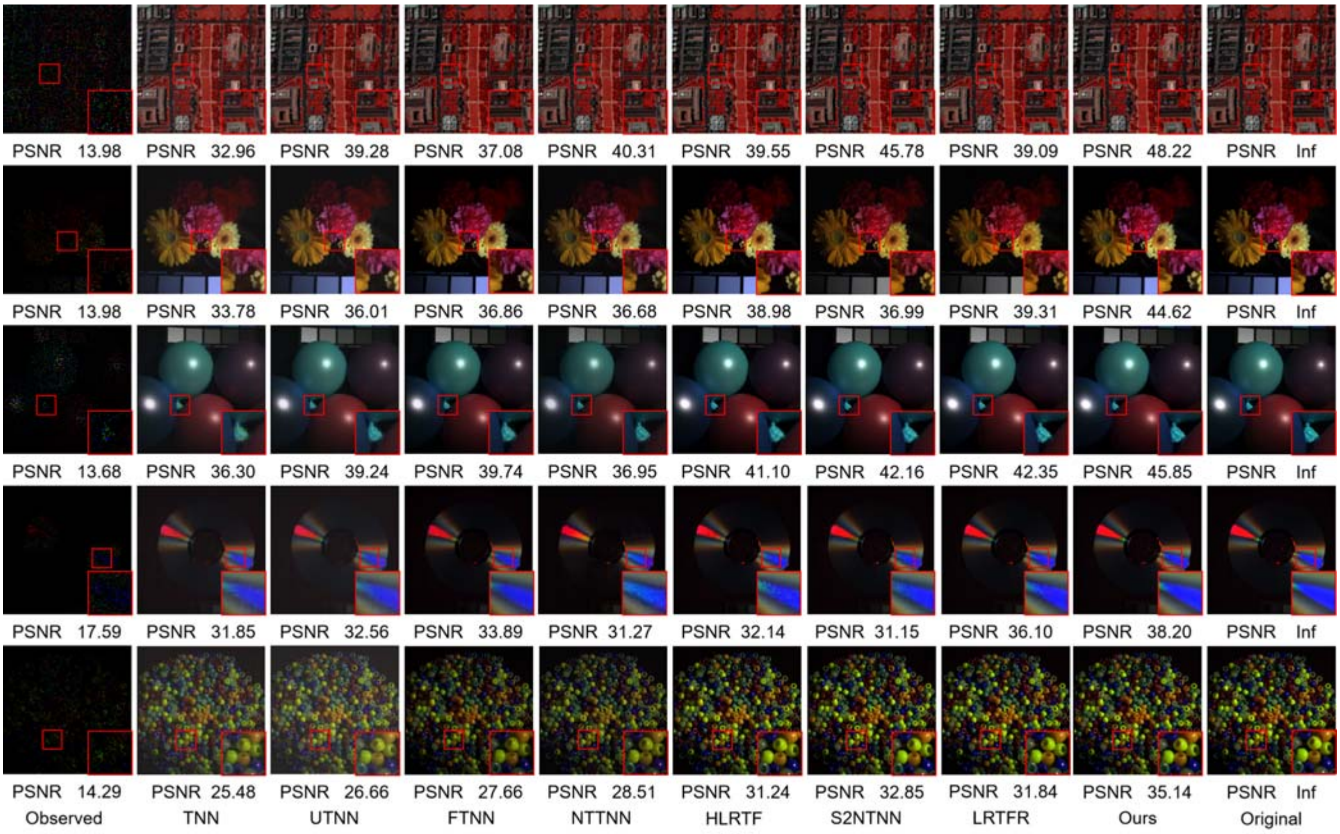


Fig. 3. Results of tensor completion by different methods on HSI *WDC* and MSI *Flowers*, *Balloons*, *Cd* and *Beans* with random missing (MR = 0.9; PSNR in dB).

