

# Reparameterized Tensor Ring Functional Decomposition for Multi-Dimensional Data Recovery

Yangyang Xu<sup>1</sup> Junbo Ke<sup>1</sup> You-Wei Wen<sup>1</sup> Chao Wang<sup>2,\*</sup>

<sup>1</sup>Key Laboratory of Computing and Stochastic Mathematics (Ministry of Education),  
 School of Mathematics and Statistics, Hunan Normal University

<sup>2</sup>Department of Statistics and Data Science, Southern University of Science and Technology

## Abstract

Tensor Ring (TR) decomposition is a powerful tool for high-order data modeling, but is inherently restricted to discrete forms defined on fixed meshgrids. In this work, we propose a TR functional decomposition for both meshgrid and non-meshgrid data, where factors are parameterized by Implicit Neural Representations (INRs). However, optimizing this continuous framework to capture fine-scale details is intrinsically difficult. Through a frequency-domain analysis, we demonstrate that the spectral structure of TR factors determines the frequency composition of the reconstructed tensor and limits the high-frequency modeling capacity. To mitigate this, we propose a reparameterized TR functional decomposition, in which each TR factor is a structured combination of a learnable latent tensor and a fixed basis. This reparameterization is theoretically shown to improve the training dynamics of TR factor learning. We further derive a principled initialization scheme for the fixed basis and prove the Lipschitz continuity of our proposed model. Extensive experiments on image inpainting, denoising, super-resolution, and point cloud recovery demonstrate that our method achieves consistently superior performance over existing approaches. Code is available at <https://github.com/YangyangXu2002/RepTRFD>.

## 1. Introduction

Low-rank tensor representations have become a fundamental tool for modeling multi-dimensional data across a wide range of computer vision and signal processing tasks, such as image and video processing [18, 20], remote sensing [5, 10], medical imaging [24], and data compression [6, 13]. By exploiting the underlying low-dimensional structures, classical tensor decompositions such as CAN-DECOMP/PARAFAC (CP) [4, 11], Tucker [7, 39], Tensor Train (TT) [27], and Tensor Ring (TR) [46] provide com-

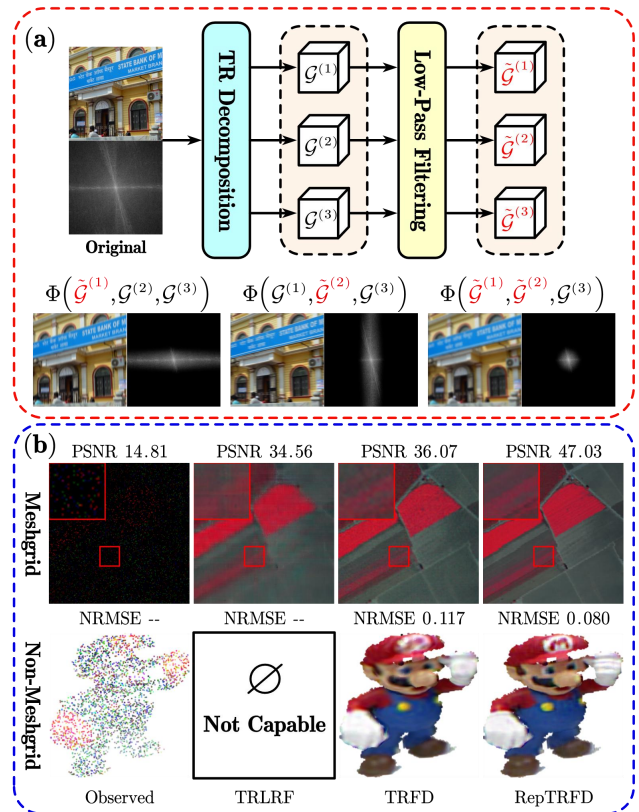


Figure 1. (a) Frequency analysis of TR factors. Low-pass filtering  $\{\mathcal{G}^{(k)}\}_{k=1}^3$  along mode-2 yields the low-frequency counterparts  $\{\tilde{\mathcal{G}}^{(k)}\}_{k=1}^3$ . Reconstructing via the contraction  $\Phi(\cdot)$  using these factors shows noticeable attenuation along the corresponding modes. (b) Qualitative comparison of TRLRF [45], TRFD, and RepTRFD for meshgrid and non-meshgrid data recovery. Compared with TRLRF, our methods (TRFD and RepTRFD) effectively handle non-meshgrid data and demonstrate marked improvements over the baseline.

pact representations and facilitate recovery from incomplete or corrupted data.

Among these, TR decomposition has been widely stud-

\*Corresponding author: [wangc6@sustech.edu.cn](mailto:wangc6@sustech.edu.cn).

ied for its ability to efficiently represent high-order tensors with compact structures [16, 28, 42, 45]. However, traditional TR formulations are inherently discrete and assume that data are defined on fixed grids, which limits their applicability to continuous signals and resolution-independent modeling. To address the limitation in discrete tensor decomposition, a general framework known as low-rank tensor functional representation has been proposed [21]. In this paradigm, tensor factors are modeled as Implicit Neural Representations (INRs) [37] that map continuous coordinates to latent factor values, effectively placing tensor decomposition in a continuous functional space. While this functional framework has been successfully extended to various decomposition formats [14, 40, 41] and improved with continuous regularizations [22, 23], extending TR decomposition to this continuous setting remains unexplored.

Inspired by existing functional tensor methods, we propose a TR Functional Decomposition (TRFD) for multidimensional image recovery. It is worth noting that establishing this decomposition is non-trivial due to the intrinsic properties of TR factors, and direct application of existing functional tensor methods often produces reconstructions dominated by low-frequency components. To further understand this limitation, we analyze TR decomposition from a frequency perspective. As illustrated in Fig. 1, replacing the original TR factors  $\{\mathcal{G}^{(k)}\}_{k=1}^3$  with their low-pass filtered versions  $\{\tilde{\mathcal{G}}^{(k)}\}_{k=1}^3$  yields reconstructions that exhibit noticeable attenuation along the corresponding modes. This finding reveals that the frequency characteristics of TR factors are directly inherited by the reconstructed tensor.

However, due to the inherent spectral bias of INRs [29], learning TR factors that capture appropriate high-frequency components remains highly challenging. Distinct from existing approaches, we take a novel perspective by revisiting the factor learning process through the lens of training dynamics. Specifically, we seek to transform the parameter space of the TR factors, so that the optimization dynamics can more effectively explore high-frequency directions. To achieve this, we propose Reparameterized TRFD (Rep-TRFD), in which each TR factor is expressed as a structured combination of a learnable latent tensor and a fixed basis. This reparameterization is theoretically shown to enhance the training dynamics, allowing the TR structure to fully exploit both meshgrid and continuous data. As illustrated in Fig. 1, it significantly improves the reconstruction of high-frequency details. By incorporating this, we offer a novel perspective for advancing the study of tensor functional representations. In summary, our main contributions are as follows:

- We extend the TR decomposition to the continuous domain and provide a frequency-domain analysis of the challenge in learning TR factors that capture high-frequency components.

- We propose a reparameterization strategy for TR factors, where each factor is expressed as a structured combination of a learnable tensor and a fixed basis, which facilitates the learning of high-frequency details.
- We theoretically show that this reparameterization enhances the training dynamics, and further derive a principled initialization scheme while providing the Lipschitz continuity guarantees. Extensive experiments demonstrate the superiority of our approach.

## 2. Related Work

**Implicit Neural Representations.** INRs have recently emerged as a powerful paradigm for representing continuous signals by mapping coordinates to values [37]. Subsequent works have sought to enhance the representational power and efficiency of INRs through improved activation functions [17, 30, 33], encoding schemes [25, 38], and training dynamics [3, 35]. Despite their expressive power, learning global mappings over entire signals still exhibits limited computational efficiency, and the lack of structural priors often limits performance in inverse problems.

**Tensor Functional Representations.** Recent works have extended low-rank tensor representations to the continuous domain. Unlike INRs that map coordinates to the entire signal, these methods learn coordinate-to-factor mappings, enabling continuous modeling with lower computational cost and inherent low-rank priors. Representative approaches include the low-rank tensor function representation (LRTFR) [21] based on Tucker decomposition, a functional transform-based factorization [41] that improves expressiveness, a parameter-efficient deep rank-one factorization (DRO-TFF) [14] that reduces storage and computation overhead, and unified functional extensions of CP and TT decompositions [40]. Moreover, these functional tensor formulations can be synergistically combined with continuous regularization methods, such as neural total variation (NeurTV) [23] and continuous representation-based nonlocal (CRNL) [22] priors, to further enhance reconstruction quality and structural consistency.

**Reparameterization and Training Dynamics.** Reparameterization techniques have been widely adopted in deep learning to improve training dynamics. Early work by Salimans et al. [31] first stabilized optimization by decoupling the magnitude and direction of weights through weight normalization. Following this, Ding et al. [8] further extended it to structural reparameterization with parallel branches. Mostafa [26] explored another perspective by decoupling weight optimization from sparse connectivity learning. More recently, Shi et al. [35] factorized network weights into a learnable matrix and a fixed Fourier basis, enhancing learning behavior in INRs. In addition, several works have explored alternative approaches to improving the training dynamics of INRs and achieved promising

results [3, 15, 36]. In this work, we take the first step toward analyzing low-rank tensor functional representations through training dynamics, highlighting the potential of reparameterization for more effective factor learning.

### 3. Methodology

#### 3.1. Notations and Preliminaries

To enhance readability, we denote scalars by lowercase letters (e.g.,  $x$ ), vectors by bold lowercase (e.g.,  $\mathbf{x}$ ), matrices by bold uppercase (e.g.,  $\mathbf{X}$ ), and tensors by calligraphic letters (e.g.,  $\mathcal{X}$ ). Given a  $d$ -th order tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d}$ , its  $\mathbf{v}$ -th entry is written as  $\mathcal{X}_{\mathbf{v}}$  where  $\mathbf{v} = [v_1, v_2, \dots, v_d]^T$ . We also define the mode- $k$  slice at index  $v_k$  as  $\mathcal{X}_{:v_k:} \in \mathbb{R}^{n_1 \times \dots \times n_{k-1} \times n_{k+1} \times \dots \times n_d}$ . In addition,  $\|\mathcal{X}\|_{\infty}$  is the infinity norm of  $\mathcal{X}$  by calculating the maximum magnitude of the tensor's entries. We next introduce two fundamental definitions used throughout this paper.

**Definition 1** (Tensor Ring Decomposition [46]). Consider a  $d$ -th order tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d}$ , TR decomposition represents  $\mathcal{X}$  using third-order cores  $\mathcal{G}^{(k)} \in \mathbb{R}^{r_k \times n_k \times r_{k+1}}$  for  $k = 1, 2, \dots, d$ , with the circular constraint  $r_{d+1} = r_1$ . The tuple  $[r_1, r_2, \dots, r_d]$  is referred to as the TR rank, which controls the representation capacity. Each element of  $\mathcal{X}$  is given by

$$\mathcal{X}_{\mathbf{v}} = \text{trace} \left( \mathcal{G}_{:v_1:}^{(1)} \mathcal{G}_{:v_2:}^{(2)} \dots \mathcal{G}_{:v_d:}^{(d)} \right), \quad (1)$$

where  $\text{trace}(\cdot)$  denotes the matrix trace. For simplicity, the decomposition is presented as

$$\mathcal{X} = \Phi(\mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \dots, \mathcal{G}^{(d)}), \quad (2)$$

where  $\Phi(\cdot)$  denotes the TR contraction operation.

**Definition 2** (Mode- $k$  Discrete Fourier Transform [13, 19]). Given a  $d$ -th order tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d}$ , its mode- $k$  Discrete Fourier Transform (DFT) is defined as

$$\mathcal{F}_k[\mathcal{X}] = \mathcal{X} \times_k \mathbf{F}_k, \quad (3)$$

where  $\mathbf{F}_k \in \mathbb{C}^{n_k \times n_k}$  is the DFT matrix, and  $\times_k$  is the  $k$ -mode product between tensor and matrix.

#### 3.2. Tensor Ring Functional Decomposition

Traditional TR decomposition assumes that each factor  $\mathcal{G}^{(k)}$  is a discrete tensor defined on a fixed grid. To extend the discrete TR model into a continuous representation, we reinterpret each TR factor as a functional tensor parameterized by INRs. Since the TR factors are inherently linked through cyclic contractions, we introduce a shared frequency embedding to enhance their consistency across modes.

As illustrated in Fig. 2, given a coordinate vector  $\mathbf{v} = [v_1, v_2, \dots, v_d]^T$ , the shared embedding for the  $k$ -th dimension is implemented by a single sinusoidal layer:

$$\mathbf{z}_k = \sin(\omega_0(\mathbf{w}v_k + \mathbf{b})) \in \mathbb{R}^h, \quad k = 1, \dots, d, \quad (4)$$

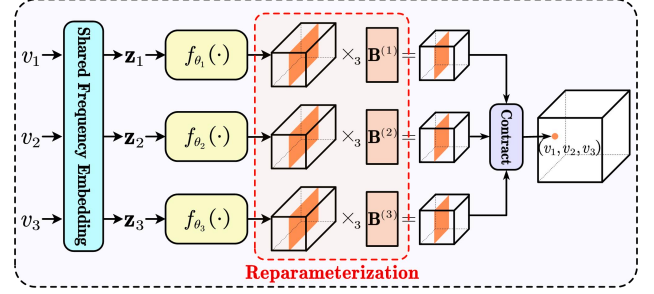


Figure 2. Illustration of proposed TRFD and RepTRFD. TRFD directly learns a continuous mapping from coordinates to TR factors. In contrast, RepTRFD reparameterizes each TR factor as a structured combination of a learnable latent tensor and a fixed basis, which improves the training dynamics and facilitates more effective learning of high-frequency components.

where  $\omega_0$  is a frequency scaling parameter,  $\mathbf{w} \in \mathbb{R}^h$  and  $\mathbf{b} \in \mathbb{R}^h$  are learnable weight and bias, and  $h$  denotes the hidden dimension of the sinusoidal layer. To generate each TR factor, we employ a separate branch network for each mode. Each branch is a multi-layer perceptron (MLP) that maps the corresponding embedding  $\mathbf{z}_k$  to a vector of length  $r_k r_{k+1}$ , which is then reshaped into the desired TR slice:

$$\mathcal{G}_{:v_k:}^{(k)} = f_{\theta_k}(\mathbf{z}_k) = g_{\theta_k}(v_k) \in \mathbb{R}^{r_k \times r_{k+1}}, \quad v_k \in \{1, \dots, n_k\}, \quad (5)$$

where  $f_{\theta_k}(\cdot)$  denotes the MLP associated with the  $k$ -th tensor mode, producing the mode-specific factor slice from the shared embedding, and  $g_{\theta_k}(\cdot)$  is the function composed with MLP and shared frequency embedding. This approach naturally extends the TR decomposition to a continuous representation, which we refer to as the Tensor Ring Functional Decomposition (TRFD).

#### 3.3. Reparameterization for Factor Learning

**Frequency Analysis of TR Factors.** While the TR functional decomposition provides a flexible and compact framework for representing high-dimensional data, its practical effectiveness is often constrained by the inherent spectral bias of the INRs used to parameterize the factors. INRs tend to prioritize low-frequency components while under-representing high-frequency details, leading to limited expressiveness within the individual TR factors. Crucially, this spectral limitation does not remain confined to the factor level, because it propagates through the cyclic contractions of the TR structure and influences the frequency characteristics of the reconstructed tensor  $\mathcal{X}$ . To better understand this phenomenon, we formally analyze how the frequency content of the TR factors propagates through the reconstruction process. The following theorem presents the corresponding characterization.