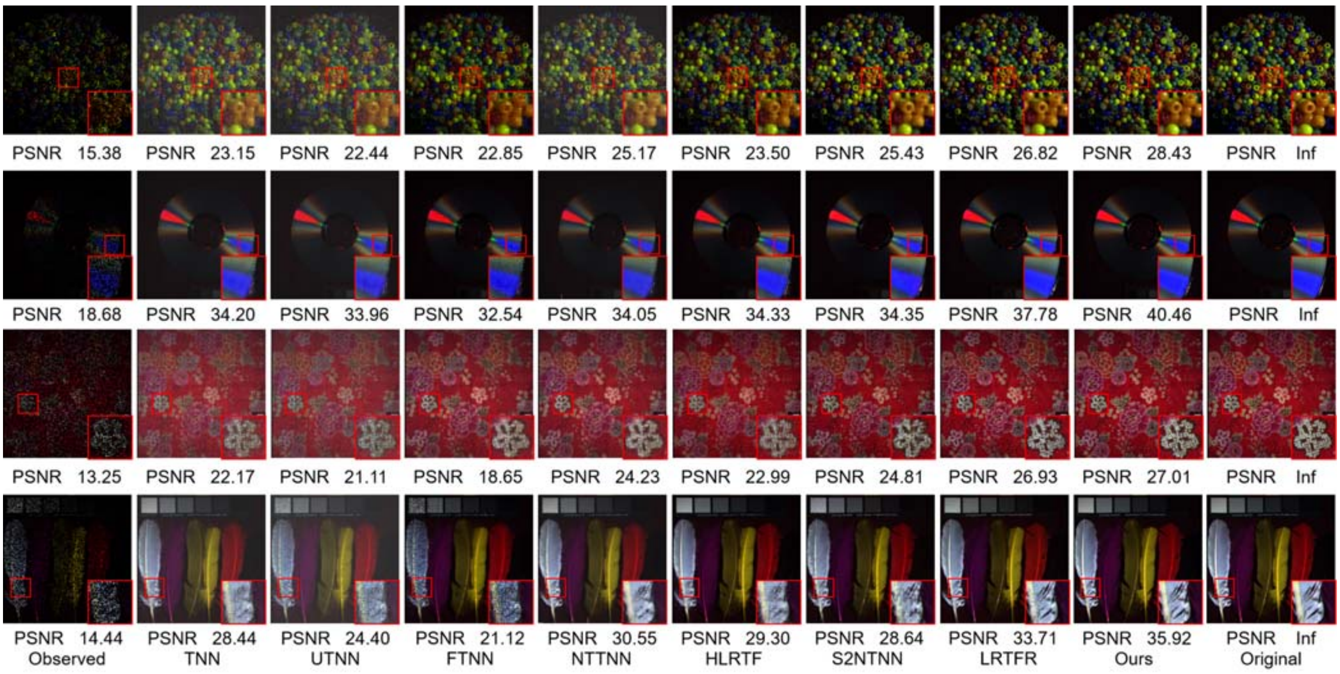


Fig. 4. Results of tensor completion by different methods on MSI *Beans*, *Cd*, *Cloth*, and *Feathers* with tube missing (MR = 0.7; PSNR in dB).

TABLE II
AVERAGE QUANTITATIVE COMPARISON OF DIFFERENT METHODS ON HSI AND MSI DATASETS (TUBE MISSING)