**Theorem 1.** Let  $\mathcal{X} = \Phi(\mathcal{G}^{(1)}, \dots, \mathcal{G}^{(d)})$  be a TR decomposition. Suppose that the mode-2 frequency components of

$\mathcal{G}^{(k)}$  beyond a threshold  $\Omega_k$  are negligible for all  $k$ , i.e.,

$$\|(\mathcal{F}_2[\mathcal{G}^{(k)}])_{:\omega_k}\|_\infty \leq \epsilon, \quad \forall |\omega_k| > \Omega_k, k = 1, 2, \dots, d,$$

where  $\epsilon > 0$  is a small constant. Then the reconstructed tensor  $\mathcal{X}$  also exhibits attenuated high-frequency content along mode  $k$ :

$$\|(\mathcal{F}_k[\mathcal{X}])_{:\omega_k}\|_\infty \leq c_k \epsilon, \quad \forall |\omega_k| > \Omega_k, k = 1, 2, \dots, d,$$

where  $c_k$  is a constant depending only on the magnitudes of the remaining cores  $\{\mathcal{G}^{(j)}\}_{j \neq k}$ .

All proofs of the theoretical results presented in this section are provided in the supplementary material. Theorem 1 indicates that the spectral content of each TR factor directly affects the corresponding dimension of the reconstructed tensor. Consequently, achieving accurate reconstruction of high-frequency tensors is contingent on the TR factors themselves containing sufficiently high-frequency components. However, due to the inherent spectral bias of standard INRs, learning such factors is challenging. This motivates the design of strategies that facilitate the acquisition of high-frequency information.

**Reparameterized Tensor Ring.** To address this challenge, we aim to transform the parameter space of the TR factors to improve the training dynamics, thereby facilitating more efficient learning of the TR factors. Specifically, by introducing a fixed basis  $\mathbf{B}^{(k)} \in \mathbb{R}^{r_{k+1} \times R_{k+1}}$ , we reparameterize each TR factor as a combination of a learnable latent tensor and this basis, which is expressed as

$$\mathcal{G}^{(k)} = \mathcal{C}^{(k)} \times_3 \mathbf{B}^{(k)}, \quad k = 1, \dots, d. \quad (6)$$

This decomposition separates the learnable component from the fixed basis, allowing each TR factor to be expressed as a structured combination of latent elements. Such a formulation improves the conditioning of the optimization and facilitates more effective learning of high-frequency components, as shown in the following theorem.

**Theorem 2.** Consider a TR factor  $\mathcal{G}$  reparameterized as  $\mathcal{G} = \mathcal{C} \times_3 \mathbf{B}$ , where  $\mathcal{C} \in \mathbb{R}^{r_1 \times n \times R_2}$  is a trainable tensor and  $\mathbf{B} \in \mathbb{R}^{r_2 \times R_2}$  is a fixed basis. Let  $\mathcal{L}(\omega)$  be the loss associated with frequency  $\omega$ . For any  $\omega_{\text{high}} > \omega_{\text{low}} > 0$ , given any  $\epsilon \geq 0$ , and fixed indices  $p, q$  with  $1 \leq p \leq r_1$  and  $1 \leq q \leq n$ , there exists a matrix  $\mathbf{B}$  such that for all  $s = 1, \dots, R_2$ ,

$$\left| \frac{\partial \mathcal{L}(\omega_{\text{high}})}{\partial \mathcal{C}_{pq s}} \right| \Big/ \left| \frac{\partial \mathcal{L}(\omega_{\text{low}})}{\partial \mathcal{C}_{pq s}} \right| \geq \max_{j=1, \dots, r_2} \left| \frac{\partial \mathcal{L}(\omega_{\text{high}})}{\partial \mathcal{G}_{pq j}} \right| \Big/ \left| \frac{\partial \mathcal{L}(\omega_{\text{low}})}{\partial \mathcal{G}_{pq j}} \right| - \epsilon.$$

Theorem 2 demonstrates that our reparameterization amplifies the gradient response to high-frequency components relative to low-frequency ones. This implies that the optimization becomes more responsive to fine-scale variations,

effectively improving the convergence behavior when learning TR factors with rich spectral content. To realize these benefits in practice, we need to specify how to determine the latent tensor  $\mathcal{C}^{(k)}$  and fixed basis  $\mathbf{B}^{(k)}$ .

Similar to TRFD, we represent the latent tensor  $\mathcal{C}^{(k)}$  as generated slice by slice through a neural mapping. Specifically, given a coordinate  $v_k$ , the corresponding slice  $\mathcal{C}_{:v_k}^{(k)}$  is produced by applying a shared frequency embedding and an MLP as:

$$\mathcal{C}_{:v_k}^{(k)} = g_{\theta_k}(v_k) \in \mathbb{R}^{r_k \times R_{k+1}}, v_k \in \{1, \dots, n_k\}. \quad (7)$$

The full latent tensor  $\mathcal{C}^{(k)} \in \mathbb{R}^{r_k \times n_k \times R_{k+1}}$  is formed by stacking all slices along the  $k$ -th mode, where  $R_{k+1} = \beta r_{k+1}$  with  $\beta \geq 1$ . For brevity, the overall mapping from the coordinate indices  $\mathbf{v}_k = [1, \dots, n_k] \in \mathbb{R}^{n_k}$  to the latent tensor is denoted as

$$\mathcal{C}^{(k)} = g_{\phi_k}(\mathbf{v}_k) \in \mathbb{R}^{r_k \times n_k \times R_{k+1}}, \quad (8)$$

emphasizing that the latent tensor is generated through a learnable neural mapping.

While Theorem 2 guarantees the existence of a basis that can amplify high-frequency gradients, an arbitrary choice may lead to poor scaling or variance inconsistency during training. To this end, we adopt a Xavier-style initialization [9] for the entries of  $\mathbf{B}^{(k)}$ , as formalized in the following theorem.

**Theorem 3.** Suppose the entries of  $\mathbf{B}^{(k)}$  in the reparameterized TR factor are sampled independently from a uniform distribution as

$$\mathbf{B}_{ij}^{(k)} \sim \mathcal{U}\left(-\sqrt{\frac{6}{r_{k+1} + R_{k+1}}}, \sqrt{\frac{6}{r_{k+1} + R_{k+1}}}\right), \forall i, j, k,$$

then the variances in both the forward and backward passes are preserved.

To summarize the above developments, we refer to the proposed framework as the Reparameterized Tensor Ring Functional Decomposition (RepTRFD), with its overall architecture illustrated in Fig. 2. While the previous analyses focus on gradient dynamics and initialization, it is also essential to understand the overall stability of the reparameterized formulation as a functional mapping. To ensure that the reparameterized formulation remains well-behaved and avoids excessive sensitivity to input perturbations, we establish its Lipschitz continuity property in the following theorem.

**Theorem 4.** Let  $g_\phi(\mathbf{v}) : \mathbb{R}^d \rightarrow \mathbb{R}$  be defined as

$$g_\phi(\mathbf{v}) = \Phi(g_{\phi_1}(v_1) \times_3 \mathbf{B}^{(1)}, \dots, g_{\phi_d}(v_d) \times_3 \mathbf{B}^{(d)}),$$

where  $\mathbf{v} = [v_1, \dots, v_d]^T \in \mathbb{R}^d$  denotes the coordinate vector, and each  $\mathbf{B}^{(k)} \in \mathbb{R}^{r_{k+1} \times R_{k+1}}$  is a fixed basis. Assume for each mode  $k = 1, \dots, d$ , the following conditions hold:

- $g_{\phi_k}(\cdot)$  is an  $L_k$ -layer MLP with activation function  $\sigma(\cdot)$  that is  $\kappa$ -Lipschitz continuous;
  - the spectral norm of each weight matrix in  $g_{\phi_k}(\cdot)$  is bounded by  $\eta$ ;
  - the output is bounded:  $\sup_{v_k} \|g_{\phi_k}(v_k)\|_F \leq C_k < \infty$ .
- Then,  $g_\phi(\mathbf{v})$  is globally Lipschitz continuous, i.e.,

$$|g_\phi(\mathbf{v}) - g_\phi(\mathbf{v}')| \leq \delta \|\mathbf{v} - \mathbf{v}'\|_2,$$

where the global Lipschitz constant  $\delta = \sqrt{\sum_{k=1}^d \delta_k^2}$ , and  $\delta_k = (\kappa\eta)^{L_k} \cdot \|\mathbf{B}_k\|_2 \cdot \left(\prod_{j \neq k} C_j\right)$ .

Given the RepTRFD representation  $g_\phi(\{\mathbf{v}_k\}_{k=1}^d) = \Phi(g_{\phi_1}(\mathbf{v}_1) \times_3 \mathbf{B}^{(1)}, \dots, g_{\phi_d}(\mathbf{v}_d) \times_3 \mathbf{B}^{(d)})$ , we introduce a general formulation on the loss designs used for various low-level vision and 3D recovery tasks. Each task is formulated as an optimization problem consisting of a data fidelity term  $\mathcal{L}_{\text{data}}$  and an optional regularization term  $\mathcal{L}_{\text{reg}}$ , expressed as:

$$\min_{\phi} \mathcal{L}_{\text{data}}(g_\phi(\{\mathbf{v}_k\}_{k=1}^d; \mathcal{O})) + \mathcal{L}_{\text{reg}}(g_\phi(\{\mathbf{v}_k\}_{k=1}^d)), \quad (9)$$

where  $\mathbf{v}$  denotes the coordinate input,  $\phi$  includes all the learnable parameters, and  $\mathcal{O}$  denotes the given observations. Depending on the specific recovery task, the regularization term can be omitted or designed to impose appropriate priors on the spatial or spectral structure [1]. For more details, please refer to the supplementary materials.

## 4. Experimental Results

In this section, we evaluate the proposed method on four representative tasks, including inpainting, denoising, super-resolution, and point cloud recovery. Training is performed using the Adam optimizer [12], and all experiments are run on a workstation equipped with an Intel Core i9-12900K CPU, 62 GB RAM, and an NVIDIA RTX 4090 GPU.

### 4.1. Parameter Setting

The main parameters of our model include the TR rank  $[r_1, \dots, r_d]$ , the latent tensor expansion factor  $\beta$ , and the sinusoidal frequency  $\omega_0$ . For all modes  $k = 1, \dots, d$ , we set  $r_k \in \{16, 20\}$  and  $R_k = \beta r_k$  with  $\beta \in \{3, 5, 10\}$ . The frequency parameter  $\omega_0$  is set to 90, 120, or 240, and each branch has one or two layers with 256 hidden units. The model is optimized with a fixed learning rate of  $3 \times 10^{-4}$ . Input coordinates are normalized to the range  $[-1, 1]$ . Detailed parameter settings for each task are provided in the supplementary material.

### 4.2. Comparisons with State-of-the-Arts

**Image and Video Inpainting Results.** We first evaluate our method on the inpainting task under various sampling ratios

Table 1. Quantitative inpainting results on all datasets.

| Data                                                                | Method  | SR = 0.1     |              | SR = 0.2     |              | SR = 0.3     |              | Time (s) |
|---------------------------------------------------------------------|---------|--------------|--------------|--------------|--------------|--------------|--------------|----------|
|                                                                     |         | PSNR         | SSIM         | PSNR         | SSIM         | PSNR         | SSIM         |          |
| Airplane<br>House<br>Pepper<br>Sailboat<br>(256 × 256 × 3)          | TRLRF   | 17.52        | 0.326        | 21.57        | 0.546        | 24.85        | 0.698        | 5.12     |
|                                                                     | FCTN    | 17.38        | 0.359        | 21.23        | 0.566        | 24.67        | 0.710        | 5.44     |
|                                                                     | HLRTF   | 20.89        | 0.544        | 25.27        | 0.754        | 28.02        | 0.841        | 3.43     |
|                                                                     | LRTFR   | 21.38        | 0.587        | 25.86        | 0.770        | 28.74        | 0.856        | 6.03     |
|                                                                     | DRO-TFF | 23.22        | 0.734        | 27.52        | 0.848        | 30.04        | 0.899        | 13.25    |
|                                                                     | NeurTV  | <u>24.16</u> | <u>0.797</u> | <u>27.81</u> | <u>0.884</u> | <u>30.28</u> | <u>0.924</u> | 31.57    |
|                                                                     | Ours    | <b>25.70</b> | <b>0.841</b> | <b>29.37</b> | <b>0.915</b> | <b>32.01</b> | <b>0.944</b> | 13.11    |
| Data                                                                | Method  | SR = 0.05    |              | SR = 0.1     |              | SR = 0.15    |              | Time (s) |
|                                                                     |         | PSNR         | SSIM         | PSNR         | SSIM         | PSNR         | SSIM         |          |
| Toys<br>Flowers<br>(256 × 256 × 31)                                 | TRLRF   | 28.95        | 0.748        | 33.40        | 0.869        | 36.17        | 0.922        | 20.16    |
|                                                                     | FCTN    | 31.71        | 0.838        | 36.77        | 0.929        | 39.33        | 0.956        | 15.34    |
|                                                                     | HLRTF   | 34.89        | 0.937        | 40.86        | 0.980        | 43.48        | 0.988        | 4.12     |
|                                                                     | LRTFR   | 36.38        | 0.952        | 40.71        | 0.980        | 42.48        | 0.984        | 7.04     |
|                                                                     | DRO-TFF | <u>38.45</u> | <u>0.982</u> | <u>42.28</u> | <u>0.992</u> | <u>45.00</u> | <u>0.994</u> | 14.57    |
|                                                                     | NeurTV  | 37.49        | 0.964        | 41.66        | 0.984        | 43.72        | 0.988        | 33.56    |
|                                                                     | Ours    | <b>39.34</b> | <b>0.984</b> | <b>44.66</b> | <b>0.993</b> | <b>47.74</b> | <b>0.995</b> | 14.49    |
| Washington DC<br>(256 × 256 × 191)<br>Botswana<br>(256 × 256 × 145) | TRLRF   | 29.64        | 0.776        | 32.24        | 0.899        | 34.72        | 0.938        | 98.45    |
|                                                                     | FCTN    | 32.42        | 0.905        | 34.94        | 0.942        | 36.30        | 0.956        | 57.55    |
|                                                                     | HLRTF   | 37.36        | 0.970        | 40.14        | 0.984        | 42.09        | 0.988        | 18.36    |
|                                                                     | LRTFR   | 36.62        | 0.962        | 39.11        | 0.976        | 42.56        | 0.986        | 17.63    |
|                                                                     | DRO-TFF | <u>38.59</u> | <u>0.972</u> | <u>41.23</u> | <u>0.984</u> | <u>42.11</u> | <u>0.986</u> | 56.58    |
|                                                                     | NeurTV  | 37.79        | <u>0.972</u> | <u>41.99</u> | <u>0.987</u> | <u>43.33</u> | <u>0.989</u> | 45.51    |
|                                                                     | Ours    | <b>40.75</b> | <b>0.984</b> | <b>44.22</b> | <b>0.990</b> | <b>45.83</b> | <b>0.992</b> | 56.45    |
| News<br>Carphone<br>(144 × 176 × 100)                               | TRLRF   | 24.59        | 0.716        | 26.58        | 0.790        | 27.95        | 0.837        | 21.30    |
|                                                                     | FCTN    | 25.63        | 0.752        | 27.41        | 0.815        | 28.73        | 0.854        | 16.88    |
|                                                                     | HLRTF   | 26.16        | 0.787        | 28.88        | 0.865        | 30.45        | 0.898        | 4.70     |
|                                                                     | LRTFR   | 26.48        | 0.798        | 28.70        | 0.865        | 29.95        | 0.889        | 6.80     |
|                                                                     | DRO-TFF | <u>28.79</u> | <u>0.892</u> | <u>30.13</u> | <u>0.915</u> | <u>31.48</u> | <u>0.932</u> | 11.83    |
|                                                                     | NeurTV  | 27.48        | 0.818        | 29.44        | 0.875        | 30.55        | 0.907        | 16.01    |
|                                                                     | Ours    | <b>29.77</b> | <b>0.917</b> | <b>31.64</b> | <b>0.939</b> | <b>32.85</b> | <b>0.951</b> | 15.59    |

(SRs) using color images, multispectral images (MSIs), hyperspectral images (HSIs), and videos. Datasets are taken from USC-SIPI<sup>1</sup>, CAVE<sup>2</sup>, Remote Sensing Scenes<sup>3</sup>, and YUV sequences<sup>4</sup>. All images are resized or cropped for consistent input sizes, and 100 consecutive frames are used for video evaluation. We compare our approach with representative tensor completion models such as TRLRF [45] and FCTN [47], as well as recent neural tensor representations including HLRTF [20], LRTFR [21], DRO-TFF [14], and NeurTV [23]. All methods are tested under identical sampling masks, and performance is assessed using peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) [44].