Method		TNN		UTNN		FTNN		NTTNN		HLRTF		S2NTNN		LRTFR		Ours	
Data	MR	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
HSI	0.9	16.19	0.1876	15.92	0.1778	14.62	0.1515	18.16	0.3915	21.50	0.4719	18.23	0.4021	19.56	0.4436	**22.91**	**0.5756**
Pavia	0.8	19.68	0.3126	19.25	0.3015	16.54	0.2196	23.53	0.5679	23.57	0.5649	21.12	0.5214	21.08	0.4973	**26.04**	**0.7647**
(200×200×80)	0.7	22.28	0.4831	21.91	0.4690	19.86	0.3861	26.06	0.7209	24.63	0.6606	23.51	0.5816	24.85	0.6965	**28.49**	**0.8541**
HSI	0.9	15.17	0.1136	16.22	0.2058	16.78	0.2532	18.08	0.3820	21.79	0.5447	19.23	0.4589	19.52	0.4825	**25.23**	**0.7161**
WDC	0.8	17.01	0.2531	18.63	0.3661	19.21	0.4118	23.38	0.6194	23.70	0.6761	22.51	0.5584	22.43	0.5571	**27.77**	**0.8158**
(256×256×191)	0.7	19.01	0.3966	20.97	0.5142	22.13	0.5812	25.41	0.7713	25.71	0.7831	22.98	0.6349	23.75	0.6632	**29.82**	**0.8671**
MSI	0.9	19.52	0.5148	17.60	0.4940	16.56	0.5750	20.40	0.5433	27.07	0.7957	23.48	0.6214	29.75	0.8112	**32.61**	**0.9063**
Flowers	0.8	25.95	0.6695	21.75	0.6029	20.06	0.6448	27.57	0.7796	31.58	0.8903	29.97	0.7896	34.55	0.9003	**35.21**	**0.9477**
(512×512×31)	0.7	30.38	0.8025	26.62	0.7277	23.11	0.7228	31.84	0.8652	34.41	0.9384	33.25	0.8680	37.95	0.9470	**37.66**	**0.9657**
MSI	0.9	17.50	0.2595	16.78	0.2638	16.62	0.3763	18.48	0.3733	20.46	0.4786	18.52	0.3958	20.18	0.4148	**24.02**	**0.6993**
Beans	0.8	20.53	0.4058	19.60	0.3898	19.76	0.5000	22.75	0.6191	21.34	0.5345	23.87	0.6258	25.22	0.6991	**25.43**	**0.7925**
(512×512×31)	0.7	23.15	0.5572	22.44	0.5424	22.85	0.6465	25.17	0.7443	23.50	0.6762	25.43	0.7548	26.82	0.7703	**28.43**	**0.8941**
MSI	0.9	24.47	0.7628	22.64	0.7461	21.84	0.7746	20.92	0.6728	26.32	0.8474	25.16	0.7840	33.45	0.9178	**35.85**	**0.9421**
Cd	0.8	30.75	0.8787	28.93	0.8532	27.69	0.8578	29.80	0.8926	31.11	0.8957	31.43	0.8879	36.07	0.9450	**37.88**	**0.9554**
(512×512×31)	0.7	34.19	0.9222	33.96	0.9213	32.54	0.9170	34.05	0.9370	34.33	0.9293	34.35	0.9250	37.78	0.9534	**40.46**	**0.9675**
MSI	0.9	18.87	0.5246	17.07	0.4434	16.60	0.6010	20.79	0.6326	23.44	0.7266	21.56	0.6443	26.93	0.7832	**29.88**	**0.9078**
Feathers	0.8	24.12	0.7148	20.61	0.6237	19.06	0.6841	27.08	0.8344	26.73	0.8308	24.63	0.7318	31.02	0.8814	**33.13**	**0.9512**
(512×512×31)	0.7	28.44	0.8355	24.40	0.7620	21.12	0.7622	30.55	0.9081	29.30	0.8818	28.64	0.8431	33.71	0.9265	**35.92**	**0.9693**
MSI	0.9	17.18	0.2329	15.57	0.2091	14.29	0.2218	19.08	0.3494	19.36	0.3490	20.91	0.4032	21.47	0.5203	**22.59**	**0.6127**
Cloth	0.8	20.13	0.3820	18.57	0.3407	16.51	0.3367	22.45	0.5523	21.26	0.5538	22.46	0.4932	24.21	0.5572	**25.02**	**0.7643**
(512×512×31)	0.7	22.17	0.5165	21.11	0.4866	18.65	0.4666	24.23	0.6678	22.99	0.6829	24.81	0.6149	26.93	0.7208	**27.01**	**0.8361**

The **best** and second-best values are highlighted.

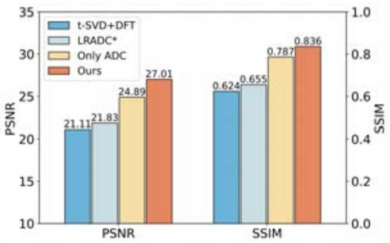


Fig. 5. Ablation study results on the MSI *Cloth* dataset under tube missing with MR = 0.7.