Fig. 3 shows the inpainting results on the color image *Airplane* and the video *News* at different SRs. For clearer comparison, zoomed-in regions and their corresponding residual maps are provided. The proposed method recovers sharper details and finer textures than existing approaches, achieving around 2 dB higher PSNR than the best competing method. Table 1 reports the quantitative inpainting results on all datasets. Our method consistently attains the highest PSNR and SSIM for all data types and SRs, while keeping computational cost at a moderate level.

<sup>1</sup><https://sipi.usc.edu/database/>

<sup>2</sup><https://cave.cs.columbia.edu/repository/Multispectral>

<sup>3</sup>[https://www.ehu.es/ccwintco/index.php/Hyperspectral\\_Remote\\_Sensing\\_Scenes](https://www.ehu.es/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes)

<sup>4</sup><http://trace.eas.asu.edu/yuv/>

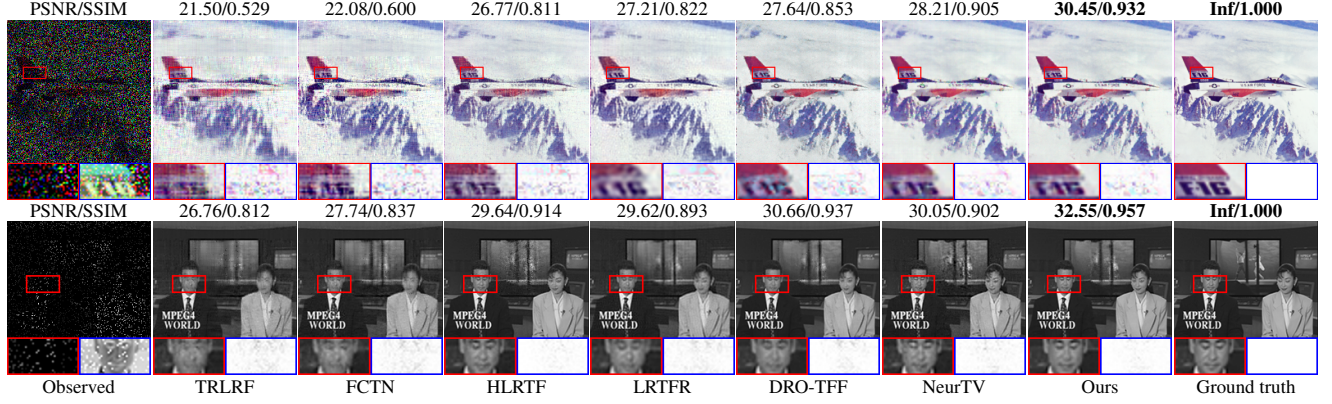


Figure 3. Visual inpainting results on color image *Airplane* (SR = 0.2) and video *News* (SR = 0.1).

Table 2. Average quantitative results for MSI and HSI denoising under different Gaussian noise levels.

| Data                                                                             | Method     | SD = 0.1     |              | SD = 0.2     |              | SD = 0.3     |              | Time (s) |
|----------------------------------------------------------------------------------|------------|--------------|--------------|--------------|--------------|--------------|--------------|----------|
|                                                                                  |            | PSNR         | SSIM         | PSNR         | SSIM         | PSNR         | SSIM         |          |
| <i>Toys</i><br><i>Face</i><br>(256 × 256 × 31)                                   | LRTDTV     | 35.85        | 0.895        | 31.27        | 0.718        | 28.45        | 0.554        | 5.97     |
|                                                                                  | DeepTensor | 36.23        | 0.914        | 32.48        | 0.833        | 30.25        | 0.702        | 47.93    |
|                                                                                  | HLRTF      | 37.09        | 0.917        | 33.28        | 0.828        | 30.88        | 0.726        | 4.55     |
|                                                                                  | LRTFR      | 37.55        | 0.935        | 33.75        | 0.864        | 31.80        | 0.819        | 5.86     |
|                                                                                  | DRO-TFF    | 38.67        | 0.959        | 34.74        | 0.911        | 32.91        | 0.881        | 7.61     |
|                                                                                  | NeurTV     | 38.61        | 0.960        | 34.98        | 0.915        | 33.37        | 0.883        | 49.20    |
|                                                                                  | Ours       | <b>39.02</b> | <b>0.964</b> | <b>35.91</b> | <b>0.933</b> | <b>34.08</b> | <b>0.905</b> | 8.79     |
| <i>Washington DC</i><br>(256 × 256 × 191)<br><i>Salinas</i><br>(217 × 217 × 224) | LRTDTV     | 37.23        | 0.941        | 34.27        | 0.895        | 32.56        | 0.857        | 21.13    |
|                                                                                  | DeepTensor | 38.13        | 0.954        | 34.69        | 0.906        | 32.81        | 0.856        | 53.67    |
|                                                                                  | HLRTF      | 38.23        | 0.958        | 35.12        | 0.921        | 33.45        | 0.893        | 17.21    |
|                                                                                  | LRTFR      | 38.14        | 0.958        | 35.50        | 0.928        | 33.76        | 0.901        | 16.32    |
|                                                                                  | DRO-TFF    | 38.52        | 0.954        | 36.35        | 0.934        | 34.95        | 0.911        | 34.23    |
|                                                                                  | NeurTV     | 39.36        | 0.963        | 36.75        | 0.932        | 34.78        | 0.899        | 259.53   |
|                                                                                  | Ours       | <b>40.37</b> | <b>0.972</b> | <b>37.63</b> | <b>0.950</b> | <b>35.98</b> | <b>0.928</b> | 35.76    |

**Multispectral Image Denoising Results.** We further evaluate the proposed method on denoising tasks with additive Gaussian noise of varying standard deviations (SDs). Six representative baselines are considered for comparison: LRTDTV [43], DeepTensor [34], HLRTF [20], LRTFR [21], DRO-TFF [14], and NeurTV [23]. Fig. 4 presents visual comparisons on the MSI *Face* and HSI *Washington DC* datasets under Gaussian noise with SD = 0.2, while quantitative results are summarized in Table 2. The proposed approach consistently attains the highest PSNR and SSIM, outperforming the strongest baseline by approximately 1 dB on average across all noise levels.

**Image Super Resolution Results.** We further evaluate our method on single-image super-resolution using samples from the DIV2K dataset [2], where high-resolution images are cropped to 768 × 768. The proposed method is compared with several INR-based baselines, including PEMLP [25], SIREN [37], Gauss [30], WIRE [33], FINER [17], and LRTFR [21]. Fig. 5 presents visual results on the *Lion* and *Parrot* images at ×4 scaling, where our method recovers sharper edges and finer textures, effectively mitigating over-smoothing and aliasing artifacts. Quantitatively, it surpasses the strongest baseline by approximately 1 dB in PSNR on average. Detailed results on *Lion* under different scaling

Table 3. Super-resolution results on *Lion* at different scales.

| Method | ×3           |              | ×4           |              | ×6           |              | Time (s) |
|--------|--------------|--------------|--------------|--------------|--------------|--------------|----------|
|        | PSNR         | SSIM         | PSNR         | SSIM         | PSNR         | SSIM         |          |
| PEMLP  | 29.26        | 0.740        | 27.84        | 0.676        | 24.28        | 0.439        | 90.15    |
| SIREN  | 31.69        | 0.866        | 29.48        | 0.779        | 27.16        | 0.634        | 125.38   |
| Gauss  | 30.27        | 0.790        | 28.23        | 0.683        | 26.73        | 0.605        | 186.92   |
| WIRE   | 32.43        | 0.869        | 30.21        | 0.788        | 27.88        | 0.683        | 423.27   |
| FINER  | 32.09        | 0.858        | 29.84        | 0.778        | 27.61        | 0.666        | 165.08   |
| LRTFR  | 30.11        | 0.805        | 28.10        | 0.701        | 25.27        | 0.577        | 10.26    |
| Ours   | <b>33.41</b> | <b>0.907</b> | <b>31.01</b> | <b>0.836</b> | <b>28.34</b> | <b>0.711</b> | 12.65    |

Table 4. NRMSE results for point cloud recovery on the SHOT dataset under SR = 0.2.

| Method | <i>Doll</i>  | <i>Duck</i>  | <i>Frog</i>  | <i>Mario</i> | <i>PeterRabbit</i> | <i>Squirrel</i> |
|--------|--------------|--------------|--------------|--------------|--------------------|-----------------|
| PEMLP  | 0.112        | 0.064        | 0.058        | 0.091        | 0.081              | 0.084           |
| SIREN  | 0.111        | 0.062        | 0.057        | 0.089        | 0.074              | 0.085           |
| Gauss  | 0.123        | 0.064        | 0.060        | 0.092        | 0.078              | 0.085           |
| WIRE   | <u>0.106</u> | 0.060        | <u>0.053</u> | <u>0.086</u> | <b>0.068</b>       | <u>0.083</u>    |
| FINER  | 0.110        | <u>0.059</u> | 0.054        | 0.088        | <u>0.073</u>       | <u>0.083</u>    |
| LRTFR  | 0.114        | 0.061        | 0.061        | 0.095        | 0.082              | 0.094           |
| Ours   | <b>0.093</b> | <b>0.053</b> | <b>0.050</b> | <b>0.080</b> | <b>0.068</b>       | <b>0.073</b>    |

factors are reported in Table 3. Overall, by leveraging a low-rank functional representation, the proposed model consistently achieves higher PSNR and SSIM across all scales while being substantially faster than INR-based baselines.

**Point Cloud Recovery Results.** This task aims to learn a continuous mapping for reconstructing complete point clouds from sparsely observed samples, where discrete tensor representations often fail to generalize. Following the setup in [22, 23], we employ the SHOT dataset [32] and evaluate performance under different SRs. The performance is measured by normalized root mean square error (NRMSE). Fig. 6 presents visual comparisons on the *Duck* (SR = 0.15) and *Mario* (SR = 0.2) point clouds, where the proposed method yields the most faithful reconstructions, accurately recovering fine geometric structures and smooth surfaces. Quantitative results summarized in Table 4 further confirm that our approach consistently achieves the lowest reconstruction error across all datasets under SR = 0.2, demonstrating its strong capability in continuous 3D signal recovery.

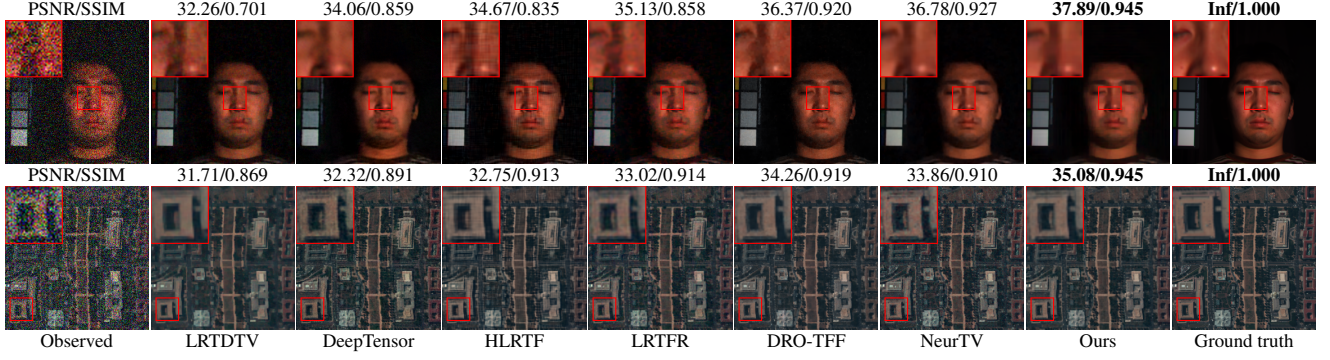


Figure 4. Visual denoising results on MSI *Face* and HSI *Washington DC* ( $SD = 0.2$ ).

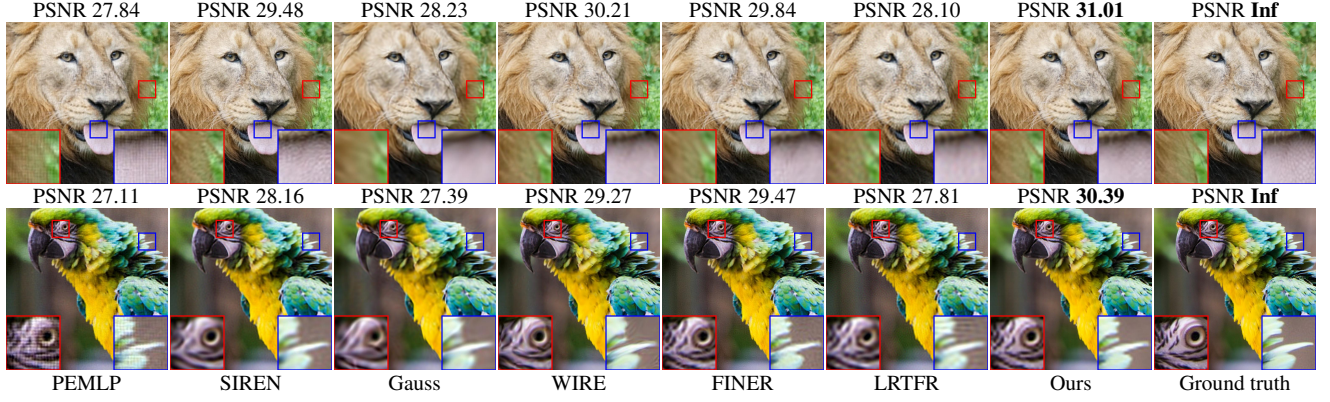


Figure 5. Visual super-resolution results at  $\times 4$  scaling on the *Lion* and *Parrot* images.

### 4.3. Discussions

To further investigate the contribution of each component within our framework, we perform a series of ablation studies under controlled conditions. All experiments are conducted with identical parameter settings and training protocols to ensure a fair comparison.

**Effect of Reparameterization.** We evaluate the effect of the proposed reparameterization on inpainting tasks using the color image *Airplane* with  $SR = 0.3$  and the MSI *Toy* with  $SR = 0.1$ . Comparisons are made between models without reparameterization (w/o Rep.) and with reparameterization (w/ Rep.). For the reparameterized models, the latent dimension is set as  $R_k = \beta r_k$ , where  $r_k$  is the original TR rank and  $\beta$  takes values 1, 3, 5, and 10. Fig. 7 shows the PSNR curves, indicating that reparameterization enhances reconstruction quality and stabilizes training, with larger  $\beta$  leading to faster convergence. Table 5 reports quantitative results across several datasets in terms of PSNR and runtime, confirming that the proposed reparameterization substantially enhances reconstruction accuracy with only a marginal increase in computational cost.