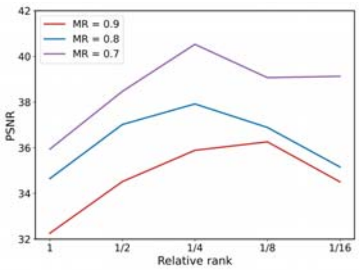
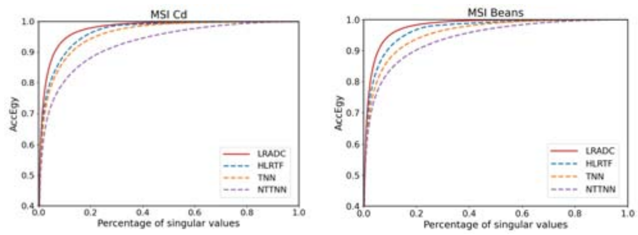


Fig. 6. Comparison of PSNR curves on the MSI *Cd* dataset under tube missing with different rank r.

the results are shown in Fig. 6. As shown, decreasing the rank can better exploit the low-rank prior and improve recovery performance. However, when the rank becomes too small, the performance starts to degrade, since an excessively low rank limits the model's expressive capacity and weakens its ability to capture global structural priors. In our experiments, we set the relative rank to $1/4$.

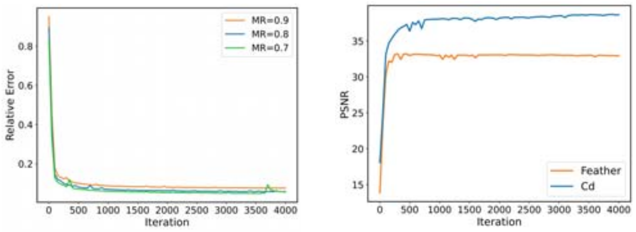
To verify the effect of low-rank tensor factorization in our method, we plot the energy accumulation curves of different methods, as shown in Fig. 7. Here, energy accumulation is defined as $\sum_{i=1}^{k}\sigma_i^2/\sum_j\sigma_j^2$, where σ_i denotes the i-th singular value. As observed, the energy of our method is more concentrated on larger singular values, thus exhibiting better low-rank performance.



(a) (b)

Fig. 7. Comparison of energy accumulation curves between LRADC and HLRTF. The horizontal axis denotes the percentage of singular values, and the vertical axis represents the cumulative energy ratio. (a) *Cd* dataset with a random missing MR = 0.7; (b) *Beans* dataset with a tube missing MR = 0.7.

3) Convergence Analysis: To demonstrate the convergence, we evaluate our model on the *Cd* and *Feathers* datasets under different missing patterns. Fig. 8 (a) shows the decreasing trend of the relative error with iterations in the *Cd* dataset under random missing rates of 0.7, 0.8, and 0.9. The results indicate that the error decreases rapidly in the early iterations and then gradually stabilizes, demonstrating that the proposed algorithm can quickly converge. Fig. 8 (b) illustrates the variation of PSNR with iterations in the *Cd* and *Feathers* datasets under the tube missing rate of 0.8. It can be observed



(a) (b)

Fig. 8. (a) Relative error curves on the *Cd* dataset with different missing rates under random missing scenario; (b) PSNR curves on the *Cd* and *Feathers* datasets under tube missing scenario with MR = 0.8.

that the PSNR for both datasets increases rapidly in the initial iterations and then reaches a stabilization phase, which verifies the good convergence and stability of the proposed method. Combining the two subfigures, it is evident that the proposed method can stably converge, reaching performance saturation within a reasonable number of iterations. This validates the effectiveness and reliability of the model optimization process.

4) Computational Efficiency Comparison: Table III compares the average running time of different methods. It shows that our method is significantly faster than NTTNN and S2NTNN, which is mainly due to the avoidance of expensive t-SVD computations. Although our method is slightly slower than HLRTF, it has clear advantages in recovery quality and accuracy. Overall, considering both running time and performance metrics, our method strikes a better balance of efficiency and effectiveness.

TABLE III
RUNNING TIME COMPARISON OF DIFFERENT METHODS.

Time(s)	MR	NTTNN	S2NTNN	HLRTF	Ours
HSI	0.9	65.21	39.48	18.21	20.23
WDC	0.8	60.12	39.14	18.95	20.56
$(256 \times 256 \times 191)$	0.7	48.43	38.87	18.73	20.18
MSI	0.9	38.51	28.76	9.72	11.29
Cd	0.8	22.26	28.69	9.52	11.53
$(512 \times 512 \times 31)$	0.7	19.31	28.71	9.49	11.28

IV. CONCLUSIONS

This paper proposes a new framework for multi-dimensional image recovery by integrating low-rank factorization and adaptive dynamic convolution (ADC), which is designed to address multi-dimensional tensor completion with high missing rates. The method leverages low-rank factorization to model global structural priors, while the ADC module dynamically enhances local details and cross-channel feature interaction. We develop an optimization algorithm based on the ADMM framework, with network parameters updated using Adam. Extensive experimental results demonstrate that LRADC outperforms state-of-the-art methods in terms of PSNR and SSIM, while exhibiting strong convergence and stability. Ablation studies further validate the necessity and complementarity of key components, confirming the effectiveness of combining global priors with adaptive feature learning.