**Sensitivity of Basis Initialization.** To evaluate the sensitivity of basis initialization, we conduct HSI inpainting experiments on the *Washington DC* and *Botswana* datasets with  $SR = 0.2$ . According to Theorem 3, the entries of the fixed

Table 5. Effect of reparameterization on PSNR and runtime for different datasets with  $SR = 0.2$ .

| Metric   | <i>Airplane</i> |              | <i>Toy</i> |              | <i>Botswana</i> |              | <i>News</i> |              |
|----------|-----------------|--------------|------------|--------------|-----------------|--------------|-------------|--------------|
|          | w/o Rep.        | w/ Rep.      | w/o Rep.   | w/ Rep.      | w/o Rep.        | w/ Rep.      | w/o Rep.    | w/ Rep.      |
| PSNR     | 27.41           | <b>30.45</b> | 29.41      | <b>48.67</b> | 29.21           | <b>45.27</b> | 26.48       | <b>34.90</b> |
| Time (s) | 12.10           | 13.11        | 13.23      | 14.49        | 40.81           | 42.26        | 14.15       | 15.59        |

Table 6. PSNR results for HSI inpainting with different basis initialization scale at  $SR = 0.2$ .

| Data                 | Initialization Scale $a$ |       |              |       |       |       |
|----------------------|--------------------------|-------|--------------|-------|-------|-------|
|                      | 0.01                     | 0.05  | 0.165 (ours) | 0.3   | 0.5   | 1     |
| <i>Botswana</i>      | 33.86                    | 43.74 | <b>45.27</b> | 44.81 | 43.15 | 38.37 |
| <i>Washington DC</i> | 33.14                    | 46.08 | <b>47.96</b> | 47.44 | 45.90 | 38.20 |

basis  $\mathbf{B}^{(k)}$  are sampled independently from a uniform distribution  $\mathcal{U}(-a, a)$ , where the theoretically derived value of  $a$  is  $\sqrt{6/(r_{k+1} + R_{k+1})} \approx 0.165$ . As shown in Table 6, extremely small or large values lead to poor reconstructions, while the proposed scheme achieves the best performance, highlighting the importance of proper basis initialization.

**Effect of Shared Frequency Embedding.** We evaluate the impact of the shared frequency embedding on denoising tasks using the MSIs *Toy* and *Face* with  $SD = 0.2$ . In the baseline, each coordinate is processed independently, whereas in our proposed design, all coordinates are passed through a shared frequency embedding network. Fig. 8

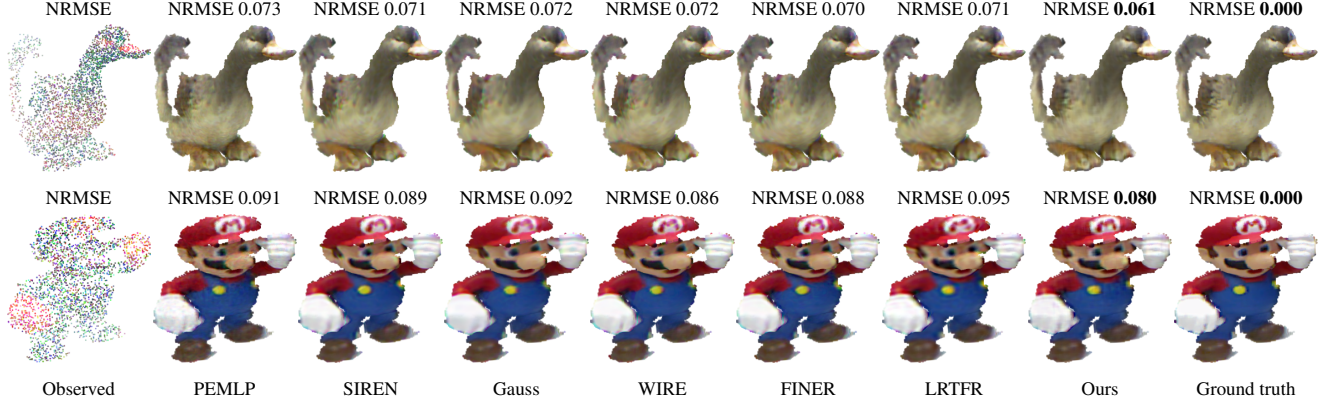


Figure 6. Visual comparisons on point cloud recovery for *Duck* (SR = 0.15) and *Mario* (SR = 0.2).

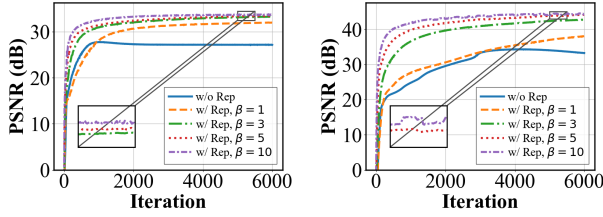


Figure 7. PSNR curves on the color image *Airplane* (SR = 0.3) (left) and *MSI Toy* (SR = 0.1) (right), comparing models without reparameterization (w/o Rep.), with reparameterization (w/ Rep.), and different latent sizes  $R_k = \beta r_k$  for  $\beta = 1, 3, 5, 10$ .

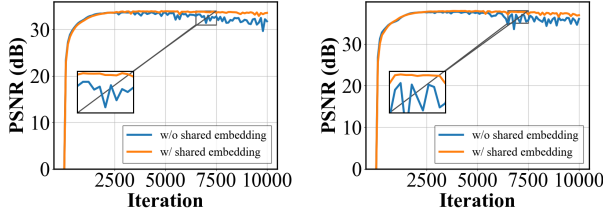


Figure 8. Effect of shared frequency embedding on denoising performance for the MSIs *Toy* (left) and *Face* (right) with SD = 0.2.

presents the PSNR curves over iterations, showing that the shared embedding produces more stable training and mitigates overfitting, resulting in more robust reconstructions.

**Model Complexity.** We vary  $\beta_k$  to obtain models with different parameter counts and compare them with LRTFR under matched complexity. For LRTFR, the MLP width is adjusted to ensure comparable parameters and FLOPs. Fig. 9 reports PSNR versus model complexity, showing that RepTRFD consistently outperforms LRTFR under similar parameter and FLOP budgets.

**Scalability to Higher-Order Tensors.** To evaluate the scalability of RepTRFD to higher-order data, we conduct inpainting experiments on color videos. Fig. 10 presents representative results on the *Container* and *Salesman* sequences. RepTRFD achieves consistently higher PSNR and SSIM values while producing visually sharper details.

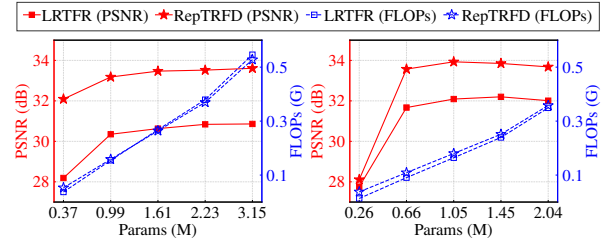


Figure 9. PSNR and FLOPs versus parameter count on *Airplane* (SR = 0.3) (left) and *Toy* (SD = 0.2) (right).

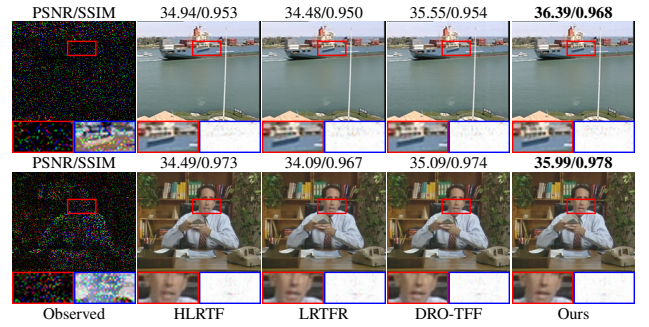


Figure 10. Visual inpainting results on color videos *Container* (SR = 0.1) and *Salesman* (SR = 0.2).

## 5. Conclusion

In this work, we presented RepTRFD for multi-dimensional data recovery. Specifically, by analyzing the TR’s frequency behavior, we identified the critical role of high-frequency components and proposed a structured reparameterization of TR factors that significantly facilitates their learning. We further established Lipschitz continuity and derived a variance-preserving initialization for the basis matrices, ensuring stable and effective training. Extensive experiments in various tasks validated our approach. Future work includes extending the framework for Tucker decomposition and block-term tensor decomposition.

## Acknowledgment

This work was partially supported by the National Natural Science Foundation of China under Grants 12571564, 12361089, and the Guangdong Basic and Applied Research Foundation 2024A1515012347.

## References

- [1] Hemant Kumar Aggarwal and Angshul Majumdar. Hyperspectral image denoising using spatio-spectral total variation. *IEEE Geoscience and Remote Sensing Letters*, 13(3):442–446, 2016. 5
- [2] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 126–135, 2017. 6
- [3] Zhicheng Cai, Hao Zhu, Qiu Shen, Xinran Wang, and Xun Cao. Batch normalization alleviates the spectral bias in coordinate networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25160–25171, 2024. 2, 3
- [4] J Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an  $N$ -way generalization of ‘Eckart-Young’ decomposition. *Psychometrika*, 35(3):283–319, 1970. 1
- [5] Yong Chen, Wei He, Naoto Yokoya, and Ting-Zhu Huang. Hyperspectral image restoration using weighted group sparsity-regularized low-rank tensor decomposition. *IEEE Transactions on Cybernetics*, 50(8):3556–3570, 2019. 1
- [6] A Cichocki, D Mandic, C Caiafa, AH Phan, G Zhou, Q Zhao, and L De Lathauwer. Tensor decompositions for signal processing applications. *IEEE Signal Processing Magazine*, 2013. 1
- [7] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. An introduction to independent component analysis. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 14(3):123–149, 2000. 1
- [8] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13733–13742, 2021. 2
- [9] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010. 4, 3
- [10] Azam Karami, Mehran Yazdi, and Grégoire Mercier. Compression of hyperspectral images using discrete wavelet transform and tucker decomposition. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(2):444–450, 2012. 1
- [11] Henk AL Kiers. Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 14(3):105–122, 2000. 1
- [12] Diederik P Kingma. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 5
- [13] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009. 1, 3
- [14] Yanyi Li, Xi Zhang, Yisi Luo, and Deyu Meng. Deep rank-one tensor functional factorization for multi-dimensional data recovery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 18539–18547, 2025. 2, 5, 6
- [15] Zhemin Li, Liyuan Ma, Hongxia Wang, Yaoyun Zeng, and Xiaolong Han. Learning input encodings for kernel-optimal implicit neural representations. In *International Conference on Machine Learning*, 2025. 3
- [16] Sheng Liu, Xi-Le Zhao, and Hao Zhang. Block tensor ring decomposition: Theory and application. *IEEE Transactions on Signal Processing*, 2025. 2
- [17] Zhen Liu, Hao Zhu, Qi Zhang, Jingde Fu, Weibing Deng, Zhan Ma, Yanwen Guo, and Xun Cao. Finer: Flexible spectral-bias tuning in implicit neural representation by variable-periodic activation functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2713–2722, 2024. 2, 6
- [18] Canyi Lu. Transforms based tensor robust PCA: Corrupted low-rank tensors recovery via convex optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1145–1152, 2021. 1
- [19] Canyi Lu, Jiashi Feng, Yudong Chen, Wei Liu, Zhouchen Lin, and Shuicheng Yan. Tensor robust principal component analysis with a new tensor nuclear norm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):925–938, 2019. 3
- [20] Yisi Luo, Xi-Le Zhao, Deyu Meng, and Tai-Xiang 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*, pages 19303–19312, 2022. 1, 5, 6
- [21] Yisi Luo, Xile Zhao, Zhemin Li, Michael K Ng, and Deyu Meng. Low-rank tensor function representation for multi-dimensional data recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3351–3369, 2023. 2, 5, 6
- [22] Yisi Luo, Xile Zhao, and Deyu Meng. Revisiting nonlocal self-similarity from continuous representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 2, 6
- [23] Yisi Luo, Xile Zhao, Kai Ye, and Deyu Meng. Neurvtv: Total variation on the neural domain. *SIAM Journal on Imaging Sciences*, 18(2):1101–1140, 2025. 2, 5, 6
- [24] Meng Lv, Wei Li, Tianhong Chen, Jun Zhou, and Ran Tao. Discriminant tensor-based manifold embedding for medical hyperspectral imagery. *IEEE Journal of Biomedical and Health Informatics*, 25(9):3517–3528, 2021. 1
- [25] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2, 6

- [26] Hesham Mostafa and Xin Wang. Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization. In *International Conference on Machine Learning*, pages 4646–4655. PMLR, 2019. 2
- [27] Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011. 1
- [28] Yuning Qiu, Guoxu Zhou, Qibin Zhao, and Shengli Xie. Noisy tensor completion via low-rank tensor ring. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1):1127–1141, 2022. 2
- [29] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310, 2019. 2
- [30] Sameera Ramasinghe and Simon Lucey. Beyond periodicity: Towards a unifying framework for activations in coordinate-mlps. In *European Conference on Computer Vision*, pages 142–158. Springer, 2022. 2, 6
- [31] Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in Neural Information Processing Systems*, 29, 2016. 2
- [32] Samuele Salti, Federico Tombari, and Luigi Di Stefano. Shot: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251–264, 2014. 6
- [33] Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G Baraniuk. Wire: Wavelet implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18507–18516, 2023. 2, 6
- [34] Vishwanath Saragadam, Randall Balestriero, Ashok Veeraraghavan, and Richard G Baraniuk. DeepTensor: Low-rank tensor decomposition with deep network priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10337–10348, 2024. 6
- [35] Kexuan Shi, Xingyu Zhou, and Shuhang Gu. Improved implicit neural representation with fourier reparameterized training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25985–25994, 2024. 2
- [36] Kexuan Shi, Hai Chen, Leheng Zhang, and Shuhang Gu. Inductive gradient adjustment for spectral bias in implicit neural representations. In *International Conference on Machine Learning*, 2025. 3
- [37] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020. 2, 6
- [38] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020. 2
- [39] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966. 1
- [40] Sai Karthikeya Vemuri, Tim Büchner, and Joachim Denzler. F-INR: Functional tensor decomposition for implicit neural representations. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6557–6568, 2026. 2
- [41] Jianli Wang and Xile Zhao. Functional transform-based low-rank tensor factorization for multi-dimensional data recovery. In *European Conference on Computer Vision*, pages 39–56. Springer, 2024. 2
- [42] Wenqi Wang, Vaneet Aggarwal, and Shuchin Aeron. Efficient low rank tensor ring completion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5697–5705, 2017. 2
- [43] Yao Wang, Jiangjun Peng, Qian Zhao, Yee Leung, Xi-Le Zhao, and Deyu Meng. Hyperspectral image restoration via total variation regularized low-rank tensor decomposition. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(4):1227–1243, 2017. 6
- [44] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 5
- [45] Longhao Yuan, Chao Li, Danilo Mandic, Jianting Cao, and Qibin Zhao. Tensor ring decomposition with rank minimization on latent space: An efficient approach for tensor completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9151–9158, 2019. 1, 2, 5
- [46] Qibin Zhao, Guoxu Zhou, Shengli Xie, Liqing Zhang, and Andrzej Cichocki. Tensor ring decomposition. *arXiv preprint arXiv:1606.05535*, 2016. 1, 3
- [47] Yu-Bang Zheng, Ting-Zhu Huang, Xi-Le Zhao, Qibin Zhao, and Tai-Xiang Jiang. Fully-connected tensor network decomposition and its application to higher-order tensor completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11071–11078, 2021. 5