REFERENCES

[1] H. Li, R. Feng, L. Wang, Y. Zhong, and L. Zhang, “Superpixel-based reweighted low-rank and total variation sparse unmixing for hyperspectral remote sensing imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 1, pp. 629–647, 2021.

[2] Z. Zhang, J. Liu, D. Yang, U. Kamilov, and G. Hugo, “Deep learning-based motion compensation for four-dimensional cone-beam computed tomography (4D-CBCT) reconstruction,” *Medical Physics*, vol. 50, no. 2, pp. 808–820, 2023.

[3] S. H. Lee, P. Oh, and M. G. Kang, “Three dimensional colorization based image/video reconstruction from white-dominant RGBW pattern images,” *Digital Signal Processing*, vol. 93, pp. 87–101, 2019.

[4] H. Chen and J. Li, “Neural tensor model for learning multi-aspect factors in recommender systems,” in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, 2020, pp. 2449–2455.

[5] J. Liu, P. Musialski, P. Wonka, and J. Ye, “Tensor completion for estimating missing values in visual data,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 208–220, 2013.

[6] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[7] L. R. Tucker, “Some mathematical notes on three-mode factor analysis,” *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[8] M. E. Kilmer and C. D. Martin, “Factorization strategies for third-order tensors,” *Linear Algebra and Its Applications*, vol. 435, no. 3, pp. 641–658, 2011.

[9] B. Madathil and S. N. George, “DCT based weighted adaptive multi-linear data completion and denoising,” *Neurocomputing*, vol. 318, pp. 120–136, 2018.

[10] D. Du, Z. Pan, P. Zhang, Y. Li, and W. Ku, “Compressive sensing image recovery using dictionary learning and shape-adaptive DCT thresholding,” *Magnetic Resonance Imaging*, vol. 55, pp. 60–71, 2019.

[11] G. Song, M. Ng, and X. Zhang, “Robust tensor completion using transformed tensor singular value decomposition,” *Numerical Linear Algebra with Applications*, vol. 27, no. 3, p. e2299, 2020.

[12] Y. Luo, X. Zhao, T. Jiang, Y. Chang, M. Ng, and C. Li, “Self-supervised nonlinear transform-based tensor nuclear norm for multi-dimensional image recovery,” *IEEE Transactions on Image Processing*, vol. 31, pp. 3793–3808, 2022.

[13] Y. Luo, X. Zhao, D. Meng, and T. Jiang, “HLRTF: Hierarchical low-rank tensor factorization for inverse problems in multi-dimensional imaging,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2022, pp. 19 303–19 312.

[14] Y. Luo, X. Zhao, Z. Li, M. Ng, and D. Meng, “Low-rank tensor function representation for multi-dimensional data recovery,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 5, pp. 3351–3369, 2024.

[15] J. Peng, Q. Xie, Q. Zhao, Y. Wang, L. Yee, and D. Meng, “Enhanced 3DTV regularization and its applications on hsi denoising and compressed sensing,” *IEEE Transactions on Image Processing*, vol. 29, pp. 7889–7903, 2020.

[16] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proceedings of the 3rd International Conference on Learning Representations(ICLR)*, 2015, pp. 1–15.

[17] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[18] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer, “Novel methods for multilinear data completion and de-noising based on tensor-svd,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2014, pp. 3842–3849.

[19] T. Jiang, M. Ng, X. Zhao, and T. Huang, “Framelet representation of tensor nuclear norm for third-order tensor completion,” *IEEE Transactions on Image Processing*, vol. 29, pp. 7233–7244, 2020.

[20] B. Li, X. Zhao, T. Ji, X. Zhang, and T. Huang, “Nonlinear transform induced tensor nuclear norm for tensor completion,” *Journal of Scientific Computing*, vol. 92, no. 3, p. 83, 2022.